Серия
“КОНСТРУИРОВАНИЕ
И ОПТИМИЗАЦИЯ ПРОГРАММ”

Выпуск 17

Серия
“КОНСТРУИРОВАНИЕ
И ОПТИМИЗАЦИЯ ПРОГРАММ”

Под редакцией
доктора физ.-мат. наук, профессора, чл.-корр. РАЕН
В. Н. Касьянова

Выпуски серии:

1. Смешанные вычисления и преобразование программ (1991)
2. Конструирование и оптимизация программ (1993)
3. Интеллектуализация и качество программного обеспечения (1994)
4. Проблемы конструирования эффективных и надежных программ (1995)
5. Оптимизирующая трансляция и конструирование программ (1997)
6. Проблемы систем информатики и программирования (1999)
7. Поддержка супервычислений и Интернет-ориентированные технологии (2001)
8. Касьянов В. Н., Мирзуитова И. Л. Slicing: срезы программ и их использование (2002)
9. Современные проблемы конструирования программ (2002)
10. Новые информационные технологии в науке и образовании (2003)
11. Программные средства и математические основы информатики (2004)
12. Методы и инструменты конструирования и оптимизации программ (2005)
13. Проблемы интеллектуализации и качества систем информатики (2006)
14. Касьянова Е. В. Адаптивные методы и средства поддержки дистанционного обучения программированию (2007)
15. Методы и инструменты конструирования программ (2007)
16. Конструирование и оптимизация параллельных программ (2008)
17. ***Евстигнеев В.А., Касьянов В.Н. Словарь по графам в информатике***

Российская академия наук
Сибирское отделение
Институт систем информатики
им. А. П. Ершова

В.А. Евстигнеев, В.Н. Касьянов

СЛОВАРЬ ПО ГРАФАМ В ИНФОРМАТИКЕ

Под редакцией
проф. Виктора Николаевича Касьянова

Новосибирск
ООО «Сибирское Научное Издательство»
2009

УДК 519.6
ББК 2218
К28

Евстигнеев В.А., Касьянов В.Н. Словарь по графам в информатике. — Новосибирск, 2009. — 300 с.
ISBN 978-591124-036-3

Книга содержит более 2500 относящихся к графам терминов вместе с их ясными и сжатыми определениями. Помимо базовой терминологии теории графов в неё включены термины и определения по информатике. Каждый термин приводится на английском и русском языках, после чего следует его описание.

Книга будет полезна студентам и преподавателям по информатике, а также всем, кто использует компьютеры дома, в лаборатории или на работе.

Издание осуществлено при частичной финансовой поддержке Российского фонда фундаментальных исследований (гранты РФФИ № 09-07-00012 и № 08-01-00673).

Р е ц е н з е н т
д.ф.-м.н., профессор В.К. Попков

ISBN 978-591124-036-3

Siberian Division of the Russian Academy of Sciences
A. P. Ershov Institute of Informatics Systems

V.A. Evstigneev, V.N. Kasyanov

DICTIONARY OF GRAPHS IN COMPUTER SCIENCE

Edited by
prof. V. N. Kasyanov

Novosibirsk
“Siberian Scientific Publishing House”
2009

Evstigneev V.A., Kasyanov V.N. Dictionary of graphs in computer science. — Novosibirsk, 2009. — 300p.
ISBN 978-591124-036-3

The book contains more than 2500 graph-related terms with their clear and succinct definitions. In addition to the base graph theory terminology, terms and definitions from computing are included. Every term is presented in English and Russian, and its description is given.

The book will be useful for students and teachers of computing, and everyone who uses computers at home, lab, or in business.

ISBN 978-591124-036-3

ISBN 978-591124-036-3

PREFACE

In 1999, in Nauka Publishing House, our “Dictionary of graph theory and its applications in informatics and programming” was issued, which covered main graph-related terms from monographs in Russian. It was the first dictionary of graphs in computing and it aroused a great interest of readers. The dictionary offered to the readers is an extended dictionary of 1999 and it includes more than 1000 new terms from journal articles whose abstracts were published in Abstract Journal “Mathematics” in section “Graph Theory”, as well as from volumes of annual conferences “Graph-Theoretic Concepts in Computer Science” and book series “Graph Theory Notes of New York”. The dictionary can be used as a usual English-Russian dictionary. For this purpose each term is accompanied by its Russian translation. In future the dictionary will be supplemented with the Russian-English dictionary and illustrations.

ПРЕДИСЛОВИЕ

В 1999 году в издательстве «Наука» вышел в свет наш «Толковый словарь по теории графов и её применении в информатике и программировании», который охватывал основные связанные с графами термины из монографий, вышедших на русском языке. Это был первый словарь по графам в информатике, и он вызвал большой интерес среди читателей. Предлагаемый читателям словарь представляет собой расширение словаря 1999 года и включает в себя более 1000 новых терминов из статей, рефераты которых публиковались в РЖ «Математика» в разделе «Теория графов», а также из томов ежегодных конференций «Graph-Theoretic Concepts in Computer Science» и книг серии «Graph Theory Notes of New York». Словарь может быть использован как обычный англо-русский словарь. Для этого каждый термин сопровождается русским переводом. В дальнейшем словарь будет дополнен русско-английским словарем и иллюстрациями.

A

Abdiff-tolerance competition graph — граф конкуренции ab-разностной толерантности.
See *Generalized competition graphs.*

Absolute hypergraph — абсолютный гиперграф.

Absolute incenter — абсолютный внутренний центр.

Absolute inner radius — абсолютный внутренний радиус.

Absolute median — абсолютная медиана.

Absolute outcenter — абсолютный внешний центр.

Absolute outer radius — абсолютный внешний радиус.

Absolute of a rooted tree — абсолют корневого дерева.

Absolute retract — абсолютный ретракт.
See *Retract.*

Absorbant set — внешне устойчивое множество, доминирующее множество.

Absorbent set — поглощающее множество.
See *Independent set.*

Abstract graph — абстрактный граф.

Abstract computer — абстрактный компьютер.
The same as *Model of computation.*

Abstract machine — абстрактная машина.
The same as *Model of computation.*

(Abstract) syntax representation — (абстрактное) синтаксическое представление.

Abstract syntax tree — абстрактное синтаксическое дерево.

Acceptable assignment — насыщающая разметка.

Access term — выражение доступа.
See *Large-block schema.*

Achromatic number — ахроматическое число.
The **achromatic number** $\psi(G)$ of G is the maximum number of sets in a partition of V into *independent* subsets $V_1, V_2, \ldots, V_k$ such that
(1) each V_i is an independent set of vertices, and
(2) for $i \neq j$, there exists $v_i \in V_i$ and $v_j \in V_j$ such that $v_iv_j \in E(G)$.

Achromatic status — ахроматический статус.
Let G be a connected graph with its *achromatic number* $\psi(G) = k$. The **achromatic status** $\sum \psi(G)$ is the minimum value of the *total*

status for a set X of k vertices each from a different set V_i in the partition of V, where minimum is taken over all possible partitions of V that satisfy (1) and (2) from the definition of achromatic number.

Acyclic chromatic number — ациклическое хроматическое число.

Acyclic colouring — ациклическая раскраска.

Acyclic dominating set — ациклическое доминирующее множество.
See *Tree dominating set.*

Acyclic domination number — ациклическое доминирующее число.
See *Tree dominating set.*

Acyclic graph — ациклический граф, бесконтурный граф.

Acyclic orientation — ациклическая ориентация.
See *Orientation of a graph.*

α**-Acyclic hypergraph** — α-ациклический граф.
See *Hypercycle.*

Additive hereditary graph property — аддитивное наследуемое свойство графа.

An **additive hereditary graph property** is a set of graphs, closed under isomorphism and under taking subgraphs and *disjoint unions.* Let $\mathcal{P}_1, \dots, \mathcal{P}_n$ be additive hereditary graph properties. A graph G has a property $(\mathcal{P}_1 \circ \dots \circ \mathcal{P}_n)$ if there is a partition $(V_1, \dots, V_n)$ of $V(G)$ into n sets such that, for all i, the induced subgraph $G[V_i]$ is in $\mathcal{P}_i$. A property $\mathcal{P}$ is **reducible** if there are properties $\mathcal{Q}$, $\mathcal{R}$ such that $\mathcal{P} = \mathcal{Q} \circ \mathcal{R}$; otherwise it is **irreducible**.

Addressable transformation graph — ациклический граф преобразований.
See *Addressing scheme.*

Addressing scheme — адресующая схема.

An **addressing scheme** for a *transformation graph* $\mathcal{G} = (V_\mathcal{G}, \Lambda_\mathcal{G})$ is a total function

$$\bar{a} : V_\mathcal{G} \to \bar{M}o(\Lambda_\mathcal{G}),$$

such that the following two properties hold:
(1) for some origin vertex $v_0 \in V_\mathcal{G}$ $v_0\bar{a} = \bar{I}d_{V_\mathcal{G}}$,
(2) for all transformations $\lambda \in \Lambda_\mathcal{G}$ and all vertices $v \in Domain(\lambda)$

$$(\lambda)\bar{a} = (v\bar{a}) \cdot \lambda;$$

the lefthand side denotes functional application, and the righthand side denotes multiplication in $\bar{M}o(\Lambda_\mathcal{G})$.

A transformation graph is **addressable** if it admits an addressing scheme.

Adjacency — смежность.

Adjacency list — список смежности.

In an **adjacency list** representation of a graph, each vertex has an associated list of its adjacent vertices. Their lists can be embodied in a table T. In order to trace the list for v_i, say, in the table, we consult $T(i, 2)$ which points to $T(T(i, 2), 1)$, where the first vertex adjacent to v_i is recorded. Then $T(T(i, 2), 2)$ points to $T(T(T(i, 2), 2), 1)$, where the second vertex adjacent to v_i is recorded, and so on. The list for v_i terminates, when a zero pointer is found. Notice the convention of numerically ordering the vertices adjacent to v_i within v_i's adjacency list; this is relevant to understanding some later examples of applying algorithms. Clearly, T has $(n + |E|)$ rows for a directed graph and $(n + 2|E|)$ for an undirected graph. In some circumstances it is additionally useful to use doubly linked lists for undirected graphs; we might also link the two occurrences of an edge (u, v), the first in u's adjacency list and the second in v's.

Adjacency matrix — матрица смежности.

The **adjacency matrix** $A(G)$ of a graph $G = (V, E)$ and an ordering $(v_1, \ldots, v_n)$ of V is the $(0, 1)$-matrix (a_{ij}) defined by

$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases}$$

Adjacency matrices are very handy when dealing with path problems in graphs. Nodes i, j are connected by a path (chain) of length k if and only if $A^k(i, j) = 1$.

Another its name is **Neighborhood matrix**.

The **augmented adjacency matrix** is formed by setting all values a_{ii} in the adjacency matrix to 1.

Adjacency operator — оператор смежности.

A directed infinite graph G is a pair of the set V of the countable vertices and the set E of the arrows (arcs) $u \leftarrow v$. Let $\mathcal{H}$ be a Hilbert space $\ell^2(G)$ on V with a canonical basis $\{e_v |\ v \in V\}$. Since every arrow $u \leftarrow v \in E$ induces a dyad $e_u \otimes e_v$, where $(x \otimes y)z = \langle z, y \rangle x$ for $x, y, z \in \mathcal{H}$, the **adjacency operator** $A(G)$ is expressed by

$$A(G) = \sum_{u \leftarrow v} e_u \otimes e_v$$

if G has a bounded degree.

Adjacency operators are classified as follows:

- A is **self-adjoint** if $A = A^*$.
- A is **unitary** if $A^*A = AA^* = I$.
- A is **normal** if $A^*A = AA^*$.
- A is **hyponormal** (resp. **co-hyponormal**) if $A^*A \geq AA^*$ (resp. $AA^* \geq A^*A$).
- A is **projection** if $A = A^* = A^2$.
- A is **partial isometry** if A^*A and AA^*.
- A is **isometry** (resp. **co-isometry**) if $A^*A = I$ (resp. $AA^* = I$).
- A is **nilpotent** if there exists a number n such that $A^n = 0$.
- A is **idempotent** if $A = A^2$.
- A is **positive** if $(Ax|x) \geq 0$ for $x \in H$.

Here G^* is the *adjoint* graph for a graph G.

Adjacency property — свойство смежности.

See *Convex bipartite graph.*

Adjacent arcs — смежные дуги.

An arc (u, v) is **adjacent to** an arc (w, z) if $v = w$, and (u, v) is **adjacent from** an arc (x, y) if $y = u$.

Adjacent edges — смежные ребра.

Two different edges which have a common vertex are called **adjacent edges**.

Adjacent faces — смежные грани.

Adjacent forest graph — смежный граф лесов.

Let G be a connected graph. Given $1 \leq \omega \leq |V(G)|-1$, the **adjacent forest graph** of G, denoted by $F^a_\omega(G)$, is defined as a *spanning* subgraph of a *forest graph* $F_\omega(G)$; its two vertices are adjacent if and only if the only two edges in the symmetric difference of their corresponding forests are adjacent in G.

See *Forest graph.*

Adjacent vertices — смежные вершины.

1. Two different vertices that incident with the same edge are called **adjacent vertices**.

2. In a digraph $G = (V, A)$, a vertex u is **adjacent to** v if $(u, v) \in A(G)$, and u is **adjacent from** w if $(w, u) \in A(G)$.

3. On adjacent vertices in a hypergraph, see *Partial edge.*

H-adjacent graphs — H-смежные графы.

Let G_1 and G_2 be two graphs of the same order and same size such that $V(G_1) = V(G_2)$, and let H be a connected graph of order 3 at least.

Two subgraphs H_1 and H_2 of G_1 and G_2, respectively, are **H-adjacent** if $H_1 \cong H_2 \cong H$ and H_1 and H_2 share some but not all edges, that is, $E(H_1 \cap E(H_2) \neq \emptyset$ and $E(H_2) - E(H_1) \neq \emptyset$. The graphs G_1 and G_2 are themselves **H-adjacent** if G_1 and G_2 contain H-adjacent subgraphs H_1 and H_2, respectively, such that $E(H_2) - E(H_1) \subseteq E(\bar{G}_1)$ and $G_2 = G_1 - E(H_1) + E(H_2)$.

See also *H-distance.*

Adjoint digraph — сопряженный орграф.

The **adjoint digraph** is defined as a graph, that is, the one whose arcs are exactly the converses for those of G. The *adjacency operator* $A(G^*)$ of G^* is the adjoint operator $A(G)^*$. Though G^* is called the *converse* digraph of G among graph theorists, the term **adjoint** is often used in this sense.

The **coadjoint graphs** are graphs G and G^* satisfying $G \cong G^*$.

Admissible sequence — допустимая последовательность,

Let $G = (V, E)$ be a simple undirected graph of order n. Let $\tau - (n_1, \dots, n_k)$ denote a sequence of positive integers such that $n_1 + \dots + n_k = n$. Such a sequence will be called **admissible** for G. If $\tau = (n_1, \dots, n_K)$ is an admissible sequence for G and there exists a partition $(V_1, \dots, V_k)$ of the vertex set V such that for each $i \in \{1, \dots, k\}$, $|V_i| = n_i$ and the subgraph induced by V_i is connected, then τ is called **realizable** in G and the sequence $(V_1, \dots, V_k)$ is said to be G-realization of τ or a realization of τ in G. A graph G is **arbitrarily vertex decomposable** if for each admissible sequence τ for G there exists a G-realization of τ.

Advancing arcs — опережающая дуга.

The same as *Forward arcs.*

Alive Petri net — живая сеть Петри.

A *Petri net* N is called **alive** iff whenever m is a reachable marking for N and t is a transition of N, it is possible for N to reach, starting from m, a marking in which t is enabled.

Algebraic connectivity — алгебраическая связность.

See *Laplacian matrix.*

Algebraic graph theory — алгебраическая теория графов.

Algebraic graph theory can be considered as a branch of the graph theory, where *eigenvalues* and eigenvectors of certain matrices associated with graphs are employed to deduce some of their properties. In fact, eigenvalues are closely related to almost all major invariants of a graph, linking one extremal property to another. These eigenvalues play a central role in our fundamental understanding of graphs. Many interesting books on the algebraic graph theory can be found, such as Biggs (1993), Cvetkovic et al. (1980), Seidel (1989), and Chung (1997).

One of the major contributions to the algebraic graph theory is due to Fiedler, where the properties of the second eigenvalue and eigenvector of the *Laplacian* of a graph have been introduced. This eigenvector, known as the Fiedler vector, is used in graph partitioning and nodal ordering.

Algorithm — алгоритм.

An **algorithm** is a specific set of instructions for carrying out a procedure or solving a problem, usually with the requirement that the procedure terminates at some point. Specific algorithms are sometimes called a method, a procedure, or a technique. The word "algorithm" is a distortion of al-Khwarizmi, a Persian mathematician who wrote an influential treatise about algebraic methods. The process of applying an algorithm to an input to obtain an output is called a **computation.**

Clearly, each *Turing machine* constitutes an algorithm in the intuitive sense. The statement that a Turing machine is a general-enough mathematical model for the intuitive notion of an algorithm is usually referred to as **Church's thesis**.

Almost cubic graph — почти кубический граф.

See *Cubic graph.*

Almost 3-regular — почти 3-регулярный граф.

See *Cubic graph.*

Alphabet — алфавит.

This is a finite nonempty set of elements called **letters** or **symbols**.

Alt — альт, альтернативный фрагмент, закрытый фрагмент.

An **alt** is a *fragment* with a single *initial node.*

Let A be a set of alts of a *cf-graph* G that contains H_1 and H_2. H_1 is **immediately embedded** in H_2 with respect to A if $H_1 \subset H_2$

and there is no alt $H_3 \in A$ such that $H_1 \subset H_3 \subset H_2$. H_1 is called an **internal** alt with respect to A if there is no alt in A immediately embedded in H, and an **external** alt with respect to A if there is no alt in A, into which H is immediately embedded.

A set of nontrivial alts A is called a **nested set of alts** (or **hierarchy of embedded alts**) of the cf-graph G if $G \in A$ and, for any pair of alts from A, either their intersection is empty or one of them is embedded in the other.

A sequence of cf-graphs $G_0, G_1, \ldots, G_r$ is called a representation of the cf-graph G in the form of a nested set of alts A (**A-representation of the cf-graph** G) if $G_0 = G$, G_r is a trivial graph and for any $i > 0$, G_i is a factor cf-graph $B_i(G)$, where B_i is the set of all external alts with respect to $\cup\{A_j : j \in [1, i]\}$ and A_j is the set of all internal alts with respect to $A \setminus (\cup\{A_k : k \in [1, i)\})$.

Alternating chain — альтернирующая цепь, чередующаяся цепь.

Given a chain $P = v_0, v_1, \ldots, v_k$, P is **alternating** with respect to a *matching* M if each edge in P that does not belong to M is followed in P by an edge that belongs to M, and vice versa. If P is alternating, the index k is odd, and $v_0, v_k \notin V(M)$, then P is **augmenting** with respect to M. A well known result due to Berge states that M is maximum if and only if M does not admit any augmenting chains.

Amalgam — амальгама.

Given two plane trees T_1 and T_2, with the same number of leaves and without degree 2 vertices, and a bijection φ between their leaf sets which preserves their order on the plane. The **amalgam** $A = \mathcal{A}(T_1, T_2, \varphi)$ is the union of the corresponding *Halin graphs* $\mathcal{H}(T_1)$ and $\mathcal{H}(T_2)$ in which the leaf vertices v and $\varphi(v)$ are identified.

Amalgamation of a graph — амальгамация графа.

Amalgamating a graph H can be thought of as taking H, partitioning its vertices, then, for each element of the partition, squashing together the vertices to form a single vertex in the amalgamated graph G. Any edges incident with original vertices in H are then incident with the corresponding new vertex in G, and any edge joining two vertices that are squashed together in H becomes a loop on the new vertex in G. The number of vertices squashed together to form a new vertex w is the **amalgamation number** $\eta(w)$ of w. The resulting graph is the **amalgamation** of the original. Formally, this is represented by a graph homomorphism $f : V(G) \to V(H)$; so for example if

$w \in V(H)$, then $\eta(w) = |f^{-1}(w)|$.

Amallamorphic graphs — амалламорфные графы.

Let M be a multigraph. Let $G(M)$ denote a graph obtained from M by replacing every multiple edge by a simple edge. Two multigraphs M_1 and M_2 are **amallamorphic** if $G(M_1)$ is isomorphic to $G(M_2)$.

Amount of a flow — величина потока.

See *Flow*.

Anarboricity of a graph — недревесность графа.

Ancestor of a vertex — предок вершины.

See *Directed tree*.

Animal — животное.

Annihilator — аннигилятор.

See *Irredundant set*.

Antichain — антицепь.

Given a poset $P = (X, <_P)$, an **antichain** of P is a subset of X consisted of pairwise incomparable elements.

Anticlique — антиклика.

The same as *Independent set* (of vertices).

Antidependence — антизависимость.

See *Data dependence*.

Antidirected Hamiltonian cycle — антинаправленный гамильтоновый цикл.

See *Antidirected path*.

Antidirected Hamiltonian path — антинаправленный гамильтоновый путь.

See *Antidirected path*.

Antidirected path — антиориентированный путь.

An **antidirected path** in a digraph is a simple path, every two adjacent arcs of which have opposite orientations, i.e. no two consecutive arcs of the path form a directed path. An **antidirected Hamiltonian path** in a digraph is a simple **antidirected path** containing all the vertices. Similarly, an **antidirected Hamiltonian cycle** is defined.

Anti-gem — антидрагоценность.

See *Gem*.

Antihole — антидыра.

See *Hole*.

Antimagic graph — антимагический граф.

An **antimagic graph** is a graph whose edges can be labeled with integers $1, 2, \ldots, e$ so that the sum of the labels at any given vertex is different from the sum of the labels of any other vertex, that is, no two vertices have the same sum. Hartsfield and Ringel conjecture that every tree other than K_2 is antimagic and, more strongly, that every connected graph other than K_2 is antimagic.

A special case is an (a, d)**-antimagic graph**. The weight $w(x)$ of a vertex $x \in V(G)$ under an edge labeling $f : E \to \{1, 2, \ldots, e\}$ is the sum of values $f(xy)$ assigned to all edges incident with a given vertex x. A connected graph $G = (V, E)$ is said to be (a, d)**-antimagic** if there exist positive integers a, d and a bijection $f : E \to \{1, 2, \ldots, e\}$ such that the induced mapping $g_f : V(G) \to W$ is also bijection, where $W = \{w(x); x \in V(G)\} = \{a, a + d, a + 2d, \ldots, a + (v - 1)D\}$ is the set of the weights of vertices.

Antiparallel arcs — антипараллельные дуги.

Given a directed graph G, **antiparallel arcs** are arcs (v, w) and (w, v), such that $v \neq w$.

Antiprism — антипризма.

The **antiprism** A_n, $n \geq 3$, is the plane regular graph of degree 4 (an Archimedean convex polytope). In particular, A_3 is the octahedron. The k**-antiprism** is the 4-regular plane graph consisting of two k-gons and $2k$ triangles such that every vertex is incident with three triangles and one k-gon.

Antisymmetric graph — антисимметричный граф.

If D is an orientaion of *underlying graph* $UG(D)$, then D is also called an **antisymmetric** digraph.

Antisymmetric relation — антисимметричное отношение.

See *Binary relation*.

α**-Approximable problem** — α-аппроксимируемая задача.

Apex graph — вершинный граф.

An **apex graph** is a graph G that has a vertex $v \in V(G)$ such that $G \setminus \{v\}$ is *planar*.

Approximate point spectrum — аппроксимирующий точечный спектр.

See *Spectrum*.

Approximation algorithm — аппроксимирующий алгоритм.

For the *travelling salesman problem*, as indeed for any other *intract-*

able problem, it is useful to have a polynomial time algorithm which will produce, within known bounds, an approximation to the required result. Such algorithms are called **approximation algorithms**. Let L be the value obtained (for example, this may be the length of a travelling salesman's circuit) by an approximation algorithm and let L_0 be an exact value. We require a performance guarantee for the approximation algorithm which could, for a minimisation problem, be stated in the form:

$$1 \leq L/L_0 \leq \alpha.$$

For a maximisation problem we invert the ratio L/L_0. Of course, we would like α to be as close to one as possible.

Unfortunately, not every heuristic produces a useful approximation algorithm.

Arbitrarily hamiltonian graph — произвольно гамильтонов граф.

Arbitrarily traceable graph — произвольно вычерчиваемый граф.

Arbitrarily traverseable graph — произвольно проходимый граф.

Arbitrarily vertex decomposable graph — произвольно вершинно разложимый граф.

A graph G of order n is said to be **arbitrarily vertex decomposable**, if for each sequence $(n_1, \ldots, n_k)$ of positive integers such that $n_1 + \ldots + n_k = n$ there exists a partition $(V_1, \ldots, V_k)$ of the vertex set of G such that, for each $i \in \{1, \ldots, k\}$, V_i induces a connected subgraph of G on n_i vertices.

See also *Admissible sequence.*

Arboreal hypergraph — древесный гиперграф.

See *Hypertree.*

Arborescence — ориентированное дерево.

This is a digraph G with a specified vertex a called a *root* such that each *point* $x \neq a$ has indegree 1 and there is a unique (a, x)-path for each point x. **Arborescence** can be obtained by specifying a vertex a of a tree and then orienting each edge e such that the unique path connecting a to e ends at the tail of e. An **inverse arborescence** is a digraph obtained from an **arborescence** by inverting its edges.

Arboricity of a graph — древесность графа.

Arc — дуга.

See *Directed graph.*

F**-Arc** — F-дуга.

See *Numbering of cf-graph.*

Arc base — база дуг.

Arc-forwarding index — прямодуговой индекс.

See *Routing.*

Arithmetic graph — арифметичский граф.

Let m be a power of a prime p, then the **arithmetic graph** G_m is defined to be a graph whose vertex set is the set of all divisors of m (excluding 1) and two distinct vertices a and b are adjacent if and only if $\gcd(a, b) = p^i$, where $i = 1 \pmod 2$.

(k, d)**-Arithmetic graph** — (k, d)-арифметичский граф.

See *Weakly* (k, d)*-arithmetic graph.*

(k, d)**-Arithmetic numbering** — (k, d)-арифметическая нумерация.

See *Weakly* (k, d)*-arithmetic graph.*

Argument — аргумент (оператора).

See *Large-block schema.*

Arrangeable alt — аранжируемый альт.

An alt is called **arrangeable** if its *arrangement* exists and **non-arrangeable** otherwise.

Arrangeable graph — аранжируемый граф.

A *cf-graph* is called **arrangeable** if its *arrangement* exists and **non-arrangeable** otherwise.

Every arc of an arrangeable graph G is either forward or backward arc. An arc of G is called a **backward** arc if it is an F*-inverse arc* for an arrangement of G and a **forward** arc if it is an F*-direct arc* for an arrangement of G.

A **depth** of an arrangeable graph G is defined as the depth of an arrangement of G.

Arrangement — аранжировка.

A *numbering* F of an *alt* C is called an **arrangement** if every simple path in C from its initial node is F*-path.*

Articulation point — точка сочленения графа, разделяющая вершина, шарнир.

A vertex $v \in V$ is an **articulation point** of a graph $G = (V, E)$ if $G(V \setminus \{v\})$ is disconnected. A graph G is **2-connected** if G has no **articulation points**. The maximal 2-connected subgraphs of G are the **blocks** of G.

Other names are **Cutpoint, Cutting vertex, Cutvertex**.

Articulation set — множество сочленения.

Given a hypergraph $\mathcal{E} = (V, \{E_1, \ldots, E_m\})$, a set $A \subseteq V$ is an

articulation set for $\mathcal{E}$ if $A = E_1 \cap E_2$ for some pair of hyperedges $E_1, E_2 \in \mathcal{E}$ and $\mathcal{E}[V \setminus A]$ has more connected components than $\mathcal{E}$.

m-Ary tree — m-арное дерево, сильно ветвящееся дерево.

Assignment problem — задача о назначениях.

Associated Cayley digraph — соотнесённый орграф Кэли.

Let Γ be a group and S be a generating set of Γ such that
(1) $e \notin S$, e is the identity in Γ,
(2) $s \in S \Leftrightarrow s^{-1} \in S$.

The **associated Cayley digraph** $Cay(\Gamma, S)$ is a digraph whose vertices are the elements of Γ and arcs are the couples (x, sx) for $x \in \Gamma$ and $s \in S$.

With this definition, $Cay(\Gamma, S)$ is a connected symmetric digraph (in fact, a strongly connected digraph).

Associated undirected graph — соотнесенный неориентированный граф.

Associative search — ассоциативный поиск.

Asteroidal set — астероидальное множество.

See *Asteroidal number*.

Asteroidal number — астероидальное число.

A set of vertices $A \subseteq V$ of a graph $G = (V, E)$ is an **asteroidal set** if for each $a \in A$ the set $A - a$ is contained in one component of $G - N[a]$. The **asteroidal number** of a graph G, denoted by $an(G)$, is the maximum cardinality of the asteroidal set in G.

Graphs with asteroidal number at most two are commonly known as **AT-free graphs**. The class of AT-free graphs contains well-known graph classes such as *interval, permutation* and *cocomparability* graphs.

Asteroidal triple — астероидальная тройка.

Three pairwise nonadjacent vertices u, v, w of G are an **asteroidal triple** of G if for any two of them there is a path connecting the two vertices which avoids the neighborhood of the remaining vertex.

Asymmetric graph — асимметричный граф.

Asymmetric relation — асимметричное отношение.

Atomic formula — атомарная формула.

See *Logic for expressing graph properties*.

Attachment graph — соединяющий граф.

Attribute grammar — атрибутная грамматика.

L-Attribute grammar — L-атрибутная грамматика.

Attribute tree — атрибутное дерево.

Augmented adjacency matrix — расширенная матрица смежности.

See *Adjacency matrix.*

Augmenting chain — увеличивающая цепь.

See *Alternating chain.*

Automata theory — теория автоматов.

In theoretical computer science, **automata theory** is the study of *abstract machines* and problems which they are able to solve. **Automata theory** is closely related to *formal language theory* as the automata are often classified by the class of *formal languages* they are able to recognize.

Automorphism — автоморфизм (ор)графа.

1. For an undirected graph, see *Isomorphic graphs.*

2. For a directed graph, **automorphism** is a permutation α of $V(G)$ such that the number of (x, y)-edges is the same as the number of $(\alpha(x), \alpha(y))$-edges $(x, y \in V(G))$. We also speak of the **automorphism** of a graph G with colored edges. This means a permutation α such that the number of (x, y)-edges is the same as the number of $(\alpha(x), \alpha(y))$-edges with any given color.

The set of all automorphisms of a (di)graph forms a permutation group $A(G)$.

Automorphism group — группа автоморфизмов.

See *Isomorphic graphs.*

Automorphism problem — проблема автоморфизма.

See *Isomorphic graphs.*

Average connectivity — средняя связность.

See *Connectivity.*

Average independent domination number — среднее число независимого доминирования.

See *Dominating set.*

Average domination number — среднее число доминирования.

See *Dominating set.*

AVL-tree — АВЛ-дерево.

A *binary search tree* is an **AVL-tree**, if for each internal node u with children v_1 and v_2,

$$|height(v_1) - height(v_2)| \leq 1.$$

See *Heightbalanced tree.*

B

Backbone — хребет.

The **backbone** β of a *caterpillar* C is a (possibly trivial) path that remains after a *pruning* of C.

Backbone coloring — хребтовая раскраска.

Consider a graph $G = (V, E)$ with a spanning tree $T = (V, E_T)$ (backbone). A vertex coloring $f : V \to \{1, 2, \ldots\}$ is proper, if $|f(u) - f(v)| \geq 1$ holds for all edges $(u, v) \in E$. A vertex coloring is a **backbone coloring** for (G, T), if it is proper and, additionally, $|f(u) - f(v)| \geq 2$ holds for all edges $(u, v) \in E_T$ in the spanning tree T.

The **backbone coloring number** $BBC(G, T)$ of (G, T) is the smallest integer ℓ for which a backbone coloring $f : V \to \{1, 2, \ldots, \ell\}$ exists.

Backbone coloring number — число хребтовой раскраски.

See *Backbone coloring*.

Back-edge — обратная дуга.

See *Depth-first search. 2*.

Backward arc — обратная дуга, дуга назад.

1. See *Basic numberings*.

2. See *Numbering of a cf-graph*.

3. See *Arrangeable graph*.

Balance of a vertex — баланс вершины.

Balanced circuit — сбалансированный цикл.

A **balanced circuit** in a hypergraph is a circuit $(x_1, E_1, \ldots, x_k, E_k)$ such that either $k = 2$ or there is an incidence $x_i \in E_j$, where $j \neq i, i-1$ and $(i, j) \neq (1, k)$.

Balanced digraph — сбалансированный орграф.

1. A digraph is **balanced**, if for every vertex v, $deg^+(v) = deg^-(v)$.

2. A directed graph is called **balanced** if each of its cycles contains equal numbers of forward and backward arcs.

3. A directed graph G is **balanced** if there exists a *homomorphism* of G to a monotone path.

Balanced graph — сбалансированный граф.

See *Density*.

w**-Balanced graph** — w-сбалансированный граф.

See *w-density*.

Balanced hypergraph — сбалансированный гиперграф.

A hypergraph is a **balanced hypergraph** if every circuit of odd length is *balanced*. A hypergraph is a **totally balanced hypergraph** if every circuit is balanced.

Balanced signed graph — сбалансированный знаковый граф.

See *Signed labeled graph*.

Ball number — шаровое число (графа).

By a family of solid balls in R^3, we mean a family of balls no two of which penetrate each other. A chain is a finite sequence of solid balls $b_1, b_2, \ldots, b_n$ in which each consecutive pair of balls is tangent. The two balls b_1, b_n are called the end balls of the chain.

Let a_1, a_2 be two solid balls. If an end of a chain is tangent to a_1, and the other end of the chain is tangent to a_2, then the chain is said to connect a_1, a_2. Let $G = (V, E)$ be a finite graph. Take a family of red solid balls $a_i,\ i \in V$. Connect each non-tangent pair a_i, a_j $(ij \in E)$ by a chain of blue solid balls so that no two distinct chains share a blue ball. Then we have a family $\mathcal{F}$ consisting of solid balls $a_i,\ i \in V$, and the solid balls making the chains. This family is called a representation of G. The **ball number** $b(G)$ of G is the minimum number of balls necessary to make a representation of G. For example, $b(K_6) = 8$, $b(K_7) = 13$.

Bandwidth — ширина полосы.

Let $G = (V, E)$ be a simple graph and let f be a numbering of vertices of G.

$$B(G, f) = \max_{(u,v)\in E} |f(u) - f(v)|$$

is called the **bandwidth** of the numbering f.

The **bandwidth** of G, denoted $B(G)$, is defined to be the minimum bandwidth of numberings of G.

The bandwidth problem for graphs has attracted many graph theorists for its strong practical background and theoretical interest. The decision problem for finding the bandwidths of arbitrary graphs is NP-complete, even for trees having the maximum degree 3, *caterpillars* with hairs of length at most 3 and *cobipartite graphs*. The problem is polynomially solvable for caterpillars with hairs of length 1 and 2, *cographs*, and *interval graphs*.

See also *Layout*, *Graceful graph*.

Bandwidth sum — широкополосная сумма.

The **bandwidth sum** of a graph G, denoted by $B_s(G)$, is defined by

$$B_s(G) = \min \sum_{uv \in E(G)} |f(u) - f(v)|,$$

where minimum is is taken for all *proper labelings* of G.

Bar-visibility graph — полосо-видимый граф.

See *Visibility graph*.

Base digraph — базовый орграф.

Base of a directed graph — база орграфа, основание орграфа.

Base of a matroid — база матроида.

See *Matroid*.

1-Base — 1-база.

Based graph — базируемый граф.

Basic block — линейный участок, луч.

1. A **basic block** is a sequence of consecutive statements (of a program) such that control flow enters the sequence at the beginning and leaves the sequence at the end without halt or possibility of branching except for the end.

2. A simple path $P = (p_1, p_2, \ldots, P_r)$, $r \geq 1$, in a *control flow graph* is called a **basic block** (or **ray**) if p_{i-1} is a single predecessor of p_i and p_j is a single successor of p_{j+1} for all $i > 1$ and $j < r$.

Basic cycle — базисный цикл, фундаментальный цикл.

Basic cycle set — базисное множество циклов, множество фундаментальных циклов.

Basic numberings — базисные нумерации.

Let G be a *cf-graph* with the *initial node* p_0. **Basic numberings** M and N of G are defined as follows.

A *numbering* F of a *cf-graph* G is called a **direct** numbering (or an M**-numbering**) of G if the following three properties hold:

(1) $F(p_0) = 1$;

(2) for any node p distinct from p_0 there is a predecessor q of p such that $F(q) < F(p)$;

(3) for any two nodes q and p, if (q, p) is an F-*direct arc* and $F[F(q)+1, F(p)-1]$ contains no predecessors of p, then $F(r) < F(p)$ for any successor r of any node from $F[F(q)+1, F(p)-1]$.

A numbering F of G is called an **inverse** numbering (or an N**-numbering**) **correlated** with a direct numbering M of G if for any two nodes q and p of G, $F(q) < F(p)$ if and only if either p is *M-reachable* from q or $M(p) < M(q)$ and q is not M-reachable from p.

Any cf-graph G has at least one pair of correlated basic numberings, but the same inverse numbering of G can be correlated with different direct numberings of G.

The set of all arcs (p, q) of G is divided with respect to its correlated basic numberings M and N into the four subclasses: **tree arcs** T, **forward arcs** F, **backward arcs** B and **cross arcs** C defined as follows:

(1) $(p, q) \in T$ iff (p, q) is M-direct arc and there is no M-direct arc (s, q) such that $M(p) < M(s)$;

(2) $(p, q) \in F$ iff (p, q) is M-direct arc and $(p, q) \notin T$;

(3) $(p, q) \in B$ iff $M(q) < M(p)$ and $N(q) < N(p)$;

(4) $(p, q) \in C$ iff $M(q) < M(p)$ and $N(p) < N(q)$.

A graph (X, T, p_0) were X is the set of vertices of G is a *spanning tree* of G with root p_0.

Every pair of correlated basic numberings can be computed in linear time by the procedure $DFS(p_0)$ (See *Depth-first search*).

Basis number — базисное число.

A basis $\mathcal{B}$ for cycle space $\mathcal{C}(G)$ is called a d-fold if each edge of G occurs in at most d of the cycles in the basis $\mathcal{B}$. The **basis number** $b(G)$ of G is the least non-negative integer d such that $\mathcal{C}(G)$ has a d-fold basis.

Berge's complete graph — полный граф Бержа.

Berge graph — граф Бержа.

1. See *Hole*.

2. A graph for which *Berge's conjecture* is fulfilled is called a **Berge graph**; clearly, each Berge graph is *perfect*.

Berge's conjecture — гипотеза Бержа.

In 1960, C.Berge conjectured that a graph is *perfect* iff none of its induced subgraphs is a C_{2k+1} or the *complement* of such a cycle, $k \geq 2$.

This conjecture is well-known as the *Strong Perfect Graph Conjecture* and is still open.

Berge's Formula — формула Бержа.

Let G be a graph and let $o(G)$ be the number of *odd components* of G.

Berge's Formula for estimating the *deficiency* of the graph:

$$def(G) = \max_{S \subset V(G)} \{o(G \setminus S) - |S|\}.$$

Biblock — библок.
Bicenter — бицентр.
Bicenter tree — бицентральное дерево.
Bicentre — бицентр.
Bicentroid of a tree — бицентроид дерева.
Bichordal bipartite graph — бихордальный двудольный граф.
A bipartite graph G is defined as **bichordal**, if any cycle C_n, $n > 4$, of G has at least two chords.
Bichromatic graph — бихроматический граф.
Bichromatic hypergraph — бихроматический гиперграф.
Biclique — биклика.
1. In a *bipartite graph* $G = (V, W, E)$, a subset $A \subseteq V \cup W$ is called a **biclique** if it induces a *complete bipartite graph.*
2. Given a graph, this is an inclusion-maximal induced *complete bipartite subgraph* of a graph.
Biclique edge cover — бикликовое покрытие ребер.
Given a *bipartite graph* $B = (U \cup V, E)$, a **biclique edge cover** for B is a covering of the edge set E by *bicliques.*
Biclique edge covering number — число бикликового покрытия ребер.
The **biclique edge covering number** of a *bipartire graph* B, $\beta^*(B)$, is defined as the minimum number of *bicliques* required to cover the edges of B.
Biclique number — бикликовое число.
The **biclique number** $w^*(B)$ of a *bipartite graph* B is defined as the cardinality of the maximum *cross-free matching* in B.
Bicoloured subgraph — двуцветный подграф.
Bicomponent — бикомпонента.
The same as *Strongly connected component.*
Biconnected component — компонента двусвязности, блок.
A **biconnected component** of a graph G is a maximal set of edges such that any two edges in the set lie on a common cycle. A **block** is a *bridge(2)* or a *biconnected component* of G.
See also *Articulation point, 2-Connected graph, Block.*

Biconvex bipartite graph — бивыпуклый двудольный граф.
See *Convex bipartite graph.*

Bicritical graph — бикритический граф.
A finite, undirected, connected and simple graph G is said to be **bicritical** if $G-u-v$ has a *perfect matching* for each pair of vertices u and v in G such that $u \neq v$.
Bicritical graphs play a central role in the decomposition theory of graphs in terms of their maximum *matchings.*

Bicubic graph — бикубический граф.
bicubic graph is a *bipartite cubic graph.*

Bidirectional arcs — бинаправленные дуги.
That is a pair of arcs (x, y), (y, x). If a directed graph D has no bidirectional arcs, then D is called an *orientation* of *underlying graph* $UG(D)$.

Bifurcant — бифуркант.

Bigraph — биграф, двудольный граф.
The same as *Bipartite graph.*

Bi-Helly family — би-Хелли семейство.
A hypergraph $\mathcal{H}$ is called a **bi-Helly family** if it satisfies the following property: if any two edges of a subhypergraph $\mathcal{H}' \subseteq \mathcal{H}$ share at least two vertices, then

$$\left| \bigcap_{H \in cal H'} H \right| \geq 2.$$

Bihypergraph — бигиперграф.
Let H^0 and H^1 be hypergraphs with the same vertex set V. An ordered pair $H = (H^0, H^1)$ is called a **bihypergraph** with the set of 0-edges $E(H^0)$ and the set of 1-edges $E(H^1)$. Every hyperedge of either H^0 or H^1 is considered as a hyperedge of H. The order of H is $n(H) = |V|$. The rank of H is $r(H) = \max\{r(H^0), r(H^1)\}$.
A bihypergraph $H = (H^0, H^1)$ is called **bipartite** if there exists an ordered partition $V^0 \cup V^1 = V(H)$ (bipartition) such that the set V^i is *stable* in H^i, $i = 0, 1$.

Binary Hamming graph — бинарный граф Хэмминга.

Binary de Bruijn graph — бинарный граф де Брёйна.
See *De Bruijn graph.*

Binary n-dimensional cube — двоичный n-мерный куб.

Binary matroid — бинарный матроид.
See *Matrix matroid.*

Binary relation — бинарное отношение.

R is a **binary relation** on V if $R \subseteq V \times V$. R is **reflexive** on V if for all $v \in V$ $(v,v) \in R$ and **irreflexive** otherwise. R is **transitive** on V, if for all $u,v,w \in V$ $(u,v) \in R$ and $(v,w) \in R$ implies $(u,w) \in R$. R is **symmetric**, if $(v,w) \in R$ implies $(w,v) \in R$, and **antisymmetric** on V, if for all $u,v \in V$ $(u,v) \in R$ and $(v,u) \in R$ implies $u = v$.

The **inverse** of a relation R, denoted by R^{-1}, is obtained by reversing each of the pairs belonging to R, so that $aR^{-1}b$ iff bRa. Let R^U denote the union and R^I the intersection of a collection of relations $\{R_k :\ k \in \mathcal{C}\}$ in S, where $\mathcal{C}$ is some nonempty index set. Then aR^Ub iff aR_kb for some k in $\mathcal{C}$, and aR^Ib iff aR_kb for each k in $\mathcal{C}$.

See also *Equivalence relation*, *Partial order*.

Binary search tree — бинарное дерево поиска.

Binary search trees (BT) are a special class of GBST (*generalized binary split trees*) with equal key and split values in all nodes.

Note that the BST's and BT's do not contain each other. The intersection of BT's and BST's is the set of *frequency-ordered binary search trees (FOBT's)*.

Binary sorting tree — бинарное дерево сортировки.

Binary split tree — бинарное расщепляемое дерево.

Binary split trees (BST's) are a data structure for storing static records with skewed frequency distribution. Each node of the tree contains two values, one of them being the key (records stored in this node are associated with this value), the other being a split value. The split value is used as a guide for further search in the tree if the key value is not equal to the search value. The reason to store two values is to "decouple" the conflicting functions of frequency ordering and subtree construction.

By separating the split value from the key value, we are allowed to store whatever key we want in the root without putting constraints on the structure of the subtree. The most reasonable choice of the root is by selecting the most frequent key in the tree.

Binary split trees are also a special class of GBST's (*generalized binary split trees*) with the constraint of decreasing frequency.

Note that the BST's and BT's do not contain each other.

Binary tree — бинарное дерево.

An n**-node binary tree** is defined to be a rooted tree where each of the n nodes has zero, one or two *descendants*, and a distinction is made between the left and right subtrees.

One class of operation which may be performed on binary trees is that of traversing the whole tree: each node in the tree is “visited”, or “processed”, exactly once in some predefined order. The three most natural traversal orders are known as *preorder, inorder and postorder* (Knuth, 1975). Preorder and postorder traversals are also commonly called *depth-first* and bottom-up *traversals*, respectively, though these latter terms are normally used in connection with more general types of trees.

Binary vertex — бинарная вершина.

See *Unary node.*

k-Binding number — k-связывающее число.

The **k-binding number** of G is defined to be

$$bind^k(G) = \min_{X \in \delta^{k-1}(G)} \left\{ \frac{|\Gamma^{k-1}(X)|}{|X|} \right\},$$

where

$$\delta^k(G) = \{X : \emptyset \neq X \subseteq V(G) \text{ and } \Gamma^k(X) \neq V(G)\}.$$

Let $k \geq 2$. The following two properties are obvious.

1. Let G be a graph with n vertices. If $diam(G) \leq k-1$, then $bind^k(G) = n-1$.

2. If a graph G has at least one isolated vertex, then $bind^k(G) = 0$.

Binode — бивершина.

See *T-Numbering.*

Binomial tree — биномиальное дерево.

There are several equivalent definitions of a **binomial tree**. One recursive definition is to define the tree B_0 as a single vertex, and then the rooted tree B_{i+1} is obtained by taking one copy of each of B_0 through B_i, adding a root, and making the old roots the children of the new root. In particular, the tree B_i has 2^i vertices.

An equivalent definition is based on the *corona* of a graph. Recall that the corona of a graph is obtained by adding a new leaf adjacent to each existing vertex. Then B_{i+1} is the corona of B_i.

Bipanpositionable graph — бипанпропозицируемый граф.

A bipartite hamiltonian graph G is **bipanpropositionable** if for any two different vertices x and y of G and for any integer k with $d_G(x,y) \leq k < |V(G)|/2$ and $(k - D_G(x,y))$ is even, there exists a hamiltonian cycle C of G such that $d - C(x,y) = k$.

Bipartite bihypergraph — двудольный бигиперграф.

See *Bihypergraph*.

Bipartite density — двудольная плотность.

Let $G = (V, E)$ be a *simple graph*. Let H be any *bipartite* subgraph of G with the maximum number of edges. Then $(\varepsilon(G) = |E(G)|)$

$$b(G) = \frac{\varepsilon(H)}{\varepsilon(G)}$$

is called the **bipartite density** of G. The problem of determining the bipartite density of a graph is *NP-complete problem*, even if G is *cubic* and *triangle-free*.

Bipartite graph — двудольный граф.

A **bipartite graph** is a graph whose vertex set can be partitioned into two nonempty subsets V and W such that every edge of G joins V and W.

Bipartite matroid — двудольный матроид.

Bipartite permutation graph — двудольный граф перестановок.

A class of **bipartite permutation graphs** is the intersection of two well studied subclasses of *perfect* graphs, namely bipartite and permutation graphs. The other name is **bipartite tolerance graphs**.

Bipartite tolerance graph — двудольный граф перестановок.

See *Bipartite permutation graph*.

Bipyramid — бипирамида.

The plane dual graph D_n^* of a *prism* D_n is the graph of a **bipyramid**. See also *Quasibipyramid*.

Bisection width of a graph — ширина бисекции графа.

The **bisection width** $bw(G)$ **of a graph** G is the minimal number of edges between vertex sets A and $\bar{A}$ of almost equal sizes, i.e. $A \cup \bar{A} = V(G)$ and $||A| - |\bar{A}|| \leq 1$. If $A \subseteq V(G)$, then $E(A, \bar{A})$ denotes the set of edges of G having one end in A and another end in $V(G) \setminus A = \bar{A}$. The **isoperimetric number** $i(G)$ of a graph G equals the minimum of the ratio $|E(A, \bar{A}|/|A|$ for all $A \subseteq V(G)$ such that $2|A| \leq |V(G)| = n$.

From the definitions, we have the following inequality for these characteristics:

$$i(G) \leq \frac{2}{n} bw(G).$$

Bisimplicial edge — бисимплициальное ребро.

An edge $e = (x, y)$ of a *bipartite graph* H is called a **bisimplicial**

edge if $N(x) \cup N(y)$ ($N(x)$ is the *neighborhood* of x) induces a *complete bipartite* subgraph of H.

Bistochastic matrix — бистохастическая матрица.

Block — блок.

See *Articulation point.*

Block duplicate graph — блочно удвоенный граф.

A **block duplicate graph** (**BD-graph**) is a graph obtained by adding zero or more *true twins* to each vertex of a *block graph* (or equivalently to each *cut-vertex*, since adding a true twin to a non-cut vertex preserves the property of being a block graph).

Block graph — блоковый граф.

A graph G is a **block graph** if G is connected and every maximal 2-connected subgraph (i.e., a *block*) is complete (i.e., a *clique*). Another name is **completed Husimi tree**.

Block of a graph — блок графа, компонента двусвязности.

For a graph G, the maximal *2-connected* subgraph of G. Another name is *Biconnected component.*

See also *Leaf, Endblock.*

Block-cutvertex tree, block-cutpoint graph — дерево (граф) блоков и точек сочленения.

The **block-cutvertex tree** of a connected graph G has a B-node for each block (biconnected component) of G, and a C-node for each *cutvertex* of G. There is an edge between a C-node u and a B-node b if and only if u belongs to the corresponding block of b in G. The **block-cutvertex tree** can be constructed in linear time.

Bondy–Chvátal closure operation — операция замыкания Бонди-Хватала.

Given a graph of order n, repeat the following operation as long as possible. For each pair of nonadjacent vertices a and b, if $d(a)+d(b) \geq n$, then add the edge ab to G. We denote by $cl(G)$ the resulting graph and call it the **Bondy–Chvátal closure** of G.

The other name is **Hamiltonian closure**.

Boundary NCE graph grammar — граничная графовая грамматика типа NCE.

An *NCE graph grammar* is **boundary** (B-NCE) if the axiom and the right hand-side of each production do not contain adjacent nonterminal nodes.

Boundary of a face — граница грани.

Boundary of a 2-mesh — граница 2-сетки.
See *n-Mesh*.

Boundary operator — граничный оператор.

Boundary node of a fragment — граничная вершина фрагмента.
See *Fragment*.

Bounded reachability matrix — матрица ограниченных достижимостей.

Bounded Petri net — ограниченная сеть Петри.
A Petri net is **bounded** if its set of reachable markings is finite. The **boundedness problem** for Petri nets is *decidable*, but it is a *PSPACE-hard problem*.

k**-Bounded Petri net** — k-ограниченная сеть Петри.
Let N be a *Petri net* with *initial marking* m_0. N is called k**-bounded** if the number of tokens in each place doesn't exceed k for any marking reachable from m_0.

Bounded reaching matrix — матрица ограниченных контрадостижимостей.

Bounded tolerance graph — ограниченный толерантный граф.
See *Tolerance graph*.

Boundedness problem — проблема ограниченности.
See *Bounded Petri net*.

Branch of a tree relative to a vertex v — ветвь к вершине v.
See *Centroid*.

Branch-weight centroid number — число центроида со взвешенными рёбрами.
See *Centroid*.

Breadth first search — поиск в ширину.
Given the adjacency lists $A(v)$ for each vertex v of a connected graph (directed or undirected), the following algorithm conducts a **breadth first search**. On completion of the search, each vertex has acquired a **breadth first index (BFI)** indicating the order in which the vertex was visited. The vertex u is visited first and $BFI(u) = 0$.

Bridge — мост.
1. A **bridge** of a *cycle* C is the shortest path in C joining nonconsecutive vertices of C which is shorter than both of the edges of C joining those vertices. Thus a *chord* is a bridge of length 1, and a graph is bridged iff every cycle of length at least 4 has a bridge.
2. A **bridge** of G is an edge whose removal disconnects G.

Bridged graph — граф с мостами.

A graph G is a **bridged graph** if each cycle C of length at least 4 contains two vertices whose distance from each other in G is strictly less than that in C. A graph is called **bridged** if every cycle of length at least 4 has a *bridge*. Observe that in a bridged graph every cycle of length 4 or 5 has a chord. Bridged graphs are in general not perfect, as the *wheel* W_7 shows.

Bridgeless graph — граф без мостов.

The same as *2-Connected graph*.

Broadcast digraph — орграф широковещания.

See *Broadcast graph*.

Broadcast graph — граф широковещания.

Let us first consider the full-duplex model (See *Broadcasting problem*.) Let G be a graph modeling an interconnection network. We will denote by $b(v)$ the broadcast time of v, that is the time to achieve broadcasting from a vertex v of G in the network. Moreover, $b(G)$, the broadcast time of G, is defined as follows:

$$b(G) = \max\{b(v) |\ v \in V(G)\}.$$

If we consider a complete graph of order n, K_n, it is not difficult to see that $b(K_n) = \lceil \log_2(n) \rceil$. Any graph G such that $b(G) = b(K_n) = \lceil \log_2(n) \rceil$ is called a **broadcast graph**. We call a **minimum broadcast graph** of order n any broadcast graph G having the minimum number of edges. This number is denoted by $B(n)$.

Similarly, a **broadcast digraph** is defined, using the half-duplex model.

Broadcasting problem — проблема широковещания.

The **broadcasting problem** is the problem of information dissemination described in a group of individuals connected by a communication network. In broadcasting, one node knows a piece of information and needs to transmit it to everyone else. This is achieved by placing communication calls over the communication lines of the network. It is assumed that a node can communicate with at most one of its neighbors at any given time, and communication between two nodes takes one unit of time. This model implies that we will deal with connected graphs without loops and multiple edges to model the communication network. Note also that, depending on their cases, we will either consider a half-duplex or a full-duplex model. In the letter, when communication takes place along a communication line, the information flows in both directions, while in the former only one direction is allowed. Hence, in the half-duplex model, we will deal with directed graphs, while we will consider undirected graphs in the full-duplex model.

See also *Gossiping problem.*

Brooks graph — граф Брукса.

A **Brooks graph** is a connected graph that is neither a complete graph nor an odd cycle.

Brooks' Theorem. The chromatic number of a Brooks graph does not exceed its maximum degree.

Brother of a vertex — брат вершины.

A **brother of a vertex** v, denoted $\beta(v)$, is a vertex in an oriented tree having the same father as v has.

Brother tree — братское дерево, HB-дерево.

A **brother tree** is a rooted oriented tree each of whose internal nodes has either one or two sons. Each unary node must have a binary brother. All external nodes are at the same *depth.* The number of internal binary nodes of a brother tree is called its size. Note that the number of external nodes of a brother tree is always by 1 greater than its size.

Another name is **HB-Tree**.

1-2 Brother tree — 1-2-братское дерево.

2-3 Brother tree — 2-3-братское дерево.

A **2-3 brother tree** is a *2-3 tree* satisfying an additional brother property: except for the sons of a binary root, each binary node has a ternary brother. Obviously, the class of 2-3 brother trees is properly contained in the class of 2-3 trees.

"Brute force" method — метод "грубой силы", перебор.

See *Exhaustive search.*

Bull — бык.

A **bull** is a (self complementary) graph with 5 vertices a, b, c, d, e and 5 edges $(a, b), (b, c), (c, d), (b, e), (c, e)$.

k**-Bunch** — k-пучок.

k**-Bunch isomorphic graph** — k-пучково изоморфные графы.

(L, Y)**-Bunch** — (L, Y)-связка.

Butterfly graph — граф-бабочка.

Let n be a positive integer. The n-level **butterfly graph** $\mathcal{B}(n)$ is a digraph whose vertices comprise the set $V_n = Z_n \times Z_2^n$. The subset $V_n^q = \{q\} \times Z_2^n$ of V_n $(0 \leq q < n)$ comprises the q^{th} level of $\mathcal{B}(n)$. The arcs of $\mathcal{B}(n)$ form directed butterflies (or, copies of the directed complete bipartite graph $K_{2,2}$) between consecutive levels of vertices, with wraparound in the sense that level 0 is identified with level n. Each butterfly connects each vertex $\langle q, \beta_0\beta_1 \cdots \beta_{q-1}\alpha\beta_{q+1} \cdots \beta_{n-1}\rangle$ on level q of $\mathcal{B}(n)$ ($q \in Z_n$; α and each β_i in Z_2) to both vertices

$$\langle q+1 \pmod n, \beta_0\beta_! \cdots \beta_{q-1}\gamma\beta_{q+1} \cdots \beta_{n-1}\rangle$$

on level $q+1 \pmod n$ of $\mathcal{B}(n)$, for $\gamma = 0, 1$.

C

Cactus — кактус, дерево Хусими.

A graph G is a **cactus** if every its edge is a part of at most one cycle in G. Cactus graphs are *outerplanar* since they cannot contain K_4 or $K_{2,3}$ as a *minor*. Cactus graphs have *treewidth* ≤ 2.

Cage number — (k, g)-клетка.

See (k, g)-*Cage*.

(k, g)**-Cage** — (k, g)-клетка.

For a given ordered pair of integers (k, g), with $k \geq 2$ and $g \geq 3$, a k-regular graph with the smallest cycle length, or *girth*, equal to g is said to be a (k, g)-graph. A (k, g)**-cage** is a (k, g)-graph having the least number, $f(k, g)$, of vertices. We call $f(k, g)$ the **cage number** of a (k, g)-graph. One readily observes that $(2, g)$-cages are cycles of length g, and $(k, 3)$-cages are complete graphs of order $k + 1$.

The unique (3,7)-cage known as the **McGee graph** is an example of a cage that is not transitive. It has 24 vertices and its automorphism group has order 32.

Call graph — граф [вызова] процедур.

Capacity of an arc — пропускная способность дуги.

See *Flow*.

Capacity of a cut-set — пропускная способность разреза.

The **capacity of a cut-set** (P, P') is defined to be the sum of the capacities of those edges incident from vertices in P and incident to vertices in P'.

Cardinal product — кардинальное произведение, прямое произведение.

The same as *Direct product*.

Cardinality constrained circuit problem — проблема цикла с ограниченной мощностью.

See *Weighted girth problem*.

Cartesian product of graphs — декартово произведение графов.

See*Product of two graphs*.

Cartesian sum of graphs — декартова сумма графов.

k**-Case term** — выражение k-выбора.

See *Large-block schema*.

Categorical product of graphs — категорийное произведение графов.

See *Product of two graphs*.

Caterpillar — гусеница.

1. A *tree* such that the removal of all *pendant vertices* or leaves (vertices with exactly one neighbor) yields a *path* is a **caterpillar**.

2. A **caterpillar** is a graph derived from a path by hanging any number of pendant vertices from vertices of the path.

3. A **caterpillar** C is a tree of order $n \geq 3$ whose *pruned tree* is a (possibly trivial) path.

Caterpillar-pure graph — гусенично-чистый граф.

A connected graph G is **caterpillar-pure** if each *spanning tree* of G is a *caterpillar*.

Case term — переключатель, слово выбора.

See *Large-block schema*.

Cayley graph — граф Кэли.

1. Let $Z_n = \{0, 1, \ldots, n-1\}$ be an additive abelian group of integers modulo n, and H be a subset of Z_n with $0 \notin H$. Then the **Cayley graph** $C_{Z_n,H}$ is an undirected graph with $V(C_{Z_n,H}) = Z_n$, and $E(C_{Z_n,H}) = \{(x, x+y) : \ x \in Z_n, y \in H$ and the addition is taken modulo $n\}$. The *adjacency matrix* of $C_{Z_n,H}$ is $n \times n$ *symmetric circulant* with entries 0 and 1.

2. Let Γ be a finite group and S be a symmetric generator set of Γ, i.e. $\langle S \rangle = \Gamma,\ s \in S \Rightarrow s^{-1} \in S$ and $1_\Gamma \notin S$. The **Cayley graph** $G_S(\Gamma)$ is defined as an undirected graph with its vertex set $V = \Gamma$ and edge set $E = \{(g,h)|\ g^{-1}h \in S\}$.

Center — центр.

See *Center vertex*.

Center of gravity of a graph — центр тяжести графа.

Center vertex — центральная вершина.

A vertex v in a connected graph G is called a **center (central) vertex** if $e(v) = rad(G)$. A subgraph induced by central vertices of G is called the **center** $C(G)$ of G. It was proved that the center of every graph H is contained in a block (a maximal 2-connected subgraph) of H.

p-**Center** — p-центр.

A p-**center** of $G = (V, E)$ is a set $C \subseteq V$ that realizes the *p-radius* of G.

The p-center problem is one of the main problems in facility location theory. For general graphs, the problem of finding p-centers is *NP*-hard. Polynomial algorithms for the p-center problem are known only for trees and almost-trees. The best known algorithms have time complexity $\mathcal{O}(|V|)$ for unweighted trees and $\mathcal{O}(|V| \cdot \log^2 |V|)$ for weighted trees.

Central distance — центральное расстояние.

The **central distance** $c(v)$ of v is the largest nonnegative integer n such that whenever $d(v, x) \leq n$ the vertex x is in the center of G.

Central fringe — центральная область.

Some central vertices of G are barely in $C(G)$, in the sense that they are adjacent to the vertices that are not central. The subgraph of $C(G)$ induced by those vertices with *central distance* 0 is called the **central fringe** of G and is denoted by $CF(G)$.

Central vertex — центральная вершина.

See *Center vertex.*

Centroid — центроид.

A **branch** of a tree T at a vertex v is a maximal subtree T_v of T, in which the degree of v is unity. Therefore, the number of branches at v is $deg(v)$. The **branch-weight centroid number** of a vertex v in a tree T, denoted by $bw(v)$ is the the maximum size of any branch at v. A vertex v of a tree T is a **centroid vertex** of T if v has minimum branch-weight centroid number. The **centroid** of T consists of its set of centroid vertices. Jordan (1869) has proved the following theorem.

Theorem. The centroid of a tree consists of either a single vertex or a pair of adjacent vertices.

See also *Slater number.*

Centroid sequence — центроидная последовательность.

Let T be a nontrivial tree; that is, a tree of order $n \geq 2$. A **centroid sequence** $A = \{a_1, a_2, \ldots, a_n\}$ of T is a sequence of the weights of the vertices of T, arranged in a non-increasing order ($a_1 \geq a_2 \geq \ldots \geq a_n$).

Centroid vertex — центроидная вершина.

See *Centroid.*

Centroidal vertex — центроидная вершина.

The same as *Centroid vertex.*

Chain — цепь, цепочка.

1. Given a poset $P = (X, <_P)$, a **chain** of P is a subset of X consisted of pairwise comparable elements.

2. See *Martynyuk schemata.*

Chain graph — цепной граф.

A *bipartite graph* $G = (P, Q, E)$ is called a **chain graph** if there is an ordering π of the vertices in P, $\pi : \{1, \ldots, |P|\} \to P$, such that

$$N(\pi(1)) \subseteq N(\pi(2)) \subseteq \cdots \subseteq N(\pi(|P|)).$$

Here $N(v)$ is a *neighborhood* of v. It is known that G is a chain graph iff it does not contain an independent pair of edges (an induced $2K_2$).

0-Chain of a graph — 0-цепь графа.

1-Chain of a graph — 1-цепь графа.

Characteristic number of a graph — характеристическое число графа.

Characteristic polynomial of a graph — характеристический полином графа.

Given a graph G, the polynomial $p_G(\lambda) = \det(\lambda I - A_G)$, where A_G is the *adjacency matrix* of G. This clearly does not depend on the labeling of vertices. The roots of the characteristic polynomial, i.e. the eigenvalues of A_G, are called the **eigenvalues** of the graph G. The **spectrum** of G is the set of solutions to $p(\lambda) = 0$ denoted by $S_p = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ with $\lambda_1 \geq \lambda_2 \geq \ldots, \lambda_n$. The first eigenvalue λ_1 is called the **index** or **spectral radius**.

Characteristic polynomial of Laplacian — характеристический полином лапласиана.

The **characteristic polynomial of Laplacian** $L(G)$ is $\mu(G, x) = \det(xI - L(G))$.

Chinese postman's problem — задача китайского почтальона.

Let $G = (V, E)$ be a connected undirected graph. A non-negative cost (or length) is associated with each edge of G. The **Chinese postman's problem** consists in determining a least-cost *traversal* of G.

Choice number — списочное хроматическое число, число выбора.

The same as *List chromatic number*.

Chomsky hierarchy — иерархия Хомского.

When Noam Chomsky first formalized grammars in 1956, he classified them into types now known as the Chomsky hierarchy. The difference between these types is that they have increasingly strict production rules and can express fewer formal languages. The **Chomsky hierarchy** consists of the following four types of grammars and languages.

(0) **Type-0 grammars** (or **unrestricted grammars**) include all formal *grammars*. They generate exactly all languages that can be

recognized by *Turing machine*. These languages are also known as the *recursively enumerable languages*.

(1) **Type-1 grammars** (**context-sensitive grammars**, or **CS-grammars**) generate the **context-sensitive languages** (or **CS-languages**). These grammars have rules of the form $\alpha A\beta \to \alpha\gamma\beta$, where A is a nonterminal and α, β, γ are strings of terminals and nonterminals. The strings α and β may be empty, but γ must be nonempty. The rule $S \to e$ is allowed if S does not appear in the right-hand side of any rule. The languages described by these grammars are exactly all languages that can be recognized by *linear bounded automata*.

(2) **Type-2 grammars** (**context-free grammars** or **CF-grammars**) generate the **context-free languages** (or **CF-languages**). These grammars contain rules of the form $A \to \alpha$, where A is a nonterminal and α is a string of terminals and nonterminals. These languages are exactly all languages that can be recognized by *non-deterministic pushdown automata*. Context-free languages are the theoretical basis for the syntax of most programming languages.

(3) **Type-3 grammars** (**regular grammars**) generate the **regular languages**. Such a grammar restricts its rules to a single nonterminal in the left-hand side and a right-hand side consisting of a single terminal, possibly followed (or preceded, but not both in the same grammar) by a single nonterminal. The rule $S \to e$ is also allowed here if S does not appear in the right-hand side of any rule. These are exactly all languages that can be recognized by *finite state automata*. Additionally, this family of formal languages can be obtained by *regular expressions*. Regular languages are commonly used to define the search patterns and lexical structure of programming languages.

Chomsky normal form — нормальная форма Хомского.

Choosability — списочное хроматическое число, выбираемость.

The same as *List chromatic number*.

k**-Choosable graph** — k-выбираемый граф.

1. A graph G is k**-choosable** if its *list chromatic number* satisfies the inequality $\chi_\ell(G) \leq k$.

2. A graph G is k**-choosable** if, whenever each vertex v is given a "list" (set) $L(v)$ of k colours, G has a *proper colouring* in which each vertex receives a colour from its own list.

m-Choosable graph with impropriety d — m-выбираемый граф с некорректностью d.

See *L-Coloring with impropriety d.*

Chord — хорда.

1. (For a subgraph G_1 of G) An edge $e \in E(G) - E(G_1)$ connecting two vertices of G_1 is called a **chord**.

2. (For a hypergraph) A **chord** of a *hypercycle* C is an edge e with $e_i \cap e_{i+1 \pmod k} \subseteq e$ for at least three indices i, $1 \le i \le k$.

Chordal graph — хордальный граф.

A graph that does not contain *chordless cycles* of length greater than three is called a **chordal** graph. This is equivalent to saying that the graph does not contain an *induced subgraph* isomorphic to C_n (i.e., a cycle of length n) for $n > 3$.

There are many ways to characterize chordal graphs. Although many of these characterizations are interesting and useful, it suffices to list only some of them. One of the most important tools is the concept of a *perfect elimination scheme*. The other way to define a chordal graph is to consider it as an *intersection graph* of a family of subtrees of a tree.

An important subclass of chordal graphs is the *interval graphs*.

Other names of a chordal graph are **Triangulated graph, Rigid circuit graph, Perfect elimination graph, Monotone transitive graph**.

Chordal bipartite graph — хордальный двудольный граф.

A graph G is a **chordal bipartite graph** if G is *bipartite* and any induced *cycle* in G is of length 4. Note that a **chordal bipartite graph** is not a *chordal graph*.

1-Chordal graph — 1-хордальный граф.

A chordal graph is called **1-chordal graph** if the maximum number of vertices common to two distinct cliques is 1.

c-Chordal graph — c-хордальный граф.

A graph is **c-chordal graph** if every induced cycle in it is of length at most c. *Triangulated graphs* are precisely 3-chordal graphs.

k-Chorded bigraph — k-хордовый двудольный граф.

A bigraph is called **k-chorded** if each of its non-*quad cycle* has at least k chords (so, for example, a 4-chorded bigraph has no 6-cycles, induced or not, and an ∞-chorded bigraph has no non-quad cycles, induced or not).

Chordless cycle — цикл без хорд.

A cycle such that two vertices of a cycle in G are adjacent if and only if the incident edge is also in the cycle. A c.c. with three vertices is called a *triangle* and a **chordless cycle** with four vertices is called a *square*.

Chromatic decomposition of a graph — хроматическое разложение графа.

Chromatic distance — хроматическое расстояние.

See *Colored distance*.

Chromatic function — хроматическая функция, хроматический полином.

The same as *Chromatic polynomial*.

Chromatic index — хроматический индекс, хроматический класс.

This is the least integer k for which the edges of G can be colored so that *adjacent* edges have different colores. We denote it by $q(G)$. Clearly, $q(G) = \chi(L(G))$. Here $L(G)$ is the *line graph* of G.

Chromatic number — хроматическое число.

This is the minimal number of colors (denoted by $\chi(G)$) for which there exists a *vertex coloring* of a graph G.

Chromatic polynomial — хроматический полином графа.

A **chromatic polynomial** $P_G(\lambda)$ of a graph G is the number of good λ-colorings of G ($\lambda = 0, 1, \ldots$). This is a polynomial in λ (for a fixed G) and so, its definition can be extended to all real (or complex) values of λ. Note that two λ-colorings differing in the labeling of colors are considered as different.

Chromatic status — хроматический статус.

See *Status of a vertex*.

k**-Chromatic graph** — k-хроматический граф.

A graph for which the *chromatic number* is equal to k is called k-**chromatic**.

k**-Chromatic hypergraph** — k-хроматический гиперграф.

n**-Chromatic number** — n-хроматическое число.

Chromatically equivalent graphs — хроматически эквивалентные графы.

Chromatically unique graph — хроматически единственный граф.

Church's thesis — тезис Черча.

See *Algorithm*.

Circuit — цикл.

1. The same as *Cycle*.

2. Given a graph G, a **circuit** is a walk $(x_1, e_1, \ldots, x_k, e_k, x_{k+1})$ such that $x_1, \ldots, x_k$ are distinct vertices, $e_1, \ldots, e_k$ are distinct edges and $x_1 = x_{k+1}$. If the graph is *simple*, we will denote it by $(x_1, \ldots, x_k)$.

3. Given a hypergraph, a **circuit** is a sequence $(x_1, E_1, \ldots, x_k, E_k)$, where $x_1, \ldots, x_k$ are distinct vertices, $E_1, \ldots, E_k$ are distinct edges and $x_i \in E_i$, $i = 1, \ldots, k$, $x_{i+1} \in E_i$, $i = 1, \ldots, k-1$, and $x_1 \in E_k$. Here k is the length of this circuit.

Circuit closed graph — ориентированно-циклически замкнутый граф, циклически замкнутый граф.

Circuit edge connected vertices — циклически-реберно связанные вершины.

Circuit isomorfic graph — циклически изоморфные графы.

Circuit of matroid — цикл матроида.

See *Matroid*.

Circuit rank — цикломатический ранг, цикломатическое число.

Circuitless graph — граф без циклов, лес.

Circulant graph — циркулянтный граф.

1. Let p be a positive integer and S be a subset of $\{1, 2, \ldots, p-1\}$, such that $i \in S$ implies $p - i \in S$. A **circulant graph** $G(p, S)$ has vertices $0, 1, \ldots, p-1$ and two vertices i and j are *adjacent* if and only if $i - j \in S$, where subtraction is carried out modulo p. The adjacency matrix of a circulant graph is a *symmetric circulant*.

2. (A circulant graph $G(n, d)$). The **circulant graph** $G(n, d)$ with $d \geq 2$ is defined as follows. The vertex set is $V = \{0, 1, 2, \ldots, n-1\}$, and the edge set is $E = \{(u, v) | \exists i, 0 \leq i \leq \lceil \log_d(n) \rceil - 1$, such that $u + d^i \equiv v \pmod n\}$.

3. (A circulant graph $G(cd^m, d)$). The **circulant graph** $G(cd^m, d)$ has cd^m vertices $(0 < c < d)$. $V = \{0, \ldots, cd^m - 1\}$ is the set of vertices, and $E = \{(v, w), v, w \in V / \exists i, 0 \leq i \leq \lceil \log_d cd^m \rceil - 1, v \pm d^i \equiv w \pmod{cd^m}\}$. An edge between v and $w = v \pm d^i$ will have the label d^i. It is easy to see that $G(cd^m, d)$ is a *Cauley graph* defined on an abelian group.

Circular-arc graph — граф дуг окружности.

A **circular-arc graph** is the intersection graph of a family of arcs on a circle; that is, for each vertex v_i there is a (closed) arc a_i of the circle such that v_i and v_j are adjacent if and only if $a_i \cap a_j \neq \emptyset$.

Circular chromatic number — цикловое хроматическое число.

See *Circular coloring of a graph.*

Circular clique number — цикловое кликовое число.

The **circular clique number** of a graph G, denoted by $\omega_c(G)$, is defined as the maximum quotient k/d such that the graph G_d^k ($k \geq 2d$) admits a homomorphism to G.

The graph G_d^k is defined as follows:

$V(G_d^k) = \{v_0, v_1, \ldots, v_{k-1}\}$,

$E(G_d^k) = \{v_i, v_j : d \leq |j - i| \leq k - d \bmod k\}$.

r**-Circular colorable graph** — r-циркулярный раскрашиваемый граф.

See *Circular coloring of a graph.*

Circular coloring of a graph — цикловая раскраска графа.

An r**-circular coloring of a graph** (r is a real number, $r \geq 2$) is a mapping $\psi : V(G) \to [0, r)$ such that $1 \leq |\psi(u) - \psi(v)| \leq r - 1$, whenever $uv \in E(G)$. A graph G is called r**-circular colorable** if it admits an r-circular coloring. The **circular chromatic number** of G, denoted by $\chi_c(G)$, is the smallest value for r such that G is r-circular colorable.

The concept of a circular coloring was first introduced in 1988 by Vince who first called it a **star coloring**, and it was given the current name by Zhu.

Circular perfect graph — цикловой совершенный граф.

A graph G is called **circular perfect** if $\omega_c(H) = \chi_c(H)$ for each induced subgraph H of G, where ω_c is the *circular clique number* and χ_c is the *circular chromatic number.*

The concept of a circular perfect graph was introduced by Zhu in 2004.

Circumference of a graph — окружение графа, окружность графа.

The length of a longest *cycle* of G (denoted $c(G)$).

Circumstance — окружение (графа).

Circumstance problem — проблема окружения.

$\mathcal{P}$ **and** $\mathcal{NP}$ **Classes** — классы $\mathcal{P}$ и $\mathcal{NP}$.

See *Complexity theory.*

Classification of Binary Trees — классификация бинарных деревьев.

The following classes of *binary trees* are considered:

BT — Binary Search Trees;

BST — Binary Split Trees;

FOBT — Frequency-Ordered Binary Search Trees;

GBST — Generalized Binary Split Trees;
MST — Median Split Trees;
OBST — Optimal Binary Split Trees;
OBT — Optimal Binary Search Trees;
OGBST — Optimal Generalized Binary Split Trees.

Claw — клешня.

A **claw** is a four-vertex star $K_{1,3}$.

Claw-free graph — граф без клешней.

A graph G is a **claw-free graph** if it contains no induced subgraph *isomorphic* to $K_{1,3}$.

Clique — клика.

This is a subgraph $G[W]$ induced by $W \subseteq V(G)$ such that every pair of vertices is adjacent. The **clique size** of a clique $G[W]$ is the number of vertices of W. The maximum clique size of a clique in G, $\omega(G)$, is called the **clique number** of G. The clique number $\Omega(G, w)$ of a *weighted graph* is defined as the minimum weight of a clique in G.

Clique cover — кликовое покрытие.

Let F be a family of cliques. By a **clique cover** we mean a *spanning subgraph* of G, each component of which is a member of F. With each element α of F we associate an indeterminate (or weight) w_α, and with each cover C of G we assosiate the weight $w(C) = \prod_{\alpha \in C} w_\alpha$.

Clique cover number, clique-covering number — число кликового покрытия.

The number $k(G)$ which is equal to the smallest number of cliques in the clique covering of $V(G)$ is called **clique cover number**.

Clique convergent — кликовая конвергенция.

See *Clique graph.*

Clique divergent — кликовая дивергенция.

See *Clique graph.*

Clique-good graph — кликово-хороший граф. See *Clique-transversal.*

Clique graph — граф клик.

The **clique graph** $k(G)$ is the *intersection graph* of the set of all cliques of G. The **iterated clique graphs** are defined recursively by $k^0(G) = G$ and $k^{n+1}(G) = k(k^n(G))$. A graph G is said to be **clique divergent** (or k**-divergent**) if

$$\lim_{n\to\infty} |V(k^n(G)| = \infty.$$

A graph G is said to be **k-convergent** if $k^n(G) \cong k^m(G)$ for some $n \neq m$; when $m = 0$, we say that G is **k-invariant**. G is **k-null** if $k^n(G)$ is trivial (one vertex) for some n (clearly, a special case of a k-covergent graph). It is easy to see that every graph is either k-convergent or k-divergent.

Clique-independence number — кликово-независимое число.

See *Clique-transversal*.

Clique-independent set —кликово-независимое множество.

See *Clique-transversal*.

Clique-transversal number — кликово-трансверсальное число.

See *Clique-transversal*.

Clique matrix — матрица клик.

Let $\mathcal{C}(G) = \{C_1, \ldots, C_k\}$ be the maximal cliques of a graph G. The **clique matrix** $C(G)$ of G is a $(0,1)$-matrix (c_{ij}) with entry

$$c_{ij} = \begin{cases} 1, \text{ if } v_i \in C_j \\ 0, \text{ otherwise} \end{cases}$$

Clique model — кликовая модель.

See *Tree model*.

Clique number — кликовое число.

See *Clique*.

Clique-partition — кликовое разбиение.

A *clique cover* Φ of G is a **clique-partition** of G if each vertex of G belongs to exactly one element of Φ.

Clique-perfect graph — кликово-совершенный граф.

See *Clique-transversal*.

Clique polynomial — кликовый полином.

Let G be a finite, simple graph and let F be a family of cliques. By a **clique cover** of G we mean a *spanning subgraph* of G, each component of which is a member of F. With each element α of F we associate an indeterminate (or weight) w_α and with each cover C of G we associate the weight

$$w(C) = \prod_{\alpha \in C} w_\alpha.$$

The **clique polynomial** of G is then:

$$K(G; \vec{w}) = \sum_C w(C),$$

where the sum is taken over all the covers C of G and $\vec{w}$ is the vector of indeterminates w_α. If, for all α, we set $w_\alpha = w$, then the resulting polynomial in the single variable w is called a **simple clique polynomial** of G.

Denote by $S(n,k)$ the Stirling numbers of the second kind. Then

$$K(K_n; w) = \sum_{k=0}^{n} S_{n,k} w^k.$$

Clique problem — проблема клики.

Clique separator — кликовый сепаратор.

In a connected graph G, $S \subseteq V$ is called a **clique separator** if S is a separator and $\langle S \rangle$ is a *clique*.

Clique size — размер клики.

See *Clique*.

Clique-transversal — кликовая трансверсаль.

A **clique-transversal** of a graph G is a subset of vertices that meets all the cliques. A **clique-independent set** is a collection of pairwise vertex disjoint cliques. The **clique-transversal number** and **clique-independence number** of G, denoted by $\tau_c(G)$ and $\alpha_c(G)$, are the sizes of a minimum clique-transversal and a maximum clique-independent set of G, respectively.

It is easy to see that $\tau_c(G) \geq \alpha_c(G)$ for any graph G. A graph G is **clique-perfect** if $\tau_c(H) = \alpha_c(H)$ for every induced subgraph H of G. If this equality holds for the graph G, we say that G is **clique-good**.

Clique tree — кликовое дерево.

Suppose G is any graph and T is a tree whose vertices — call them *nodes* to help avoid confusing them with the vertices of G — are precisely the *maxcliques* of G. For every $v \in V(G)$, let T_v denote a subgraph of T induced by those nodes that contain v. If every such T_v is connected — in other words, if every T_v is a subtree of T — then call T a **clique tree** for G.

Clique-width — кликовая ширина.

The **clique-width** of a graph G, denoted $cwd(G)$, is defined as a minimum number of labels needed to construct G, using four graph operations: creation of a new vertex v with a label i (denoted $i(v)$), disjoint union of two labeled graphs G and H (denoted $G \oplus H$), connecting vertices with specified labels i and j (denoted $\eta_{i,j}$) and

renaming labels (denoted ρ). Every graph can be defined by an algebraic expression using the four operations above. For instance, a graph consisted of 2 isolated vertices x and y can be defined by an expression $1(x) \oplus 1(y)$, and a graph consisted of two adjacent vertices x and y can be defined by an expression $\eta_{1,2}(1(x) \oplus 2(y))$.
With any graph G and an algebraic expression T which defines G we associate a tree (denoted by $tree(T)$) whose leaves are the vertices of G, and the internal nodes correspond to operations $\oplus$, η and ρ in T. Given a node a in $tree(T)$, we denoted by $tree(a, T)$ the subtree of $tree(T)$ rooted at a. The label of a vertex v of G at the node a of $tree(T)$ is defined as the label that v has immediately before the operation a is applied.

Closed hamiltonian neighbourhood — замкнутая гамильтонова окрестность.
See *Hamiltonian neighbourhood.*

Closed neighbourhood — замкнутая окрестность.
See *Neighbourhood.*

Closed semiring — замкнутое полукольцо.

Closed walk — замкнутый маршрут.
A **closed walk** in a mixed graph is a cycle which may visit vertices, edges and arcs multiple times.

Closure of graph — замыкание графа.

k**-Closure of a graph** — k-замыкание графа.
The k**-closure** $G_k(G)$ of a graph G is obtained from G by recursively joining pairs of non-adjacent vertices whose degree-sum is at least k until no such pair remains. It is known that if $G_n(G)$ is *complete*, then G contains a *Hamiltonian cycle.* The k-closure of a graph can be computed in $\mathcal{O}(n^3)$ time in the worst case.

Cluster — кластер.
See *Graph clustering problem.*

Clustered graph — кластерный граф.
See *Hierarchical graph.*

Clutter — клаттер.
See *Hypergraph.*

Coadjoint graphs — косопряженные графы.
See *Adjoint graph.*

Coadjoint pair — сопряженная пара.
A pair of operators (A, P) is a **coadjoint pair** if A is an *adjacency*

operator $A(G)$ for a graph G and $P = \sum_{v \in V(G)} \varphi(v) \otimes v$ is a permutation on $V(G)$ satisfying

$$A(G)^* = P^* A(G) P.$$

Moreover, the bijection φ on $V(G)$ satisfies $\varphi^2 = 1$, or $P^2 = 1$. In this case, P is called a **transposition symmetry**. Like this case, if a graph G has a **coadjoint pair** (A, S) such that S is a transposition symmetry, then G is called **strongly coadjoint**. Needless to say, undirected graphs are all strongly coadjoint and strongly coadjoint graphs are all coadjoint.

Coalescing of two rooted graphs — срастание двух корневых графов.

A **coalescing of two rooted graphs** (G, u) and (H, v), denoted $G.H$, is defined by Schwenk as a graph obtained by identifying the two roots so that $u = v$ becomes a *cut-vertex* of $G.H$.

Coarseness — крупность, зернистость, шероховатость.

Cobase of a matroid — кобаза матроида.

See *Dual matroid.*

Coboundary of a graph — кограница графа.

Coboundary operator — кограничный оператор.

Cochromatic number — число кохроматическое.

Cocircuit of a graph — коцикл графа.

See *Edge cut.*

Cocircuit of a matroid — коцикл матроида.

See *Dual matroid.*

Cocomparability graph — граф косравнимости.

Graph which is the *complement* of a *comparability graph* is called a **cocomparability graph**.

The class of cocomparability graphs consists of *perfect graphs* and contains the property set of all *cographs, permutation graphs* and *interval graphs.*

Cocomparability number — число косравнимости.

The **cocomparability number** of a graph G, denoted $ccp(G)$, is the smallest integer k such that G admits a k-CCPO (a k-cocomparability ordering). Note that $ccp(G) = 1$ if and only if $\bar{G}$ (the complement of G) is a comparability graph.

Cocomparability ordering — косравнимое упорядочение.

A graph G has a **cocomparability ordering** if there exists a linear order $<$ on the set of its vertices such that for every choice of vertices

u, v, w the following property holds

$$u < v < w \wedge (u,w) \in E \Rightarrow (u,v) \in E \vee (v,w) \in E.$$

A graph is a cocomparability graph if it admits a cocomparability ordering.

k-Cocomparability ordering — k-косравнимое упорядочение.

Let $G = (V, E)$ be a graph and k a positive number. A **k-cocomparability ordering** (or **k-CCPO**) of G is an ordering of its vertices such that for every choice of vertices u, v, w we have the following:

$$u < v < w \wedge d(u,w) \leq k \Rightarrow d(u,v) \leq k \vee d(v,w) \leq k.$$

A graph G is called a **k-cocomparability graph** if it admits a k-CCPO.

Cocycle — коцикл.

Given a graph $G = (V, E)$ and a subset $W \subseteq V$ of its vertices, the set of edges in G linking a vertex of W to a vertex outside W is called a **cocycle**.

Cocycle basis — базис коциклов.

Cocyclic matrix — матрица коциклов.

Cocycle vector — вектор-коцикл.

Cocyclic rank of a graph — коциклический ранг графа, коцикломатическое число.

Code of a tree — код дерева.

t-Code (in a graph) — t-код (в графе).

A set $C \subseteq V(G)$ is a **t-code** in G if $d(u,v) \geq 2t+1$ for any two distinct vertices $u, v \in C$; t-codes are known as *$2t$-packings*. In addition, C is called a **t-perfect code** if for any $u \in V(G)$ there is exactly one $v \in C$ such that $d(u,v) \leq t$; 1-perfect codes are also called **efficient dominating sets**.

A set $C \subseteq V(G)$ is a 1-perfect code if and only if the *closed neigbourhoods* of its elements form a partition of $V(G)$.

Codependent set of a matroid — козависимое множество матроида.

Codiameter — кодиаметр.

Let $u, v \in V(G)$ be any two distinct vertices. We denote by $p(u,v)$ the length of the longest path connecting u and v. The **codiameter** of G, denoted by d^*, is defined to be $\min\{p(u,v) | \; u, v \in V(G)\}$. A graph G of order n is said to be **Hamilton-connected** if $d^*(G) = n - 1$, i.e. every two distinct vertices are joined by a *Hamiltonian path*.

Codistance — корасстояние (между вершинами графа).

Let x, y be distinct vertices of a graph G. We define the **codistance** $d^*_G(x, y)$ between x and y to be the maximum length of an (x, y)-path in G.

See also *Codiameter*.

Cograph — кограф.

1. G is a **cograph** if G is the *comparability graph* of a *series-parallel poset*. The class of cographs should not be confused with the class of series-parallel graphs.

The following recursive definition describes also the cographs:

(1) a one-vertex graph is a cograph;.

(2) if $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are cographs, then $G = (V_1 \cup V_2, E_1 \cup E_2)$ is also a cograph;

(3) if $G = (V, E)$ is a cograph, then $\bar{G} = (V, \bar{E})$ is also a cograph;

(4) there are no other cographs.

2. A **cograph** is a graph without P_4.

Cographic matroid — кографический матроид.

See *Graphic matroid.*

Coindependent set of a matroid — конезависимое множество матроида.

Collapsible graph — разборный граф, складной граф

1. A *cf-graph* G is called a **collapsible** one if it can be transformed to a trivial one upon repeated application of transformations T_1 and T_2 desribed below.

Let (n, n) be an arc of G. The transformation T_1 is removal of this edge.

Let n_2 not be the initial node and have a single predecessor, n_1. The transformation T_2 is the replacement of n_1, n_2 and (n_1, n_2) by a single node n. The predecessors of n_1 become the predecessors of n. The successors of n_1 or n_2 become the successors of n. There is an arc (n, n) if and only if there was formerly an edge (n_2, n_1) or (n_1, n_1).

2. A graph H is called **collapsible** if for every even subset $S \subseteq V(H)$, there is a subgraph T of H such that $H - E(T)$ is connected and the set of odd degree vertices of T is S.

Color graph of a group — цветной граф группы.

Color requirement of a vertex — цветовое требование к вершине.

See *Multi-coloring.*

t-Color-critical — t-цвето-критическиий.

If $\chi(G) = t$ and $\chi(H) < t$ for every proper subgraph H of G, then G is said to be **t-color-critical**.

G is **t-immersion-critical** if $\chi(G) = t$ and $\chi(H) < t$ whenever H is properly immersed in G.

Coloration — раскраска.

The same as *Coloring*.

Colored distance — раскрашенное расстояние.

The **colored distance** of a colored graph G is introduced as the sum of distances between all unordered pairs of vertices having different colors. The **chromatic distance** of G, denoted by $d_{ind}(G)$, is the minimum colored distance of a proper coloring of the vertex set.

Colored graph — раскрашенный граф.

Colored multigraph — раскрашенный мультиграф.

Coloring, colouring — раскраска.

1. Let G be a graph and S be a set of colors. A **coloring** of G is a mapping $f : V(G) \to S$ from the vertex set $V(G)$ into S such that no two adjacent vertices have the same color, i.e., $f(x) \neq f(y)$ whenever x and y are adjacent. This **C.** is called a **proper** (**valid**, **legitimate**, **good**) coloring.

f is called **k-coloring** of G if f is a coloring of G into k colors, i.e. $k = |\{f(p) : p \in V(G)\}|$.

It is NP-complete to decide if a given graph G admits a k-coloring for a given k except for the cases $k = 1$ and $k = 2$. Graph coloring remains NP-complete even on planar graphs of degree at most 4.

2. See *Large-block schema*.

Coloring number — число раскрашивания.

The **coloring number** of G, denoted $col(G)$, is defined as the largest integer k such that G has a subgraph of minimum degree $k - 1$.

k-Colorable graph — k-раскрашиваемый граф.

This is a graph which has a good k-coloring.

k-Colorable hypergraph — k-раскрашиваемый гиперграф.

k-Colorable map — k-раскрашиваемая карта.

k-Colored graph — k-раскрашенный граф.

Let k be an integer. A **k-colored graph** is a graph $G = (V, E)$ together with a vertex *coloring* which is a mapping $f : V \to S$ such that

(1) each vertex is colored with one of the colors such that no two

adjacent vertices have the same color (i.e., $f(x) \neq f(y)$ whenever x and y are adjacent),

(2) $|S| = k$ and each color is used at least once (i.e., f is surjective).

k-Colored hypergraph — k-цветной гиперграф.

k-Coloring — k-раскраска.

See *Coloring*.

(k,d)-Coloring — (k,d)-раскраска.

Let k and d be positive integers such that $k \geq 2d$. A **(k,d)-coloring** of a graph $G = (V,E)$ is a mapping $c : V \to Z_k = \{0,1,\ldots,k-1\}$ such that, for each edge $(u,v) \in E$, $|c(u) - c(v)|_k \geq d$, where $|x|_k = \min\{|x|, k - |x|\}$. This generalizes a usual notion of a *k-coloring*: an ordinary k-coloring of G is just a $(k,1)$-coloring.

L-Coloring with impropriety d — L-раскраска с некорректностью d.

A **list assignment** of G is a function L which assigns a list of colors $L(v)$ to each vertex $v \in V(G)$. An **L-coloring with impropriety** d, or simply **$(L,d)^*$-coloring**, is a mapping λ which assigns to each vertex $v \in V(G)$ a color $\lambda(v)$ from $L(v)$ so that v has at most d neighbours colored with $\lambda(v)$. For $m \in N$, the graph is **m-choosable with impropriety** d, or simply **$(m,d)^*$-choosable**, if there exists an $(L,d)^*$-coloring for every list assignment L with $|L(v)| \geq m$ for each $v \in V(G)$. For an improper coloring of a graph G, the number of neighbours of $v \in V(G)$ colored with the same color as itself is called the **impropriety** of v and is denoted by $im(v)$. The smallest m for which G is $(m,d)^*$-choosable is called the **d-improper list chromatic number** of G and is denoted by $\chi_l^*(G,d)$.

Coloured class — цветной класс.

Coloured Petri net — раскрашенная (или цветная) сеть Петри.

See *High-level Petri nets*.

3-Combination problem — задача о трехмерном сочетании.

Combinatorial dual graph, combinatorically dual graph — комбинаторно двойственный граф.

A graph G is a **combinatorial dual graph** to a graph G^*, if there exists a one-to-one mapping $e : E(G) \to E(G^*)$ of the edge set of G onto the edge set of G^* such that C is a *circuit* of G if and only if $e(C)$ is a *cocircuit* of G^*.

Combinatorial Laplacian — комбинаторный лапласиан.

Let $G = (V,E)$ be a locally finite graph without isolated vertices. Let $L^2(G)$ be the space of all R-valued functions on $V(G)$. The

combinatorial Laplacian $\Delta_G : L^2(G) \to L^2(G)$ of G is given by

$$\Delta_G f(x) = f(x) - \frac{1}{m_G(x)} \sum_{y \sim_G x} f(y)$$

for any $f \in L^2(G)$, $x \in V(G)$. Here $m_G(x)$ is the degree of a vertex $x \in V(G)$ and we write $y \sim_G x$ if the vertices y and x are adjacent in G. Inasmuch as G is a discrete analogue of a Riemannian manifold, Δ_G is a discrete analogue of the ordinary Laplace–Beltrami operator in Riemannian geometry. This analogy has been widely exploited both in the development of a harmonic analysis on graphs and within the spectral geometry of graphs.

Comet — комета.

Common ancestor — общий предок.

See *Directed tree.*

Common minimal dominating graph — общий минимальный доминирующий граф.

A **common minimal dominating graph** of G is defined as the graph having the same vertex set as G with two vertices adjacent if there is a minimal dominating set containing both vertices.

Common receiver — общий приемник.

See *Directed graph.*

Common server — общий сервер.

See *Directed graph.*

Compact — компакт.

See *Separator.*

Compact closed class of graphs — компактно замкнутый класс графов.

A class $\mathcal{C}$ of graphs is said to be **compact closed** if, whenever a graph G is such that each of its finite subgraphs is contained in a finite induced subgraph of G which belongs to the class $\mathcal{C}$, the graph G itself belongs to $\mathcal{C}$. We will say that a class $\mathcal{C}$ of graphs is **dually compact closed** if, for every infinite $G \in \mathcal{C}$, each finite subgraph of G is contained in a finite induced subgraph of G which belongs to $\mathcal{C}$.

Comparability graph — граф сравнимости.

Let $G = (V, E)$ be an undirected graph and let F be an *orientation* of its edges (i.e. (V, F) is the resulting oriented graph). F is called a **transitive orientation** of G if the following properties hold:

$$F \cap F^{-1} = \emptyset \text{ and } F + F^{-1} = E \text{ and } F^2 \subseteq F,$$

where $F^2 = \{(a,c)|\ \exists_{b\in V}(a,b) \in F \wedge (b,c) \in F\}$.

A graph G which admits a **transitive orientation** of its edges is called a **comparability graph**.

If graph G is a **comparability** graph, then this also holds for every induced subgraph of G.

The other name is a **Transitively orientable graph**.

Comparable vertices — сравнимые вершины.

Competition graph (of a tournament) — граф конкуренции.

Let $O(x)$ be the set of vertices that x beats. Given a tournament T with a vertex set $V(T)$, the **competition graph** of T, denoted $C(T)$, is the graph on $V(T)$ with an edge between vertices x and y if and only if $O(x) \cap O(y) \neq \emptyset$. It is known that the *complement* of the **competition graph** is the *domination graph*.

For an arbitrary acyclic digraph D, the **competition graph** of D has the same set of vertices as D and an edge between vertices u and v if and only if there is a vertex x in D such that (u,x) and (v,x) are arcs of D. The **competition number** of a graph G, denoted by $k(G)$, is the smallest number k such that G with k isolated vertices is a competition graph of an acyclic digraph.

The **competition-common enemy graph** of D has the same set of vertices as D and an edge between vertices u and v if and only if there are vertices w and x in D such that (w,u), (w,v), (u,x) and (v,x) are arcs in D. The **double competition number** of a graph G, denoted by $dk(G)$, is the smallest number k such that G with k isolated vertices is a competition-common enemy graph of an acyclic digraph. It is known that $dk(G) \leq k(G) + 1$ for any graph G.

Competition number — число конкуренции.

See *Competition graph.*

p-Competition graph — граф p-конкуренции.

See *Generalized competition graphs.*

Competition-common enemy graph — граф животных с общей конкуренцией.

See *Competition graph.*

Compilation problem — проблема компиляции.

Complement of a graph, complementary graph — дополнение графа.

The **complementary graph** $\bar{G} = (V, \bar{E})$ of a graph $G = (V, E)$ is defined by $\bar{E} = \{(x,y) : x, y \in V \text{ and } x \neq y \text{ and } (x,y) \notin E\}$.

Given a simple digraph G, the simple digraph $\bar{G}$ is defined by

$$V(\bar{G}) = V(G),$$
$$E(\bar{G}) = V(G) \times V(G) - E(G).$$

Complement-reducible graph — дополнительно сводимый граф.

Complement-reducible graph can be characterized as a graph with no induced paths of length three.

Complete bipartite graph — полный двудольный граф.

A bipartite graph $G = (X, Y, E)$, denoted $K_{m,n}$, in which every vertex of X is adjacent to every vertex of Y. Here $m = |X|$ and $n = |Y|$. The *treewidth* of **complete bipartite graph** is $\min(m, n)$.

Complete coloring — полная раскраска.

Complete directed graph — полный орграф.

Complete graph — полный граф.

A graph for which every pair of distinct vertices defines an edge is called a **complete graph**. The complete graph with n vertices is denoted by K_n.

Complete homomorphism of order n — полный порядка n гомоморфизм.

Complete multipartite graph — полный многодольный граф.

By **complete multipartite graph** we mean any graph whose complement is a disjoint union of at least three complete graphs.

Complete k-partite graph — полный k-дольный граф.

Complete product — полное произведение.

The **complete product** $G_1 \nabla G_2$ of graphs G_1 and G_2 is the graph obtained from the *union* of graphs $G_1 \cup G_2$ by joining every vertex of G_1 with every vertex of G_2.

Complete rotation — полное вращение [орграфа].

Let $G = Cay(\Gamma, S)$ be a Cayley digraph with $|S| = d$. (See also *Associated Cayley digraph*).

A **complete rotation** of G is a group automorphism ω of Γ such that for some ordering $s_0, s_1, \ldots, s_{d-1}$ of the elements of S, we have $\omega(s_i) = s_{i+1}$ for every $t \in Z$.

Clearly, a rotation is a graph automorphism. A Cayley digraph with a complete rotation is called a **rotational Cayley digraph**.

Complete set of graph invariants — полный набор инвариантов.

Complete system of equivalent transformations — полная система эквивалентных преобразований.

See *Yanov schemata.*

Complete tree — завершенное дерево.

Complete k-uniform hypergraph — полный k-униформный гиперграф.

***NP*-complete problem** — *NP*-полная задача.

See *Complexity theory*.

Completed Husimi tree — завершёное дерево Хусими.

See *Block graph*.

Completely triangular graph — полностью триангулированный граф.

See *Triangular vertex*.

Complex windmill — ветрянная мельница.

See *Windmill*.

Complexity of RAM — сложность РАМ.

Complexity theory — теория сложности.

The theory of classifying problems based on how difficult they are to solve. A problem is assigned to the ***P*-problem (polynomial-time) class** if the number of steps needed to solve it is bounded by some power of the problem's size. A problem is assigned to the ***NP*-problem (nondeterministic polynomial-time) class** if if it is solvable in polynomial time by a *nondeterministic Turing machine*.

A problem is called **intractable** if it is not a P-problem. The class of *P*-problems is a subset of the class of *NP*-problems, but there also exist problems which are not *NP*.

The ***P* versus *NP* problem** is the determination of whether all *NP*-problems are actually P-problems. If $P \neq NP$, then the solution of *NP*-problems requires (in the worst case) an *exhaustive search*, while if they are, then asymptotically faster algorithms may exist. The answer is not currently known, but determination of the status of this question would have dramatic consequences for the potential speed with which many difficult and important problems could be solved.

A problem C is said to be ***NP*-hard** if every problem from *NP* is reducible to C in polynomial time. A problem which is both *NP* and *NP*-hard is called an ***NP*-complete problem**. Examples of *NP*-complete problems include the *Hamiltonian cycle*, *traveling salesman problems*, *Hamiltonian path problem*, *subgraph isomorphism problem*, *clique problem*, *vertex cover problem*, *independent set problem*, *dominating set problem*, *graph coloring problem*.

Thus, if some *NP*-complete problem is a *P*-problem then $P{=}NP$, and,vice versa, if some problem from *NP*-problem class is *intractable*, then all *NP*-complete problems are also intractable.

***NP*-Complete language** —*NP*-полный язык.

***NP*-Complete problem** —*NP*-полная задача, *NP*-полная проблема.
See *Complexity theory.*

Component design method — метод построения компонент.

Component index — индекс компонент.

Component number — число компонент.
See *Component of a graph.*

Component of a graph — компонента графа.
A component H of G is **odd** (**even**) if $|V(H)|$ is odd (even). The **component number** of G is denoted by $c(G)$, and the **odd component number** of G is denoted by $o(G)$.

Composite hammock — составной гамак.
See *Hammock.*

Composition of graphs — граф-композиция.

Compound dependency graph — граф составной зависимости.

Compound graph — составной граф.

Computation — вычисление.
See *Algorithm.*

Concatenation — конкатенация, сцепление.
1. See *String.*
2. See *Formal language.*

Concatenation closure — итерация (языка).
See *Formal language.*

Condensation — конденсация, граф Герца.
See *Strongly connected component.*

Conditional connectivity — условная связность.

Cone over a graph — конус над графом.
The **cone over a graph** G is the graph $K_1 \nabla G$ obtained from G by adding a vertex adjacent to all vertices of G.

Conflict — конфликт.
See *Petri net.*

Confluent NCE graph grammar — конфлуентная графовая грамматика типа NCE.
An *NCE graph grammar* $\mathcal{G}$ is **confluent** (C-NCE) if for every graph G derivable (see *Derivation*) from the axiom of $\mathcal{G}$, all nonterminal nodes u, v in G, and all productions $(\phi(u), H, D)$, $(\phi(v), J, F)$ in $\mathcal{G}$

we have:

$$G[u/_D H][v/_F J] = G[v/_F J][u/_D H].$$

In the confluent graph grammar, the order in which the productions are applied is irrelevant for the resulting graph.

Conformal hypergraph — конформальный гиперграф.

A *hypergraph* $\mathcal{H}$ such that every *clique* C in a *2-section graph* $2EC(\mathcal{H})$ is contained in an edge $e \in \mathcal{H}$.

Conjunction of graphs — конъюнкция графов.

See *Product of two graphs.*

Connected component — компонента связности.

A **connected component** of a graph G is a maximal connected subgraph of G. Any two connected components of G are vertex-disjoint and each vertex (and edge) belongs to one of them. Their number is denoted by $c(G)$.

Connected component of a hypergraph — связная компонента гиперграфа.

Let $\mathcal{E} = (V, \{E_1, \ldots, E_m\})$ be a hypergraph. A sequence $(E_1, \ldots, E_k)$ of distinct hyperedges is a **path** of length k if for all i, $1 \leq i < m$, $E_i \cap E_{i+1} \neq \emptyset$. Two vertices $x \in E_1$, $y \in E_k$ are connected (by the path $(E_1, \ldots, E_k)$), and E_1 and E_k are also connected. A set of hyperedges is **connected** if every pair of hyperedges in the set is connected. A **connected component of a hypergraph** is a maximal connected set of hyperedges.

Connected domination number — число связного доминирования.

See *Dominating set.*

Connected dominating set — связное доминирующее множество.

See *Dominating set.*

Connected graph — связный граф.

A graph G is a **connected graph** if for all $u, v \in V(G)$, $u \neq v$, there is a *chain* $(v_1, \ldots, v_k)$ in G with $\{v_1, v_k\} = \{u, v\}$, (the chain connects u and v). Otherwise, the graph is called **disconnected**.

A graph $G = (V, E)$ is **maximum edge-connected** (in short, **max-**λ) if $\lambda = [2q/p]$, where $p = |V|$, $q = |E|$ and $\lambda = \lambda(G)$ is *edge-connectivity* of G. Note that the set of edges adjacent to a point u of degree λ is certainly a minimum edge-disconneting set. Similarly, G is **maximum point-connected** (in short, **max-**κ) if $\kappa = [2q/p]$, where $\kappa = \kappa(G)$ is the point-connectivity of G. Also, the set of points adjacent to u of degree κ is certainly a minimum point-disconnecting set. In this context, such an edge or a point set is called trivial. A graph G is called **super edge-connected** if G is max-λ and every minimum edge-disconnecting set is trivial. Analogously, G is **super point connected** if G is max-κ and every minimum point-disconnecting set is trivial.

Connected hierarchical graph — связанный иерархический граф.

See *Hierarchical graph.*

Connected set of vertices — связное множество вершин.

Connected vertices — связные вершины.

2-Connected graph — двусвязный граф.

See *Articulation point.*

k**-Connected component** — компонента k-связности.

k**-Connected graph** — k-связный граф.

A graph G is k**-connected** if there exist k internally node-disjoint chains between every pair of distinct nodes in G. A k-connected graph G is **minimal** if for any edge $e \in E$, $G-e$ is not k-connected. Usually, 2-connected graphs are called *biconnected graphs* and 3-connected graphs are called **triconnected graphs**.

k**-Connected vertices** — k-связные вершины.

See *Connectivity.*

P_4**-Connected graph** — P_4-связный граф.

A graph $G = (V, E)$ is P_4**-connected graph** if for every partition of V into nonempty disjoint sets V_1 and V_2, some chordless path on four vertices and three edges (i.e. P_4) contains vertices from both V_1 and V_2.

The concept of P_4-connectedness leads to a structure theorem for arbitrary graphs in terms of P_4-connected components and suggests, in a quite natural way, a tree representation unique up to *isomorphism.* The leaves of the resulting tree are P_4-connected components and weak vertices, that is, vertices belonging to no P_4-connected component. The structure theorem and the corresponding tree representation provide tools for the study of graphs with a simple P_4-structure, such as P_4-reducible, P_4-extendible, P_4-sparse graphs.

Connected hypergraph — связный гиперграф.

A hypergraph such that it is not representable as $\mathcal{H}_1 \cup \mathcal{H}_2$, where $\mathcal{H}_1, \mathcal{H}_2$ are vertex-disjoint non-empty hypergraphs is called **connected**. Note that if $\emptyset \in E(\mathcal{H})$, $\mathcal{H}$ is not connected.

Connected to relation — отношение связности "к", (достижимость) в гиперграфе.

The relation **connected to**, which is denoted by the symbol $\succ$, is defined for a given subset R of nodes and a node y; we say that R is **connected to** y and write $R \succ y$ if and only if a *directed hyperpath* exsists in a hypergraph from R to the node y.

It is easy to check that the relation $\succ$ satisfies the following set of **connectivity axioms**:

(1) $y \in R \subseteq V \Rightarrow R \succ y$ (reflexivity);

(2) $R \succ y$ and $Z \subseteq V \Rightarrow (R \cup Z) \succ y$ (augmentation);

(3) $R \succ y,\ \forall y \in Y,$ and $Y \succ z \Rightarrow R \succ z$ (transitivity).

H**-Connected graphs** — H-связные графы.

See *H-distance.*

Connective index — индекс связности (вершины).

Connectivity — связность.

The best known and most useful of the measures of how well a graph is conected is the **connectivity**, defined to be the minimum number of vertices in a set whose deletion results in a disconnected or trivial graph.

Two vertices u and v in a graph G are said to be k-connected if there are k or more pairwise internally disjoint paths between them. The (u, v)-connectivity of G, denoted $k_G(u, v)$, is defined to be the maximum value of k for which u and v are k-connected.

If the order of G is p, then the **average connectivity** of G, denoted $\bar{k}(G)$, is defined to be

$$k_G = \frac{\sum_{u,v} k_G(u,v)}{\binom{p}{2}}.$$

(The expression $\sum_{u,v} k_G(u, v)$ is sometimes referred to as the **total connectivity** of G.) In contrast to the connectivity, which gives the smallest number of vertices whose failure disconnects some pair of vertieces, the average connectivity gives the expected number of vertices that must fail in order to disconnect an arbitrary pair of nonadjacent vertices.

See *Edge connectivity, Vertex connectivity.*

Connectivity function — функция связности.

Connectivity matrix — матрица смежности.

Connectivity axioms — аксиомы связности.

See *Connected to relation.*

Consecutive adjacent graph — последовательный граф смежности.

See *Semigraph.*

Consecutive labeling — последовательная разметка.

A labeling is said to be **consecutive** if, for every number s, the weights of all s-sided faces constitute a set of consecutive integers.

Constructible graph — конструируемый граф.

Graph G is **constructible** if it can be built vertex-by-vertex so that a vertex x can be added to the currently constructed induced subgraph G_x of G if there exists a vertex y of G_x which is adjacent in G to x and to all neighbors of x belonging to G_x.

A graph G is said to be **constructible** if there is a well-order $\leq$ on $V(G)$ such that every vertex x which is not the smallest element of $(V(G), \leq)$ is *dominated* by some vertex $y \neq x$ in the subgraph of G induced by the set $\{z \in V(G) : z \leq x\}$. The well-order $\leq$ on $V(G)$, and the enumeration of the vertices of G induced by $\leq$, will be called a **constructing order** and a **constructing enumeration**, respectively.

See also *Dismantlable graph.*

Constructing enumeration — конструктивная нумерация.

See *Constructible graph.*

Constructing order — конструктивный порядок.

See *Constructible graph.*

Containment graph — граф содержимого.

See *Intersection graph.*

Context-free grammar — контекстно-свободная грамматика.

See *Chomsky hierarchy.*

Context-free language — контекстно-свободный язык.

See *Chomsky hierarchy.*

Context-sensitive grammar — контекстно-зависимая грамматика, неукорачивающаяся грамматика.

See *Chomsky hierarchy.*

Context-sensitive language — контекстно-зависимый язык.

See *Chomsky hierarchy.*

Contrabasis — антибаза.

Contractable edge — стягиваемое ребро.

An edge e in a 3-connected graph G is **contractable** if the contraction

G/e is still 3-connected.

Contracted visibility graph — стягиваемый граф видимости.

See *Visibility graph*.

Contracting edge, contraction of an edge — стягивание ребра.

If G is a graph and (u, v) is an edge of G, the graph obtained by **contracting edge** (u, v) is the graph obtained from $G[V \setminus \{u, v\}]$ by adding a new vertex z and adding edges (z, w) for all $w \in (N(u) \cup N(v)) \setminus \{u, v\}$, where $N(u)$ is the *neighborhood* of u. **Contracting a subgraph** means contracting all edges of it (the order in which the contraction is made is irrelevant). Note that multiple edges may appear.

Contraction of an even pair — стягивание четной пары.

The **contraction of an even pair** (u, v) is an operation that consists in replacing the two vertices u, v by a unique vertex t whose *neighborhood* is $N_G(u) \cup N_G(v) - \{u, v\}$: the resulting graph is denoted by G_{uv}. Contracting an even pair preserves the *chromatic number* and *clique number*. Thus, successive contraction of even pairs could possibly be used to reduce a given graph G to a smaller, simpler graph with the same parameters χ and ω. In the case where the final graph is a *clique*, G is called **even contractile**; whenever this reduction can be performed not only for the graph G itself, but also for every one of its induced subgraphs, G is called **perfectly contractile**.

Contraction of a graph — стягивание графа.

Contrafunctional graph — контрафункциональный граф.

Control dependence — зависимость по управлению.

Control flow graph — управляющий граф, управляющий граф, граф потока управления, граф переходов.

A program can be represented as a directed graph (called **control flow graph** or **cf-graph**), in which vertices (or **nodes**) correspond to program statements and arcs reflect possible transfers of control between the corresponding statements. There are **initial** (or **entry**) and **terminal** (or **exit**) nodes in the graph that correspond to input and output statements of the program. If there is an arc (p, q), then p is called a **predecessor** of q and q is called a **successor** of p.

It is assumed that a control flow graph G is a **proper** one, i.e. G has a single initial node without predecessors and a single terminal node without successors, and every node of G lies on at least one of the paths from the initial node to the terminal node.

k-**Convergent** — k-ковергенция.

See *Clique graph.*

Converse digraph — обратный орграф.

Let D be a digraph. The **converse** of D, denoted D', is a digraph obtained from D by reversing the direction of each arc of D. A digraph is called **self-converse** if it is *isomorphic* to its *converse.*

Convex bipartite graph — выпуклый двудольный граф.

Let $G = (X, Y, E)$ be a *bipartite graph.* An ordering of X has the **adjacency property** if for each $y \in Y$ the neighbors of y in X are consecutive in the ordering of X. A bipartite graph G is called **convex bipartite** if there is an ordering of X or of Y that has the adjacency property. A bipartite graph $G = (X, Y, E)$ is **biconvex** if there is an ordering of X and Y with the adjacency property. Convex graphs contain the *bipartite permutation graphs.*

Convex dominating set — выпуклое доминирующее множество.

A set $X \subseteq V(G)$ is **convex** in G if vertices from all $(a-b)$-geodesics belong to X for any two vertices $a, b \in X$. A set X is a **convex dominating set** if it is convex and dominating. The **convex domination number** $\gamma_{con}(G)$ of a graph G is the minimum cardinality of a convex dominating set in G.

Convex domination number — число выпуклого доминирования. See *Convex dominating set.*

Convex linear graph — выпуклый прямолинейный граф.

Convex set in G — выпуклое множество в графе G.

See *Convex dominating set.*

m-**Convex set in** G — m-выпуклое множество в графе G.

A path P in G is called m-path if the graph induced by the vertex set $V(P)$ of P is P. A subset C of $V(G)$ is said to be m-**convex set** if, for every pair of vertices $x, y \in C$, the vertex set of every $x - y$ m-path is contained in C. The cardinality of a maximal proper m-convex set in G is the m-**convexity number** of G.

m-**Convexity number** — число m-выпуклости.

See *m-Convex set in G.*

Coordinated graph — координатный граф.

A graph G is **coordinated** if the cardinality of a maximum set of *cliques* of H with a common vertex is equal to the cardinality of a minimum partition of the cliques of H into *clique-independent sets*, for every induced subgraph H of G.

Corank function of a matroid — коранговая функция матроида.

E_k**-Cordial graph** — E_k-сердечный граф.

Let f be an edge labelling of a graph $G = (V, E)$, such that

$$f : E(G) \to \{0, 1, \ldots, k-1\}$$

and the induced vertex labelling is given as

$$f(v) = \sum_{\forall u} f(u, v) \pmod{k},$$

where $v \in V$ and $\{u, v\} \in E$. f is called an E_k**-cordial labelling** of G, if the following conditions are satisfied for $i, j = 0, 1, \ldots, k-1$, $i \neq j$.

$$1)\ |e_f(i) - e_f(j)| \leq 1,$$

$$2)\ |v_f(i) - v_f(j)| \leq 1,$$

where $e_f(i)$, $e_f(j)$ denote the number of edges, and $v_f(i)$, $v_f(j)$ denote the number of vertices labelled with i's and j's respectively. The graph G is called E_k**-cordial** if it admits an E_k-cordial labelling. See also *Edge-graceful graph.*

Core — ядро.

1. Given a graph G, a **core** of G is a subgraph C of G such that G is *homomorphic* to C, but C fails to be homomorphic to any proper subgraph of G. This notion of a core is due to Hell and Nešetřil (1992). A graph G is a **core** if G is a core for itself. It is known, that in general, the problem of deciding whether G is a core is NP-complete, but there exists a polinomial time algorithm to decide if G is a core for the particular case when G has the *independence number* $\alpha(G) \leq 2$. Finally, it is known that "almost all graphs" are cores.

2. Let $\Omega(G)$ denote the family of all maximum stable sets. Then the **core** is defined as $core(G) = \cap\{S : S \in \Omega(G)\}$. Thus, the core is the set of vertices belonging to all maximum stable sets.

Corona — корона.

1. The **corona** $coro(G)$ of a graph G is a graph obtained from G by adding a *pendant edge* to each vertex of G. See also *Crown of graphs.*

2. Let $\Omega(G)$ denote the family of all maximum stable sets of the graph G. We define $corona(G) = \cup\{S : S \in \Omega(G)\}$ as the set of vertices belonging to some maximum stable sets of G.

3. $G \circ H$ is called the **corona** of graphs, if it is obtained from the disjoint union of G and n copies of H (where $n = |V(G)|$) by joining a vertex x_i of G with every vertex from i-th copy of H, for each $i = 1, 2, \ldots, n$.

Let k be a fixed integer, $k \geq 1$, k**-corona** $kG \circ H$ is a graph obtained from k copies of G and $|V(G)|$ copies of H with appropriate edges between each vertex x_i^j of the copy G^j and all vertices of the copy of H_i.

The **2-corona** of a graph H is a graph of order $3|V(H)|$ obtained from H by attaching a path of length 2 to each vertex so that the attached paths are vertex disjoint.

Cospectral graphs — коспектральные графы.

Cotree, co-tree — кодерево (графа).

Let T be a *spanning tree* of G. A **cotree** of G is a graph induced by edges that do not belong to T. Any edge of a cotree is called a *chord* of the spanning-tree.

k**-Cover of a (hyper)graph** — k-покрытие (вершинное) графа (гиперграфа).

This is a collection of points such that each edge contains at least k of them. 1-cover is simply a vertex (point) cover. A k-cover can also be regarded as a mapping $t: V(G) \to \{0, 1, \ldots\}$ such that $\sum_{x \in E} t(x) \geq k$ for each edge E.

(t, i, j)**-Cover** — (t, i, j)-покрытие.

Let $G = (V(G), E(G))$ be a graph. The set S of vertices is called a (t, i, j)**-cover** if every element of S belongs to exactly i balls of radius t centered at elements of S and every element of $V \setminus S$ belongs to exactly j balls of radius t centered at elements of S.

Covering cycle — покрывающий цикл.

The same as *Dominating cycle*.

Covering graph — покрывающий граф, накрывающий граф.

Covering set of vertices — покрывающее множество вершин.

Covering vertex set — накрывающее множество вершин.

H**-Covering,** H**-Covering set** — H-покрытие.

Let H and F be two hypergraphs on the same vertex set. A subset C of F is said to be H**-covering** if every edge in H meets some edge from C (in other words, the union of C is a *vertex cover* for H).

Countable graph — счетный граф.

Counter automation — покрывающее дерево счетчиковый автомат.

Coverability tree — покрывающее дерево (сети Петри).

CPM — метод критического пути.

The same as *Critical path method*.

Critical edge — критическое ребро.

Critical graph — критический граф.

See *Edge-critical graph, Point-critical graph*.

$k-\gamma$**-Critical graph** — $k-\gamma$-критический граф.

See $k-\gamma$*-Domination critical graph*.

Critical pair — критическая пара.

Critical path — критический путь.

An important parameter in any *PERT* digraph is the length of the longest path from the start to the termination vertex. Such a path is called a **critical path**, and its length represents the shortest time within which the overall task can be completed. For this reason the analysis is sometimes called **CPM (Critical path method)**.

Critical path method — метод критического пути.

See *Critical path*.

Critical kernel-imperfect digraph — критический ядерно-недостаточный орграф.

See *Kernel*.

Critical set — критическое множество.

See *Forcing set*.

Critical tournament — критический турнир.

Given a tournament $T=(V,A)$, a subset X of V is an **interval** of T provided that for every $a,b\in X$ and $x\in V-X$, $(a,x)\in A$ if and only if $(b,x)\in A$. For example, $\emptyset$, $\{x\}$ $(x\in V)$ and V are intervals, called **trivial intervals**. A tournament all intervals of which are trivial is called **indecomposable**; otherwise, it is **decomposable**. An indecomposable tournament $T=(V,A)$ is then said to be **critical** if for each $x\in V$, $T(V-\{x\})$ is decomposable and if there are $x\neq y\in V$ such that $T(V-\{x,y\})$ is indecomposable.

Critical vertex — критическая вершина.

p**-Critical graph** — p-критический граф.

A graph G is p**-critical** if G is not *perfect* but every proper induced subgraph of G is perfect. The celebrated *Strong Perfect Graph Conjecture* (SPGC) of C. Berge states that p-critical graphs are only C_{2n+1} and C^c_{2n+1}, $n\geq 2$.

Criticality index — индекс критичности.

The **criticality index** of an edge $e \in E(\bar{G})$ is $ci(e) = \gamma_t(G) - \gamma_t(g+e)$. Note that $ci(e) \in \{0,1,2\}$. Let $E(\bar{G}) = \{e_1, \ldots, e_{\bar{m}}$ and $S = \sum_{j=1}^{\bar{m}} ci(e_j)$. Then the **criticality index** of G is $ci(G) = s/\bar{m}$.

Critically k-connected graph — критический k-связный граф.

A k-connected graph G is said to be **critically k-connected** if $G - v$ is not k-connected for any $v \in V(G)$.

Cross — скрещивание.

Given a bipartite graph $B = (U \cup V, E)$, two non-adjacent edges $e, e' \in E$ with $e = (u_1, v_1)$ and $e' = (u_2, v_2)$ are said to form a **cross** if $(u_1, v_2) \in E$ and $(u_2, v_1) \in E$. Two edges are said to be **cross-adjacent** if either they are adjacent (i.e. share a common node) or they form a cross. A **cross-free matching** in B is a set of edges $E' \subseteq E$ with the property that no two edges $e, e' \in E'$ are cross-adjacent. A **cross-free coloring** of B is a *coloring* of the edge set E subject to the restriction that no pair of cross-adjacent edges has the same color.

The **cross-chromatic index**, $\chi^*(B)$, of B is the minimum number of colors required to get a cross-free coloring of B. The **cross-free matching number** of B, $m^*(B)$, is defined as the edge cardinality of the maximum cross-free matching in B.

Cross arc — поперечная дуга.

See *Basic numberings.*

Cross product — поперечное произведение.

See *Product of two graphs.*

Cross-adjacent edges — кросс-смежные рёбра.

See *Cross.*

Cross-chromatic index — кросс-хроматический индекс.

See *Cross.*

Cross-free coloring — кросс-свободная раскраска.

See *Cross.*

Cross-free matching — кросс-свободное паросочетание.

See *Cross.*

Cross-free matching number — число кросс-свободного паросочетания.

See *Cross.*

Crossing number — число скрещиваний.

The **crossing number** of a graph is the minimum number of crossings of edges for the graph drawn on a plane. It is not the same as its *genus*. The genus of a graph will not exceed its crossing number. Genus and crossing number have obvious implications for the manufacture of electrical circuits on planar sheets.

Crown — корона.

For positive integers $k \leq n$, the **crown** $C_{n,k}$ is a graph with a vertex set $\{a_1, \ldots, a_n, b_1, \ldots, b_n\}$ and an edge set $\{a_i b_j : \ 1 \leq i \leq n, j = i+1, i+2, \ldots, i+k-1 \pmod n\}$.

Crown of graphs — корона графов.

For positive integers $k \leq n$, the **crown of graphs** $C_{n,k}$ is a graph with a vertex set $\{a_1, \ldots, a_n, b_1, \ldots, b_n\}$ and an edge set $\{a_i b_j : \ 1 \leq i \leq n, j = i+1, i+2, \ldots, i+k \pmod n\}$. For any positive integer λ, let $\lambda C_{n,k}$ denote a multiple graph obtained from the crown $C_{n,k}$ by replacing each edge e by λ edges with the same end vertices as those of e. We call $\lambda C_{n,k}$ a **multicrown**.

See also *Corona*.

n**-Cube graph** — куб n-мерный.

Consider the set $Q^n = \{(x_1, x_2, \ldots, x_n) | \ x_i \in \{0,1\}, \ i = 1, \ldots, n\}$. For $u, v \in Q^n$ the Hamming distance $\rho(u,v)$ is defined as the number of entries where u and v differ. An n**-cube graph** is a graph on the vertex set Q^n, where two vertices u, v are *adjacent* iff $\rho(u,v) = 1$. The n**-cube graph** is a *regular graph* with a degree $n-1$.

Other names are **Hypercube**, n**-Dimensial hypercube**

Cubic graph — кубический граф.

This is a *regular graph* with a vertex degree 3.

A graph is called **almost cubic** (or **almost 3-regular**) if one of its vertices has degree $3+e, \ e \geq 0$, and the others have degree 3.

Cubical graph — кубовой граф.

A graph G is called a **cubical graph** if it is embeddable in an *n-cube graph* Q_n, i.e. G is a subgraph of some Q_n.

Cut of a layout — разрез укладки.

See *Layout*.

(a,b)**-Cut** — (a,b)-разрез.

Given a graph G, (a,b)**-C.** is a set F of edges representing (covering) all (a,b)-paths.

Cut-edge — ребро-разрез.

See *Cutset, Bridge*.

Cutpoint — вершина-разрез.

The same as *Articulation point.*

Cutpoint graph — граф точек сочленения.

Cutset — разрез, сечение.

A set of points (edges) in a connected graph whose removal results in a disconnected graph is called a **cutset**. A *cutpoint* (**cut-edge**) is a point (edge) forming a cutset itself.

Other name is **Separating set**.

Cutset matrix — матрица разрезов.

Cutset matroid — матроид разрезов.

Cutting set — сечение.

Cutting vertex — разрезающая вершина, вершина-разрез.

The same as *Articulation point.*

Cutvertex — разрезающая вершина, вершина-разрез.

The same as *Articulation point.*

Cutwidth of a graph — разрезающая ширина графа.

See *Layout.*

Cutwidth of a layout — разрезающая ширина укладки.

See *Layout.*

Cyclability — цикличность.

A subset S of vertices of a graph G is called **cyclable** in G if there is in G some cycle containing all the vertices of S. It is known that if G is a 3-connected graph of order n and if S is a subset of vertices such that the degree sum of any four independent vertices of S is at least $n+2\alpha(S,G)-2$, then S is cyclable. Here $\alpha(S,G)$ is the number of vertices of a maximum independent set of $G[S]$.

See also *Pancyclable graph.*

k**-Cyclable graph** — k-цикловой граф.

The graph g is k**-cyclable** if any k vertices of G lie on a common cycle. It is easy to see that every k-connected graph is k-cyclable.

Cycle — замкнутый маршрут, цикл, контур.

1. A closed *walk*, i.e. a walk such that the starting and ending vertices are the same and all vertices in the walk are distinct is called is a **cycle**.

Another name is **Circuit**.

2. For a directed graph, a closed path, i.e. a path $v_0, \ldots, v_K$ is a **cycle** if $k > 1$ and $v_0 = v_k$.

3. The **inverse cycle** of a cycle $C = (v, v_1, \ldots, v_{n-1}, v)$ is the cycle

$C^{-1} = (v, v_{n-1}, \ldots, v_1, v)$. Two cycles $C_1 = (v_1, \ldots, v_m)$ and $C_2 = (w_1, \ldots, w_m)$ are called **equivalent** if $w_j = v_{j+k}$ for all j. The inverse cycle of C is not equivalent to C. Let $[C]$ be the equivalence class which contains a cycle C. Let B^r be the cycle obtained by going r times around a cycle B. Such a cycle is called a multiple of B. A cycle C is **primitive** if both C and C^2 have no backtracking (a backtracking is a subsequence of the form $..., x, y, x, ...$), and it is not a multiple of a strictly smaller cycle.

Cycle basis — база циклов.

See *Independent circuits*.

Cycle complementary — циклическое дополнение.

A digraph D is **cycle complementary** if there exist two vertex-disjoint cycles C and C' such that $V(D) = V(C) \cup V(C')$.

Cycle cover problem — задача о покрытии графа циклами.

Let $G = (V, E)$ be a connected undirected graph. A non-negative cost or length is associated with each edge. The **cycle cover problem** consists in determining a least cost cover of G with simple cycles, each containing at least three different edges.

Cycle embedding matrix — матрица вложенности контуров.

Cycle-factor — цикличский фактор.

A **cycle-factor** of a digraph D is a *spanning subdigraph* consisting of disjoint cycles.

Cycle isomorphism — циклический изоморфизм.

A bijection f between the vertex sets V_1 and V_2 of two *sigraphs* S_1 and S_2, respectively, is called f **cycle isomorphism** (or **weak isomorphism**) between S_1 and S_2 if f preserves both vertex adjacencies and cycle signs of S_1 and S_2.

Cycle matrix — матрица циклов.

Cycle matroid — матроид циклов.

Let $E(G)$ be the edge-set of a graph G and C be the set of cycles. The cycles satisfy the circuit postulates. Thus, we obtain a *matroid* related to the graph. We denote this matroid by $M(G)$ and call it the **cycle matroid** of G. The bases of $M(G)$ are the *spanning trees*. The *rank* of $M(G)$ is less by 1 than the number of vertices.

Cycle space — пространство циклов.

Given a graph G, let $e_1, e_2, \ldots, e_{|E(G)|}$ be an ordering of its edges. Then a subset S of $E(G)$ corresponds to a (0,1)-vector $(b_1, \ldots, b_{|E(G)|})$ in the usual way with $b_i = 1$ if $e_i \in S$, and $b_i = 0$ if $e_i \notin S$.

These vectors form an $|E(G)|$-dimensional vector space, denoted by $(Z_2)^{|E(G)|}$, over the field of integers modulo 2. The vectors in $(Z_2)^{|E(G)|}$ which correspond to the cycles in G generate a subspace called the **cycle space** of G denoted by $\mathcal{C}(G)$. It is known that

$$\dim \mathcal{C}(G) = |E(G)| - |V(G)| + r$$

where r is the number of connected components.

See also *Basis number*.

Cycle spectrum — цикловой спектр.

For a graph G, we define the **cycle spectrum** $CS(G)$ of G as the sequence $\ell_1 < \cdots < \ell_r$ of lengths of cycles in G.

Cycle vector — вектор-цикл.

Cyclic alternating chain — чередующийся цикл.

Cyclic chromatic number — циклическое хроматическое число.

Cyclic edge — ориентированно-циклическое ребро.

Cyclic edge connected vertices — ориентированно-циклически-реберно связные вершины.

Cyclic graph — циклический граф.

The **cyclic graph** $C(n,k)$ have a point set $V = \{0, 1, \ldots, n-1\}$ and lines $\{i, i+1\} \pmod n$ and $\{i, i+k\} \pmod n$ $(i = 1, \ldots, n)$, where k is an integer with $2 \leq k \leq n-2$. If $G(n,k)$ is a *circulant graph*, then $C(n,k) \simeq G(n,S)$ with $S = \{1, \ldots, \min\{k, n-k\}\}$. The graphs $C(n,k)$ are point-transitive 3-regular if $n = 2k$, and 4-regular otherwise.

Cyclic matroid — циклический матроид.

Cyclic sequence — циклический маршрут.

Cyclic vector of a graph — циклический вектор графа.

k-Cyclic chromatic number — k-циклическое хроматическое число.

The **k-cyclic chromatic number** $\chi_k(G)$ of a plane graph is the smallest number of colours in a vertex colouring of G such that no face of size at most k has two boundary vertices of the same colour. It is easy to see that the Four Colour Theorem may be stated in the form:

$$\chi_3(G) \leq 4$$

for every plane graph G.

The number $\chi_k(G)$ was introduced explicitly by Ore and Plummer (1969).

k-Cyclic coloring — k-циклическая раскраска.

α-Cyclic hypergraph — α-циклический гиперграф.
Cyclomatic complexity of a program — цикломатическая сложность программы.
Cyclomatic matrix — цикломатическая матрица.
Cyclomatic number — цикломатическое число, цикломатический ранг.

D

DAG (Directed Acyclic Graph) — бесконтурный орграф.

A **directed acyclic graph**, also called a **DAG**, is a *directed graph* without *cycles*.

The reachability relation in a DAG forms a partial order, and any finite partial order may be represented by a DAG using reachability. DAGs may also be used to model expressions and basic blocks.

A DAG presentation for an expression identifies the common subexpression of the expression. Like a *syntax tree*, an **expression dag** has a node for every subexpression of the expression; an interior node represents an operator and its children represent its operands. The difference is that a node in a dag representing a common subexpression has more than one "parent", in a syntax tree, the common subexpression would be represented as a duplicated subtree.

DAG is a useful data structure for implementing transformations on basic blocks. A DAG representation for a basic block gives a picture of how the value computed by each statement in a basic block is used in subsequent statements of a block. A **dag for a basic block** is a directed acyclic graph with the following labels on nodes.

(1) Leaves are labeled by unique identifiers, either variable names or constants. From the operator applied to a name, we determine whether the l-value or r-value of a name is needed; most leaves represent r-values. The leaves represent the initial values of names, and we subscript them with 0 to avoid confusion with labels denoting "current" values of names as in (3) below.

(2) Interior nodes are labeled by an operator symbol.

(3) Nodes are also optionally given a sequence of identifiers for a label. The intention is that interior nodes represent computed values, and the identifiers labeling a node are deemed to have that value.

(4) Certain nodes are designated output nodes. These are the nodes whose variables are live on exit from the block; that is, their values may be used later, in another block of the flow graph.

It is important not to confuse dags with *flow graphs*. Each node of a flow graph can be represented by a dag, since each node of the flow graph stands for a basic block.

Dag for basic block — дэг луча.

See *DAG*.

Dag of control flow graph — каркас уграфа.

A **dag** of a *cf-graph* G with an initial node p is an acyclic cf-graph D with the initial node p such that $V(G) = V(D)$, $A(D) \subseteq A(G)$ and for any arc $u \in A(G)\backslash A(D)$ the graph $D \bigcup \{u\}$ has a cycle. That is, D is a maximal acyclic subflowgraph.

Data dependence — зависимость по данным.

In general terms, a statement T **depends** on a statement S, denoted by $S\delta T$, if there exist an instance S' of S, an instance T' of T, and a memory location M, such that the following properties hold:
(1) both S' and T' are references to M, and at least one of those references is a “write”;
(2) in a serial execution of the program, S' is executed before T'; and
(3) in the same execution, M is not written between the time S' finishes and the time T' starts.
The following three types of dependence between the statements S and T based upon the types of the two references to M are considered:
(1) T is **flow (true) dependent** on S, ($S\delta T$), if S' writes to M and then T' reads it.
(2) T is **anti-dependent** on S, ($S\bar{\delta}T$), if S' reads M and then T' writes to it.
(3) T is **output dependent** on S, ($S\delta^o T$), if S' writes to M and then T' writes to it again.

Data dependence graph — граф зависимости по данным.

By the **data dependence graph** of the program we mean a digraph, where the vertices correspond to the (assignment) statements in the program, and there is an arc from a vertex S to a vertex T iff T depends on S; i.e. $S\delta T$, $S\bar{\delta}T$ or $S\delta^o T$. Each arc may be labeled with information about the type of dependence and other information.

Data flow analysis frameworks — схема с разметкой, схема свойств состояний.

Data flow analysis problem — задача анализа свойств состояний, задача потокового анализа, задача глобального анализа потока данных.

Data set — информационное множество.

Data term — простое выражение, слово значения.

See *Large-block schema.*

De Bruijn graph — граф де Брёйна.

The **binary De Bruijn graph** $\mathcal{D}(n)$ is a directed graph of order 2^n

whose vertices comprise the set Z_2^n. The arcs of $\mathcal{D}(n)$ connect each vertex αx, where $\alpha \in Z_2$ and $x \in Z_2^{n-1}$, to vertices $x0$ and $x1$.

The **De Bruijn graph** of d symbols is a directed graph $B(d, n)$ representing overlaps between n-sequences of d symbols. $B(d, n)$ has d^n vertices from $Z_d^n = \{(1, 1, \ldots, 1, 1)(1, 1, \ldots, 1, 2), \ldots, (1, 1, \ldots, 1, d)$ $(1, 1, \ldots, 2, 1), \ldots, (d, d, \ldots, d, d)\}$. The arcs of $B(d, n)$ connect each vertex $(v_1, v_2, \ldots, v_{n-1}, v_n)$ to a vertex $(w_1, w_2, \ldots, w_{n-1}, w_n)$ such that $v_2 = w_1$, $v_3 = w_2, \ldots$, $v_n = w_{n-1}$.

The **De Bruijn undirected graph**, denoted $UB(d, n)$, is obtained from $B(d, n)$ by deleting the orientation of all directed edges and omitting multiple edges and loops. Clearly, $UB(d, 1)$ is a complete graph of order d.

De Bruijn undirected graph — неориентированный граф де Брёйна.

See *De Bruijn graph*

Decay number — число распада.

Given a connected graph G, the **decay number** $\zeta(G)$ is the smallest number of components a *cotree* of G can have. That is

$$\zeta(G) = \min\{c(G - E(T))|\ T \text{ is a spanning tree of } G\},$$

where $c(H)$ denotes the number of connected components of a graph H.

Decision problem — задача распознавания свойств.

A **decision problem** is one that asks only for a yes-or-no answer: Can this graph be 5-colored? Is there a set of 67 independent vertices? Because of this, it is traditional to define the decision problem equivalently as: the set of inputs for which the problem returns "yes". These inputs can be natural numbers, but also other values of some other kind, such as strings of a formal language. Using some encoding, such as Godel numbers, the strings can be encoded as natural numbers. Thus, a decision problem informally phrased in terms of a formal language is also equivalent to a set of natural numbers. To keep the formal definition simple, it is phrased in terms of subsets of natural numbers.

A decision problem A is called **decidable** or **effectively solvable** if A is a **recursive set**, i.e. if there is an algorithm which terminates in a finite time and correctly decides whether or not a given number belongs to the set A.

A problem is called **partially decidable**, **semidecidable**, **solvable**, or **provable** if A is a **recursively enumerable set**, i.e. there is an algorithm that, when given an input number, eventually halts if and only if the input is an element of A.

Partially decidable problems and any other problems that are not decidable are called **undecidable**.

Decidable problem — (алгоритмически) разрешимая задача.

See *Decision problem.*

Decision problem DIM_k — задача распознавания DIM_k.

See *k-Dimensional poset.*

Decision tree — дерево решений.

Decomposable hammock — разложимый гамак.

See *Hammock.*

Decomposable graph — разложимый граф.

We write $G = H_1 \oplus H_2$ if G is the edge disjoint union of its subgraphs H_1 and H_2. If $G = H_1 \oplus \cdots \oplus H_k$, where $H_1, \ldots, H_k$ are all isomorphic to H, then G can be **decomposed** into subgraphs isomorphic to H; we say that G is H**-decomposable** and that $\{H_1, \ldots, H_k\}$ is H**-decomposition** of G. In particular, G is C_m**-decomposable** if it can be decomposed into subgraphs isomorphic to an m-cycle.

Decomposable tournament — разложимый турнир.

See *Critical tournament.*

Decomposition — разложение, декомпозиция.

A **decomposition** of a graph H consists of a set of edge-disjoint subgraphs of H, which partition the edges of H. If each of the subgraphs in the decomposition is isomorphic to some graph G, then the decomposition is called a G-decomposition of H, or a decomposition of H into copies of G.

Decomposition dimension — декомпозитная размерность.

A decomposition $\mathcal{F} = \{F_1, \ldots, F_r\}$ of the edge set of a graph G is called a resolving r-decomposition, if for any pair of edes e_1 and e_2 there exists an index i such that $d(e_1, F_i) \neq d(e_2, F_i)$, where $d(e, F)$ denotes the distance from e to F. The **decomposition dimension** $dec(G)$ of a graph G is the least integer r such that there exists a resolving r-decomposition. See also *Metric dimension.*

d**-Decomposition** — d-разложение, d-декомпозиция.

See *Maximal packing.*

H**-decomposition number** — число H-разложения.

For a fixed graph H without isolated vertices, the H**-decomposition number** $d_H(G)$ of a graph is the minimum number of vertices that must be added to G to produce a graph that can be decomposed into copies of H. (Any number of edges may be added incident with the new vertices.)

Defect n-extendable graph — дефектно n-расширяемый граф.

If G is a connected graph and any n independent edges in G are contained in a *near perfect matching* of G where $n \leq (|V(G)|-2)/2$, then G is **defect n-extendable**.

Deficiency — дефицит.

The **deficiency** $def(\mathcal{A})$ of a family $\mathcal{A}$ of hypergraphs is the minimal natural number d such that the *matching width* $mw(\mathcal{B})$ satisfies the condition

$$mw(\cup\mathcal{B}) \geq |\mathcal{B}| - d$$

for every subfamily $\mathcal{B}$ of $\mathcal{A}$.

Deficiency of a bipartite graph — дефицит двудольного графа.

For a vertex v, the **deficiency of a bipartite graph** $D(v)$ is a set of pairs defined by $D(v) = \{(x, y)|v \in N(x), v \in N(y), x \notin N(y), x \neq y\}$. So a vertex v is *simplicial* if $D(v) = \emptyset$.

For an edge (x, y), the **deficiency of a bipartite graph** is the set of pairs defined by $D(x, y) = \{(a, b)|a, b \in N(x) \cup N(y), (a, b) \notin E\}$. So an edge (x, y) is *bisimplicial* if $D(x, y) = \emptyset$.

Here $N(v)$ denotes a *neighbourhood* of v.

Deficiency of a graph — дефицит графа.

Let M be a *matching* in a graph G. A vertex v is **saturated** by M if an edge of M is incident with v, otherwise v is said to be unsaturated. The **deficiency** $def(G)$ of the graph G is the number of vertices unsaturated by a maximum *matching*. Thus, if $def(G) = 0$, then G has a *perfect matching*.

Defining set — определяющее множество.

See *Forcing set*.

Degenerate tree — вырожденное (тривиальное, пустое) дерево.

Degenerate tree is a tree with one vertex.

See also *Trivial tree*.

Degree balanced digraph — степенно-балансированный граф.

A digraph G is called **degree balanced** if $d^+_G = d^-_G$ for all vertices $v \in V(G)$.

Degree of an edge — степень ребра.

Degree of a graph — степень графа.

This is the largest of the *degrees* of vertices of a graph G. It is denoted by $\Delta(G)$.

Degree of a graph group — степень группы графа.

Degree of a hypergraph edge — степень ребра гиперграфа.

Degree of a vertex — степень вершины.

1. (For an undirected graph). This is the number of edges (denoted by $deg\ v$) incident to v. A graph $G = (V, E)$ has a **bounded degree** B if each vertex $v \in V$ has the **degree** up to B, and G is B**-regular** if each vertex $v \in V$ has exactly the degree B.

2. (For a directed graph.) The sum of the *in-degree* and *out-degree* of a vertex v, $deg\ v = deg^{-}v + deg^{+}v$, is called its **degree**.

Degree pair of a vertex — степенная пара вершины.

Let G be a digraph and v be a vertex of G.

The ordered pair $[outdegree(v), indegree(v)]$ is called the **degree pair** of the vertex v.

Degree sequence — степенная последовательность.

For a graph G, its **degree sequence** $d_1 \geq d_2 \geq \ldots \geq d_n$ is the ordered sequence of the degrees of its vertices. A sequence $d_1 \geq d_2 \geq \ldots \geq d_n$ with $n - 1 \geq d_1$ is a **graphic sequence of numbers** if there is a graph having $D_1, \ldots, d_n$ as its degree sequence.

See also *Havel – Hakimi criterion, Erdös – Gallai criterion.*

Dense tree — плотное дерево, густое дерево.

See *r-dense tree.*

r**-Dense tree** — r-плотное дерево.

An m-ary tree T is said to be r**-dense tree**, where r is a natural number with $1 \leq r \leq m - 1$, iff the following properties hold:

(1) the root of T is at least binary,

(2) each unsaturated (the number of sons is less than m) node different from the root has either only saturated brothers and at least one such brother or at least r saturated brothers,

(3) all leaves have the same *depth.*

A class of m-ary trees is called **dense** if it is a class of r-dense m-ary trees for some r. In particular, we speak of **weakly dense** m**-ary trees** and **strongly dense** m**-ary trees**, respectively, if we have in mind the classes of r-dense and $(m-1)$-dense m-ary trees, respectively.

Observe that there is only one class of dense binary trees. This class coincides with the class of *brother trees.*

Density — плотность.

Let G be a graph with a vertex set $V(G)$ and an edge set $E(G)$. The **density** of G is defined by

$$d(G) = \frac{|E(G)|}{|V(G)|}.$$

G is said to be **balanced** if for each subgraph H of G we have $d(H) \leq d(G)$, where V(H) is assumed to be nonempty. If G is not balanced, then it contains a subgraph with greater density than that of G. In particular, we use $m(G)$ to denote the maximum density of a subgraph of G, i.e.

$$m(G) = \max_{H \subseteq G} d(H).$$

$m(G)$ is called the **global density** of G.

w**-Density** — w-плотность.

The w**-density** of G is defined by

$$wd(G) = \frac{w^E(G)}{w^V(G)}.$$

(See *Weighted graph.3.*) A weighted graph is called w**-balanced**, if for each subgraph H of G, we have $wd(H) \leq wd(G)$, where $V(H)$ is assumed to be nonempty. If G is not w-balanced, then there exists a subgraph with greater w-density than that of G. Let $wm(G)$ denote the maximum w-density of a subgraph of G, i.e.

$$wm(G) = \max_{H \subseteq G} wd(H).$$

$wm(G)$ is called the **global** w**-density** of G.

Dependent set of a matroid — зависимое множество матроида.

See *Matroid.*

Depth of an arrangement — глубина аранжировки.

Depth of an arrangeable graph — глубина аранжируемого графа.

See *Arrangeable graph.*

Depth of a flow graph — глубина управляющего графа.

Given a *depth-first spanning tree* for a *flow graph*, the **depth** is the largest number of *retreating edges* on any cycle-free path. Here retreating edges are those going from a node m to an ancestor of m. It is interesting and useful fact that if the flow graph is *reducible*, then the retreating edges are exactly the back edges of the flow graph, independent of the order in which successors are visited. For any flow graph, every back edge is retreating, although, if the graph is *nonreducible*, there will be some retreating edges that are not back edges.

Depth of a numbering — глубина нумерации.

See *Numbering of cf-graph.*

Depth of a tree — глубина дерева.

Depth-first search (DFS) — поиск в глубину.

1. Let G be a directed graph.

It is convenient to formulate DFS as a recursive procedure $DFS(v)$ with a vertex v as a parameter. In general, we search for unexplored vertices by traversing an unexplored arc from the most recently reached vertex which still has unexplored arcs. The set $REACH$ contains the explored vertices. On these conditions DFS has the following main structure

procedure $DFS(v : V)$
 begin
  add v to $REACH$;
  for $\forall w$ with $(v, w) \in E$ do
   if $w \notin REACH$ then $DFS(w)$ fi
  od
end

The procedure starts with

$$REACH \leftarrow \emptyset;$$
$$DFS(r);$$

and marks all vertices reachable from the start vertex r. But DFS gives some further information about the digraph G. In particular, DFS computes the so-called **depth-first search tree** (or DFS-**tree**) with a root r, which consists of all vertices reachable from the vertex r. If G is a *cf-graph* with an initial vertex r, then DFS-tree is a *spanning tree* of G. DFS can be easily extended in such a way that for any cf-graph G with an initial vertex r, $DFS(r)$ computes in a linear time two correlated *basic numberings* of G and makes the corresponding partition of all arcs of G into the four classes of the *tree, forward, backward* and *cross* arcs with respect to the basic numberings.

2. Let G be an undirected graph.

Suppose that in a depth-first search of an undirected graph we are currently visiting a vertex v. The general step in the search then requires that the next visited is a vertex adjacent to v which has not yet been visited. If no such vertex exists, then the search returns to the vertex visited just before v and the general step is repeated until every vertex in that component of the graph becomes visited. Such a search cannot revisit a vertex except by returning to it via edges that have been used since the previous visit. Hence, the edges traversed in a depth-first search form a *spanning tree* for each separate component of the graph. This set of trees is called a **depth first spanning forest** F. Thus, DFS partitions the edges E into two sets, F and $B = E \setminus F$. The edges in B are called **back-edges**.

The time complexity of *DFS* in a general case is $\mathcal{O}(n+m)$

Depth-first search tree — дерево поиска в глубину.

See *Depth-first search.*

Depth-first spanning forest — глубинный остовный лес.

See *Depth-first search. 2.*

Depth-first spanning tree — глубинное остовное дерево.

A **depth-first spanning tree** (or **DFS-tree**) is a *spanning tree* which is found by the *depth-first search.*

Depth of a DAG — глубина дэга.

See *Depth of a vertex.*

Depth of a vertex — глубина вершины.

The **depth of a vertex** v in a directed acyclic graph G is the length of the longest path in G from an *input vertex* to v, and the **depth** of G is the maximum depth of any of its vertices.

The **depth of a vertex** v in a rooted tree is its distance from the root, i.e. the number of edges in the path from the root to v.

Derivation — вывод (в грамматике).

The concept of derivation is a central concept in the theory of formal grammars and languages. Let $G = (V_N, V_T, R, S)$ be a grammar and α, β be two strings over the alphabet $V = V_N \bigcup V_T$.

A **derivation** of β from α in G is a finite sequence of words over V

$$w_0, w_1, \ldots, w_{j-1}, w_j, \ldots, w_m,$$

such that $m \geq 0$, $w_0 = \alpha$, $w_m = \beta$ and for any $j \in [1, m]$, $w_{j-1} \Rightarrow w_j$, i.e. w_j immediately derives from w_{j-1}. m is called the **length of the derivation**.

Derivations are also written as

$$w_0 \Rightarrow w_1 \Rightarrow \ldots, \Rightarrow w_m.$$

Derivation tree — дерево вывода.

Let G be a CF-grammar and x be a sentence form in G.

All equivalent derivations of x can be represented by its **derivation tree** that is an ordered rooted tree labeled by elements from $V \bigcup \{e\}$, such that the following properties hold. The root of the tree has a label S, labels of all leaves enumerated according to their ordering forms x and for any internal node q of the tree with a list of all sons $q_1, q_2, \ldots, q_r$, $r \geq 0$, enumerated according to their ordering, there is such a production $A \rightarrow a_1 a_2 \ldots a_r$ of the grammar that either $a_1 a_2 \ldots a_r \neq e$ and the nodes $q, q_1, q_2, \ldots, q_r$ have labels $A, a_1, a_2, \ldots, a_r$, respectively, or $a_1 a_2 \ldots a_r = e$, $r = 1$, q has a label A and q_1 has the empty string e as a label.

Other names are **Parse tree**, **Syntax tree**.

Derived graph — производный граф.

Given *cf-graph* G, a **derived graph** is *1-derived* graph of G.

Other name is *Interval graph 2*. See *Reducible (control) flow graph.*

Derived sequence — последовательность сведения.

k-Derived graph — k-производный граф.

See *Reducible (control) flow graph.*

Descendant — потомок.

See *Directed tree.*

Descendance graph — граф потомства.

Descendant of a vertex — потомок вершины.

Deterministic automaton — детерминированный конечный автомат.

See *Model of computation.*

Deterministic pushdown automaton — детерминированный автомат с магазинной памятью.

See *Model of computation.*

Deterministic Turing machine — детерминированная машина Тьюринга.

See *Model of computation*.

DFS — поиск в глубину.

See *Depth-first search*.

Detour — обходной путь.

Let $P_G(v_0, v_1, \ldots, v_p)$ be the shortest path from v_0 to v_p in a biconnected graph G. A **detour** from v_i to v_p, denoted by $P_{G-e}(v_i, v_p)$, is the shortest path from v_i to v_p that does not contain the edge $e = (v_i, v_{i+1})$.

Detour center — центр обхода.

The **detour center** of G is a subgraph induced by the vertices of G having the detour eccentricity $rad_D(G)$.

Detour diameter — диаметр обхода.

The **detour diameter** $diam_D(G)$ is the maximum detour eccentricity among the vertices of G.

Detour distance — расстояние обхода.

Let G be a nontrivial connected graph. For distinct vertices u and v of G, the **detour distance** $D(u, v)$ between u and v is the length of the longest $u - v$ path in G. Thus $1 \leq D(u, v) \leq n - 1$, where $D(u, v) = 1$ if and only if uv is a *bridge* of G and $D(u, v) = n - 1$ if and only if G contains a hamiltonian $u - v$ path.

Detour dominating set — обходное доминирующее множество.

For a vertex v in G, define

$$D^-(v) = \min\{D(u, v) : u \in V(G) - \{v\}\}.$$

A vertex u ($\neq v$) is called a **detour neighbor** of v if $D(u, v) = D^-(v)$. The **detour neighborhood** $N_D(v)$ of a vertex v is the set of detour neighbors of v, and its **closed detour neighborhood** is $N_D[v] = N_D(v) \cup \{v\}$. A vertex v is said to detour-dominate a vertex u if $u = v$ or u is a detour neighbor of v.

Detour eccentricity — эксцентриситет обхода.

The **detour eccentricity** $e_D(v)$ of a vertex v in G is the maximum detour distance from v to a vertex of G.

Detour order — порядок обхода.

The **detour order** of G, denoted $\tau(G)$, is the order of the longest path in G.

Detour periphery — периферия обхода.

The **detour periphery** of G is a subgraph induced by the vertices of G having the *detur eccentricity* $diam_D(G)$.

Detour radius — радиус обхода.

The **detour radius** $rad_D(G)$ of G is the minimum *detour eccentricity* among the vertices of G.

DFS-tree — дерево поиска в глубину.

See *Depth-first search*, *Depth-first spanning tree.*

Diagonal of a block — диагональ блока.

Diameter — диаметр (графа).

1. See *Eccentricity of a vertex.*

2. The maximum distance between points of a graph is called its **diameter**.

Diameteral chain — диаметральная цепь.

See *Eccentricity of a vertex.*

k-**Diameter** — k-диаметр.

Let $\mathcal{P}_k(u,v) = \{P_1, P_2, \cdots, P_k\}$ be a family of k vertex disjoint paths between u and v with lengths $|P_1| \geq |P_2| \geq \cdots \geq |P_k|$. The k-**distance** between u and v is the minimum $|P_1|$ among all $\mathcal{P}_k(u,v)$, and the k-**diameter** $d_k(G)$ of G is

$$d_k(G) = \max\{d_k(u,v) : \ u \neq v \text{ and } u,v \in V(G).$$

The concept of k-diameter naturally arises from the study of routing, reliability, randomized routing, fault tolerance and other communication protocols in the parallel architecture and distributed computer networks.

Diamond — алмаз.

This is a graph obtained from K_4 by deleting any edge.

Difference digraph — разностный граф.

A digraph G is a **difference digraph** iff there exists an $S \in IN^+$ such that G is isomorphic to the digraph $DD(S) = (V, A)$, where $V = S$ and $A = \{(i,j) : i,j \in V \wedge i - j \in V\}$.

Difference of graphs — разность графов.

Differential of a graph — дифференциал графа.

Let $B(X)$ be the set of vertices in $V - X$ that have a neighbor in the set X. We define the **differential of a set** X to be $\partial(X) = |B(X)| - |X|$, and the **differential of a graph** to equal the $\max\{\partial(X)\}$ for any subset X of V.

Differential of a set — дифференциал множества.
See *Differential of a graph*.

Digraph — орграф.
The same as *Directed graph*.

Dijkstra's algorithm — алгоритм Дейкстры.

Dimension of a poset — размерность чу-множества.
See *Linear extension*.

d**-Dimensional** c**-ary array** — d-мерный c-арный массив.

d**-Dimensional** c**-ary clique** — d-мерная c-арная клика.

n**-Dimensional hypercube** — n-мерный гиперкуб.
The same as *n-Cube graph*.

d**-Dimensional lattice** — решетчатый d-мерный граф.

k**-Dimensional poset** — k-мерное чу-множество.
The **order dimension** $\dim(P)$ of a poset $P = (V, <)$ is the smallest number of linear extensions $L_1, \ldots, L_k$ of P, $L_i = (V, <_i)$, whose intersection is P, i.e. $a < b$ iff $a <_i b$ for all $i = 1, \ldots, k$. A partial order P is k**-dimensional poset** if $\dim(P) \leq k$. See also *n-Mesh*.
The decision problem $DIM_k = \{P : \dim(P) \leq k\}$ is NP-complete already for $k = 3$, whereas 2-dimensional posets can be recognized in a polynomial time.

Dinitz's algorithm — алгоритм Диница.
This is the algorithm for finding maximum *flows* in undirected graphs that repeatedly augments the current flow by a blocking flow in the graph induced by the residual arcs on shortest paths from s to t. It is known that **Dinitz's algorithm** terminates in $\min(n^{2/3}m^{1/2})$ iterations.

F**-Direct arc** — F-прямая дуга.
See *Numbering of cf-graph*.

Direct product — прямое произведение [графов].
See *Product of two graphs*.

Direct search — перебор.
See *Exhaustive search*.

Directed acyclic graph — ориентированый ациклический граф, ориентированный бесконтурный граф, дэг.
See *DAG*.

Directed edge — ориентированное ребро, дуга.
The same as *Arc*.

Directed graph — ориентированный граф, орграф.

A **directed graph** G, or simply a **d-graph, digraph**, consists of a finite set of vertices (or **nodes**) V, a finite set of **arcs** (directed edges) E, and two mappings s and t from E to $V \times V$, assigning to each arc e its **source, positive end** and **target, negative end** nodes, respectively. The digraph G will be denoted by $G = (V, E, s, t)$ or simply $G = (V, E)$. Let $G = (V, E, s, t)$ be a digraph; we define the **underlying (undirected) graph** of G as $Und(G) = (V, E, Ends)$ with:

$$\forall e \in E,\ Ends(e) = \{s(e), t(e)\}.$$

If $s(e) = u$ and $t(e) = v$ for some $e \in E$, then u is a **server** of v, and v is a **receiver** of u. A vertex w is called a **common server** of u and v, if w is a server of u and v. Similarly w is called a **common receiver** of u and v, if w is a receiver of u and v.

Directed hypergraph — ориентированный гиперграф.

A **directed hypergraph** H is a pair (N, E), where N is a nonempty set of **nodes** or vertices and E is a set of **hyperarcs**; a hyperarc e is an ordered pair (T, h), with $T \subseteq N$, $T \neq \emptyset$ and $h \in N$; h and T are called the **head** and the **tail** of the hyperarc e and will be denoted with $Head(e)$ and $Tail(e)$, respectively.

The **size** of a directed hypergraph can be defined as a sum of the cardinalities of its hyperarcs:

$$size(H) = \sum_{e \in E} |T_e|.$$

Directed hyperpath — ориентированный гиперпуть.

A **directed hyperpath** P_{Rt} from the root set R $(\subseteq V)$ to the *sink* t $(\in V)$ in H is a minimal acyclic sub-hypergraph of H containing both the nodes of R and node t, such that each node, with exception of the nodes in R, has exactly one entering hyperarc.

The definition of a hyperpath can be extended as follows. A (directed) **hypertree** T_R rooted at R in H is an acyclic sub-hypergraph of H containing the nodes in R, such that each node, with the exception of the nodes in R has exactly one entering hyperarc.

The set R is called the root set, while the remaining nodes are called the non-roots. Any non-root v not contained in the tail of any hyperarc of T_R is said to be a **leaf** of the hypertree. By definition, for any non-root v there is a unique directed hyperpath in T_R from R to v.

An **undirected hyperpath** (**hypertree**) is a “permutation” of a hyperpath P_{Rt} (hypertree T_R), i.e. it is obtained by a permutation of some of the hyperarcs on P_{Rt} (T_R), where the permutation of a hyperarc e is a hyperarc e' such that $T'_e \cup h'_e = T_e \sup h_e$.

Directed hypertree — ориентированное гипердерево.

See *Directed hyperpath.*

Directed tree — ориентированное дерево, ордерево.

A **directed tree** T is an *acyclic digraph* with one distinguished vertex r called the **root**, such that there is a simple path of length greater than 0 from r to any vertex v ($r \xrightarrow{+} v$) for all vertices $v, v \neq r$, and no arcs enter r. In a tree exactly one arc enters every other vertex as the root. A tree vertex with no existing edges is a **leaf**. Let (v, w) be a tree arc, then v is a **father** of w and w is a **son** of v. If there exists a path from v to w, then v is an **ancestor** of w and w is a **descendant** of v.

A tree node z is called a **common ancestor** of nodes v and w if and only if there are paths from z to v and from z to w. The vertex z is the **nearest common ancestor** of v and w iff there is no other common ancestor x of v and w with $x \neq z$ and there is a path from z to x. The **hight** of a node v in T is the length of a path with the maximum length from v to a *leaf* in the subtree of v.

A **sink-tree** S is a tree in which arcs are directed from a son to its father or, in other words, there is a path from w to the root r for eath node w.

F**-Direct arc** — F-прямая дуга.

See *Numbering of cf-graph.*

Directable graph — ориентируемый граф.

Directed forest — ориентированный лес.

Directed multigraph — ориентированный мультиграф.

Directed sequence — ориентированный маршрут.

Diregular digraph — дирегулярный орграф.

Let $\rho \in \{1, 2, \ldots\}$. A digraph D is called ρ**-diregular** if every vertex of D has the degree pair (ρ, ρ). Hence, if a ρ-diregular oriented graph has n vertices, then $\rho \leq \frac{n-1}{2}$. Moreover, a digraph is called **diregular** if it is ρ-diregular for some ρ.

Dirichlet eigenvalue problem — проблема Дирихле о собственных значениях.

The **Dirichlet eigenvalue problem** can be introduced by restricting the eigenfunctions of the graph *Laplacian* to f such that $f(u_0) = 0$ for all boundary vertices $u_0 \in \partial V$.

Disc — диск, окрестность порядка k.

Let v be a vertex of G. A **disc** centered at v with a *radius* k is the set of all vertices whose *distance* to v is at most $k : N^k[v] = \{u : u \in V \text{ and } d(u, v) \leq k\}$.

See also *Neighbourhood.*

Disconnected graph — несвязный граф.

See *Connected graph.*

Discrete matroid — свободный матроид.

Disjunct union of graphs — дизъюнктное объединение графов.

Dismantlable graph — разборный граф.

A graph G is said to be **dismantlable** if its vertices can be removed one after another in such a way that a vertex x can be taken off the currently remaining subgraph G_x of G if there exists a vertex y in G_x which is adjacent to x and to all neighbors of x in G_x.

A graph G is said to be **dismantlable** if there is a well-order $\preceq$ on $V(G)$ such that every vertex x which is not the greatest element of $(V(G), \preceq)$, if such a greatest element exists, is dominated by some vertex $y \neq x$ in a subgraph of G induced by the set $\{z \in V(G) : x \preceq z\}$. The well-order $\preceq$ on $V(G)$, and the enumeration of the vertices of G induced by $\preceq$, will be called a **dismantling order** and a **dismantling enumeration**, respectively.

See also *Constructible graph.*

Dismantling enumeration — разбирающая нумерация.

See *Dismantlable graph.*

Dismantling order — разбирающий порядок.

See *Dismantlable graph.*

Disorientation of an arc — дезориентация дуги.

Dissection — рассечение.

Distance — расстояние.

1. The **distance** between two vertices u, v, denoted $d(u, v)$, is the length of the shortest path between them. If no path of G connects u to v, their distance is ∞.

2. If H_1 and H_2 are subgraphs of a graph G, the **distance** between

H_1 and H_2 is defined as

$$\min\{d_G(x, y)|\ x \in V(H_1), y \in V(H_2)\}.$$

3. For a vertex subset S of $V(G)$, the **distance** of S, denoted by $d(S)$, is equal to the sum of the distances between all pairs of distinct vertices of S. In particular, $d(V(G)) = d(G)$.

k-Distance — k-расстояние.

See *k-Diameter*.

Distance-hereditary graph — дистанционно-наследуемый граф.

A graph is **distance-hereditary graph** if the *distance* stays the same between any of two vertices in every connected induced subgraph containing both. **distance-hereditary graph** form a subclass of *perfect graphs*. Two well-known classes of graphs, trees and *cographs*, both belong to **distance-hereditary graph**.

Distance-transitive graph — дистанционно-транзитивный граф.

H-distance — H-расстояние.

Let G_1 and G_2 be two graphs of the same order and size such that $V(G_1) = V(G_2)$, and let H be a connected graph of order at least 3. A $G_1 - G_2$ **H-path** is a sequence $G_1 = F_0, F_1, \ldots, F_k = G_2$ of graphs of the same order and the same size such that F_i is *H-adjacent* to F_{i+1} for $i = 0, 1, \ldots, k-1$. The graphs G_1 and G_2 are **H-connected** if there exists a $G_1 - G_2$ H-path. For H-connected graphs G_1 and G_2, the **H-distance** $d_H(G_1, G_2)$ from G_1 to G_2 is the minimum number of H-adjacencies required to transform G_1 into G_2.

k-Divergent graph — k-дивергентный граф.

See *Clique graph*.

Divider — делитель.

See *Separator*.

Ditree — ордерево.

Domatic number — доматическое число.

The **domatic number** $d(G)$ (or $dom(G)$) of G is the maximum cardinality of a *domatic partition* of G. The **domatic number** is one of the numerous domination *invariants*. It was introduced by Cockayne and Hedetniemi in 1977. Clearly, any graph G satisfies $d(G) \leq \delta(G) + 1$ ($\delta(G)$ is a minimal degree of G). Graphs for which $d(G)$ achieves this upper bound $\delta(G) + 1$ are called **domatically full**.

Domatically full graph — доматически полный граф.

See *Domatic number*.

Domatic partition — доматическое разбиение.

A **domatic partition** of G is a partition $\mathcal{D} = \{D_1, \ldots, D_l\}$ of $V(G)$ into (pairwise) disjoint *dominating sets*. The **domatic number** $d(G)$ of G is the maximum cardinality of a domatic partition of G.

Dominance number — число доминирования.

Dominant set — доминирующее множество.

See *Independent set*.

Dominant-covering graph — доминантно-покрывающий граф.

A graph G is called a **dominant-covering graph** if $\gamma(H) = \tau(H)$ for every isolate-free induced subgraph H of G. Here $\gamma(G)$ is the *domination number*, and $\tau(G)$ is the *vertex-coverung number* of G.

Dominant-matching graph — домиинантный граф паросочетания.

A graph G is called a **dominant-matching graph** if $\gamma(H) = \mu(H)$ for every isolate-free induced subgraph H of G. Here $\gamma(G)$ is the *domination number*, and $\mu(G)$ is the *matching number* of G.

Dominating cycle — доминирующий цикл.

1. A cycle C in G is called a **dominating cycle** if the vertices of the graph $G - C$ are *independent*.

2. A cycle C in G is called a **dominating cycle** if $V(C)$ is a dominating set of G.

3. In some papers, a **dominating cycle** is defined as a cycle such that every edge in G is incident with a vertex in C.

Other name is **Covering cycle**.

***f*-Dominating cycle** — f-доминирующий цикл.

Let f be a non-negative integer-valued function defined on $V(G)$. Then a cycle C is called an ***f*-dominating cycle** if $d_G(C) \leq f(v)$ for every $x \in V(G)$. By taking an appropriate function as f, we can give a unified view to many cycle-related problems. If f is a constant function taking the value 0 (resp. 1), then an f-dominating cycle is a *Hamiltonian cycle* (resp. a dominating cycle) of G.

Dominating function — доминирующая функция.

A **signed dominating function** of G is defined as $g: V \to \{\pm 1\}$ satisfying $g(N[v]) \geq 1$ for all $v \in V$. A signed dominating function g is minimal if there does not exist a signed dominating function $h \neq g$ satisfying $h(v) \geq g(v)$ for every $v \in V$. The **signed domination number** of a graph G is defined as $\gamma_s(G) = \min\{g(V)|\ g$ is a minimal signed dominating function of $G\}$.

A **minus dominating function** is defined as a function $g: V \to \{0, \pm 1\}$ such that $g(N[v]) \geq 1$ for all $v \in V$. Similarly, we can define a minimal minus dominating function, the minus domination number $\gamma^{-}(G)$ of G.

A **majority dominating function** of a graph G is defined as a funmction $g: V \to \{\pm 1\}$ such that for at least half the vertices $v \in V$, $g(N[v]) \geq 1$. Similarly, a minimal majority dominating function and the majority domination number $\gamma_{maj}(G)$ of G are defined.

Dominating graph — доминирующий граф.

A **dominating graph** $D(G)$ is the graph with $V(D(G)) = V(G) \cup S(G)$, where $S(G)$ is the set of all minimal dominating sets of G, and two vertices $u, v \in V(D(G))$ are adjacent if $u \in V(G)$ and $v = D$ is a minimal dominating set containing u.

Dominating path — доминирующий путь.

A path P in G is called **dominating** if the vertices of the graph $G - P$ are *independent*.

Dominating set — доминирующее множество.

A set $S \subseteq V$ is a **dominating set** of G if for all $v \in V \setminus S$ there is a vertex $u \in S$ such that $(u, v) \in E(G)$. The minimum cardinality of a **dominating set** of G is called the **domination number** of G. It is well-known that determining the domination number of a graph is NP-hard.

A dominating set S is called **independent** if the induced subgraph $\langle S \rangle$ is empty; **total** if $\langle S \rangle$ has no isolated vertex and **connected** if $\langle S \rangle$ is connected. The minimum cardinality taken over all minimal independent (total/connected) dominating sets in G is called the **independent** (**total**/**connected**) domination number of G and is denoted by γ_i (γ_t/γ_c).

For a vertex v of a graph $G = (V, E)$, the **domination number** $\gamma_v(G)$ **of** G **relative to** v is the minimum cardinality of a dominating set in G that contains v. The **average domination number** of G is

$$\gamma_{av}(G) = \frac{1}{|V|} \sum_{v \in V} \gamma_v(G).$$

The **independent domination number** $i_v(G)$ **of** G **relative to** v is the minimum cardinality of a maximal independent set in G that contains v. The **average independent domination number** of G is

$$i_{av}(G) = \frac{1}{|V|}\sum_{v\in V} i_v(G).$$

A **dominating set of a digraph** $\vec{G}$ is a set S of vertices such that for every vertex $v \notin S$ there exists some $u \in S$ with $(u,v) \in E(\vec{G})$. The **domination number** $\gamma(\vec{G})$ **of** $\vec{G}$ is defined as the cardinality of the smallest dominating set.

The dominating set problem is *NP*-complete on arbitrary graphs. It is also *NP*-complete on several classes of graphs, including planar graphs, bipartite graphs and *chordal* graphs. The problem can be solved in polynomial time on, for example, AT-free graphs, *permutation graphs*, *interval graphs*, and trees.

See also *k-restricted total dominating number.*

Dominating vertex — доминирующая вершина.

Dominating walk — доминирующий маршрут.

A **dominating walk** W in a graph G is a walk such that for each $v \in V(G)$, either $v \in V(W)$ or v is adjacent to a vertex of W.

k-**Dominating cycle** — n-доминирующее множество.

See *Weak k-covering cycle.*

n-**Dominating set** — n-доминирующее множество.

A set D of vertices in a graph G is defined to be an n-**dominating set** of G if every vertex of $V(G) - D$ is within distance n from some vertex of D. The minimum cardinality among all n-dominating sets of a graph G is called the n-**domination number** of G and is denoted by $\gamma_n(G)$, while the maximum cardinality among all minimal n-dominating sets of a graph G is called the **upper** n-**domination number** of G and is devotedly $\Gamma_n(G)$. A set D of vertices in a graph G is called n-**independent** if $d(u,v) > n$ for all $u,v \in D$. The **independent** n-**domination number** of a G, denoted by $i_n(G)$, is the minimum cardinality among all maximal n-independent sets of a graph G, while the n-**independence number** of G, denoted by $\beta_n(G)$, is the maximum cardinality among all maximal n-independent sets of a graph G.

Domination graph — граф доминирования.

Let D be a digraph with a vertex set $V(D)$ and an arc set $A(D)$. If $(x,y) \in A(D)$, then x dominates y. A vertex is also considered to dominate itself. The **domination graph** of D, $dom(D)$, is the graph, where $V(dom(D)) = V(D)$ and $\{x,y\} \in E(dom(D))$, whenever x

and y dominate all other vertices in D.

Domination graph (of a tournament) — граф доминирования.

The vertices x and y dominate a *tournament* T if for all vertices $z \neq x, y$, either x beats z or y beats z. Let $\mathrm{dom}(T)$ be a graph on the vertices of T with edges between pairs of vertices that dominate T. This graph is called a **domination graph**. It is known that $\mathrm{dom}(T)$ is either an odd *cycle* with possible *pendant vertices* or a *forest* of *caterpillars*. Also a **domination graph** is the *complement* of the *competition graph* of the tournament.

Domination number — число доминирования.

See *Dominating set*.

Domination number relative to v — число доминирования относительно вершины v.

See *Dominating set*.

Domination perfect graph — совершенный граф доминирования.

See *Hereditary class of graphs*.

Domination subdivision number — доминирующее число подразбиения.

The **domination subdivision number** $sd_\gamma(G)$ of a graph G is the minimum number of edges that must be *subdivided* (where an edge can be subdivided at most once) in order to increase the *domination number*. It is known that this number is at most 3 for any tree.

See also *Independence subdivision number*.

γ**-Domination critical graph** — γ-доминирующий критический граф.

A graph is said to be γ**-domination critical graph**, or just γ-critical, if $\gamma(G) = \gamma$ и $\gamma(G + e) = \gamma - 1$ for every edge e in the complement $\bar{G}$ of G.

n**-Domination number** — число n-доминирования.

See *n-Dominating set*.

Dominator — обязательный предшественник, доминатор.

Given a *cf-graph* G with the *initial* vertex r and two vertices x and y in G. The vertex x is a **dominator** of the vertex y (x **dominates** y) if any path $P(r, y)$ contains x; if $x \neq y$ then x is a **proper dominator** of y. We denote as $DOM(x)$ the set of all dominators of the vertex x. The vertex x is the **immediate dominator** of y, denoted as $IDOM(y)$, if x is a proper dominator of y and no vertex $z \notin \{x, y\}$ exists such that $x \in DOM(z)$ and $z \in DOM(y)$.

Dominator tree — дерево доминаторов, доминаторное дерево.

The **dominator tree** T_D of a dag G is a concise representation of the dominance relationship, where, for each vertex x in G, the parent of x in T_D corresponds to $DOM(x)$. Hence, a vertex x dominates a vertex y in G iff x is an *ancestor* of y in T_D. Given a set $U \subseteq V$, a vertex $d \in V$ is the **nearest common dominator** of U if the following two conditions hold: (1) d dominates all vertices of U; (2) there is no vertex $d' \neq d$ that dominates all vertices of U and is dominated by d.

Domino — домино.

1. A graph which consists of two cycles of length 4 with a common edge is called a **domino**.

2. A **domino** is a graph in which every vertex is contained in at most two maximal cliques. The class of such graphs properly contains the line graphs of bipartite graphs. Every **domino** is a *line graph* of a *multigraph*, and hence *claw-free*. The class of the **domino** graphs is hereditary; i.e. if G is a domino, then every induced subgraph of G is also a domino.

Domsaturation number — число доминирующего насыщения.

See *Restricted domination number*.

Double competition number — число двойной конкуренции.

See *Competition graph*.

Double dominating set — двойное доминирующее множество.

A set $S \subset V(G)$ is a **double dominating set** for G if every vertex in V is dominated by at least two vertices in S. The minimum cardinality of a double dominating set is a **double domination number**, denoted $dd(G)$. Obviously, every **double dominating set** is also a *dominating set*. Note that the concept of double domination can be extended to **multiple domination (*h*-tuple domination)** by requiring that each vertex in V be dominated at least h times.

The concept of double domination in graphs was defined by Harary and Haynes (2002).

A double dominating set is **exact** if every vertex of G is dominated exactly twice. The problem of existence of an exact double dominating set is an NP-complete problem.

Double domination number — двойное число доминирования.

See *Double dominating set*.

Double edge dominating set — двойное рёберное доминирующее множество.

A set $F \subseteq E$ is a **double edge dominating set** of G if each edge in E is adjacent to at least two edges in F. The **double edge domination number** $dd_e(G)$ of G is the minimum cardinality of a double edge dominating set of G.

Double edge domination number — двойное рёберное доминирующее множество.

See *Double edge dominating set.*

Double rotation — двойное вращение.

Double ray — двойной луч.

See *Ray.*

Double star — двойная звезда.

A **double star** $S_{k,k}$ is a tree on $2k+2$ vertices consisting of two adjacent vertices u and v of degree $k+1$ and 2k end vertices.

Doubly chordal graph — дважды хордальный граф.

A vertex v of a graph G is **doubly simplicial** if v is *simplicial* and has a *maximum neighbor*. A linear ordering $(v_1, \ldots, v_n)$ of the vertices of G is **doubly perfect** if for all $i \in \{1, \ldots, n\}$ v_i is a doubly simplicial vertex of G_i (a subgraph induced by $v_i, \ldots, v_n$). A graph G is **doubly chordal** if it admits a doubly perfect ordering. It is known that the powers of a **doubly chordal graph** are **doubly chordal**.

Doubly perfect ordering — двойной совершенный порядок.

See *Doubly perfect graph.*

Doubly regular tournament — двойной регулярный турнир.

See *Tournament.*

Doubly simplicial vertex — двойная симплициальная вершина.

See *Doubly chordal graph.*

Dual hypergraph — двойственный гиперграф.

A **dual hypergraph** $\mathcal{H}^*$ has $\mathcal{H}$ as its vertex set and $\{e \in \mathcal{H};\ v \in e\}$ $(v \in V)$ as its edges.

Doubly stochastic matrix — бистохастическая матрица.

Dual graph — двойственный граф.

Dual hypertree — двойственное гипердерево.

See *Hypertree.*

Dual map — двойственная карта.

A **dual map** of a connected planar graph G is the map G^* constructed as follows. We select a point x_F in each of the faces F of G; these will be the vertices of G^*. Also we select a point p_e on each edge e of G. We connect each point p_e to the points $x_F, x_{F'}$ by Jordan curves J_e, J'_e interior to F and F', respectively, where F, F' are two faces adjacent to e. If $F = F'$ (i.e. the same face of G bounds e from both sides), then J_e, J'_e should connect p_e to x_F such that they leave p_e on different sides of e (this happens, if e is a *cutting edge*. Moreover, let us choose J_e, J'_e such that the arcs J_e connecting X_F to points p_e on the boundary of F should have no point in common other than x_F. Set $e^* = J_e \cup J'_e$ and $E(G^*) = \{e^* :\ e \in E(G)\}$. Then G^* is a planar map and is, essentially, uniquely defined, i.e. if $\hat{G}^*$ is another dual map of G, then there is a homeomorphism φ of the plane onto itself such that $\varphi(x) = x$ for each $x \in V(G)$, $\varphi(e) = e$ for each $e \in E(G)$, $\varphi(V(G^*)) = V(\hat{G}^*)$ and if $\hat{e}^*$ is the edge corresponding to e^* in $\hat{G}^*$, then $\varphi(e^*) = \hat{e}^*$. The dual of G^* is G. The above construction and these last assertions involve much from plane topology that we accept here without proof.

Dual matroid — матроид двойственный

For a *matroid* $\mathcal{M}$ on a set E with a family $\mathcal{B}$ of *bases*, another family $\mathcal{B}^*$ defined by

$$\mathcal{B}^* = \{E \setminus B : B \in \mathcal{B}\}$$

is shown to be the family of bases of another matroid $\mathcal{M}^*$ on the same set E, which is called the **dual matroid**. Obviously, $(\mathcal{M}^*)^* = \mathcal{M}$.

A *base* and a *circuit* of $\mathcal{M}^*$ are called a **cobase** and a **cocircuit** of $\mathcal{M}$, respectively.

Dual tournament — двойственный турнир.

Let $T = (V, A)$ be a finite *tournament* with n vertices. The **dual** of T is the tournament $T^* = (V, A^*)$, defined by: for all $x, y \in V$, $(y, x) \in A^*$ if and only if $(x, y) \in A$.

The tournament T is **selfdual**, when T is isomorphic to T^*.

Dually chordal graph — двойственно-хордальный граф.

A vertex $u \in N[v]$ ($N[v]$ is a *closed neighborhood* of v) is a **maximum neighbor** of v if for all $w \in N[v]$, $N[w] \subseteq N[u]$ holds (note that $u = v$ is not excluded). A linear ordering $(v_1, \ldots, v_n)$ of V is a **maximum neighborhood ordering** of G if for all $i \in \{1, \ldots, n\}$, there is a maximum neighbor $u_i \in N_i[v_i]$; i.e.,

$$\text{for all } w \in N_i[u_i],\ N_i[w] \subseteq N_i[u_i] \text{ holds.}$$

The graphs with maximum neighborhood ordering are dual to chordal graphs and called **dually chordal graph**.

The k**-th power** G^k, $k \geq 1$, of G has the same vertices as G, and two distinct vertices are joined by an edge in G^k if and only if their distance in G is at most k. It is known that any power of a dually chordal graph is dually chordal.

Dually compact closed class of graphs — двойственно-компактно замкнутый класс графов.

See *Compact closed class of graphs*.

Dudeney set — множество Дьюдене.

Dudeney set in K_n is a set of *Hamilton cycles* with the property that every path of length two (2-path) in K_n lies on exactly one of the cycles. We call the problem of construction a Dudeney set in K_n for all natural numbers **"Dudeney's round table problem"**.

Dudeney's round table problem — проблема Дьюдене круговых таблиц.

See *Dudeney set*.

E

Eccentric graph — граф эксцентриситетов.

A graph G is an **eccentric graph** if every vertex of G is an eccentric vertex. If G is self-centered, then G is an eccentric graph.

Eccentric sequence — последовательность эксцентриситетов.

For a graph G, the **eccentric sequence** of G is the sequence of the *vertex eccentricities* in ascending order. A vertex v is a **mode vertex** if the eccentricity of v appears at least as often in the eccentric sequence as any other eccentricity. The **mode** of a graph G is the subgraph induced by its mode vertices.

If u is a farthest vertex from v, then u is an **eccentric vertex** of v, and we say that u is an **eccentric vertex** if it is an eccentric vertex of at least one vertex of G. In a graph G with a vertex set $V(G)$, vertices in a set $X \subseteq V(G)$ are **mutually eccentric** if for all pairs $u, v \in X$, u is eccentric for v and v is eccentric for u.

Another way of describing an eccentric vertex is to say that u is an eccentric vertex of v if $d(u, v) = e(v)$. If each vertex of G has exactly one eccentric vertex, then G is **unique eccentric point graph**. A simple class of unique eccentric point graphs are the paths P_{2n} on an even number of vertices.

Eccentricity of a vertex — эксцентриситет вершины.

Let $d(x, y)$ be the distance in a graph G. Then the **eccentricity** $e(v)$ **of a vertex** v is the maximum over $d(v, x),\ x \in V(G)$. The minimum over the eccentricities of all vertices of G is the **radius** $rad(G)$ of G, whereas the maximum is the **diameter** $diam(G)$ of G. A pair x, y of vertices of G is called **diametral** iff $d(x, y) = diam(G)$. A chain in G which length is equal to $diam(G)$ is called a **diametral chain**.

See also *Quasi-diameter, Quasi-radius.*

Edge — ребро.

Edge adding — добавление ребра.

(a, d)**-Edge-antimagic total graph** — (a, d)-рёберно-антимагический тотальный граф.

See *Super* (a, d)*-edge-antimagic total labeling.*

(a, d)**-Edge-antimagic total labeling** — (a, d)-рёберно-антимагическая тотальная раскраска.

See *Super* (a, d)*-edge-antimagic total labeling.*

Edge chromatic number — реберно-хроматическое число.

Edge clique cover — покрытие рёбер кликами.

An **edge clique cover** of G is a collection of cliques that covers all edges of G. The minimum number of cliques in an edge clique cover is called the **edge clique cover number** and denoted by $\theta_e(G)$.

Edge clique cover number — число рёберно-кликового покрытия.

See *Edge clique cover*.

Edge colourable graph — реберно-раскрашиваемый граф.

Edge k-colourable graph — реберно-k-раскрашиваемый граф.

Edge k-colouring — реберная k-раскраска.

k-Edge-connected graph — k -рёберно-связный граф.

A graph G is **k-edge-connected graph** if there is no edge cut set of G of cardinality less than k.

Edge connectivity — реберная связность.

We define the **edge-connectivity**, $K_e(G)$ or K_e, for the connected graph G to be the size of the smallest *cut-set* of G. G is said to be **h-edge-connected** for any positive integer h satisfying $h \leq K_e(G)$. We denote the smallest degree of any vertex in a graph by δ. Since the set of edges incident with any vertex forms a cut-set, we have $\delta \geq K_e(G)$.

For any connected graph G:

$$K_v(G) \leq K_e(G) \leq \delta.$$

$K_v(G)$ is the vertex-connectivity.

The **local-edge-connectivity** $\lambda(u, v)$ of two vertices u and v in a graph or digraph D is the maximum number of edge-disjoint $u-v$-paths in D, and the edge-connectivity of D is defined as $\lambda(D) = \min\{\lambda(u,v) | u, v \in V(D)\}$.

Edge connectivity number — число реберной связности.

Edge-cordial graph — рёберно-сердечный граф.

See *Edge-cordial labeling*.

Edge-cordial labeling — рёберно-сердечная разметка.

Yilmaz and Cahit introduced in 1997 **edge-cordial labeling** as a weaker version of *edge graceful labeling*. Let f be a binary edge labeling of graph $G = (V, E)$; that is, $f:\ E \to \{0, 1\}$. Let the induced vertex labeling be given by $f^+(v) = \sum_{uv \in E} f(uv) \pmod 2$, where $v \in V$. The function f is called an edge-cordial labeling of G if the following two properties hold:

(1) $|e_f(0) - e_f(1)| \leq 1$, and

(2) $|v_f(0) - v_f(1)| \leq 1$,
where $e_f(0)$ and $e_f(1)$ denote the number of edges, and $v_f(0)$ and $v_f(1)$ denote the number of vertices labeled with 0 or 1, respectively. The graph G is called **edge-cordial** if it admits an edge-cordial labeling.

Edge core — реберное ядро.

Edge covering — реберное покрытие.

Edge critical graph — реберно-критический граф.

Edge cut, edge cut set — разрез.

A set X of edges of a graph G is called an **edge cut** if $G \setminus X$ has more connected components than G. An **edge cut** of G minimal under inclusion is called a **cocircuit** of G. A **vertex star**, which is the set of edges in G incident to that vertex, is associated with each vertex in G.

Edge-degree — рёберная степень.

The **edge-degree** $\xi_G(e)$ of the edge $e = (uv) \in E(G)$ is defined by

$$\xi_G(e) = deg(u) + deg(v) - 2.$$

Edge density — рёберная плотность.

Given an undirected graph $G = (V, E)$ and a nonempty subset $X \subseteq V$, the **edge density** of X is given by

$$\rho(X) = |V||E_X|/|X||V \setminus X|,$$

where E_X is the set of all edges with one end in X and the other end in $V \setminus X$.

Edge dominating set — рёберное доминирующее множество.

A set F of edges in a graph G is an **edge dominating set** if every edge in $E - F$ is adjacent to at least one edge in F. The **edge domination number** $\gamma^1(G)$ is the minimum cardinality of the edge dominating set of G.

Edge domination number — рёберное доминирующее множество.

See *Edge dominating set.*

Edge-graceful labeling — рёберно-грациозная разметка.

Edge-graceful labeling of graphs was introduced by Lo in 1985. Let $G(V, E)$ be a simple graph with $|V| = p$ and $|E| = q$. Then, G is said to be edge-graceful if there exits a bijection $f: E \to \{1, 2, \ldots, q\}$ such that the induced mapping $f^+: V \to \{0, 1, 2, \ldots, p-1\}$, defined by

$$f^{+}(v) = \sum_{uv \in E} f(uv) \pmod p$$

is also a bijection.

Edge graph — реберный граф.

Edge group of a graph — реберная группа графа.

Edge incidency matrix — матрица смежности ребер.

Edge-independent number — число рёберной независимости.

The same as *Matching number*.

Edge isomorphic graphs — реберно изоморфные графы.

Edge-isoperimetric problem — рёберно-изопериметрическая задача.

Given a graph $G = (V, E, \partial)$ having the vertex-set V, edge-set E and boundary-function $\partial : E \to \binom{V}{2}$ which identifies the pair of vertices incident to each edge, we define

$$\Theta(S) = \{e \in E : \partial(e) = \{v, w\}, v \in S \& w \notin S\}.$$

Then, the **edge-isoperimetric problem** is to minimize $|\Theta(S)|$ over all $S \subseteq V$ such that $|S| = k$ for a given $k \in Z^{+}$.

Edge kernel — реберное ядро.

Edge-labeling — разметка рёбер.

See *Labeling*.

Edge list — список ребер.

Edge-magic total graph — реберно-магический тотальный граф.

An **edge-magic total labeling** on G will mean a one-to-one map λ from $V(G) \cup E(G)$ onto the integers $1, 2, \ldots, v + e$ with a property that for any edge (x, y)

$$\lambda(x) + \lambda(x, y) + \lambda(y) = k$$

for some constant k. It will be convenient to call $\lambda(x)+\lambda(x,y)+\lambda(y)$ the edge sum of (x, y), and k (constant) a magic sum of G. A graph is called **edge-magic total** if it admits any edge-magic total labeling. It is known that *caterpillars* and all cycles C_n are edge-magic total. See also *Vertex-magic total graph.*

Edge-magic total labeling — реберно-магическая тотальная раскраска.

See *Edge-magic total graph.*

Edge monochromatic class — реберный цветной класс.

Edge of attachment — соединяющее ребро.

Edge of a hypergraph — ребро гиперграфа.

See *Hypergraph*.

Edge-ordering — рёберное упорядочение.

See *Oscillation of graph*.

Edge-pancyclic graph — рёберно-панциклический граф.

See *Edge-pancyclicity*.

Edge-pancyclicity — рёберная панцикличность.

A graph G is called **edge-pancyclic** if every edge of G lies on a cycle of every length from 4 to n.

Edge path cover — дуговое путевое покрытие.

Let $\mathcal{P} = \{P_1, \ldots, P_k\}$ be a set of paths in a digraph D. $\mathcal{P}$ is an **edge path cover** of D iff $\{E(P_1), \ldots, E(P_k)\}$ is a partition of $E(D)$.

$$pn_e(D) = \min\{|\mathcal{P}| : \mathcal{P} \text{ is an edge path cover of } D\}$$

is the **edge path number** of D.

Edge path number — дуговое путевое покрытие.

See *Edge path cover*.

Edge ranking number — рёберно-ранговое число.

See *Edge t-ranking*.

Edge reconstructibility — реберная реконструируемость.

Edge regular graph — реберно-регулярный граф.

Let X be a graph and $A = Aut(X)$ be the full automorphism group of X. Let G be a subgroup of the full automorphism group A of the graph X. We call X G-edge regular if the action of G on $E(X)$ is regular. When $G = Aut(X)$, we remove the prefix $G-$ and call X **edge regular**.

Edge symmetric graph — реберно-симметрический граф.

k**-Edge connected graph** — k-реберно связный граф.

A graph G is k**-edge connected** if there exist k internally edge-disjoint chains between every pair of distinct nodes in G.

Edge-cover — реберное покрытие.

Given a graph (hypergraph), an **edge-cover** is a set of edges containing all vertices.

Edge-critical graph — реберно-критический граф.

A graph G is called **edge-critical** with respect to a property P if G has it but, on removing any edge, the resulting graph will not have the property P. **Point-critical** is defined analogously.

Edge-forwarding index — рёберно-продвигающий индекс.

See *Routing*.

Edge-graceful graph — реберно-грациозный граф.

A graph $G(V, E)$ is said to be **edge-graceful** if there exists a bijection

$$f : E \to \{1, 2, \ldots, |E|\}$$

such that the induced mapping

$$f^+ : V \to \{0, 1, \ldots, |V| - 1\}$$

given by

$$f^+(x) = \sum\{f(xy)|xy \in E\} \pmod{|V|}$$

is a bijection.

One of the well known conjectures came from Lee in 1989:

Conjecture (Lee). Every tree with an odd number of vertices is edge-graceful.

This conjecture has not been proved yet.

Edge-integrity — реберная целостность.

The **edge-integrity** of a graph G is

$$I'(G) = \min\{|S| + m(G - S) : S \subset E\},$$

where $m(H)$ denotes the maximum order of a component of H.

Edge-ranking of a graph — реберное упорядочение графа.

Edge t-ranking — реберное t-ранжирование.

Let $G = (V, E)$ be a graph and t be a positive integer. An **edge t-ranking** is an edge coloring $c' : E \to \{1, \ldots, t\}$ such that for every pair of edges e and f with $c'(e) = c'(f)$, there is an edge g on every path between e and f with $c'(g) > c'(e)$. The **edge ranking number**, $\chi'_r(G)$, is the smallest value of t such that G has an edge t-ranking.

See also *Vertex t-ranking*.

Edge space — пространство рёбер.

The **edge space** $\mathcal{E}(G)$ of a simple graph $G = (V(G), E(G))$ is a power set of its edges $E(G)$ endowed with the structure of a vector space over the two-element field $F_2 = \{0, 1\}$. Addition in $\mathcal{E}(G)$ is a symmetric difference of sets, and zero is the empty set.

Edge-superconnectivity — рёберная суперсвязность.

Superconnectivity is a stronger measure of connectivity. A maximally edge-connected graph is called **super-**λ if every *edge cut* $(C, \bar{C})$ of cardinality $\delta(G)$ satisfies either $|C| = 1$ or $|\bar{C}| = 1$. In order to measure the super edge-connectivity, we use the following parameter:

$$\lambda_1(G) = \min\{|(C, \bar{C})|,\ (C, \bar{C}) \text{ is a nontrivial edge cut}\}.$$

We define the **edge-superconnectivity** of a graph G as the value of $\lambda_1(G)$.

Efficient dominating set — эффективно-доминирующе множество.

See *t-code (in graph).*

Effectively solvable problem — частично разрешимая задача.

See *Decision problem.*

Eigenvalue of a graph — собственное значение графа.

See *Characteristic polinomial of a graph.*

1-edge hamiltonian graph — 1-рёберно-гамильтоновый граф.

See *1-hamiltonian graph.*

Element of a graph — элемент графа.

A vertex or an edge (arc) of a graph (digraph).

Elementary homomorphism — элементарный гомоморфизм.

Elementary Petri net — элементарная сеть Петри.

Embedding of a graph — укладка графа, вложение графа.

An **embedding** of a graph G (into a *complement* $\bar{G}$) is a permutation σ on $V(G)$ such that if an edge xy belongs to $E(G)$, then $\sigma(x)\sigma(y)$ does not belong to $E(G)$. If there exists an embedding of G, we say that G is embeddable or that there is a **packing** of two copies of the graph G (of order n) into the complete graph K_n.

Emptiness problem — проблема пустоты.

Empty deadend — пустой тупик.

Empty deadlok — пустой тупик.

Empty graph — пустой граф, вполне несвязный граф, регулярный степени 0 граф.

This is a graph which has one vertex and no edges.

Empty hypergraph — пустой гиперграф.

This is a hypergraph which has no points and no edges.

Empty loop — пустой цикл.

Empty marking problem — проблема нулевой разметки.

The **empty marking problem** for Petri nets consists in finding an algorithm for deciding whether or not the marking $(0, \ldots, 0)$ is a reachable one for a given Petri net.

Empty string — пустая цепочка.

See *String*.

Empty subgraph — пустой подграф.

See *Independent set*.

Empty symbol — пустой символ.

Empty tree — пустое дерево, вырожденное дерево.

See *Empty graph, Trivial tree, Degenerate tree*.

Enabled transition — разрешенный переход, переход, готовый сработать.

See *Petri net*.

Endblock — концевой блок, висячий блок.

An **endblock** of G is a *block* containing exactly one *cutvertex* of G.

End-edge — концевое ребро, висячее ребро.

Enclosure transition — объемлющий переход.

Endline graph — концевой граф.

Let $G = (V, E)$ be a graph and $V(G) = \{v_1, \ldots, v_n\}$. We added to G n new vertices and n edges $\{u_i, v_i\}$, $(i = 1, 2, \ldots, n)$, where u_i are different from any vertex of G and from each other. A new graph G^+ with $2n$ vertices is called the **endline graph** of G.

See also *Middle graph*.

Endomorphism — эндоморфизм.

Given a graph (digraph) G, **endomorphism** is a homomorphism of G into itself. The set of all **endomorphisms** of G with composition as multiplication, forms a semigroup denoted by $\mathrm{End}(G)$.

Endpoint, end-vertex — висячая вершина.

Given a graph G, a vertex with *degree* 1 is called an **endpoint** (or **end-vertex**).

Endpoints of a path (chain) — концевые вершины пути (цепи).

For a given path (chain) $P_m = v_1, v_2 \cdots v_m$, vertices v_1 and v_m are the **endpoints** of that path (chain).

Energy of graph — энергия графа.

Let G be a graph possessing n vertices and m edges. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$, be the *eigenvalues* of the *adjacency matrix* of G. The **energy** of G is defined as follows

$$\mathcal{E}(G) = \sum_{i=1}^{n} |\lambda_i|.$$

The eigenvalues of the n-vertex complete graph K_n are $\lambda_1 = n-1$, $\lambda_2 = \lambda_3 = \ldots = \lambda_n = -1$. Therefore, the energy $\mathcal{E}(K_n) = 2n-2$. Graphs with the property $\mathcal{E}(G) > 2n-2$ are called **hyperenergetic graphs**; such graphs exist for all $n \geq 8$.

Entire choice number — число целого выбора.

See *Entire colouring.*

Entire chromatic number — целое хроматическое число.

See *Entire colouring.*

Entire colouring — целая раскраска.

An **entire colouring** of a plane *multigraph* G is a map $\sigma: V \cup E \cup F \to S$, where S is a set of "colours", such that $\sigma(X) \neq \sigma(Y)$ whenever X, Y are incident or adjacent elements, i.e. a pair of adjacent vertices, a vertex-edge pair with the edge incident on the vertex, an edge-face pair with the edge on a boundary of the face, etc.; a face touching either another face or an edge only in a vertex is not considered as adjacency.

The **entire chromatic number**, χ_{vef}, is the least size of S admitting such a coloring. For example, if G consists of two copies of K_3 joined in a single common vertex, then $\chi_{vef}(G) = 6$. The **entire choice number**, $\hat{\chi}_{vef}$, is the least integer k such that if $L(A)$ is a set (*list*) of size k for each $A \in V \cup E \cup F$, then there exists an entire colouring σ of G with $\sigma(A) \in L(A)$ for each such A.

Entry — вход.

Entry node of a fragment — входная вершина фрагмента.

See *Fragment.*

Entry vertex — стартовая вершина, входная вершина.

Entry vertex of a subgraph — входная вершина подграфа.

Environment of a vertex — окружение вершины, окрестность вершины.

Equally coloured vertices — соцветные вершины.

Equiseparable trees — эквисепарабельные деревья.

Let T be a tree and e an arbitrary edge of T. Then $T - e$ consists of two components with $n_1(e)$ and $n_2(e)$ vertices. Conventionally, $n_1(e) \leq n_2(e)$. If $T^{'}$ and $T^{''}$ are two trees of the same order n and if their edges can be labelled so that $n_1(e_i^{'}) = n_1(e_i^{''})$ holds for all $i = 1, 2, \ldots, n-1$, then $T^{'}$ and $T^{''}$ are said to be **equiseparable**.

Equistable graph — эквиустойчивый граф.

A graph $G = (V, E)$ is **equistable** if there is a non-negative weight function w on V such that a set $S \subseteq V$ satisfies $w(S) = \sum_{v \in S} w(e) = 1$ if and only if S is maximal stable. The problem of recognizing equistable graphs in polynomial time is still open.

A graph G is **strongly equistable** if for each set $\emptyset \neq T \subseteq V$ such that T is not maximal stable, and for each constant $c \leq 1$, there is a non-negative weight function w on V such that $w(S) = 1$ for each maximal stable set S, and $w(T) \neq c$.

Equitable partition — справедливое разбиение.

Let G be a simple graph with a vertex set $V(G)$. A partition $\pi = (C_1, \ldots, C_t)$ of $V(G)$ is said to be **equitable** if, for all i and j, the number c_{ij} of edges from a vertex in C_i to C_j does not depend on the choice of the vertex in C_i. For an equitable partition π, we denote the quotient graph with respect to π by G/π. The matrix (c_{ij}) is called the adjacency matrix of the quotient graph G/π.

k-Equitable graph — k-справедливый граф.

See *k-Equitable labeling.*

k-Equitable labeling — k-справедливая разметка.

Let $G = (V, E)$ be a graph. A *labeling* of G is a function $f : V \to \{0, 1, \ldots, n\}$ such that for each edge $e = (u, v) \in E$, $f(e) = |f(u) - f(v)|$. Such a labeling is said to be **k-equitable** if it is a labeling of the vertices with the numbers 0 through $k-1$ such that, if v^i is the number of vertices labeled i, and e^i is the number of edges labeled i, then $|v^i - v^j| \leq 1$ and $|e^i - e^j| \leq 1$ for all i, j. A graph is said to be **k-equitable** if it has k-equitable labeling.

Equivalence of languages problem — проблема эквивалентности языков.

Equivalence relation — отношение эквивалентности.

A relation is an **equivalence relation** if it is *reflexive, symmetric*, and *transitive*. For an equivalence relation R in a set S, the **equivalence class** of a with respect to R is the set of all elements b in S such that aRb. When the relation R is understood, the equivalence class of a is denoted by $[a]$. The equivalence classes of an equivalence relation in S form a partition of S. If an equivalence relation R_1 is contained in another equivalence relation R_2 (i.e., if aR_1b implies aR_2b), then the partition formed by the equivalence classes with respect to R_1 is finer than the partition formed by the classes with respect to R_2.

Equivalent cycles — эквивалентные циклы.

See *Cycle.*

Equivalent derivations — эквивалентные выводы.

Equivalent grammars — эквивалентные грамматики.

Two grammars are **equivalent** iff they generate the same language.

Equivalent program schemata — эквивалентные схемы программы.

See *Program schemata.*

Equivalent programs — эквивалентные программы.

See *Program equivalence.*

Erdös–Gallai criterion — критерий Эрдёша–Галлаи.

A sequence of integers $d_1, \ldots, d_n$ with $n-1 \geq d_1 \geq \ldots \geq d_n$ is a *graphic sequence of numbers* iff
(1) $\sum_{i=1}^{n} d_i$ is even, and
(2)$\sum_{i=1}^{r} d_i \geq r(r-1) + \sum_{i=r+1}^{n} \min\{r, d_i\}$ for $r = 2, \ldots, n-1$.
(the r-th Erdös-Gallai inequality).

Essential arc — существенная дуга.

Essential independent set — существенное независимое множество.

An independent set Y in G is called an **essential independent set** (or **essential set** for simplicity), if there is $\{y_1, y_2\} \subseteq Y$ such that $dist(y_1, y_2) = 2$.

Euler graph — эйлеров граф.

The **Euler graph** is an infinite directed graph such that at level n there are $n+1$ vertices labelled from $(n, 0)$ through (n, n). The vertex (n, k) has $n+2$ total edges leaving it, with $k+1$ edges connecting it to vertex $(n+1, k)$ and $n-k+1$ edges connecting it to vertex $(n+1, k+1)$.

Eulerian chain — эйлерова цепь.

Eulerian circuit — эйлеров цикл.

An **Eulerian circuit** of a connected graph G is a circuit (cycle) that traverses each edge of G exactly once, although it may visit a vertex more than once.

Eulerian cycle — эйлеров контур, эйлеров цикл.

Eulerian digraph — эйлеров орграф.

Eulerian graph — эйлеров граф.

A connected undirected graph such that there exists a *traversal* of all its edges using each edge exactly once (i.e. *Eulerian circuit*) is called **Eulerian**. It is well-known that G is Eulerian (unicursal) if and only if all its vertices have even *degree.*

Eulerian trail — эйлеров маршрут.

Eulerian tour — эйлеров обход.

The same as *Eulerian circuit*.

Evaluated graph — перенумерованный граф.

Evaluation of a graph — укладка графа.

Even component — чётная компонента.

See *Odd component*.

Even contractile graph — чётно стягиваемый граф.

See *Contraction of even pair*.

Even graph — четный граф.

Even pair — четная пара.

Two nonadjacent vertices u, v of G form an **even pair** if no odd chordless path joining them exists in G.

Event — событие.

Event condition — условие события.

Event realization — реализация события.

Event-condition system — условно-событийная система.

Exact double dominating set — точно двойное доминирующее множество.

See *Double dominating set*.

Exact n-step domination graph — граф точного n-шагового доминирования.

A vertex u in a graph G is said to n-step dominate a vertex v if $d(u, v) = n$. If there exists a subset $S \subseteq V(G)$ such that each $v \in V(G)$ is n-step dominated by exactly one vertex in S, then G is an **exact n-step domination graph** and S is called an **exact n-step dominating set**.

Exact n-step dominating set — точно n-шаговое доминирующее множество.

See *Exact n-step dominating graph*.

$\gamma_t(G)$-Excellent graph — $\gamma_t(G)$-превосходный граф.

A graph G is called **$\gamma_t(G)$-excellent graph** if every vertex of G belongs to some *total dominating set* of minimal cardinality. A family of **$\gamma_t(G)$-excellent trees** (trees where every vertex is in some minimum dominating set) is properly contained in the set of i-excellent trees (trees where every vertex is in some minimum *independent dominating set*).

In general, for a graph G, let $\mathcal{P}$ denote a property of sets $S \subseteq V$ of vertices. We call a set S with the property $\mathcal{P}$ having { minimum, maximum } cardinality $\mu(G)$ a $\mu(G)$-set. A vertex is called μ-good if it is contained in some $\mu(G)$-set and μ-bad otherwise. A graph G is called **μ-excellent** if every vertex in V is μ-good.

μ-Excellent graph — μ-превосходный граф.
See *$\gamma_t(G)$-Excellent graph.*

Exceptional graph — исключительный граф.
A finite graph is said to be **exceptional** if it is connected, has at least eigenvalue $\lambda_{min} \geq -2$ and is not a generalized *line graph.*

Exclusion operation — операция исключения.

Execution of large-block schema — выполнение крупноблочной схемы.
See *Value of schema under interpretation*

Execution of Petri net — выполнение сети Петри.
See *Petri net.*

Execution sequence — последовательность исполнения (операторов), цепочка исполнения (операторов).
See *Value of schema under interpretation.*

Exhaustive search — перебор.
For discrete problems in which no efficient solution method is known, it might be necessary to test each possibility sequentially in order to determine if it is the solution. Such exhaustive examination of all possibilities is known as **exhaustive search**, **direct search**, or the **"brute force" method**. Unless it turns out that *NP*-problems are equivalent to *P*-problems, which seems unlikely but has not yet been proved, *NP*-problems can only be solved by exhaustive search in the worst case.

Exit — выход.

Exit node of a fragment — выходная вершина фрагмента.
See *Fragment.*

Exponent of a digraph — Экспонента орграфа.
See *Primitive directed graph.*

Expression dag — дэг выражения.
See *DAG.*

n-Extendable graph — n-расширяемый граф.
Let G be a graph on v vertices and n be an integer such that $0 \leq n \leq (v-2)/2$. Then G is **n-extendable** if every *matching* of size n in G is contained in a *perfect matching* of G. Every n-extendable graph is also $(n-1)$-extendable and also any 2-extendable graph is

either 1-extendable bipartite or bicritical.

Extended odd graph — расширенный нечетный граф.

Extended regular expression — расширенное регулярное выражение.

Exterior face — внешняя грань.

See *Planar graph*.

Exterior of a cycle — внешность цикла.

External input place — внешнее входное место.

External output place — внешнее выходное место.

External place — стороннее место.

External stability set — внешне устойчивое множество, доминирующее множество.

External vertex — висячая вершина.

F

Face — грань.

See *Planar graph.*

(a, d)**-Face antimagic graph** — (a, d)-граневый антимагический граф.

A connected plane graph $G = (V, E, F)$ is said to be (a, d)**-face antimagic** if there exist positive integers a, b and a bijection

$$g: E(G) \to \{1, 2, \ldots, |E(G)|\}$$

such that the induced mapping $w_g^*: F(G) \to W$ is also a bijection, where $W = \{w^*(f): f \in F(G)\} = \{a, a+d, \ldots, a+(|F(G)|-1)d\}$ is the set of weights of a face. If $G = (V, B, F)$ is (a, d)-face antimagic and $g: E(G) \to \{1, 2, \ldots, |E(G)|\}$ is the corresponding bijective mapping of G, then g is said to be an (a, d)-face antimagic labeling of G.

The weight $w^*(f)$ of a face $f \in F(G)$ under an edge labeling

$$g: E(G) \to \{1, 2, \ldots, |E(G)|\}$$

is the sum of the labels of edges surrounding that face.

Facet — грань (плоского графа).

Facial cycle — цикл грани.

The boundary of a face of a plane graph is called a **facial cycle**.

Factor-critical graph — фактор-критический граф.

A graph $G = (V, E)$ is **factor-critical** if $G - v$ has a *perfect matching* for every vertex $v \in V(G)$.

1-Factor — 1-фактор.

For a given graph, **1-factor** is 1-regular *spanning subgraph.* The **1-factor** is often referred to as *perfect matching.* The existence of perfect matchings in bipartite graphs is a subject of the celebrated König-Hall theorem. See also *k-Factor.*

k**-Factor of a graph** — k-фактор графа.

A *spanning subgraph* H of a graph G is a k**-factor** if all vertices of H have degree k. A graph G is k**-factorable** (factorizable) if its edges can be partitioned into k-factors. A classical result of Petersen (1891) states that any $2k$-regular graph is 2-factorable.

Let g and f be integer-valued functions defined on $V(G)$ such that $0 \le g(x) \le f(x)$ for every $x \in V(G)$. A (g, f)**-factor** of G is a spanning subgraph F of G such that $g(x) \leq d_F(x) \leq f(x)$ for every $x \in V(G)$. If G itself is a (g,f)-factor, then G is called a (g,f)**-graph**. A (g,f)**-factorization** $\mathcal{F} = \{F_1, \ldots, F_m\}$ of G is a partition of G into edge-disjoint (g,f)-factors $F_1, \ldots, F_m$. Let H be a subgraph of G with m edges, then $\mathcal{F}$ is called **orthogonal** to H if each F_i has exactly one edge in common with H.

(g,f)**-Factor** — (g,f)-фактор.

See *k-Factor of a graph.*

k**-Factorable graph** — k-факторизуемый граф.

See *k-Factor of a graph.*

k**-Factorizable graph** — k-факторизуемый граф.

The same as *k-Factorable graph.*

(g,f)**-Factorization** — (g,f)-факторизация.

See *k-factor of a graph.*

Factor-critical graph — фактор-критический граф.

A graph G is called **factor-critical** if G has no 1-factor but $G - x$ has 1-factor for each vertex x.

Factor-cf-graph — фактор-уграф.

The same as *Factor-control-flow-graph.*

Factor-control-flow-graph — фактор-уграф.

Let R be a set of alts of a *cf-graph* G such that every node of G belongs to a single alt from R, i.e. R forms a partition of $V(G)$.

The cf-graph G' in which $V(G') = R$ is said to be **obtained by reduction of alts** R **into nodes** (notation $G' = R(G)$) if the following properties hold:

(1) for any $C_1, C_2 \in R$, $(C_1, C_2) \in A(G')$ if and only if there are $p_1 \in C_1$ and $p_2 \in C_2$ such that $(p_1, p_2) \in A(G)$,

(2) $C \in R$ is the initial node of G' if and only if C contains the initial node of G,

(3) $C \in R$ is the terminal node of G' if and only if C contains the terminal node of G.

The cf-graph G' is called a **factor-control-flow-graph** (or **factor-cf-graph**) of the cf-graph G with respect to R.

Let R be a set of mutually disjoint alts of a cf-graph G that form a partitation of some subset $Y \subset V(G)$. Then $R(G)$ is defined as cf-graph $R'(G)$, where $R' = R \bigcup \{\{p\} : p \in V(G) \setminus Y\}$.

Factor-graph — фактор-граф, граф Герца.

The same as *Condensation.*

Factorization — факторизация.

Factorization of a graph — факторизация графа.

1-Factorization of K_{2n} — один-факторизация графа K_{2n}.

A **one-factorization** of K_{2n} is a partition of the edge-set of K_{2n} into $2n-1$ *one-factors*. A **perfect one-factorization (P1F)** is a one-factorization in which every pair of distinct one-factors forms a Hamiltonian cycle of K_{2n}. P1Fs of K_{2n} are known to exist when $2n-1$ or n is prime, and for $2n \in \{16, 28, 36, 40, 50, 126, 170, 244, 344, 730, 1332, 1370, 1850, 2198, 3126, 6860, 12168, 16808, 29792\}$.

It has been conjectured that a perfect one-factorization of K_{2n} exists for all $n \geq 2$.

n-Factorization of a graph — n-факторизация графа.

l-Fan — l-веер.

FAS-problem — проблема разрывающих дуг.

See *Feedback arc set*.

Father of a vertex — отец (непосредственный предок) вершины.

See *Directed tree*.

Feedback arc set — разрывающее множество дуг.

Given a digraph $G = (V, A)$, an arc set $B \subseteq A$ is called a **feedback arc set** if the digraph $G - B$, resulting from G by deleting all the arcs of B from G, is *DAG*. **FAS-problem** is the problem of looking for a minimum feedback arc set.

Feedback vertex set — разрывающее множество вершин.

Given a digraph $G = (V, A)$, a subset $F \subseteq V$ is called a **feedback vertex set** if a digraph G', induced by the vertex set $V \setminus F$, is *DAG*. **FVS-problem** is the problem of looking for a minimum feedback vertex set. It is well known that *FAS-problem* can be reduced to the FVS-problem and vice versa.

F-Heap — куча Фибоначчи.

See *Fibonacci heap*.

Fibonacci heap — куча Фибоначчи.

A **heap-ordered tree** is a rooted tree containing a set of items, one item in each node, with the items arranged in a **heap order**: If x is any node, then the key of the item in x is no less than the key of the item in its *parent* $p(x)$, provided x has a parent. The fundamental operation on heap-ordered trees is linking, which combines two item-disjoint trees into one. Given two trees with roots x and y, we link them by comparing the keys of the items in x and y. If the item in

x has the smaller key, we make y a child of x; otherwise, we make x a child of y.

A **Fibonacci heap** (**F-heap**) is a collection of item-disjoint heap-ordered trees.

Fibre — слой.

Filter — фильтр.

Final substring — конечная подцепочка.

See *String*.

Finish vertex — конечная вершина.

Finishing node of fragment — финишная вершина фрагмента.

See *Fragment*.

Finite automaton — конечный автомат.

See *Model of computation*.

Finite graph — конечный граф.

The graph $G = (V, E)$ is said to be **finite** if both $n = |V|$ and $m = |E|$ are finite.

Finite tree — конечное дерево.

The tree $T = (V_T, E_T)$ is said to be **finite** if both $n = |V_T|$ and $m = |ET|)$ are finite.

Γ**-Finite graph** — Γ-конечный граф.

Γ^{-1}**-Finite graph** — Γ^{-1}-конечный граф.

Finiteness problem — проблема конечности.

First Order formula — формула первого порядка.

See *Logic for expressing graph properties*.

Five-color theorem — теорема о 5 красках.

The following result belongs to Heawood.

Theorem. Every planar map is 5-face-colorable.

Fixed vertex — неподвижная вершина.

Flag — флаг.

A **flag** is a graph obtained from C_4 by adding a vertex adjacent to exactly one vertex of C_4.

Flat forest — плоский лес.

A. **flat forest** of a graph $G = (V, E)$ is a *forest* $F = (V, E')$ such that $E' \subset E$ and each tree in F has height at most 1. Any zero-degree node in G is zero-degree in F, too.

Flow — поток.

Given a *network* $G = (V, A; s, t)$ with the *source* s and the *sink* t, let each arc $e \in A$ have a nonnegative integer number $c(e)$, the

capacity of e, associated with it. A (**feasible**) **flow**, of **magnitude** (or **amount**) Φ, from s to t in G is an integer-valued function f with the domain A that satisfies the linear equations and inequalities

$$0 \le f(e) \le c(e),$$

$$\sum_{w \in V} f(w, v) = \sum_{w \in V} f(v, w), v \ne s, t.$$

The value

$$\Phi(f) = \sum_{w \in V} f(s, w) = \sum_{w \in V} f(w, t)$$

is called the **magnitude** of f. The **maximum flow problem** is that of constructing a flow f that maximizes Φ.

Flow augmenting path — путь (цепь), увеличивающий поток.

For a given flow $f(N)$ of a net N, a **flow augmenting path** is a path Q of N such that for each $(v_i, v_{i+1}) \in Q$:
(a) if (v_i, v_{i+1}) is a *forward-edge*, then:

$$\Delta_i = c(v_i, v_{i+1}) - f(v_i, v_{i+1}) > 0$$

and
(b) if (v_i, v_{i+1}) is *reverse-edge*, then:

$$\Delta_i = f(v_i, v_{i+1}) > 0.$$

If Q is an augmenting path, then we define Δ as follows:

$$\Delta = \min \Delta_i > 0.$$

If an augmenting path Q exists, then we can construct a new flow $f'(N)$ such that the value of $f'(N)$ is equal to the value of $f(N)$ plus Δ. We do this by changing the flow for each (v_i, v_{i+1}) of Q as follows:
(a) if (v_i, v_{i+1}) is a *forward-edge*, then:

$$f(v_i, v_{i+1}) \leftarrow f(v_i, v_{i+1}) + \Delta$$

and
(b) if (v_i, v_{i+1}) is *reverse-edge*, then:

$$f(v_{i+1}, v_i) \leftarrow f(v_{i+1}, v_i - \Delta.$$

If Q is an augmenting path, then we define Δ as follows:

$$\Delta = \min \Delta_i > 0.$$

Flow control — потоковое управление.

Flow dependence — потоковая зависимсоть.

See *Data dependence.*

Flow-diagram — управляющий граф, граф потока управления, граф переходов.

See *Large-block schema.*

Flow graph — управляющий граф.

Given a program, its **flow graph** is a directed graph with *basic blocks* as vertices; one node is distinguished as **initial**; it is the block whose leader is the first statement. There is a directed edge or arc from a block B_1 to a block B_2 if B_2 can immediately follow B_1 in some execution sequence; that is, if (1) there is a conditional or unconditional jump from the last statement of B_1 to the first statement of B_2, or (2) B_2 immediately follows B_1 in the order of the program, and B_1 does not end in an unconditional jump.

We say that B_1 is a **predecessor** of B_2, and B_2 is a **successor** of B_1.

5-Flow conjecture — гипотеза о 5-потоке.

The conjecture is that every **bridgeless graph** has a **nowhere-zero 5-flow**. The **Petersen graph** does not have a **nowhere-zero 4-flow**, which shows that this conjecture (if true) is best possible.

k**-Flow** — k-поток.

Flow-equivalent graphs — потоко-эквивалентные графы.

Let $F(G, \lambda)$ be a polynomial in λ which gives the number of nowhere-zero λ-flows in G independent of the chosen orientation. Two graphs R and S are said to be **flow-equivalent** if $F(R, \lambda) = F(S, \lambda)$.

Flower — цветок.

A **flower** F is a connected graph of order 8 which contains a cut-vertex w such that each component of $F \setminus w$ has order at most 3. The vertex w is called a **stamen** of flower and the components of flower are called its **petals**.

n**-Folded Petersen graph** — n-складной граф Петерсена.

Forbidden subgraph — запрещенный подграф.

1. If $H_1, \ldots, H_k$ $(k \geq 1)$ are graphs, then a graph G is said to be $H_1, \ldots, H_k$-free if G contains no copy of any of the graphs $H_1, \ldots, H_k$ as an induced subgraph; the graphs $H_1, \ldots, H_k$ will be also referred to in this context as **forbidden subgraphs**.

2. A *cf-graph* G with the initial node p_0 has a **forbidden subgraph**

if there exist distinct nodes p_1,p_2 and p_3 and simple paths $P_{0,1}$, $P_{1,2}$, $P_{1,3}$, $P_{2,3}$, $P_{3,2}$, where $P_{i,j}$ denotes a path from p_i to p_j, that do not intersect on internal nodes.

Forcing number — форсированное число паросочетаний.

Let G be a graph that admits a *perfect matching*. A **forcing set** for a perfect matching M of G is a subset S of M, such that S is contained in no other perfect matching of G. The cardinality of a forcing set of M with the smallest size is called the **forcing number** of M and is denoted by $f(G, M)$.

Forcing set — форсированное множество.

See *Forcing number*.

Ford-Fulkerson's theorem — теорема Форда-Фалкерсона, теорема о максимальном потоке и минимальном разрезе.

The same as *Max-flow min-cut theorem*.

Forest — лес.

A **forest** is *undirected graph* such that it contains no *cycle*. The connected components of a forest are trees.

Forest graph — граф лесов.

Given $1 \leq \omega \leq |V(G)| - 1$, the **forest graph** of G denoted by $F_\omega(G)$ is defined on the set of *spanning forests* of G with ω components; two vertices are adjacent if and only if the symmetric difference of their corresponding forests has exactly two edges. See *Tree graph*, *adjacent forest graph*.

Forest-perfect graph — лесо-совершенный граф.

The class of **forest-perfect graphs** contains all forests and their compliments, all *tree-cographs*, and of course all *tree-perfect graphs*. Another subclass of forest-perfect graphs consists of the *P_4-reducible graphs*. Forest-perfect graphs are *weakly triangulated*.

H-forming number — число H-формирования.

See *H-Forming set*.

H-forming set — H-формирующее множество.

For graphs G and H, a set $S \subseteq V(G)$ is an **H-forming set** of G if for every $v \in V(G) - S$, there exists a subset $R \subseteq S$, where $|R| = |V(H)| - 1$, such that the subgraph induced by $R \cup \{v\}$ contains H as a subgraph (not necessarily induced). The minimum cardinality of an H-forming set of G is the **H-forming number** $\gamma_{\{H\}}(G)$. The H-forming number of G is a generalization of the *domination number* $\gamma(G)$.

Formal language — формальный язык.

Let $\sum$ be an alphabet and $\sum^*$ be the set of all strings over an alphabet $\sum$.

Subsets of $\sum^*$ are referred to as **formal languages** — or, briefly, **languages** — over the alphabet $\sum$.

Regarding languages as sets, we may immediately define the Boolean operations of union, intersection, complementation, and difference in a natural way.

The **concatenation** (or **product**) of two languages L_1 and L_2, is defined by

$$L_1L_2 = \{\alpha\beta : \alpha \in L_1 \text{ and } \beta \in L_2\}.$$

The notation L^i is extended to apply it to concatenation of languages. By definition, $L^0 = \{e\}$.

The **concatenation closure** (**Kleene closure** or **Kleene star**) of the language L, denoted L^*, is defined as the union of all powers of L:

$$L^* = \bigcup_{n \geq O} L^n.$$

The **e-free concatenation closure** of the language L, denoted L^+, is defined as the union of all positive powers of L:

$$L^* = \bigcup_{n > O} L^n.$$

It is clear that $L^+ = LL^* = L^*L$ and $L^* = L^+ \cup \{e\} = \{e\} \cup L^+$.

Formal language theory — теория формальных языков.

Forcing set — вынуждающее множество.

Let G be a graph that admits a *perfect matching.* A **forcing set** for a perfect matching M of G is a subset S of M, such that S is contained in no other perfect matching of G. This notion originally arose in chemistry in the study of molecular resonance structures. Similar concepts have been studied for block designs and graph colorings under the name **defining set** and for Latin squares under the name **critical set**.

See *Global forcing set.*

Forward arc — прямая дуга, дуга вперед.

1. See *Basic numberings.*

2. See *Numbering of cf-graph.*

3. See *Arrangeable graph.*

Foundtion of G-trade — основание G-трейда.

See *G-trade.*

Fractional-coloring — дробная раскраска.

A **fractional-coloring** is a mapping c from the collection $\mathcal{S}$ of independent sets of a graph G to the interval $[0,1]$, if for every vertex x of G we have

$$\sum\{c(S) : S \in \mathcal{S} \text{ such that } x \in S\} = 1.$$

The value of a **fractional-coloring** c is

$$\sum_{S \in \mathcal{S}} c(S).$$

The **fractional-chromatic number** $\chi_f(G)$ of G is the infimum of the values of fractional-colorings of G.

Fractional-chromatic number — дробно-хроматическое число.

See *Fractional-coloring.*

Fractional clique number — дробно-кликовое число.

For a graph G, its **fractional clique number** is the maximum total weight $\sum_{v \in V(G)} w(v)$ that can be assigned to the vertices v of G so that each independent set X has the total weight $\sum_{v \in X} w(v)$ at most 1.

Fractional k-factor —дробный k-фактор.

Let g and f be two integer-valued functions defined on $V(G)$. Let $h : E(G) \to [0,1]$ be a function. A function h is called a fractional (g,f)-factor if $g(x) \leq h(E_x) \leq f(x)$ holds for any vertex $x \in V(G)$, where $h(E_x) = \sum_{e \in E_x} h(e)$ and $E_x = \{e \in E(G) | e$ is incident with x in $E(G)\}$. A fractional (g,f)-factor is called a fractional $[a,b]$-factor if $g(x) \equiv a$ and $f(x) \equiv b$, where a and b are two integers such that $a \leq b$. A fractional $[a,b]$-factor is called a **fractional k-factor** if $a = b = k$. In particular, a fractional $[0,1]$-factor is also called a **fractional matching** and fractional 1-factor is also called a **fractional perfect matching**.

Fractional matching — дробное паросочетание.

See *k-Matching.*

Fractional matching number — чисо дробного паросочетания.

See *k-Matching.*

Fragment — фрагмент.

A subgraph of a *control flow graph* G is called a **fragment**.

A fragment A is a **subfragment** of B, if $A \subseteq B$; it is a **proper subfragment** if $A \neq B$.

A node p of a fragment A is called **initial** (respectively, **output** or **exit**) if either p is the initial node of G (respectively, p is the terminal node of G) or an arc of G not belonging to A enters p (respectively, leaves P).

A node p of a fragment A is called its **entry** if there is a part from the initial node of G to p that includes no arcs of the fragment A. p is called a **terminal** node of a fragment A if p does not belong to A and is a successor of a node of A.

A node p of a fragment A other than the initial and terminal nodes of G is called a boundary of A if p is the initial or output node of A. Let p be a **boundary** node of a fragment A. It is called **starting** of A if A contains no predecessors of p or all successors of p. It is called **finishing** of A if A contains all predecessors of p or no successors of p.

Frame — фрейм.

See *Framing number*.

Framing number — фреймовое число.

A graph G is **homogeneously embedded** in a graph H if for every vertex x of G and every vertex y of H there exists an embedding of G in H as an induced subgraph with x at y. A graph F of minimum order in which G can be homogeneously embedded is called a **frame** of G, and the order of F is called the **framing number** $fr(G)$ of G. For graphs G_1 and G_2, the **framing number** $fr(G_1, G_2)$ is defined as the minimum order of a graph F such that G_i $(i = 1, 2)$ can be homogeneously embedded in F. The graph F is called a **frame** of G_1 and G_2. Frames and framing numbers for digraphs were defined similarly.

Fraternal orientation — братская ориентация.

See *Fraternally oriented digraph*.

Fraternally orientable digraph — братски ориентируемый орграф.

See *Fraternally oriented digraph*.

Fraternally oriented digraph — братски ориентированный граф.

A digraph D is called a **fraternal orientation** of G_D if $(u, w) \in A(D)$ and $(v, w) \in A(D)$ implies $(u, v) \in A(D)$ or $(v, u) \in A(D)$. If

D is a fraternal orientation of G_D, we will say that D is a **fraternally oriented digraph**, G is **fraternally orientable** if it admits a fraternal orientation.

e**-Free concatenation closure** — позитивная итерация (языка).

See *Formal language*.

AT-Free graph — AT-свободный граф.

See *Asteroidal number*.

e**-Free grammar** — грамматика без e-правил.

HHD**-free graph** — HHD-свободный граф.

A graph which does not contain a *hole*, a *house* or a *domino* as an induced subgraph is called a HHD**-free graph**. The class of HHD-free graphs contains such well-known graph classes as *chordal graphs* and *distance-hereditary graphs*.

Free language — свободный язык.

Free tree — свободное дерево.

Free Petri net — свободная сеть Петри.

Free-choice Petri net — сеть Петри свободного выбора.

Frequency-ordered binary search tree — частотно-упорядоченные бинарные деревья поиска.

Frequency-ordered binary search tree (or **FOBT**) is a binary search tree that satisfies the condition that the root of a subtree must have the highest frequency. In other words, the frequencies of nodes along any path (from root top leaf) must be decreasing (or not-increasing). It has been shown that the ratio between the access cost of a FOBT and the optimal tree may be as high as $n/(4 \log n)$.

Fully disconnected graph — вполне несвязный граф, регулярный степени 0 граф, пустой граф.

A graph that contains only *isolated* vertices is **fully disconnected**.

Functional directed graph — функциональный орграф.

Functional vertex — функциональная вершина.

Functionally equivalent program schemata — функционально эквивалентные схемы программы.

See *Program schemata*.

Functionally equivalent programs — функционально эквивалентные программы.

See *Program equivalence*.

Fundamental circuit — фундаментальный цикл.

It is easy to see that the addition of a *chord* to a *spanning tree* of a graph creates precisely one circuit which is called a **fundamental circuit**. In a graph the collection of these circuits with respect to a particular spanning tree is called a **set of fundamental circuits**. Any arbitrary circuit of the graph may be expressed as a linear combination of the fundamental circuits using the operation of *ring-sum*. In other words, the fundamental circuits form a basis for the circuit space.

Fundamental cutset matrix — матрица фундаментальных разрезов.

Fundamental cycle matrix — матрица фундаментальных циклов.

Fundamental set of circuits — фундаментальная система циклов.
See *Fundamental circuit*.

Fundamental set of cutsets — фундаментальная система разрезов.
Let T be a *spanning tree* of a connected graph G. Any edge of T defines a partition of the vertices of G, since its removal disconnects T into two components. There will be a corresponding cut-set of G producing the same partition of vertices. This cut-set contains precisely one edge and a number *chord* of T. This cut-set is called a **fundamental cut-set** of G with respect to T. For the graph G and spanning tree T, a corresponding **set of fundamental cut-sets** and some other cut-sets can be expressed as linear ring-sums of fundamental cut-sets.

FVS-problem — проблема разрезающих вершин.
See *Feedback vertex set*.

G

Game-chromatic number — игровое хроматическое число.

In 1991 Bodlaender introduced the game-coloring problem on graphs. Let G be a graph and let $X = \{1, \dots, k\}$ be a set of colors. Consider a two-person game on G as follows: Players 1 and 2 make alternate moves with player 1 moving first. Each move consists of choosing an uncolored vertex and coloring it with a color from X, so that in the subgraph H of G induced by the colored vertices the adjacent vertices have distinct colors. The game ends when one of the two players can no longer execute a move. Player 1 wins if all the vertices of G are colored, otherwise player 2 wins. A graph G is called **k-game-colorable** if player 1 has a winning strategy for $|X| = k$, and the **game-chromatic number** $\chi_g(G)$ of G is the least integer k such that G is k-game-colorable.

k-Game-colorable graph — k-игровой раскрашиваемый граф.

See *Game chromatic number*.

Game domination number — игровое число доминирования.

We define a 'domination parameter' of an undirected graph G as the domination number of one of its orientations determined by the following two player game. Players A and D orient the unoriented edges of the graph G alternately with D playing first, until all edges are oriented. Player D is trying to minimize the *domination number of the resulting digraph*, while player A tries to maximize the domination number. This game gives a unique number depending only on G, if we suppose that both A and D play according to their optimal strategies. We call this number the **game domination number of** G and denote it by $\gamma_g(G)$.

Gem — драгоценный камень.

A **gem** is a graph obtained from P_4 by adding a vertex adjacent to all four vertices of P_4. A *complement* of a gem is called **anti-gem**.

General graph — общий граф, граф общего вида.

General phrase-structure grammar — грамматика с фразовой структурой.

See *Grammar*.

Generalized de Bruijn graph — обобщенный граф де Брюйна.

The **generalized de Bruijn digraph** $G_B(n,d)$ is defined by congruence equations.

$$\begin{cases} V(G_B(n,d)) = \{0,1,2,\ldots,n-1\}, \\ A(G_B(n,d)) = \{(x,y) | \, y \equiv dx+i \pmod n, 0 \le i < d\} \end{cases}$$

If $n = d^D$, $G_B(n,d)$ is the de Bruijn digraph $B(n,D)$.

See *DeBruijn graph*.

Generalized binary split tree — обобщенное бинарное расщепляемое дерево.

A **generalized binary split tree** (or **GBST**) is a *binary split tree* except that the constraint of decreasing frequency is removed.

Generalized competition graph — обобщенный граф конкуренции.

A different type of generalization of *competition graphs* was given in 1989 by Kim, McKee, McMorris, and Roberts. They defined the ***p*-competition graph** G of a digraph D as the graph with the same vertex set as D and two vertices adjacent if and only if they compete in D for at least p distinct species.

The most recent generalization is the **ϕ-tolerance competition graph** defined in 1995. Here ϕ is a non-negative valued symmetric function whose two arguments are usually assumed (but not required) to be non-negative integers. A graph $G=(V,E)$ is a **ϕ-tolerance competition graph** if each vertex x can be assigned a value (tolerance) t_x such that there exists a collection of at most $|V|$ subsets of V having the property that an edge xy is in G if and only if x and y lie together in at least $\phi(t_x,t_y)$ subsets. It is known that any graph can be transformed into a ϕ-tolerance competition graph by adding isolated vertices, and a minimal number of such vertices, required to accomplish this, is known as the **ϕ-tolerance competition number**. Of course, this number is 0 if the graph is a ϕ-tolerance competition graph.

A graph $G=(V,E)$ is **abdiff-tolerance competition graph** if for each vertex i a non-negative integer t_i can be assigned and at most $|V|$ subsets S_j of V can be found such that $xy \in E$ if and only if x and y lie in at least $|t_x - t_y|$ sets S_j.

Generalized interval order — обобщенный интервальный порядок.

Generalized Kautz digraph — обобщенный орграф Каутца.

The **generalized Kautz digraph** $G_K(n,d)$ is defined by the following congruence equations:

$$\begin{cases} V(G_K(n,d)) = \{0,1,2,\ldots,n-1\}, \\ A(G_K(n,d)) = \{(x,y) | \, y \equiv -dx - i \pmod{n}, 0 < i \leq d\} \end{cases}$$

If $n = d(d-1)^{D-1}$, $G_K(n,d)$ is the *Kautz digraph* $K(d-1,D)$.

Generalized Petersen graph — обобщённый граф Петерсена.

See *Petersen graph*.

Generalized reducible graph — обобщенно сводимый граф.

See *Regularizable graph*.

Generalized semiorder — обобщенный полупорядок.

Generating function — производящая функция.

A **generating function** of a sequence $a_0, a_1, \ldots, a_n, \ldots$ is the function

$$f(x) = \sum_{n=0}^{\infty} a_n x^n.$$

Exponential generating function is the function

$$f(x) = \sum_{n=0}^{\infty} \frac{a_n}{n!} x^n.$$

Genus of a graph — род графа.

A graph G has **genus** k if it can be embedded crossing-free in a surface of genus k. Thus $k = 0$ corresponds to *planar graphs* and $k = 1$ corresponds to *toroidal graphs*.

Geodesically convex set of vertices — геодезически выпуклое множество вершин.

Given a graph $G = (V,E)$, a subset $S \subseteq V$ is **geodesically convex** if for any two vertices $u, v \in S$ all vertices on the *shortest paths* between u and v are also contained in S.

Geodetic chain — геодезическая цепь.

Geodetic graph — геодезический граф.

G is a **geodetic graph** if for every pair of vertices there is a unique path of minimal length between them. See also *Weakly geodetic graph*.

l-Geodetic graph — l-геодезический граф.

Geometric dual graph — геометрически двойственный граф.

Plane representations Γ and Γ^* of G and G^*, respectively, are called **geometric duals** if an edge of Γ crosses the corresponding edge of Γ^* and intersects no other edges of Γ^*. It follows that the vertices of Γ and Γ^* are in a one-to-one correspondence with the faces of Γ^* and Γ, respectively.

Geometric realization of graph — геометрическая реализация графа.

Let $G = (V, E)$ be an undirected graph with weights $1/c_e$ for each edge $e \in E$. The **geometric realization of** G is the metric space $\mathcal{G}$ consisting of V and arcs of length c_e glued between u and v for each edge $e = (u, v) \in E$. The **volume** $\mu(G)$ is the Lebesgue measure of $\mathcal{G}$, i.e.

$$\mu(G) = \sum_{e \in E} c_e.$$

Girth — обхват.

The **girth** $g(G)$ of the graph G is the length of its shortest cycle. The **girth** is 1 iff G has a loop and it is 2 iff G has multiple edges.

Global density — глобальная плотность.

See *Density*.

Global w-density — глобальная w-плотность.

See *w-Density*.

Global forcing number — глобальное вынуждающее число.

See *Global forcing set*.

Global forcing set — глобальное вынуждающее множество.

Any set $S \subseteq E(G)$, such that the restriction of $f := \mathcal{M}(G) \to [0,1]^{|E(G)|}$ on S is an injection, is called a **global forcing set** of G. A global forcing set of the smallest cardinality is called a minimal global forcing set, and its cardinality is a **global forcing number** of G. For a given graph G, we denoted its global forcing number by $\varphi_g(G)$.

Global irregularity of a digraph — глобальная иррегулярность орграфа.

See *Irregularity of a digraph*.

Global strong alliance number — число глобального строгого альянса.

A global strong defensive alliance in a graph $G = (V, E)$ is a *dominating set* S of G satisfying the condition that, for every vertex $v \in S$, the number of neighbors v has in S is at least as large as the number of neighbors it has in $V - S$. The **global strong alliance number** is the minimum cardinality of a global strong defensive alliance in G.

s-Gonal tree — s-угольное дерево.

See *Polygonal tree*.

Gossip graph — граф распространения слухов.

(See *Gossiping problem*). Let us consider a full-duplex model. Let $g(G)$ be the time to gossip in a graph G. It is known that for the complete graph K_n we have:

— $g(K_n) = \lceil \log_2(n) \rceil$ for any even n;

— $g(K_n) = \lceil \log_2(n) \rceil + 1$ for any odd n.

Any graph G, such that $g(G) = g(K_n)$, is called a **gossip graph**. We call a **minimum gossip graph** of order n any gossip graph with a minimum number of edges. This number is denoted by $G(n)$.

Gossiping problem — задача распространения слухов.

The **gossiping problem** is the problem of information dissemination described in a group of individuals connected by a communication network. In gossiping, every node knows a piece of information and needs to transmit it to anyone else. This is achieved by placing communication calls over the communication lines of the network. It is assumed that a node can communicate with at most one of its neighbors at any given time, and a communication between two nodes takes one unit of time. This model implies that we will deal with connected graphs without loops and multiple edges to model the communication network. Note also that, depending on their cases, we will either consider a half-duplex or a full-duplex model. In the latter, when communication takes place along a communication line, the information flows in both directions, while in the former only one direction is allowed. Hence, in the half-duplex model, we will deal with directed graphs, while we will consider undirected graphs in the full-duplex model.

See also *Broadcasting problem*.

Graceful graph — грациозный граф.

A graph $G = (V, E)$ is said to be a **graceful graph** if there is an injection f (labelling)

$$f:\ V(G) \to \{0, 1, \ldots, q\}$$

such that the induced function

$$f^*:\ E(G) \to \{1, 2, \ldots, q\}$$

defined by

$$f^*(xy) = |f(x) - f(y)| \text{ (for all } xy \in E(G))$$

is an injection.

The images of f and f^* are called respectively vertex and edge labels.

Graceful labellings were first considered by Rosa in 1966.

(p, q)**-Graceful signed graph** — (p, q)-грациозный знаковый граф.

Let $S = (G, s)$ be a *sigraph* and s be a function which assigns a positive or a negative sign to each edge of G. Let the sets E^+ and E^- consist of m positive and n negative edges of G, respectively, where $m+n = q$. Given positive integers k and d, S said to be (k, d)**-graceful** if the vertices of G can be labeled with distinct integers from the set $\{0, 1, \ldots, k + (q-1)d\}$ such that, when each edge uv of G is assigned the product of its sign and the absolute difference of the integers assigned to u and v, the edges in E^+ and E^- are labeled with $k, k+d, k+2d, \ldots, k+(m-1)d$ and $-k, -(k+d), -(k+2d), \ldots, -(k+(n-1)d)$, respectively.

Graft — графт.

Let $G = (A, B; E)$ be a bipartite connected graph and $T \subseteq A \cup B$ be a subset of vertices of even cardinality. The pair (G, T) is called a bipartite **graft**. An edge set $F \subseteq E$ is a T**-join** if $T = \{v \in A \cup B : d_F(v) \text{ is odd}\}$. The minimum size of a T-join is denoted by $\tau(G, T)$.

We mention that a bipartite graft (G, T) always contains a T-join.

Grammar — грамматика.

A grammar is a 4-tuple $G = (V_N, V_T, R, S)$, where V_N and V_T are disjoint alphabets called the **nonterminal** and **terminal** alphabet, respectively, $S \in V_N$ is the **start** (or **sentence**) **symbol**, and R is a finite subset of $(V^*V_NV^*)\times V^*$, where $V = V_N \bigcup V_T$ is the grammar's **alphabet**. The set R consisting of couples of strings (u, v) is the essential component of a formal grammar and it is called the set of **productions** (or **grammar rules**).

$u \to v$ is a notation for $(u, v) \in R$. The elements V_N and V_T are referred to as **nonterminal** and **terminal** symbols, respectively.

A string $y \in V^*$ **immediately** (or **directly**) **derives** from a string $w \in V^*$ if there exist $x, z \in V^*$ and the production $(u, v) \in R$ such that $w = xuz$ and $y = xvz$. The relation just described is usually denoted by $\Rightarrow_G$, that is, $w \Rightarrow_G y$ iff y **immediately** derives from w.

The language **generated** by the grammar G is defined by

$$L(G) = \{w \in V_T^* : S \Rightarrow_G^* w\}$$

where $\Rightarrow_G^*$ is the reflexive transitive close of the binary relation $\Rightarrow_G$.

For any $x, y \in V^*$ the relation $x \Rightarrow_G^* y$ is true iff there is a derivation of y from x in G. A string $x \in V^*$ derivable from S is called a **sentence form**. A sentence form that does not contain nonterminals is called a **sentence**.

Grammars generate exactly all languages that can be recognized by *Turing machines*. These languages are also known as the **recursively enumerable languages**.

Other names are **Phrase-structure grammar, Unrestricted grammar, Production grammar, Grammar of type 0, General phrase-structure grammar**.

CF-grammar — КС-грамматика, контекстно-свободная грамматика.

See *Chomsky hierarchy*.

CS-Grammar — КЗ-грамматика, контекстно-зависимая грамматика.

See *Chomsky hierarchy*.

Grammar of type 0 — грамматика типа 0.

The same as *Grammar*.

Grammar rule — продукция, правило вывода.

See *Grammar*.

Graph, undirected graph, nonoriented graph — граф, неориентированный граф.

A **graph** consists of a finite set of **vertices** or **nodes** V, a finite set of **edges** E, and a mapping $Ends$ from E to $V \times V$ assigning to each edge e two, not necessarily distinct, vertices of V (the extremities of e). The graph G will be denoted by $G = (V, E, Ends)$ or simply $G = (V, E)$. It is easy to see that an edge $e = (v, w)$ corresponds to an undirected pair (v, w) of vertices. If we consider ordered pairs of vertices (v, w), we say about a *directed graph* with the set of *arcs* $A = \{(v, w)\}$.

Cf-Graph — уграф, управляющий граф, граф переходов, граф потока управления.

The same as *Control flow graph.*

Graph automorphism group — группа автоморфизмов графа, группа графа, вершинная группа графа.

Graph bundle — связка графов.

Let B and F be graphs. A graph G is a **(cartesian) graph bundle** with *fibre* F over the base graph B if there is a mapping $p: G \to B$ which satisfies the following conditions:

(1) it maps adjacent vertices of G to adjacent or identical vertices in B,

(2) the edges are mapped to edges or collapsed to a vertex,
(3) for each vertex $v \in V(B)$, $p^{-1}(v) \cong F$, and for each edge $e \in E(B)$, $p^{-1}(e) \cong K_2 \Box F$ ($\Box$ denotes the cartesian product of graphs).

Graph capacity — емкость графа.

Graph circuit space — пространство циклов графа.

Graph Clustering Problem — задача кластеризации графа.

The **graph clustering problem** (or **GCP**) is defined as follows:
Input: a sequence $(G_1, \ldots, G_m; m_1, \ldots, m_c)$, where each G_i is an n-node graph, each m_j is a positive integer, and it holds that $\sum_{j=1}^{c} m_j = m$.
Question: is there a partition $(C_1, \ldots, C_c)$ of $\{G_1, \ldots, G_m\}$ such that $|C_j| = m_j$, for all $j \in [1, c]$, and the following properties hold
(1) for any $i \in [1, c]$ and any $G_k, G_l \in C_i$, the graphs G_k and G_l are isomorphic;
(2) for any $i, j \in [1, c]$ such that $i \neq j$, and any $G_k \in C_i$ and $G_l \in C_j$, the graphs G_k and G_l are not isomorphic.
Each set C_j, $j \in [1, c]$, will be called a **cluster**.

Graph cutset space — пространство разрезов графа.

Graph enumeration — перечисление графов.

Graph grammar — графовая грамматика.

Graph isomorphism — изоморфизм графов.

Graph labeling — разметка графа.

Graph Minor Theorem — теорема о графовых минорах.

Theorem (Robertson, Seymour). For every minor closed class $\mathcal{C}$ of graphs, there is a finite set $\mathcal{F}$ of graphs such that

$$\mathcal{C} = \{\mathcal{G} |\ \forall \mathcal{H} \in \mathcal{F} : \mathcal{H} \not\preceq \mathcal{G}\}.$$

Graph morphism — графовый морфизм.

Graph of finite automata — граф конечного автомата.

Graph of function — граф функции.

Let X and Y be arbitrary nonempty sets and a function

$$f:\ M \subseteq X \to Y.$$

By $gr(f)$ we denote the **graph of funciton** f, i.e.

$$gr(f) = \{(x, f(x)) |\ x \in M\}$$

.

Graph of a partial order — граф частичного порядка.
Graph of a strict partial order — граф строгого частичного порядка.
Graph representation — задание графа.
Graph rewriting system (with priorities) — система переписывания графов (с приоритетами).

Let $C = (C_V, C_E)$ be a color alphabet. A rewriting rule over C consists of a connected C-labeled digraph D and a partial labeling function of D denoted by λ. We write $r = (D, \lambda)$. Note that such a rewriting rule can be more classically viewed as a pair of graphs (the left-hand-side and the right-hand-side of the rule) given by $(D, \lambda D)$.

A **graph rewriting system** with priorities is defined by a triple $(C, P, >)$, where:
(1) $C = (C_V, C_E)$ is the color alphabet,
(2) P is a finite set of rewriting rules over C,
(3) $>$ is a *partial order* on the rules of P.

Graph of reachable markings — граф достижимых разметок.
Graph spectral theory — спектральная теория графов.

The **graph spectral theory** studies graph eigenvalues and eigenvectors. See, for instance, the book
D.Cvetcović, M. Doob, and H. Sachs. *Spectra of Graphs — Theory and Application*, Barth, Heidelberg (1995).

Graph symmetry number — число симметрий графа.
Graph transformation rule — правило преобразования графа.

A **graph transformation rule** $r : L \to R$ is a *partial graph morphism* from L, the left-hand side, to R, the right-hand side of the rule r. A **redex** of r in a graph G is a *total graph morphism* $m : L \to G$ from the left-hand side of the rule to G.

Graph union — объединение графов.

A **graph union** $G_1 \cup G_2$ of the graphs G_1 and G_2 is a graph with vertices in $V(G_1) \cup V(G_2)$ and edges in $E(G_1) \cup E(G_2)$.

Graph with boundary — граф с границей.

A **graph with boundary** is a graph $G(V_0 \cup \partial V, E_0 \cup \partial E)$ with *interior vertices* V_0, *boundary vertices* ∂V and edge set $E_0 \cup \partial E$. Each edge $e \in E_0$ (*interior edge*) joins two interior vertices, each edge $e \in \partial E$ (*boundary edge*) connects an interior vertex with a boundary vertex.

A **d-regular tree with boundary** is a tree, where all interior edges have length 1, all boundary edges have length ≤ 1, and where all interior vertices have degree d and all boundary vertices degree 1. The set of interior vertices is not empty, i.e. $|V_0| \geq 1$.

(g, f)-Graph — (g, f)-граф.

See *k-Factor of a graph.*

θ-Graph — θ-граф.

Graphic sequence of numbers — графическая последовательность чисел.

See *Degree sequence.*

Graphical (graphic) matroid — графический матроид.

A *matroid* which is obtained from a graph $G = (V, A)$ with the vertex set V and arc set A. Let $\mathcal{I}(A)$ be the family of all the subsets of A which do not include arcs forming a *cycle*. A **graphical matroid** obtained from the graph G is also isomorphic to the *matrix matroid* obtained from the *incidence matrix* of G and is representable over any field, i.e. **graphical matroid** is *regular.*

Thus, a matroid is called **graphic** if it can be represented as the *cycle matroid* of a graph. A matroid is called **cographic** if it is the *dual* of a graphic matroid. Graphic and cographic matroids are representable over every field. The reader may wonder if there is a way of determining whether or not a matroid is graphic. Tutte (1960) gave a polinomial-time algorithm for determining whether a binary matroid is graphic. If the binary matrix is graphic, then the algorithm returns the incidence matrix of a graph, otherwise it concludes that the matrix is not graphic. Later (1981) Seymour gave an algorithm for determining whether any matroid, not necessarily a binary matroid, is graphic.

Graphical partition of a number — графическое разбиение числа.

Graphical sequence of numbers — графическая последовательность чисел.

Graphical trade — графический трейд. See *G-trade.*

Graphoid — графоид.

Graphs union — объединение графов, соединение графов.

Greedy algorithm — жадный алгоритм.

Grid graph — граф решетки.

A **grid graph** is a finite node-induced subgraph of the infinite grid G^∞. The node set of G^∞ consists of all points of the plane with integer coordinates. Two nodes are connected iff their Euclidean distance is equal to 1. A **grid graph** is completely specified by its node set. A **grid graph** $R(m,n)$ is call **rectangular** if its node set $V(R(m,n)) = \{v : 1 \leq v_x \leq m, 1 \leq v_y \leq n\}$, where v_x and v_y are the x and y coordinates of v respectively.

The other name is **Lattice graph**.

Group graph — граф группы.

Group of a directed graph — группа орграфа.

Growing tree — растущее дерево.

Grundy colouring — Гранди раскраска.

Grundy number — число Гранди.

Gupta scheme — код Гапта для 2-3-деревьев.

H

Halin graph — граф Халина.

G is a **Halin graph** if G is formed by embedding a tree having no degree-2 vertices in the plane, and connecting its *leaves* by a *cycle* that crosses none of its edges. **Halin graph**s are 3-connected and have a *Hamiltonian circuit*.

Hamiltonian center — гамильтонов центр.

A vertex can be called a **Hamiltonian center** if *Hamiltonian chains* come out from it to all other vertices.

Hamiltonian chain — гамильтонова цепь.

For given vertices a and b, a chain containing all vertices of a graph is called **Hamiltonian chain**.

Hamiltonian circuit — гамильтонов цикл.

Given a graph, a circuit containing all its vertices is called **Hamiltonian circuit**.

Hamiltonian connected graph — гамильтоново-связный граф.

A graph G is **Hamiltonian connected** if there is a *hamiltonian chain* joining every pair of vertices in the graph.

Hamiltonian cycle — гамильтонов контур.

Given a digraph, a cycle containing all its vertices is called **Hamiltonian cycle**.

Hamiltonian decomposable graph — гамильтоново разложимый граф.

A graph G is **Hamiltonian decomposable** if either the *degree* of G is $2k$ and the edges of G can be partitioned into k *hamiltonian cycles*, or the degree of G is $2k+1$ and the edges of G can be partitioned into k hamiltonian cycles and a *1-factor*. If G is a **Hamiltonian decomposable graph**, then G is *loopless*, *connected*, and *regular*. For a graph G to have a hamiltonian decomposition that the graph G, it should have a *hamiltonian cycle*.

Hamiltonian digraph — гамильтонов орграф.

A digraph is called **Hamiltonian digraph** if it has a *Hamiltonian cycle*.

Hamiltonian graph — гамильтонов граф.

This is a graph which has a *Hamiltonian circuit*. A **Hamiltonian graph** G of order n is k-ordered, $2 \leq k \leq n$, if for every sequence $v_1, \ldots, v_k$ of k distinct vertices of G, there exists a *Hamiltonian* *circuit* that encounters $v_1, \ldots, v_k$ in this order.

Hamiltonian path — гамильтонов путь.

Given a digraph, a path containing all its vertices is called **Hamiltonian path**.

Hamming distance — расстояние Хэмминга.

See *Hamming graph*.

Hamming graph — граф Хэмминга.

Given a graph $G = (V, E)$, G is called **Hamming graph** if each vertex $x \in V$ can be labeled by a word $a(x)$ (of fixed length) over some symbol set Σ such that

$$H(a(x), a(y)) = d_G(x, y)$$

for arbitrary $x, y \in V$. Here $a(x)$ is termed the address of x, and $H(a, b)$ stands for the **Hamming distance** of the addresses a and b, that is, the number of positions k such that the k-th symbol in a differs from the k-th symbol in b. Futher, $d_G(x, y)$ denotes the length of a shortest chain in G between x and y. G is called a **binary Hamming graph** if $\Sigma = \{0, 1\}$.

Hammock — гамак.

A **hammock** is an *alt*, the set of terminal nodes of which is empty or consists of one node that is a successor of each output node of the alt and is not a predecessor of its initial node.

A hammock is called **decomposable** if it can be presented as a union of two disjoint hammocks and indecomposable (or **prime**) otherwise. A maximal decomposable hammock is called **composite**.

Hammock presentation — гамачное представление.

Let G be a cf-graph and Γ_G denote the set of all nontrivial prime and composite hammocks of G. Γ_G -representation of G (i.e. *A-representation* of G for $A = \Gamma_G$) is called a **hammock representation** of G.

Handshake's lemma — лемма о рукопожатиях.

NP-Hard language —*NP*-трудный язык.

NP-Hard problem — *NP*-трудная задача.

See *Complexity theory.*

PSPACE-hard problem — *PSPACE*-трудная задача.

Harmonious chromatic number — гармоническое хроматическое число.

Head of a hyperarc — начало гипердуги.

See *Directed hypergraph.*

Head place — головное место.

Heap — куча.

A **heap** is an abstract data structure consisting of a set of items, each with a real-valued key, subject to the following operations: *make heap, insert(i,h), find min(h), delete min(h)*. See also *Fibonacci heap*.

Heap order — кучевой порядок.

See *Fibonacci heap*.

Heap-ordered tree — кучево-упорядоченное дерево.

See *Fibonacci heap*.

Height of a branch of a tree — высота ветви дерева.

Height of a tree — высота дерева.

Height of a vertex — высота вершины.

See *Directed tree*.

Helly hypergraph — гиперграф Хелли.

A *hypergraph* $\mathcal{H}$ whose edges satisfy the Helly property; i.e., any subfamily $\mathcal{H}' \subseteq \mathcal{H}$ of pairwise intersecting edges has a nonempty intersection.

Hereditary class of graphs — наследственный класс графов.

Let $ISub(G)$ be the set of all induced subgraphs of a graph G. A class of graphs P is called **hereditary class of graphs**, if $ISub(G) \subseteq P$ for every $G \in P$. The following hereditary class is associated with any class Q: $Perf(Q) = \{G : ISub(G) \subseteq Q\}$. For example, if $Q = \{G : i(G) = \gamma(G)\}$, then $Perf(Q)$ is a well-known class of **domination perfect** graphs, which was defined by Sumner and Moore in 1979 and characterized in terms of 17 forbidden induced subgraphs.

Hereditary dually chordal graph — наследственно-двойственный хордальный граф.

A *dually chordal graph* G such that any induced subgraph of G is dually chordal.

Hereditary property of a graph — наследственное свойство графа.

Hierarchical graph — иерархический граф.

Let G be a graph of some type, e.g. G can be an undirected graph, a digraph or a hypergraph, and let F be a set of its fragments such that $G \in F$ and, for any $C_1, C_2 \in F$, just one of the following properties holds: (1) $C_1 \subset C_2$, (2) $C_2 \subset C_1$, (3) $C_1 \cap C_2 = \varnothing$.

$H = (G, T)$, where $T = (F, I)$ is a directed tree with a root G such that I represents an immediate inclusion relation between fragments of F, is called a **hierarchical graph**; G is called the **underlying**

graph of H, and T is called the **inclusion tree** of H.

A hierarchical graph H is called a **connected** one, if each fragment from F is a connected graph, and a **simple** one, if all fragments from F are induced subgraphs of G.

A simple hierarchical graph $H = (G, T)$ such that G is an undirected graph and the leaves of T are exactly the trivial subgraphs of G is called a **clustered graph**.

Hierarchical Petri net — иерархическая сеть Петри.

See *High-level Petri nets.*

Hierarchy of embedded alts — иерархия вложенных альтов.

See *Alt.*

Hierarchy of embedded zones — иерархия вложенных зон.

The same as *Nested set of zones.*

High-level Petri nets — сети Петри высокого уровня.

The term of **high-level Petri net** is used for many formalisms that extend the basic Petri net formalism; this includes **coloured Petri nets**, **well-formed Petri nets**, **hierarchical Petri nets**, **prioritised Petri nets**, **timed Petri nets**, **stochastic Petri nets**, **dualistic Petri nets** and all other extensions.

Hilbert's problem — проблема Гильберта.

Hole — дыра.

An odd cycle without diagonals C_{2k+1}, where $2k+1 \geq 5$. Its *complement* is called an **antihole**. A graph is said to be a **Berge graph** if it does not contain **hole** and antiholes.

Homeomorphical graphs — гомеоморфные графы.

Homeomorphically irreducible tree — гомеоморфно несводимое дерево.

Homogeneously embedded graph — однородно вложенный граф.

See *Framing number.*

Homomorphic image of a graph — гомоморфный образ графа.

Homomorphism of a graph — гомоморфизм графа.

For given graphs G_1 and G_2, a **homomorphism** from G_1 to G_2 is a mapping $\varphi : V(G_1) \to V(G_2)$ such that if $(x, y) \in E(G_1)$, then $(\varphi(x), \varphi(y)) \in E(G_2)$.

Honest graph — честный граф.

A graph G is called **honest** if its *edge-integrity* is the maximum possible; that is, equal to the *order* of the graph. It is known that every graph of diameter 2 is honest. It is easy to see that only 3-, 4-, and 5-cycles are honest 2-regular graphs.

House — дом.

This is a graph consisting of one cycle of length 3 and one of length 4 having a common edge. A **house** is the complement of the path on five vertices.

Hungarian method — венгерский алгоритм.

Hyperarc — гипердуга.

See *Directed hypergraph.*

Hyper de Bruijn graph — многомерный граф де Брёйна.

The **hyper de Bruijn graph** $HD(m, n)$ is $Q^m \times D_n$, where Q^m is m-cube graph and D_n is the *binary de Bruijn graph* of order 2^n.

Hyper Petersen graph — многомерный граф Петерсена.

The **hyper Petersen graph** HP_n is $Q^{n-3} \times P$, where Q^{n-3} is $n-3$-cube graph and P is Petersen graph.

Hypercube — гиперкуб.

The same as *n-Cube graph.*

Hypercycle — гиперцикл.

A sequence $C = (e_1, \ldots, e_k, e_1)$ of edges is a **hypercycle** if $e_i \cap e_{i+1 \pmod k} \neq \emptyset$ for $1 \leq i \leq k$. The **length** of C is k. A hypergraph is **α-acyclic** if it is *conformal* and contains no chordless hypercycles of length at least 3.

Hypergraph — гиперграф.

A finite **hypergraph** $\mathcal{H}$ is a family of nonempty subsets (the **edges** of $\mathcal{H}$) from some finite underlying set V (the **vertices** of $\mathcal{H}$).

Hypertree — гипердерево.

This is a *hypergraph* (called **arboreal hypergraph**) such that there is a *tree* T with a vertex set V such that every edge $e \in \mathcal{H}$ induces a *subtree* in T (T is then called an **underlying vertex tree** of $\mathcal{H}$). A hypergraph $\mathcal{H}$ is a **dual hypertree** if there is a tree T with a vertex set $\mathcal{H}$ such that for all vertices $v \in V$ $T_v = \{e \in \mathcal{H} : v \in e\}$ induces a subtree of T (T is then called the **underlying hyperedge tree** of $\mathcal{H}$).

Observe that $\mathcal{H}$ is a hypertree if and only if $\mathcal{H}^*$ is a dual hypertree.

Hypohamiltonian graph — гипогамильтоновый граф.

I

Identical group of a graph — тождественная группа графа.

r**-Identifying code** — r-идентифицирующий код.

Consider a connected undirected graph $G = (V, E)$, a subset of vertices $C \subseteq V$, and an integer $r \geq 1$; for any vertex $v \in V$, let $B_r(v)$ denote the ball of radius r centered at v. If for all vertices $v \in V$, the sets $B_r(v) \cap C$ are all nonempty and different, then C is called an r**-identifying code**.

Immediate dominator — непосредственный доминатор, непосредственный обязательный предшественник.

See *Dominator*.

Immediate postdominator — непосредственный обязательный преемник, непосредственный постдоминатор.

Immersion — вложение, погружение.

A pair of adjacent edges (u, v) and (v, w), with $u \neq v \neq w$, is lifted by deleting the edges (u, v) and (v, w), and adding the edge (u, w). A graph H is said to be **immersed** in a graph G if and only if a graph isomorphic to H can be obtained from G by lifting pairs of edges and taking a subgraph.

Immovable vertex — неподвижная вершина.

d**-improper list chromatic number** — d-неправильное списковое хроматическое число.

See *L-coloring with impropriety d.*

Impropriety — неправильность.

See *L-Coloring with impropriety d.*

Incenter — внутренний центр.

Incidence function labelling — помечающая функция инцидентности.

Incidence graph — граф инцидентности.

The (bipartite vertex-edge) **incidence graph** $\mathcal{IG}(\mathcal{H}) = (V, \mathcal{H}, E)$ of the *hypergraph* $\mathcal{H}$ is a *bipartite* graph with a vertex set $V \cup \mathcal{H}$, where two vertices $v \in V$ and $e \in \mathcal{H}$ are *adjacent* if and only if $v \in e$.

Incidence matrix — матрица инцидентности.

1. (For a graph) The (vertex-edge) **incidence matrix** $I(G)$ of a graph $G = (V, E)$, $V = \{v_1, \ldots, v_n\}$, $E = \{e_1, \ldots, E_m\}$, is $n \times m$ $(0, 1)$-matrix with entries

$$i_{kl} = \begin{cases} 1, & \text{if } v_k \in e_l \\ 0, & \text{otherwise} \end{cases}$$

2. (For a directed graph) The **incidence matrix** of a directed graph D is a $p \times q$ matrix $\{b_{i,j}\}$ where p and q are the numbers of vertices and edges respectively, such that $b_{i,j} = 1$ if the edge x_j leaves vertex v_i, 1 if it enters vertex v_i and 0 otherwise. (Note that many authors use the opposite sign convention.)

3. (For a hypergraph) let $\mathcal{H}$ be a *hypergraph*, $E = \{e_1, \dots, e_m\}$ be its edge set and $V = \{v_1, \dots, v_n\}$ be its vertex set. The **incidence matrix** $\mathcal{IM}(\mathcal{H})$ of the hypergraph $\mathcal{H}$ is a matrix whose (i, j) entry is 1 if $v_i \in e_j$ and 0 otherwise. Note that the transposed matrix $\mathcal{IM}(\mathcal{H})^T$ is the **incidence matrix** of the *dual hypergraph* $\mathcal{H}^*$. A **totally balanced matrix** is **incidence matrix** of a *totally balanced hypergraph*.

Incidence relation — отношение инцидентности.

Incidentor — инцидентор.

Incidency — инцидентность.

Inclusion of languages problem — проблема включения языков.

Inclusion of schemas — вычисляемость схем.

See *Large-block schema*.

Inclusion tree — дерево вложенности.

See *Hierarchical graph*.

Incomparable vertices — несравнимые вершины.

Incompatibility graph — граф несовместимости.

Increment operator — оператор прибавления единицы.

Indecomposable tournament — неразложимый турнир.

See *Critical tournament*.

Indegree, in-degree — полустепень захода вершины.

The **indegree** of the vertex v in a digraph G is the number of distinct arcs with the *target* v and is denoted by $in(v, G)$.

Indegree matrix — матрица полустепеней захода.

Independence graph of a graph — граф независимости графа.

Maximum independent sets in G will be also called α–sets in G. The independence number $\alpha(G)$ of a graph G is the cardinality of an α–set in G. Let $\mathcal{S}$ be the set of α–sets of G. Then the **independence graph** $Ind(G)$ of G is the graph with $V(Ind(G)) = \mathcal{S}$, and $S_1, S_2 \in \mathcal{S}$ are adjacent whenever $S_1 \cap S_2 = \emptyset$.

Independence number — число независимости, число внутренней устойчивости, неплотность.

For a graph G, **independence number** $\beta(G)$ is the size of a maximum *independent set* of G. The **lower independence number** $i(G)$ is the minimum cardinality of a maximal independent set. (Because an independent set is maximally independent if and only if it is dominating, $i(G)$ is also called the independent-domination number of G.)

The other name is **Stability number**.

See also *Transversal*, *Local independence number*.

Independence polynomial — многочлен независимости.

For a graph G with *independence number* β, let i_k denote the number of independent sets of vertices of cardinality k in G ($k = 0, 1, \ldots, \beta$). The **independence polynomial** of G,

$$i(G, x) = \sum_{k=0}^{\beta} i_k x^k,$$

is the generating polinomial for the independence sequence

$$(i_1, i_2, \ldots, i_\beta).$$

Independence subdivision number — независимое число подразбиения.

The **independence subdivision number** $sd_\beta(G)$ of a graph G is the minimum number of edges that must be *subdivided* (where an edge can be subdivided at most once) in order to increase the *independence number*. It is known that for any graph G of order $n \geq 2$, either $G = K_{1,m}$ and $sd_\beta(G) = m$, or

$$1 \leq sd_\beta(G) \leq 2.$$

See also *Domination subdivision number*.

n-Independence number — число n-независимости.

See *n-Dominating set*.

Independent circuits — независимые циклы.

A set $\mathcal{C}$ of *circuits* of G is called **independent** if for every nonempty subset $\mathcal{A}$ of $\mathcal{C}$ the symmetric difference of the circuits in $\mathcal{A}$ is not empty. A maximal independent set of circuits of G is called a **cycle basis** of G. It is easy to see that every cycle basis of G has $|E(G)| - |V(G)| + c(G)$ circuits, where $c(G)$ is the number of components of G. See also *Cyclomatic number*.

Independent dominating set — независимое доминирующее множество.

See *Dominating set.*

Independent dominating number — число независимого доминирования.

See *Dominating set.*

Independent domination number relative to v — число независимого доминирования относительно v.

See *Dominating set.*

Independent n**-domination number** — число независимого n-доминирования.

See n*-Dominating set.*

Independent edges — независимые ребра.

Given a graph (digraph, hypergraph) G, edges such that no two have an endpoint in common are called **independent**.

Another name is **Matching**.

Independent paths — независимые пути.

Given a graph (digraph), paths having no points in common except, possibly, their endpoints are called **independent**.

Independent set — независимое множество.

Let G be an undirected graph. $V' \subseteq V$ is an **independent set** or **stable set** in G (or **empty subgraph**) if for all $u, v \in V'$ $(u, v) \notin E$. S is a **maximal independent set** if S is independent and for each vertex $u \in V(G) - S$ the set $S \cup \{u\}$ is not independent.

Let G be a directed graph. A set of vertices $W \subseteq V$ is called **independent** if for every pair of vertices $u, v \in W$ neither (u, v) nor (v, u) is present in the digraph. $W \subseteq V$ is **absorbent** if for each $u \in V \setminus W$ there exists $(u, v) \in A(G)$ with $v \in W$ and **dominant** if for each $v \in V \setminus W$ there exists $(u, v) \in A(G)$ with $u \in W$. A set $W \subseteq V$ is a **kernel** of G if W is independent and absorbent and W is a **solution** of G if W is independent and dominant.

Independent set Y is called an **essential independent set** if there is $\{y_1, y_2\} \subseteq Y$ such that $dist(y_1, y_2) = 2$.

Independent sets of a matroid — независимые множества матроида.

See *Matroid.*

Independent vertex set of a hypergraph — независимое множество вершин гиперграфа.

Independent F-matching width — ширина независимого F-паросочетания.

See *F-width.*

Independent F-width — независимая F-ширина.

See *F-Width.*

Independent matching width — независимая ширина паросочетания.

See *F-Width.*

Independent width — независимая ширина.

See *F-Width.*

n-Independent set — n-независимое множество.

See *n-Dominating set.*

Index — индекс.

1. See *Primitive directed graph.* 2.

2. See *Characteristic polynomial of a graph.*

Indirect addressing graph — граф косвенной адресации.

Indifference digraph — индифферентный орграф.

Indifference graph — индифферентный граф.

Induced matching partition number — число разбиения индуцированного паросочетания.

The **induced matching partition number** of a graph G, denoted $imp(G)$, is the minimum integer k such that $V(G)$ has a k-partition $(V_1, \ldots, V_k)$ such that, for each i, $1 \leq i \leq k$, $G[V_i]$, the subgraph of G induced by V_i, is 1-regular graph. It is known that, if G is a graph which has a *perfect matching*, then $imp(G) \leq 2\Delta(G) - 1$, where $\Delta(G)$ is the maximum degree of a vertex of G. Furthermore, $imp(G) = 2\Delta(G) - 1$ if and only if G is isomorphic to either K_2 or C_{4k+2} or the *Petersen graph*, where C_n is the cycle of length n.

Induced path number — число порождённых путей.

The **induced path number** $\rho(G)$ of a graph G is defined as the minimum number of subsets into which the vertex set $V(G)$ of G can be partitioned such that each subset induces a path.

If G is a graph such that $\rho(G) = k$ and $\rho(G - v) = k - 1$ for every $v \in V(G)$, then we say that G is **k-minus-critical**.

Induced (with vertices) subgraph — порождённый (вершинами) подграф.

See *Subgraph* (in a strong sense).

Inductive graph — индуктивный граф.

Infinite graph — бесконечный граф.

Inflation — инфляция.

1. The **inflation** G_l of a graph G is the *line graph* of a *subdivision* of G.

2. Let F and G be (simple) graphs such that $V(G) = \{v_1, \dots, v_n\}$. We say that F is an **inflation** if $V(F)$ can be written as a *disjoint union* $V(F) = V_1 \cup \cdots \cup V_n$ in such a way that if $x \in V_i$ and $y \in V_j$, then $xy \in E(F)$ if and only if $i = j$ or $v_i v_j \in E(G)$. (So to **inflate** a graph is to replace each vertex by a complete graph.) If $|V_i| = t$ for all t, then we write $F = G_{(i)}$ and call it a **uniform inflation** of G. (Another way of looking at $G_{(i)}$ is considering it as a *lexicographic product* or *composition* $G[K_t]$ of G and K_t, also called the *wreath product* $G * K_t$).

Information flow — информационный связь.

See *Value of schema under interpretation.*

Information graph — информационный граф.

Informationally connected operands — информационно связные операнды.

See *Large-block schema.*

Informationally incompatible operands — информационно несовместимые операнды.

See *Large-block schema.*

Inheritance graph — граф наследования.

An **inheritance graph** is a directed acyclic multigraph $H = (X, E)$ with a least element denoted by 0_H and a greatest element denoted by 1_H. The *transitive closure* H^{tc} of H induces a *partial order*. In the context of object oriented languages, this order is called the **inheritance relation**.

Inheritance relation — отношение наследования.

See *Inheritance graph.*

Inhibitor arc — ингибиторная дуга.

Inhibitor Petri net — ингибиторная сеть Петри.

Initial marking — начальная разметка.

See *Petri net.*

Initial node — начальная вершина.

1. See *Control flow graph.*

2. See *Fragment.*

Initial state — начальное состояние.

Initial string — начальная подцепочка.
See *String.*

Initial symbol — начальный символ.

(Informationally) incompatible operands — (информационно) несовместимые операнды.
See *Large-block schema.*

(Informationally) connected operands — (информационно) связные операнды.
See *Large-block schema.*

In-neighborhood — входящая окрестность.
See *Neighborhood of a vertex.*

In-neighbour — входящий сосед.
See *Neighbourhood of a vertex.*

Inner vertex — внутренняя вершина.

Inorder traversal — симметричный обход.

Input — вход.
Given a digraph G, its **input** is a vertex of *in-degree* zero.

Input arc — заходящая дуга.

Input directed spanning tree — входящий оркаркас.

Input place — входное место.

Input tree — входящее дерево.
See *In-tree.*

Inradius — внутренний радиус.

In-semicomplete digraph — полуполный по входу орграф.
See *Neighbourhood of a vertex.*

Inseparation number — число внутреннего разделения.

Inset — заходящее множество.
The **inset** $N_-(x)$ of a vertex x is the set of vertices dominating x. See also *Outset.*

Integer distance graph — граф целочисленных расстояний.
An **integer distance graph** is a graph $G(D)$ with the set of all integers as a vertex set and two vertices $u, v \in Z$ are adjacent if and only if $|u - v| \in D$, where the distance set D is a subset of positive integers set.

Integral graph — целочисленный граф.
A graph is called **integral** if all of its *eigenvalues* are integers. Such graphs are rare. It is known that there are only 263 non-isomorphic connected integral graphs with up to 11 vertices.

Integrity — целостность (графа).

The **integrity** $i(G)$ of G is defined as

$$i(G) = \min\{|X| + m(G - X) : \ X \subset V\},$$

where $m(G-X)$ stands for the maximum number of vertices among all components of the graph $G - X$.

The integrity parameter was proposed by Barefoot et al. (1987) as a vulnerability measure of a graph.

See also *Toughness of a graph*.

Internal transition — внутренний переход.

Interpretation — интерпретация.

See *Large-block schema*, *Program schemata*.

Intersection graph — граф пересечений.

The **intersection graph** of a set of items is a graph formed by associating each item with a vertex and adding an edge between two vertices if the associated items have a nonempty intersection. The **containment graph** is formed in a similar fashion, but there is an edge between two vertices if one of the items contains the other.

Intersection of graphs — пересечение графов.

Intersection number — число пересечений.

(X, Y)**-Intersection graphs** — графы (X, Y)-пересечений.

Interval — интервал.

An **interval** is such an alt I that its initial node belongs to each strongly connected subgraph of I. The initial node of interval I is also called a **header** node.

An interval I is **maximal** if there is no such an interval Z that I is a proper subfragment of Z.

For a given *control flow graph* G with its initial node n_0 and a given node n of G, the maximal interval with the header n, denoted $I(n)$, can be constructed by the following three rules: (1) n is in $I(n)$; (2) if all the predecessors of some node $m \neq n_0$ are in $I(n)$, then m is in $I(n)$; (3) nothing else is in $I(n)$.

The set of all maximal intervals of a *cf-graph* G form a partition of the set of its nodes.

A node p is a head of some maximal interval of a cf-graph G if and only if either p is the initial node of G or p is a terminal node of another maximal interval of G.

Interval $I(u, v)$ — интервал $I(u, v)$.

The **Interval** $I(u, v)$ between two vertices u and v in G is the set of all vertices on shortest paths between u and v.

Interval chromatic number — интервальное хроматическое число.

See *Interval coloring.*

Interval coloring — интервальная раскраска.

An **interval coloring** of a *weighted graph* (G, w) maps each vertex x to an open interval I_x on a real line, of width $w(x)$, such that adjacent vertices are mapped to disjoint intervals. The **total width** of an **interval coloring** is defined to be $|\cup_x I_x|$. The **interval chromatic number** $\chi(G, w)$ is the least total width needed to color the vertices with intervals.

A graph G is called **superperfect** if for every non-negative weight function w, $\Omega(G, w) = \chi(G, w)$.

Interval function — интервальная функция.

Interval graph — интервальный граф.

1. An **interval graph** is a graph for which one can associate with each vertex an interval on the real line such that two vertices are *adjacent* if and only if their corresponding intervals have a nonempty intersection.

The following characterization of an **interval graph** was found by P.C. Gilmore and A.J. Hoffman in 1964.

Theorem. The following statements are equivalent:

(1) G is an interval graph,

(2) G is chordal and its *complement* $\bar{G}$ is a *comparability graph*,

(3) there is an *interval ordering* of the maximal *cliques* of G.

The interval graphs are an important subclass of the *chordal* graphs. It is known that a graph G with n vertices and m edges can be tested for being an interval graph in $\mathcal{O}(n + m)$.

2. *1-derived graph* of a given *cf-graph* is called an **interval graph**. Another name is **Derived graph**.

Interval hypergraph — гиперграф интервалов.

Interval of a graph — интервал графа.

Interval of a tournament — интервал тунира.

See *Critical tournament.*

Interval order — интервальный порядок

1. An ordering $(X_1, \ldots, X_n)$ of the maximal *cliques* of a graph G such that for every vertex the maximal cliques containing it occur consecutively in the ordering is called **interval order**.

2. Let $I = (I_j)_{j=1}^n$ be a finite collection of intervals of the real line and let (I, L) be a poset such that $I_i L I_j$ iff I_i is completely to the left of I_j. $(P, <)$ is an **interval order** if there is a collection I of intervals such that $(P, <)$ is isomorphic to (I, L). It is easy to see that the *comparability graphs* of **interval orders** are exactly the *cointerval graphs*.

Intractable problem — трудноpешаемая задача.

See *Complexity theory*.

In-tree — входящее ордерево.

An **in-tree** is a directed tree in which precisely one vertex has zero *out-degree*. The other name is **Input tree**.

Invariant of a graph — инвариант графа.

k**-invariant graph** — k-инвариантный граф.

See *Clique graph*.

F**-Inverse arc** — F-обратная дуга.

See *Numbering of cf-graph*.

Inverse arborescence — обратная древесность.

See *Arborescence*.

Inverse cycle — обратный цикл.

See *Cycle*.

Inverse relation — обратное отношение.

See *Binary relation*.

Irreducible additive hereditary graph property — свойство несводимой аддитивной наследуемости графов.

See *Additive hereditary graph property*.

Irreducible graph — несводимый граф.

Irredundance number — число несводимости.

See *Irredundant set*.

Irredundance perfect graph — неизбыточно совершенный граф.

A graph G is an **irredundance perfect graph**, if for every *induced subgraph* H of G holds the equality $ir(H) = \gamma(H)$, where $ir(H)$ is the *irredundance number* and $\gamma(H)$ is the *domination number*. A graph G is called k**-irredundance perfect** $(k \geq 1)$ if $ir(H) = \gamma(H)$ for every induced subgraph H of G with $ir(H) \leq k$.

A graph G is **minimal irredundance imperfect** if G is not irredundance perfect and $ir(H) = \gamma(H)$ for every proper induced subgraph H of G.

The first sufficient condition, for a graph to be irredundance perfect, in terms of a family of forbidden induced subgraphs is due to Bollobás and Cockayne (1979).

k-Irredundance perfect graph — неизбыточно совершенный граф.

See *Irredundance perfect graph.*

k-Irredundance perfect graph — k-неизбыточно совершенный граф.

Irredundant Petri net — неизбыточная сеть Петри.

See *Petri net.*

Irredundant set — неизбыточное множество (вершин).

A set I of vertices of G is an **irredundant set**, if every vertex x of I that is not *isolated* in I has at least one external I-private neighbor (or I-EPN), that is a vertex of $V - I$ that is *adjacent* to x but to no other vertex of I. The minimum cardinality of the maximal **irredundant set** is called an **irredundance number** and denoted by $ir(G)$. A vertex is an **annihilator** of a vertex x of an irredundant set I (x not isolated in I) if it dominates all the I-EPN's of x.

Irreflexive relation — антирефлексивное отношение.

Irregular digraph — иррегулярный орграф.

A digraph is called **irregular** if its distinct vertices have distinct *degree pair.*

Irregular graph — иррегулярный граф.

See *Regular graph.*

Irregularity of a digraph — нерегулярность орграфа.

An **irregularity** of a digraph D is defined as $i(D) = \max|d^+(x) - d^-(y)|$ over all vertices x and y of D (possibly $x = y$). There are two other measures of regularity, namely, the **local irregularity** of a digraph D, which is $i_l(D) = \max|d^+(x) - d^-(x)|$ over all vertices x of D and **global irregularity** of D, which is $i_g(D) = \max\{d^+(x), d^-(x) : x \in V(D)\} - \min\{d^+(y), d^-(y); y \in V(D)\}$. Clearly,

$$i_g(D) \geq i(D) \geq i_l(D).$$

Irregularity strength — степень иррегулярности.

The **irregularity strength** $s(G)$ of a graph G is defined as the minimum integer t, for which the edges of G can be weighted with $1, 2, \ldots, t$ in such a way that the weighted degrees, i.e. the sum of weights of the adjacent edges in each vertex, are distinct numbers.

It is known that the **irregularity strength** of any tree with no vertices of degree 2 is its number of *pendant vertices.*

Isolated vertex — изолированная (голая) вершина.

Given a graph, a vertex adjacent to no edges is called **isolated**.

Isolated vertex of a hypergraph — изолированная вершина гиперграфа.

Isolated vertex subset — изолированное подмножество вершин.

A vertex subset is called **isolated** if the subset contains a vertex which has no neigbours in the subset.

Isometric subgraph — изометрический подграф.

An induced subgraph in G is an **isometric subgraph** in G if the distance between any two vertices in the subgraph equals their distance in the graph. See also *Distance-hereditary graph*.

Isomorphic decomposition — изоморфное разложение.

See *Maximal packing*.

Isomorphic directed graphs — изоморфные орграфы.

Isomorphic embedding problem — проблема изоморфной вложимости.

Isomorphic graphs — изоморфные графы.

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are **isomorphic graphs** (denoted $G_1 \cong G_2$ or $G_1 \sim G_2$) if there is a bijective function f from V_1 onto V_2 such that $(u, v) \in E_1$ iff $(f(u), f(v)) \in E_2$ holds. The function f is called **isomorphism** between two graphs G_1 and G_2. If $G_1 = G_2$, then f is an **automorphism**. The set of automorphisms of a graph forms a group which is called an **automorphism group**. The **isomorphism problem** is to determine all isomorphisms between two given graphs G_1 and G_2. If $G_1 = G_2$, then the isomorphism problem is the **automorphism problem**.

Isomorphic labeled graphs — изоморфные помеченные графы.

See *Labeled graph*.

Isomorphic matroids — изоморфные матроиды.

P_4**-Isomorphic graphs** — P_4-изоморфные графы.

The graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ with the same vertex set V are P_4**-isomorphic graphs** if for any set $S \subseteq V$ of 4 vertices it holds that S induces P_4 in G_1 iff S induces P_4 in G_2.

In 1984, V.Chvatal conjectured that if a graph is P_4-isomorphic to a perfect graph, then it is perfect. This conjecture is known as **Semi-Strong Perfect Graph Conjecture**. In 1987, B.Reed showed that this conjecture is true.

Isomorphic posets — изоморфные частично-упорядоченные множества.

See *Partial order relation.*

Isomorphism problem — проблема изоморфизма.

See *Isomorphic graphs.*

Isoperimetric number — число изопериметричности.

See *Bisection width of a graph.*

Isospectral graphs — изоспектральные графы.

Graphs with the same *spectrum* are called **isospectral**. It is well known that *switching* a regular connected graph into another regular connected graph of the same degree produces an isospectral graph.

Isotropic coloring — изотропная раскраска.

The problem of finding a (t, i, j)-cover of a graph G is equivalent to the following vertex-coloring problem of G. Color the vertices of G in two colors, black and white, such that black vertices correspond to the centers of the covering balls. Thus, the t-neighborhood of every black vertex contains exactly i black vertices and the t-neighborhood of every white vertex contains exactly j black vertices. A similar coloring problem was introduced by V. Vizing. He considered a **distributive** or **isotropic coloring** of a graph, that is, a coloring in which the number of vertices of a fixed color in the 1-neighborhood of any vertex depends only on the color of this vertex.

Let $\varphi : V(G) \to \{0, 1\}$ be a coloring of $V(G)$. We call a vertex $u \in V(G)$ *black* if $\varphi(u) = 1$ and we call a vertex u *white* if $\varphi(u) = 0$. For $u \in V(G)$ and $k \in \{0, 1\}$, let $N_t^k(u)$ be the set of vertices of color k in the t-neighborhood of u.

A coloring φ of G is (t, i, j)**-isotropic** if every black vertex has exactly i black vertices within distance t and every white vertex has exactly j black vertices within distance t.

Iterated clique graph — итерационный граф клик.

See *Clique graph.*

k**-Iterated line digraph** — k-итерационный рёберный орграф.

Setting $L^0G = G$, for any integer $k \geq 1$ the k**-iterated line digraph**, L^kG, is defined recursively by $L^kG = LL^{k-1}G$.

n**-Iterated line graph** — k-итерационный рёберный орграф.

See *Line graph.*

Iteration operation — операция итерации.

J

Join of graphs — соединение графов.

The **join** of the graphs G and H is the graph with a vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H) \cup \{(u,v) : u \in V(G), v \in V(H)\}$.

The join of digraphs D_1 and D_2 consists of $D_1 \cup D_2$ together with all *bidirectional arcs* between any vertex of D_1 and any vertex of D_2.

T**-Join** — T-соединение.

See *Graft*.

Join operation — операция присоединения.

l**-Joinability** — l-соединимость.

Joined vertices — смежные вершины.

Jump graph — прыгающий граф, граф скачков.

Let G be a graph of size q, where $q \geq 1$, and let F and H be edge-induced subgraphs of size k ($1 \leq k \leq q$) in G. We say that H is obtained from F by an **edge jump** if there exist four distinct vertices v, u, w and x in G such that $(u,v) \in E(G)$, $(w,x) \in E(G) - E(F)$, and $H = F - (u,v) + (w,x)$. For any two subgraphs F and H in the graph G, we say that F is j**-transformed** into H if H can be obtained from F by a sequence of j edge jumps. The minimum number of edge jumps required to transform F into H is called the **jump distance** $d_j(F,H)$ from F to H. The k**-jump graph** $J_k(G)$ of G is such a graph whose vertices are the $\binom{q}{k}$ edge-induced subgraphs of size k, that two vertices F and H of $J_k(G)$ are adjacent if $d_j(F,H) = 1$. If $k = 1$, then we refer to $J_k(G) = J_1(G)$ as a **jump graph** of G and denote this more simply by $J(G)$. It is known that $J(G)$ is the complement of the *line graph* $L(G)$ of G.

Jump distance — расстояние скачков.

See *Jump graph*.

k**-Jump graph** — граф k-скачков.

See *Jump graph*.

Justified tree — выровненное дерево.

выровненное дерево.

K

Karp-Miller tree — дерево Карпа-Миллера.

The same as *Coverability tree.*

Kautz digraph — орграф Каутца.

See *Generalized Kautz digraph.*

Kernel — ядро.

An independent set S of vertices in a digraph is called a **kernel** if for each $x \in V(G) \setminus S$, there exists $y \in S$ such that $(x, y) \in E(G)$. See also *Independent set*, *Semikernel.*

When every induced subdigraph of a digraph D has a kernel, D is said to be **kernel-perfect**. D is a **critical kernel-imperfect digraph** if D does not have a kernel but every proper induced subdigraph of D has at least one.

$(k, k-1)$**-Kernel** — $(k, k-1)$-ядро.

A subset $J \subset V(G)$ is said to be a $(k, k-1)$**-kernel** of G if the following properties hold:

(1) for each two distinct vertices $x, y \in J$, $d_G(x, y) \geq k$,

(2) for each $x' \in V(G) \setminus J$, there exists $x \in J$ such that $D_G(x', x) \leq k-1$.

In addition, a subset containing only one vertex is also called a $(k, k-1)$**-kernel** of G.

Note that for $k = 2$ the definition reduces to the definition of a *kernel* of a graph G.

Kernel-perfect digraph — ядро-совершенный граф.

See *Kernel.*

Keyed access method — ассоциативный поиск.

Kings graph — королевский граф.

The **kings graph** K_n is a graph whose vertex set consists of the squares of $n \times n$ chessboard, where two vertices are adjacent if and only if a king can move from one square to the other in a single move according to the chess rules. More formally, $V(K_n) = \{(i, j) : 1 \leq i, j \leq n\}$, where distinct (i, j) and (i', j') are adjacent if and only if $|i - i'|, |j - j'| \leq 1$

Kirchoff matrix — матрица Кирхгофа.

Let us consider the general problem of counting the number of *spanning trees* for an arbitrary *multi-graph* G. This requires that we first concentrate on digraphs and counting the number of *spanning* *out-trees* rooted at a particular vertex. We introduce the so-called **Kirchoff** or **in-degree matrix** $K(G)$. The elements of K are defined as follows:

$$K(i,j) = \begin{cases} d_-(v_i), & i = j, \\ -k, & i \neq j \end{cases},$$

where k is the number of edges from i to j. Thus, the number of spanning out-trees rooted at r in a finite digraph G is equal to $\det(K_r(G)$.

Kleene closure — замыкание Клини.

See *Formal language*.

Kleene star — операция навешивания звездочки Клини.

See *Formal language*.

Knödel graph – граф Кнёделя.

The **Knödel graph** on $n \geq 2$ vertices (n even) and of maximum degree $\Delta \geq 1$ is denoted $W_{\Delta,n}$. The vertices of $W_{\Delta,n}$ are the couples (i,j) with $i = 1,2$ and $0 \leq j \leq \frac{n}{2} - 1$. For every j, $0 \leq j \leq \frac{n}{2} - 1$, there is an edge between vertex $(1,j)$ and every vertex $(2, j + 2^k - 1$ (mod $\frac{n}{2}$), for $k = 0, \ldots, \Delta - 1$.

Knot graph — узловой граф.

König's problem — проблема Кенига.

Königsberg's bridges problem — задача о кенигсбергских мостах.

In 1736 Euler solved a recreational puzzle interesting to the inhabitants of Königsberg (now Kaliningrad). Kaliningrad sits across the river Pregel with seven bridges connecting the various banks and islands of the river as shown. The problem is whether or not it is possible to follow a circular walk starting and finishing at the same river bank and crossing each bridge precisely once.

See also *Eulerian graph*.

Krausz dimension of a graph — Краусова размерность графа.

See *Krausz partition of a graph*.

Krausz partition of a graph — разбиение Крауса графа, Краусово разбиение графа.

A **Krausz partition** of G is a partition of the edge set $E(G)$ into complete subgraphs (that are also called the clusters of the partition). The number of clusters containing a vertex v is called the **order** of v (in the Krausz partition). The order of the partition is the maximum order over all vertices of G. The **Krausz dimension** of G is defined as the minimum partition order over all Krausz partition of G.

Kronecker product — кронекерово произведение, прямое произведение.

See *Product of two graphs*.

Kruskal's algorithm — алгоритм Краскала.

The following algorithm due to Kruskal finds a minimum-weight *spanning-tree*, MWT, of a weighted undirected graph $G = (V, E)$. It is known that it operates in polynomial time.

1. Relabel the elements of E so that
   if $w(e_i) > w(e_j)$ **then** $i > j$
2. $MWT \leftarrow \varnothing$
3. **for** $i = 1$ **to** $|E|$ **do**
   if $MWT \cup \{e_i\}$ **is acyclic then**
   $MWT \leftarrow MWT \cup \{e_i\}$

Kuratowski's criterion — критерий Куратовского.

See *Planarity criteria*.

Kuratowski's theorem — теорема Куратовского.

Theorem. A graph G is planar iff it does not contain a *subdivision* of K_5 and $K_{3,3}$, i.e. iff it does not contain the *minors* K_5 and $K_{3,3}$. The other name is **Pontrjagin-Kuratowski's theorem**.

L

Label — метка.

Labeled graph, labelled graph — помеченный граф.

Let $C = (C_E, C_V)$ be a pair of distinct sets of labels; C_V (resp. C_E) stands for node (resp. edge or arc) labels. A **labeled graph** over C, or simply a C-graph, consists of a graph $G = (V, E, Ends)$ and a labeling function which is a pair of mappings $l = (l_V, l_E)$ such that:
$l_V : V \to C_V$ is the node-labeling function,
$l_E : E \to C_E$ is the edge-labeling function.
Similarly, a labeled directed graph, or simply a $C-d$-graph, is defined as a directed graph with a labeling function l defined as above.
A labeled graph $G = (V, E, \lambda)$, where λ is a labeling function, is **isomorphic** to a labeled graph $G' = (V', E', \lambda')$ if there is a pair $\phi = (\phi_V, \phi_E)$ of one-to-one mappings $\phi_V : V \to V'$ and $\phi_E : E \to E'$ such that $\phi_E((v, w)) = (\phi_V(v), \phi_V(w))$ for every (v, w) in $E(G)$ and $\lambda_{V'} \circ \phi_V = \lambda_V$ and $\lambda_{E'} \circ \phi_E = \lambda_E$.
If a subgraph G of G' is isomorphic to a labeled graph G'', G is called an **occurrence** of G'' in G'. Two occurrences $O = (V_O, E_O, \lambda)$ and $O' = (V_{O'}, E_{O'}, \lambda')$ are overlapping in G, if the set of vertices $V_O \cap V_{O'}$ is not empty.

Labeled Petri net, Labelled Petri net — помеченная сеть Петри.

Let A be an alphabet of action labels. An **labelled Petri net** is a pair (N, L), where N is a Petri net and $L : T \to A \bigcup \{e\}$ assigns to each transition either an action label or the empty string.

Labeled tree, Labelled tree — помеченное дерево.

A **labeled tree** is a tree whose nodes are labeled from a finite alphabet Σ. An **unordered labeled tree** is just a *rooted* labeled tree. An **ordered** labeled tree is a rooted labeled tree in which the *children* of each node are ordered, i.e., if a node has k children, then we can specify them as the first child, the second child, $\dots$, and the kth child.

Labeling — разметка.

A **labeling** (or **valuation**) of a graph is any map that carries some set of graph elements to numbers (usually to the positive or non-negative integers). If the domain is the vertex-set, the edge-set, or the set $V(G) \cup E(G)$, labelings are called respectively **vertex**-labelings, **edge**-labelings or **total** labelings.

Every vertex-labeling induced a natural labeling of the edges: the label of an edge uv is the absolute value of the difference of the labels of u and v.

Labeling of type (a, b, c) — разметка типа (a, b, c).

A **labeling of type** (a, b, c) assigns labels from the set

$$\{1, 2, 3, \ldots, a|V(G)| + b|E(G)| + c|F(G)|\}$$

to the vertices, edges and faces of G such that each vertex receives a labels, each edge receives b labels and each face receives c labels and each number is used exactly once as a label. Labelings of type $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ are also called vertex, edge and face labelings, respectively.

Ladder — лестница.

The **ladder graphs** (or **ladders**), are the graphs $P_2 \times P_n$ of order $2n$. The n copies of P_2 that connect the two copies of P_n are called rungs. Since these graphs are also called $(2, n)$-*meshes*, we denote them by $M_{2,n}$.

Language — язык.

See *Formal language.*

CF-Language — КС-язык.

See *Chomsky hierarchy.*

CS-language — КЗ-язык.

See *Chomsky hierarchy.*

P-Language — Р-язык.

Laplacian matrix — лапласиан.

Let G be a simple graph on n vertices. Let deg_i denote the degree of a vertex v_i, $i = 1, 2, \ldots, n$. Let $A(G)$ denote the adjacency matrix of G and $D(G) = diag(deg_1, \ldots, deg_n)$ be the diagonal matrix of vertex degrees. The **Laplacian matrix** of G is then $L(G) = D(G) - A(G)$. The eigenvalues of $L(G)$, denoted by $\mu_1(G), \mu_2(G), \ldots, \mu_n(G)$, labeled so that $\mu_1(G) \geq \mu_2(G), \ldots, \mu_n(G)$, are called the **Laplacian eigenvalues** of G. These eigenvalues form the **Laplacian spectrum** of G. Because $L(G)$ is a positive semidefinite symmetric matrix, the Laplacian eigenvalues are non-negative real-valued numbers.

For a weighted graph G on vertices labelled $1, \ldots, n$, the **Laplacian**

matrix of G is the $n \times n$ matrix L with

$$L_{ij} = \begin{cases} -\theta, \text{ if } i \neq j \text{ and } (i,j) \text{ is an edge of } G \text{ with weight } \theta, \\ 0, \text{ if } (i,j) \text{ and } (i,j) \text{ is not an edge of } G, \\ \text{the sum of weights of edges incident with } i, \text{ if } i = j \end{cases}$$

It is well-known that if G is connected, then the *nullity* of L is 1, and the *null space* of L is *spanned* by the all ones vector, 1_n. The second smallest eigenvalue of L is known as the **algebraic connectivity** of G.

Laplacian eigenvalues — лапласианово собственное значение.
See *Laplacian matrix.*

Laplacian spectral radius — лапласианов спектральный радиус.
The **Laplacian spectral radius** is the largest eigenvalue of its *Laplacian matrix.*

Laplacian spectrum — лапласианов спектр.
See *Laplacian matrix.*

Large-block program — крупноблочная программа.
Such a large-block schema α that any two its interpretations are equal on the set Σ_α, where Σ_α denotes the subset of Σ used in α is called a **large-block program**.

Large-block program execution — исполнение крупноблочной программы.
This is such a *large-block schema* α that any two its interpretations are equal on the set $\Sigma_\alpha \setminus X_\alpha$, where Σ_α denotes the subset of Σ used in α.

Large-block program schemata — крупноблочные схемы программ.
See *Large-block schema.*

Large-block schema — крупноблочная схема.
Large-block schema is an abstract model of imperative programs that is based on the notion of a program as a finite set of structured statements processing a finite set of structured variables. The class of large-block schemata includes (as proper subslasses) other models of imperative programs such as *Martynyuk schemata* and *standard program schemata.* It is complete with respect to *simulation* and allows the program optimizations affecting both program memory and control structure to be investigated.
Let Σ be an alphabet which consists of mutually disjoint sets of variables (denoted by X), constants, operator symbols, predicate symbols, access symbols and k-case symbols for any integer $k > 1$.

A **large-block program schema** α is a triple $(G_\alpha, R_\alpha, \Omega_\alpha)$, where G_α is a flow-diagram, R_α is a coloring and Ω_α is a nonempty set of interpretations.

A **flow-diagram** (or **control-flow graph**) G_α is an ordered directed graph whose nodes are instructions (or statements). It has exactly one START instruction and a nonempty set of STOP instructions. An instruction with k outgoing arcs is called a k**-recognizer** (or **recognizer**) if $k > 1$ and a **transformer** if $k = 1$.

Every instruction S has a set of **operands** (divided into four disjoint subsets: **strong inputs**, **nonstrong inputs**, **strong outputs**, **nonstrong outputs**); the set for the START instruction is empty, and the set for each STOP consists of only nonstrong inputs. There are two relations on the set of operands: an equivalence relation which divides the set into subsets of **(informationally) connected operands** and a symmetric relation which consists of pairs of so called **(informationally) incompatible operands**. It is assumed that there is no such pairs of operands which are both connected and incompatible, and the outputs of every instruction are mutually incompatible.

Let S be a statement of G_α distinct from START and STOP. Every output d of S has a **data term** Φ of S; Φ is built up using the inputs of S and the constants, and applying the operation symbols to them. If d is a nonstrong output of S and has Φ, d has also an **access term** of the form $g(\Phi_1, \Phi_2, \ldots, \Phi_n, \Phi, d)$, where g is an $(n+2)$-ary access symbol and $\Phi_1, \Phi_2, \ldots, \Phi_n$ are data terms of S. S can have a **predicate term** being a predicate symbol applied to data terms of S, all inputs of which are strong ones. If S is a k-recognizer, S has a k**-case term** being a k-case symbol applied to data terms of S.

The **colouring** R_α is such a mapping of the set of all operands on some subset X_α (called the **memory** of the schema α) of X that $R_\alpha(a) = R_\alpha(b)$ for any two connected operands and $R_\alpha(a) \neq R_\alpha(b)$ for any two incompatible operands. If d is a strong or nonstrong input (output, respectively) of a statement S, then $x = R_\alpha(d)$ is called a **strong** or **nonstrong argument** (**result**, respectively) of S.

An interpretaion $I \in \Omega_\alpha$ consists of

(1) a nonempty set D_α called the domain of I;

(2) an assignment of an element $I(c) \in D_I$ to each variable and constant c;

(3) an assignment of a function $I(f) : D_I^n \to D_I$ to every n-ary operator symbol f;
(4) an assignment of a predicate $I(h) : D_I^n \to \{$ **true, false**$\}$ to every n-ary predicate symbol h;
(5) an assignment of a (access) function $I(g) : D_I^n \to D_I$ to every n-ary access symbols g, such that for any access functions $F_1, F_2, \ldots, F_m$ and for any elements $a_1^1, \ldots, a_{k_1-1}^1, a_1^2, \ldots, a_{k_1-1}^2, \ldots, a_1^m, \ldots, a_{k_1-1}^m, a$ from D_I if $F_1 = F_m$, $a_1^1 = a_1^m$, $\ldots$ $a_{k-1}^1 = a_{k_m-1}^m$, then $F_1($ a_1^1, $\ldots$, $a_{k_1-1}^1$, $F_2($ a_1^2, $\ldots$, $a_{k_1-1}^2$, $\ldots$, $F_{m-1}($ a_1^{m-1}, $\ldots$, $a_{k_1-1}^{m-1}$, $F_m($ $a_1^m, \ldots, a_{k_1-1}^m, a)) \ldots)) = F_1($ a_1^1, $\ldots$, $a_{k_1-1}^1$, $F_2($ a_1^2, $\ldots$, $a_{k_1-1}^2$, $\ldots$, $F_{m-1}($ a_1^{m-1}, $\ldots$, $a_{k_1-1}^{m-1}, a)$ $\ldots))$;
(6) an assignment of a function $I(T) : D_I^n \to \{1, 2, \ldots, k\}$ to every n-ary k-case symbol T.

Inclusion and equivalence of schemata are defined as follows:
(1) α **includes** (or **extends**) β iff $\Omega_\alpha \subseteq \Omega_\beta$ and for any $I \in \Omega_\alpha$, $val(\alpha, I) = val(\beta, I)$ whenever $val(\alpha, I)$ (see *Value of schema under interpretation*) is defined;
(2) α and β are **(strongly) equivalent** iff either of them includes the other.

Lattice graph — граф решётки.

The same as *Grid graph.*

Lavrov schemata — схемы Лаврова.

Layout — укладка, нумерация.

A **layout** (or **linear layout, linear arrangement**) of a graph $G = (V, E)$ is an assignment of distinct integers from $\{1, \ldots, n\}$ to the elements of V. Equivalently, a **layout** $\mathcal{L}$ may be thought of as an ordering $\mathcal{L}(1)$, $\ldots$, $\mathcal{L}(n)$ of V, where $|V| = n$. A **tree layout** is a linear arrangement of a tree. If p is a non-integer point on x-axis, then the **cut** of the layout $\mathcal{L}$ at p, denoted $cut_{\mathcal{L}}(p)$ is the number of edges that cross over p, i.e. the number of edges $(u, v) \in G$ with $\mathcal{L}(u) < p < \mathcal{L}(v)$. The **cutwidth** of a layout $\mathcal{L}$, denoted $cut(\mathcal{L})$, is the maximum cut of $\mathcal{L}$ over all possible values; namely,

$$cut(\mathcal{L}) = \max_{1<p<|V|} cut_{\mathcal{L}}(p).$$

The **cutwidth of a graph** G, denoted $cut(G)$, is the minimum cutwidth of any linear arrangement of G.
The **width** of a layout $\mathcal{L}$, $b(G, \mathcal{L})$, is the maximum of $|\mathcal{L}(u) - \mathcal{L}(v)|$ over all edges (u, v) of G. That is, it is the length of the longest edge

in the layout. The **bandwidth** of G, $bw(G)$, is the minimum width over all layouts. A **bandwidth layout** for a graph G is a layout satisfying $b(G, \mathcal{L}) = bw(G)$.

See also *Bandwidth, Separation-width.*

Leaf — лист.

1. See *Directed tree.*

2. A **leaf** is a vertex of degree one.

3. See *Directed hyperpath.*

4. A **leaf** is any 2-edge-connected subgraph, trivial or not, maximal with respect to inclusion. Thus every vertex belongs to a unique *leaf* of a graph. The number of leaves of H is denoted by $l(H)$.

Leaf density — листовая плотность.

The **leaf density** $\zeta(G)$ of G is defined as

$$\zeta(G) = \frac{l(G) - s(G)}{n}.$$

Here, $l(G)$ is the number of leaves in G, $s(G)$ is the number of stems (a vertex that is adjacent to a leaf is called a **stem**).

Least upper bound — наименьшая верхняя грань.

Lee scheme — коды Ли.

Left-derivation tree — дерево левых выводов.

Left linear tree — левостороннее дерево.

Leftmost derivation — левый вывод.

Left-sided balanced tree — левостороннее балансированное дерево.

See *Height balanced tree.*

Length of an arc — длина дуги.

Length of a chain — длина цепи.

Length of a circuit — длина цикла.

See *Circuit.*

Length of a cycle — длина контура.

The **length** $|C|$ **of a cycle** C is the number of its vertices.

Length of a hypercycle — длина гиперконтура.

See *Hypercycle.*

Length of a path — длина пути.

In an unweighted graph, the **length of a path** is the number of arcs in the path. In a weighted graph, the **length of a path** is the sum of the weights of all edges of the path. The **shortest-path distance** between u and v is the minimum length of a path from u to v. The path of minimum length is called a **shortest path** between u and v.

Length of a string — длина цепочки.

See *String*.

Length of a vertex — длина вершины [в мультираскраске].

See *Multi-coloring*.

Letter — буква.

See *Alphabet*.

Level representation of rooted trees — уровневые коды корневых деревьев.

Lexicographic order — лексикографический порядок.

For $1 \le u \le m$, define a relation $<_u$ in R^m by requiring that two vectors $\vec{i} = (i_1, \dots, i_m)$ and $\vec{j} = (j_1, \dots, j_m)$ satisfy

$$\vec{i} <_u \vec{j} \text{ iff } i_1 = j_1, \dots, i_{u-1} = j_{u-1}, \text{ and } i_u < j_u.$$

Let $<$ denote the union of the relations $<_u$ for $1 \le u \le m$; it is called the **lexicographic order** in R^m.

Lexicographic product — лексикографическое произведение, композиция графов.

Given graphs G and H, the **lexicographic product** $G[H]$ has a vertex set $\{(g, h) : g \in V(G),\ h \in V(H)\}$ and two vertices (g, h), (g', h') are *adjacent* if and only if either (g, g') is an edge of G or $g = g'$ and (h, h') is an edge of H.

The other name is **Wreath product**.

Light edge — лёгкое ребро.

Let α denote an *average degree*, and δ denote the *minimum degree* of a graph. An edge is **light** if both its *endpoints* have a degree bounded by a constant depending only on α and δ. A graph is **degree-constrained** if $\alpha < 2\delta$. It is known that every degree-constrained graph has a light edge.

Light graph — лёгкий граф.

A graph H is defined to be **light** in a family $\mathcal{H}$ of graphs if there exists a finite number $w(H, \mathcal{H})$ such that each $G \in \mathcal{H}$, which contains H as a subgraph, contains also a subgraph $K \cong H$ such that the sum of degrees (in G) of the vertices of K (that is, the weight of K in G) is at most $w(H, \mathcal{H})$. Otherwise, we call the graph **heavy**.

Limit flow graph — предельный граф.

See *Interval graph.II.*

Line — ребро.

Linear component — линейная компонента.

A **linear component** of a *cf-graph* G with the initial node p_0 and the

terminal node q_0 is defined as a hammock C such that the following properties hold:
(1) the initial and terminal (if it exists) nodes of C belongs to every path in G from p_0 to q_0,
(2) the initial node of C is not reachable in G from the terminal node of C,
(3) there is no proper subfragment A of C such that A is a hammock and have the first two properties.

Linear-bounded automation — линейно-ограниченный автомат.
See *Model of computation.*

Line-chromatic number — реберно-хроматическое число.

Line covering — реберное покрытие.

Line-covering number — число реберного покрытия.

Line digraph — реберный орграф.
Given a digraph $G = (V, A)$, the digraph $LG = (V(LG), A(LG))$ where each vertex represents an arc of G, that is,

$$V(LG) = \{uv|(u, v) \in A(G)\}$$

is called a **line digraph**. A vertex uv is adjacent to a vertex wz if and only if $v = w$, that is, whenever the arc (u, v) of G is adjacent to the arc (w, z). The maximum and minimum out- and in-degrees of LG are equal to those of G. Therefore, if G is *d-regular* with the *order* n, then LG is d-regular and has the order dn. If G is a *strongly connected digraph* different from a directed cycle, then the *diameter* of LG is the diameter of G plus one.

Line graph — реберный граф.
The **line graph** $L(G)$ of a graph G is that whose vertices are the edges of G and two vertices of $L(G)$ are *adjacent* iff the correspondent edges of G have an end vertex in common.
The n**-iterated line graph** $L^n(G)$ of a graph G is defined to be $L(L^{n-1}(G))$, where $L^1(G)$ denotes the line graph $L(G)$ of G, and $L^{n-1}(G)$ is assumed to have a nonempty edge set.

Line graph of a hypergraph — реберный граф гиперграфа.
The **line graph** $L(\mathcal{H}) = (\mathcal{H}, E)$ **of a hypergraph** $\mathcal{H}$ is the intersection graph of $\mathcal{H}$, i.e., $ee' \in E$ if and only if $e \cap e' \neq \emptyset$.

Line graph of a mixed graph — рёберный граф смешанного графа.
Let $G = (V(G), E(G))$ be a *mixed graph* without loops. The **line graph** of G is defined to be $G^l = (V(G^l), E(G^l))$, where $V(G^l) =$

$E(G)$. For $e_i, e_j \in V(G^l)$, $e_i e_j$ is an unoriented edge in G^l if e_i, e_j are unoriented edges in G and have a common vertex, or one of e_i, e_j is an oriented edge in G and their common vertex is the positive end of the oriented edge, or both e_i and e_j are oriented edges in G and their common vertex is their common positive (or negative) end; $e_i \to e_j$ is an oriented edge in G^l, where e^i and e_j are the positive and negative ends of $e_i \to e_j$, respectively, if e_i is an unoriented edge, e_j is an oriented edge in G and their common vertex is the negative end of e_j, or both e_i and e_j are oriented edges in G and their common vertex is the positive and negative ends of e_i and e_j, respectively.

Line group of a graph — реберная группа графа.

Line incident with a vertex — ребро, инцидентное вершине.

Line-independence number — реберное число независимости.

Line-symmetric graph — реберно-симметрический граф.

F**-Line** — F-линия.

Linear k**-arboricity of a graph** — линейная древесность графа.

The **linear** k**-arboricity of a graph** G, denoted by $la_k(G)$, is the least integer m such that G can be edge-partitioned into m *linear* k*-forests*. Clearly, $la_1(G)$ is the edge chromatic number, or *chromatic index* $\chi'(G)$ of G.

The linear k-arboricity of a graph was first introduced by M. Habib and P. Péroche (1982).

Linear arrangement — линейная укладка, линейное упорядочение.

See *Layout*.

Linear bounded automaton — линейно ограниченный автомат.

Linear component — линейная компонента.

Linear extension of a poset — линейное расширение чу-множества.

Given a poset $P = (X, \leq)$, a **linear extension** of P is a poset $L = (X, \leq')$ [note, that ground sets are the same] with the properties that (a) L is a linear order and (b) for all $x, y \in X$, if $x \leq y$, then $x \leq' y$ [extension]. A family $\mathcal{R} = \{L_1, L_2, \ldots, L_t\}$ of linear extensions of $P = (X, \leq)$ is called a **realizer** of P provided that, for all $x, y \in X$, $x \leq y$ if and only if $x \leq_i y$ for all $i = 1, 2, \ldots, t$. The **dimension** of a poset P, denoted by $\dim P$, is the smallest size of a realizer of P.

Linear forest — линейный лес.

A **linear forest** is a disjoint union of paths and isolated vertices.

Linear k**-forest** — линейный k-лес.

The **linear k-forest** is a graph whose components are paths of length at most k.

Linear hypergraph — линейный гиперграф.

A hypergraph is **linear** if no two edges intersect in more than one vertex.

Linear layout — линейная укладка.

See *Layout*.

Linear matroid — линейный матроид.

See *Matrix matroid*.

Linear NCE graph grammar — линейная графовая грамматика типа NCE.

An NCE graph grammar is **linear** (or **L-NCE**) if the axiom and the right hand-side of each production have at most one nonterminal node.

Linear order — линейный порядок.

See *Partial order relation*.

Linear scheme (code, presentation) — линейный код.

Linear subgraph of a directed graph — линейный подграф орграфа.

Linear subgraph of a graph — линейный подграф графа.

Linear tree — линейное дерево.

Linear vertex arboricity — линейная вершинная древесность.

A subset of $V(G)$ is called an LV-set if it induces a *linear forest* in G. A partition of V is called an LV-partition if every subset in the partition is an LV-set. **Linear vertex arboricity** of G, denoted by $\rho'(G)$, is the smallest number of subsets into which the vertex set V can be partitioned so that the partition is an LV-partition.

(a,b)**-Linear class of graphs** — (a,b)-линейный класс графов.

Given a and $b \in Q^+$, we define the (a,b)**-linear class**, denoted by $L(a,b)$, to be the set of all connected graphs such that $m = an - b$. The (1,1)-linear coclass coincides with the set of all trees and $L(1,0)$ characterizes the set of connected graphs with only one cycle.

Liouville property of an operator on graphs — свойство Лиувиля оператора на графе.

Given an operator $\mathcal{L}$ on a graph and a class $\mathcal{F}$ of solutions of $\mathcal{L}$, by the **Liouville property** of the pair $(\mathcal{L},\mathcal{F})$ we mean that the dimension of the space of all solutions of the operator $\mathcal{L}$ in $\mathcal{F}$ is at most one.

List assignment — приписывание цветов.

See *L-coloring with impropriety d*.

List chromatic number — предписанное хроматическое число.

The **list chromatic number** $l\chi(G)$ of the graph G is the smallest k such that whenever each vertex $v \in V(G)$ is assigned a list $\Phi(v)$ of k admissible colours, there exists such a proper colouring f of $V(G)$ that each vertex v is coloured by a colour $f(v) \in \Phi(v)$. The **list edge chromatic number** $le\chi(G)$ is defined analogously. We can also say that $le\chi(G)$ is the **list chromatic number** of the *line graph* $L(G)$. Other names are **Prescribed chromatic number, Choice number, Choosability**.

List coloring — предписанная раскраска.

Let $V = \{v_1, \ldots, v_n\}$ be the vertices of G, L_i denote the list (= a set of admissible colors) associated with v_i, and $IL = L_1 \cup \cdots \cup L_n$. A mapping $\varphi : V \to IL$ is a **list coloring**, if φ is a *proper coloring* and $\varphi(v_i) \in L_i$ holds for all $1 \leq i \leq n$. See also *List chromatic number.*

List edge chromatic number — предписаное рёберное хроматическое число.

See *List chromatic number.*

List edge-coloring problem — задача предписанной раскраски рёбер.

See *List chromatic number, List total coloring problem.*

List homomorphism — предписанный гомоморфизм.

Given graphs H, G, and lists $L(v) \subseteq V(G)$, $v \in V(H)$, a **list homomorphism** of H to G, with respect to the lists L, is a homomorphism $f : H \to G$, such that $f(v) \in L(v)$ for all $v \in V(H)$. For a fixed graph G, the **list homomorphism problem** L-HOM G asks, whether or not an input graph H with lists L admits a list homomorphism of H to G.

List total coloring — предписанная тотальная раскраска.

Suppose that a set $L(x)$ of colors, called a list of x, is assigned to each element $x \in V(G) \cup E(G)$. Then a total coloring φ of G is called a **list total coloring** of G for L if $\varphi(x) \in L(x)$ for each element $x \in V(G) \cup E(G)$, where $\varphi(x)$ is the color assigned to x by φ. The list total coloring φ is simply called an **L-total coloring**. An ordinary total coloring is an L-total coloring for which all lists $L(x)$ are same. Thus an L-total coloring is a generalization of a total coloring. The **list total coloring problem** asks whether a graph G has an L-total coloring for given G and L. The problem is NP-complete in general, because the ordinary total coloring problem is NP-complete. The **list vertex-coloring problem** and **list edge-coloring problem** are similarly defined. The list vertex-coloring problem can be solved in polynomial time for *partial k-trees* and hence for *series-parallel graphs.*

List total coloring problem — задача предписанной тотальной раскраски.

See *List total coloring.*

List vertex-coloring problem — задача предписанной раскраски вершин.

See *List coloring, List total coloring.*

Live transition — живой переход.

Liveness problem — проблема живости.

Local computation on graphs — локальные вычисления на графах.

Local-edge-connectivity — локально рёберная связность.

See *Edge connectivity.*

Local exponent of digraph — локальная экспонента орграфа.

See *Primitive directed graph.*

Local input place — локальное входное место.

Local irregularity of a digraph — локальная иррегулярность орграфа.

See *Irregularity of a digraph.*

Local isomorphism — локальный изоморфизм.

A **local isomorphism** of a directed graph H is an isomorphism of a finite induced subgraph of H to a finite induced subgraph of H.

Local independence number — локальное число независимости.

The **local independence number** $\alpha_i(G)$ of a graph G at a distance i is the maximum number of independent vertices at distance i from any vertex.

Local output place — локальное выходное место.

Local place — внутреннее место.

Local replacement method — метод локальной замены

Local tree-width — локальная древесная ширина.

We define the r-neighborhood $N_r(v)$ of a vertex $v \in V(G)$ to be the set of all vertices $w \in V(G)$ of distance at most r from v, and we let $\langle N_r(v)\rangle$ denote the subgraph induced by G on $N_r(v)$. Then, denoting the *tree-width* of a graph H by $tw(H)$, we let

$$ltw^G(r) = \max\{tw(\langle N_r(v)\rangle)|\ v \in V(G).$$

$ltw^G(r)$ is called **local tree-width**.

Locally k-connected graph — локально k-связный граф.

Let M and H be two subgraphs of G with $V(H) \cap V(M) = \emptyset$. We say that H is **locally k-connected to M in G** if G contains k pairwise disjoint $(x, V(M))$ paths for every vertex $x \in H$. Let now M be a cycle in G, and let H be a subgraph of $G - V(M)$. We say that M is **locally longest with respect to H in G** if we cannot obtain a cycle longer than M by replacing a *segment* $M[u, v]$ by a (u, v)-path of G through H.

Locally countable graph — локально счетный граф.

Locally finite graph — локально конечный граф.

A graph is called **locally finite** if every its vertex has a finite degree (valency). In other words, a graph is **locally finite** if every vertex has a finite *indegree* and *outdegree*.

A graph is called **almost locally finite** if only finitely many of its vertices have infinite degrees.

Locally longest with respect to M cycle — локально длиннейший относительно M цикл.

See *Locally k-connected graph*.

Locally restricted graph — локально ограниченный граф.

A graph $G = (V, A)$ is called **locally restricted** if it has a bounded degree, i.e. if there is a constant $M > 0$ such that $deg(v) \leq M$ for any vertex $v \in V$.

Locally semicomplete digraph — локально полуполный орграф.

See *Neighborhood of a vertex*.

Locating-dominating set — размещённое доминирующее множество.

Slater (1987) defined a **locating-dominating set**, denoted by an LD-set, in a connected graph G to be a dominating set D of G such that for every two vertices u and v in $V(G) - D$, $N(u) \cap D \neq N(v) \cap D$. The **location-domination number** $\gamma_L(G)$ is the minimum cardinality of an LD-set for G.

Locating set — размещённое множество.

Let $S = \{v_1, \ldots, v_k\}$ be a set of vertices in a connected graph G and let $v \in V(G)$. The k-vector (ordered k-tuple) $c_S(v)$ of v with respect to S is defined by

$$c_S(v) = (d(v, v_1), \ldots, d(v, v_k)),$$

where $d(v, v_i)$ is the distance between v and v_i $(1 \leq i \leq k)$. The set S is called a **locating set** if the k-vectors $c_S(v)$, $v \in V(G)$, are distinct.

The **location number** $loc(G)$ of G is the minimum cardinality of a locating set in G.

Location number — число размещения.

See *Locating set*.

Location-domination number — число размещённого доминирования.

See *Locating-dominating set*.

Logic for expressing graph properties — логика для выражения свойств графа.

Any labelled graph may be defined as a logical structure

$$(V_G, E_G, (lab_{a,G})_{a \in C_V}, (edg_{b,G})_{b \in C_E})$$

where V_G is the set of vertices, E_G is the set of edges, C_V is a set of vertex labels and C_E is a set of edge labels, moreover, the meaning of the predicates is the following:
(1) $lab_{a,G}(v)$ is true iff the vertex v has a-label in G,
(2) $edg_{b,G}(e, v, v')$ is true iff e is an edge (v, v') in G and has a b-label in G.

To define the sets of graphs, one considers formulas built by using individual variables (vertex variables or edge variables), set variables (sets of vertices or sets of edges) and binary relation variables (subsets of $V_G \times V_G$ or $V_G \times E_G$ or $E_G \times E_G$).

Atomic formulas are the following:
(1) $x = x'$, where x, x' are two vertices or two edges;
(2) $lab_a(v)$, where v is a vertex;
(3) $edg_b(e, v, v')$, where e is an edge and v, v' are two vertices;
(4) $x \in X$, where X is a set of vertices or a set of edges;
(5) $(x, y) \in R$, where R is a binary relation included in a cartesian product $X \times Y$ with X which is a set of vertices or a set of edges, the same for Y.

A **First Order formula** is a formula formed with the above atomic formulas numbered from (1) to (3) together with boolean connectives **OR, AND, NOT**, the individual quantifications $\forall x, \exists x$ (where x is a vertex or an edge).

A **Monadic Second Order formula** is a formula formed with the above atomic formulas numbered from (1) to (4) together with boolean connectives **OR, AND, NOT**, the individual quantifications $\forall x, \exists x$ (where x is a vertex or an edge) and the set quantifications $\forall X, \exists X$ (where X is a set of vertices or a set of edges).

A **Second Order formula** is a formula formed with the above atomic formulas numbered from (1) to (5) together with boolean connectives **OR, AND, NOT**, the individual quantifications $\forall x, \exists x$ (where x is a vertex or an edge), the set quantifications $\forall X, \exists X$ (where X is a set of vertices or a set of edges) and the binary relations quantifications $\forall R, \exists R$ (where R is a binary relation).

Loop — петля, цикл.

An arc of the form (v, v) is called a **loop**. The other name is **self-loop**.

l-**Loop** — l-цикл.

Loop of matroid — цикл матроида.

Loop region — циклический участок.

Lower independence number — нижнее число независимости.

See *Independence number*.

M

Magic labeling — магическая разметка.

Magic labeling is one-to-one map onto the appropriate set of consecutive integers starting from 1, satisfying some kind of "constant-sum" property. A **vertex-magic** labeling is one in which the sum of all labels associated with a vertex is a constant independent of the choice of the vertex. **Edge-magic** labelings are defined similarly. **Vertex-magic total labeling** is a one-to-one mapping

$$\lambda :\ E \cup V \rightarrow \{1, 2, \ldots, |V| + |E|\}$$

with the property that there is a constant k such that at any vertex x

$$\lambda(x) + \sum \lambda(xy) = k$$

where the sum is over all vertices y adjacent to x. For any labeling we call the sum of the appropriate labels at a vertex the **weight** of the vertex, denoted $wt(x)$; so for vertex-magic total labelings we require that the weight of all vertices be the same, namely k and this number is called the **magic constant** for the labeling.

Magnet in a graph — магнит в графе.

A **magnet** in a graph $G = (V, E)$ is defined as a pair (a, b) of adjacent vertices with the same weight and such that each vertex in $N_G(a) \setminus N_G(b)$ is adjacent to each vertex in $N_G(b) \setminus N_G(a)$. In other words, the two endpoints of an edge induce a magnet in a graph G if and only if this edge is not the middle edge of any P_4 in G.

Magnitude of a flow — мощность потока, величина потока.

See *Flow.*

Main eigenvalue — главное собственное значение.

An eigenvalue is **main** if it has an associated eigenvector the sum of whose entries is not equal to zero.

Majority dominating function — функция мажоритарного доминирования.

See *Dominating function.*

Majority domination number — число мажоритарного доминирования.

See *Dominating function.*

Map — карта.

Mark — пометка.
Marked graph — маркированный граф.
Marked trap — размеченная ловушка.
Marker — маркер.
Marking — разметка, маркировка.

1. A **marking** of a *sigraph* S is an assignment of positive and negative signs to the vertices of S. That is, a marking is a function from the vertex set of S to the set $\{-1, 1\}$.

2. See *Petri net*.

Marking operation — операция разметки.
Marriage problem — задача о свадьбах.
Martynyuk schemata — схемы Мартынюка.

Martynuk schemata do not contain any information about a program except for a **control flow graph**. An identity relation can be introduced between vertices of the control flow graph. Two Martynuk schemata are regarded as **equivalent** if they have the same set of so called **chains** (i.e. paths in the control flow graph from the entry to the exit vertices). The problem of recognition of equivalence is decidable here, since the set of chains of an oriented graph with specified entry and exit vertices is a regular event.

The class of Martynyuk schemata is a proper subclass of *large-block schemata*. It consists of all such large-block schemata α that the following properties hold:

(1) $X_\alpha = \{x\}$,

(2) Ω_α is the set of all possible interpretations,

(3) every transformer has two operands: strong input and nonstrong output,

(4) every recognizer has one operand: strong input.

Matching — паросочетание.

For a graph $G = (V, E)$, a subset $E' \subseteq E$ such that for all edges $e, e' \in E'$ with $e \neq e'$ $e \cap e' = \emptyset$ holds. A **maximum-cardinality matching** is a matching which contains a maximum number of edges. A **perfect matching** is a matching in which every vertex of the graph is an end-point of some element of the matching. Not every graph contains a perfect matching.

Matching equivalent — эквивалентность по паросочетаниям.

Two graphs are said to be **matching equivalent** if they have the same *matching polynomial*.

Matching number — число паросочетания.

The **matching number** $v(H)$ of a hypergraph H is the maximal size of a *matching* in H.

Another name is the **Edge-independent number**.

Matching polynomial — полином паросочетаний.

Let $p(G, k)$ be the number of matchings of the graph G with k edges. Then the **matching polynomial** of G is

$$\mu(G, x) = \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k p(G, k) x^{n-2k}.$$

It is known that $\mu(G, k)$ has only real roots.

Matching width — ширина паросочетания.

See *F-width*.

k-Matching — k-паросочетание.

A *k-matching* in a *hypergraph* G is a collection of edges of G such that each point belongs to at most k of them (note that repetition of edges is allowed). A 1-matching is also called *matching*. A k-matching can be considered as a mapping $w : E(G) \to \{0, 1, \ldots\}$ such that

$$\sum_{E \ni x} w(E) \leq k$$

for every point x ($w(E)$ is the multiplicity with which E occurs in the matching). A **perfect k-matching** is a k-matching such that each edge belongs to exactly k members of it (note the difference between this and a *k-factor*!). A **fractional matching** is an assignment of a non-negative real weight $w(E)$ to each edge E such that

$$\sum_{E \ni x} w(E) \leq 1$$

for every point x.

The **fractional matching number** of G is the supremum of

$$\sum_{e \in E(G)} w(e)$$

over all fractional matchings w.

F-Matching width — ширина F-паросочетания.

See *F-width*.

G-Matching function — G-отображающая функция.

G-отображающая функция.

Matrix-tree theorem — матричная теорема о деревьях.

Matrix graph — граф матрицы.

For an $n \times n$ real symmetric matrix A, the graph $G(A) = (V, E)$ of A is defined by: $V(G) = \{1, \ldots, n\}$ and $E(G) = \{(i, j) : \ i \neq j, \ a_{ij} \neq 0, \ i, j = 1, \ldots, n\}$.

Matrix matroid — матричный матроид.

Let M be an $m \times n$ matrix over some field K, $E(M)$ the set of all the column vectors of M, and $\mathcal{I}(M)$ the family of all the linearly independent sets of column vectors of M, where we assume the empty set $\emptyset \in \mathcal{I}(M)$. Then, $\mathcal{M}(M) = (E(M), \mathcal{I}(M))$ is a matroid. A matroid obtained in this way is called a **matrix matroid** (or a **linear matroid**) and is called (linearly) representable over the field K. A matroid representable over the field $GF(2)$ is said to be **binary**, and one representable over any field, **regular**.

Matroid — матроид.

A **matroid** $\mathcal{M} = (E, \mathcal{I})$ is a pair of a finite set E and a family $\mathcal{I}$ of elements of E such that

(I0) $\mathcal{I}$ is non-empty;

(I1) if $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$;

and

(I2) if $I, J \in \mathcal{I}$ and $|I| < |J|$, then there is an element $e \in J - I$ such that $I \cup \{e\} \in \mathcal{I}$.

An element of $\mathcal{I}$ is called an **independent set of a matroid** $\mathcal{M}$, and an element in $2^E \setminus \mathcal{I}$ is called a **dependent set**, where 2^E is the set of all the subsets of E.

The system of postulates (I0) -(I2) is equivalent to that of (I0), (I1) and

(I2') for $X \subseteq E$, if I and J are two maximal independent subsets of X, then $|I| = |J|$.

A maximal independent set in $\mathcal{I}$ is called a **base** of $\mathcal{M}$, and a minimal dependent set a **circuit**.

By (I2'), any maximal independent subset of a subset X of E has a common cardinality, which is called the **rank** of X and denoted by $\rho(X)$, i.e.

$$\rho(X) = \max\{|I| : \ I \subseteq X, I \in \mathcal{I}\} \ (X \subseteq E).$$

$\rho(E)$ is called the **rank of a matroid** $\mathcal{M}$. The function $\rho : 2^E \to N_+$

(where N_+ is the set of non-negative integers) is called the **rank function** of $\mathcal{M}$.

Matroid cocycle space — пространство коциклов матроида.

Matroid connectivity — связность матроида.

For $X \subseteq E$, the **connectivity function**, λ, is defined by

$$\lambda(X) = r(X) + r(E - X) - r(M).$$

Observe that $\lambda(X) = \lambda(E - X)$. For $j \geq 1$, a partition $(X, E - X)$ of E is called a j**-separation** if $|X| \geq j$, $|E - X| \geq j$, and $\lambda(X) \leq j - 1$. When $\lambda(X) = j - 1$, we call $(X, E - X)$ an **exact** j**-separation**. For $k \geq 2$ we say matroid M is k**-connected** if M has no j-separation for $j \leq k - 1$. This definition of connectivity is refered to as Tutte k-connectivity to distinguish it from other types of k-connectivity. Tutte connectivity is invariant under duality. Moreover, from the definition it is clear that matroid connectivity begins with 2-connectivity. So when we say "connected matroid" we mean 2-connected matroid. The next two results compare matroid connectivity with graph connectivity.

Theorem 1. Let G be a graph with at least three vertices and no isolated vertex. Then $M(G)$ is 2-connected if and only if G is 2-connected and has no loop.

Theorem 2. Let G be a graph with at least three vertices, no isolated vertex, and $G \not\approx K_3$. Then $M(G)$ is 3-connected if and only if G is 3-connected and has no loop or parallel edges.

Matroid cycle space — пространство циклов матроида.

Matthews graph — граф Метьюза.

The **Matthews graph** is the line graph obtained by subdividing a *perfect mathching* in the *Petersen graph.*

Maxclique — максимальный полный подграф.

See also *Clique.*

Max-flow min-cut theorem — теорема о наибольшем потоке и наименьшем разрезе.

Theorem. For any network, the maximum amount of flow from source to sink is equal to the minimum capacity of all cuts separating source and sink.

Another name is **Ford-Fulkerson's theorem**.

Maximal complete subgraph — максимальный полный подграф.

See also *clique.*

Maximal dominating set — максимальное доминирующее множество.
See *Maximal domination number*.

Maximal domination number — максимальное доминирующее число.
A *dominating set* D of G is a **maximal dominating set** of G if $V(G) - D$ is not a dominating set of G. The **maximal domination number** $\gamma_m(G)$ of G is the minimum cardinality of a maximal dominating set of G.

Maximal exclusion graph — максимальный граф исключения.

Maximal flow — наибольший (максимальный) поток.

Maximal independence number — число максимальной независимости.
See *Independence number*.

Maximal packing — максимальная упаковка.
A **maximal packing** of a digraph $D = (V, A)$ with isomorphic copies of a digraph d is a set $\{d_1, d_2, \ldots, d_n\}$, where $d_i \cong d$ and $V(d_i) \subset V(D)$ for all i, $A(d_i) \cap A(d_j) = \emptyset$ if $i \neq j$, $\cup_{i=1}^n d_i \subset D$ and

$$|A(D) \setminus \cup_{i=1}^n A(d_i)|$$

is minimal.
A maximal packing of D with isomorphic copies of d such that $\cup_{i=1}^n d_i = D$ is an **isomorphic decomposition** of D into copies of d (or a **"d-decomposition of D"** for short).
Packings and decompositions of undirected graphs are similarly defined.

Maximal singular graph —максимальный сингулярный граф.

Maximal strongly singular graph — максимальный сильно сингулярный граф.

Maximal subnet — максимальная подсеть.

Maximal tree — максимальное дерево.

Maximally irregular graph — максимально иррегулярный граф.
See *Regular graph*.

Maximum-cardinality matching — паросочетание максимальной мощности.
See *Matching*.

Maximum edge-connected graph — максимальный рёберно-связный граф.
See *Connected graph*.

MAXIMUM FLOW problem — проблема MAXIMUM FLOW.
See *Flow*.

Maximum hyperflow problem — задача о максимальном гиперпотоке.

See *Hyperflow*.

MAXIMUM INDEPENDENT SET problem — задача о нахождении наибольшего независимого множества.

The **MAXIMUM INDEPENDENT SET problem** (or **MISP**) consists in finding an independent set of the largest cardinality. This problem is known to be *NP*-hard, approximable within factor $\mathcal{O}(|V|/(\log |V|)^2)$, and not approximable within factor $|V|^{1-\epsilon}$ for any $\epsilon > 0$.

Maximum matching graph — граф наибольших паросочетаний.

The **maximum matching graph** of a graph G has a vertex for each maximum matching and an edge for each pair of maximum matchings which differ by exactly one edge.

Maximum neighbour — максимальный сосед.

See *Dually chordal graph*.

Maximum neighbourhood ordering — упорядочение максимального соседства.

See *Dually chordal graph*.

Maximum point-connected graph — максимальный точечно-связный граф.

See *Connected graph*.

McGee graph — граф МакГи.

See *(k, g)-Cage*.

Mean diameter — средний диаметр.

Median generalized binary split tree — медианное обобщенное бинарное расщепляемое дерево.

A **median generalized binary split tree** is constructed by selecting the median to be the split values.

Median graph — медианный граф.

A connected graph G is a **median graph** if for every triple u, v, w of its vertices

$$|I(u, v) \cap I(u, w) \cap I(v, w)| = 1.$$

See *Interval $I(u, v)$*.

Median split tree — медианное расщепляемое дерево.

Median split tree (MST) selects the median (w.r.t. the lexical ordering) of the remaining keys as the split value.

Membership problem — проблема принадлежности.

Memory state — состояние памяти.

See *Value of schema under interpretation.*

Menger's theorem — теорема Менгера.

1. (An edge-form of the theorem). Let G be an unoriented graph with two distinguished vertices s and t. The maximum number of edge-disjoint paths joining s and t is equal to the minimum number of edges in a cut separating s and t.

2. (A vertex-form of the theorem). The minimum number of vertices separating two nonadjacent nodes s and t is equal to the maximum number of vertex disjoint $s-t$ paths.

Mergeable heap — сливаемое дерево.

n**-mesh** — n-сеть.

An n**-dimensional mesh** (abbreviated n-mesh) is the *Cartesian product* of n path graphs $P_{r_1}, \ldots, P_{r_n}$ of orders r_i and is denoted by $M(r_1, \ldots, r_n)$. Thus, $V(M) = \{(a_1, \ldots, a_n)| \ (1 \leq a_i \leq r_i)\}$ and for $x, y \in V(M)$, $(x, y) \in E(M)$ if and only if

$$\sum_{i=1}^{m} |x_i - y_i| = 1.$$

A mesh $M(2, b)$ is called a **ladder**.

The **boundary** of a 2-mesh $M(a, b)$ is defined as the outer cycle of $M(a, b)$, it has length $2a+2b-4$ and is the cycle through the vertices of degree 2 or 3 in $M(a, b)$. A submesh $M(c, d)$ of the mesh $M(a, b)$ such that $c = a$ or $b = d$ is called a **contraction** of $M(a, b)$ and $M(a, b)$ is said to be contracted to $M(c, d)$.

Metric dimension — метрическая размерность.

Let G be a graph. For a pair of vertices v_1 and v_2 of G, let $d(v_1, v_2)$ denote the length of a shortest path from v_1 to v_2. A vertex set $S \subset V(G)$ is called a **metric basis** of G if for any pair of vertices x, y of G, there exists a vertex $v \in S$ such that $d(x, v) \neq d(y, v)$. The **metric dimension** $\beta(G)$ is the cardinality of a smallest metric basis of G. See also *Decomposition dimension.*

Metric-locating-dominating set — метрически размещённое доминирующее множество.

The concepts of a *locating set* and a *dominating set* merge by defining the **metric-locating-dominating set**, denoted by an MLD-set, in a connected graph G to be a set of vertices of G that is both a dominating set and a locating set in G. We define the **metric-**

location-domination number $\gamma_M(G)$ of G to be the minimum cardinality of an MLD-set in G.
See also *Locating-dominating set.*

Metric-location-domination number — метрически размещённое доминирующее множество.
See *Metric-locating-dominating set.*

Middle graph — серединный граф.
The **middle graph** of a graph G is the graph obtained from G by inserting a new vertex into every edge of G and joining by edges those pairs of these new vertices which lie on adjacent edges of G.
Let us denote the *line graph* of a graph G by $L(G)$. Then, from the definition of the *endline graph* and the middle graph of a graph G, we have $L(G^+) = M(G)$.

MIDS problem — проблема MIDS, проблема минимального независимого доминирования.
See *Minimal independent dominating set problem.*

Minimal connected graph — минимально связный граф.

Minimal dominating graph — минимальный доминирующий граф.
The **minimal dominating graph** of G is the *intersection graph* on the minimal dominating sets of vertices of G.

Minimal flow — минимальный поток.

Minimal imperfect graph — минимальный несовершенный граф.
A graph is called **minimal imperfect graph** if it is not *perfect* but every its proper *induced subgraph* is. The **strong perfect graph conjecture** made by C. Berge states that the only **minimal imperfect graphs** are the chordless odd cycles of length at least five and their *complements*. The chordless odd cycles of length five and their complements are often referred to as the odd *holes* and odd *antiholes*, respectively. Until now, the strong perfect graph conjecture is unsettled.

Minimal irredundance imperfect graph — минимальный неизбыточный несовершенный граф.
See *Irredundance perfect graph.*

Minimal separator — минимальный сепаратор.
See *Separator.*

Minimal triangulation — минимальная триангуляция.
Given a graph G of *treewidth* k, a **triangulation** of G into a triangulated graph H is such that the following three properties hold:

(1) the maximal *clique size* is equal to $k+1$,

(2) if a and b are nonadjacent vertices in H, then every minimal (a,b)-*separator* of H is also a minimal (a,b)-separator in G,

(3) if S is a minimal separator in H and C is the vertex set of a connected component of $H[V \setminus S]$, then C also induces a connected component in $G[V \setminus S]$.

Every graph has a minimal triangulation.

Minimum broadcast graph — минимальный граф широковещания.

See *Broadcast graph.*

Minimum cost hyperflow problem — задача о гиперпотоке минимальной стоимости.

See *Hyperflow.*

MINIMUM FILL-IN problem — проблема MINIMUM FILL-IN.

See *Triangulation of a graph.*

Minimum gossip graph — минимальный граф сплетен.

See *Gossip graph.*

MINIMUM GRAPH COLORING problem — задача о минимальной раскраске графа.

The **MINIMUM GRAPH COLORING problem** (or **MGCP**) consists in finding a coloring with the smallest number of colors. This problem is known to be NP-hard, approximable within factor $\mathcal{O}(|V|(\log\log|V|)^2/(\log|V|)^3)$ and not approximable within factor $|V|^{1-\epsilon}$ for any $\epsilon > 0$.

Minimum independent dominating set problem — задача о минимальном независимом доминирующем множестве.

Given a graph $G=(V,e)$, the **minimum independent dominating set problem** (or **MIDS**) is the problem of finding the smallest possible set $S \subseteq V$ of vertices such that for all $u \in V - S$ there is $v \in S$ for which $(u,v) \in E$, and such that no two vertices in S are joined by an edge in E. Variation in which the degree of G is bounded by a constant B is denoted by MIDS-B.

Minimum separator — минимальный сепаратор.

See *Separator.*

Minimum t-spanner problem — задача нахождения минимального t-стягивателя.

A t-*spanner* is called a minimum t-spanner of a weighted graph G, if it has the minimum total edge weight among all t-spanners of G. The **minimum t-spanner problem** is formulated as follows.

Input: A graph G with associated (positive real valued) edge weights and a positive real value W.

Question: Does G contain a t-spanner with a total edge weight at most W?

MINIMUM VERTEX COVER problem — задача о наименьшем вершинном покрытии.

The **MINIMUM VERTEX COVER problem** (or **MVCP**) consists in finding a cover of the smallest cardinality. This problem is known to be NP-hard, approximable within factor 2 and not approximable within factor 1.1666.

Minor of a graph — минор графа.

A graph H obtained from G by a series of vertex deletions, edge deletions and *contractions of an edge*. A class of graphs $\mathcal{F}$ is called **minor-closed** if for every graph G in $\mathcal{F}$, every **minor** of G is also a member of $\mathcal{F}$. For a class of graphs $\mathcal{F}$, a finite **obstruction set** S is a finite set of minors such that a graph is a member of $\mathcal{F}$ if and only if it does not contain an element of S as a minor.

The graphs in every minor-closed class of graphs are recognizable in $\mathcal{O}(n^3)$ time.

Minor-closed class of graphs — минорно замкнутый класс графов.

See *Minor of a graph.*

Minsky machine — машина Минского.

k-Minus-critical graph — k-минус-критический граф.

See *Induced path number.*

Minus dominating function — функция минус-доминирования.

See *Dominating function.*

Minus domination number — число минус-доминирования.

See *Dominating function.*

Mixed graph — смешанный граф.

A **mixed graph** D consists of a vertex set $V(D)$ and a set of *edges* and *arcs* $E(D)$. A mixed graph without edges is a *digraph.* A mixed graph D is connected, if the *underlying graph* of D is connected.

Mode — **1.** метод, способ, образ действия, форма, вид. **2.** мода.

See *Eccentric sequence.*

Mode vertex — модная вершина.

See *Eccentric sequence.*

Model of computation — модель вычисления.

A **model of computation** is a formal, abstract definition of a

computer. Using a model, one can easily analyze the intrinsic execution time or memory space of an algorithm while ignoring many implementation issues. There are many models of computations which differ in computing power (that is, some models can perform computations impossible for other models) and the cost of various operations. The best-known example of a model of computation is the **Turing machine**. As all our models, a Turing machine also operates in discrete times. It consists of a finite state machine controller, a read-write head, and an unbounded sequential tape. Depending on the current state and symbol read on the current sell of the tape, the machine can change its state, write a symbol in the current sell and move the head to the left or right. Unless otherwise specified, a Turing machine is **deterministic**, i.e. permits at most one next action at any step in a computation.
The input-output format of a Turing machine is specified as follows. The machine begins its computation by scanning the leftmost symbol of a given input word in a specific initial state. The input is accepted iff the computation reaches a specific final state. If the machine is viewed as a translator rather than an acceptor, then the word on the tape, after machine has reached a final state, constitutes the output to the given input. Some of the symbols on the tape might thus be disregarded.
Let T be a **nondeterministic Turing machine**. When scanning a specific symbol in a specific state, T may have several possibilities for its behavior. Otherwise, a nondeterministic Turing machine is defined as a deterministic one. A word α is accepted iff it gives rise to an accepting computation, independently of the fact that it might also give rise to computations leading to failure. Thus, as in connection with nondeterministic machines in general, all roads to failure are disregarded if there is one possible road to success.
The tape of a Turing machine can be viewed both as an input and output channel and as a potentially infinite external memory. The basic difference between Turing machines and other types of automata can be briefly described as follows. A **finite automaton** has only an internal memory determined by its finite state set; the input tape is not used as an additional memory. A finite automaton just reads the input in one sweep from the left to right. In a **linear-bounded automaton**, the external memory is bounded from above

by the size of the input word (or by a linear function of it, which amounts to the same thing). In a **pushdown automaton**, the access to the information in the infinite external memory is very limited and is based on the principle "first in-last out"; a pushdown automaton is a finite automaton combined with a potentially infinite pushdown tape.

Hence, clearly, a Turing machine is more general than the other model of computation we have considered. It is also a **general model**: every algorithm (in the intuitive sense) can be realized as a Turing machine (**Church's thesis**).

Another well-known example of a general model of computation is a **random access machine** (or **RAM**) whose memory consists of an unbounded sequence of registers, each of which may hold an integer. In this model, arithmetic operations are allowed to compute the address of a memory register.

Other names are **Abstract machine** and **Abstract computer**.

Module of a graph — модуль графа.

Monadic Second Order formula — монадическая второго порядка формула.

See *Logic for expressing graph properties*.

Monge graph — граф Монжа.

Monge graph is a complete undirected weighted graph $G = (V, E)$ whose distance matrix $C = (c_{ij})$ has a property that

$$c_{ij} + c_{k,l} \leq c_{il} + c_{kj}$$

for all $1 \leq i < k \leq n$, $1 \leq j < l \leq n$, $i \neq j$, $k \neq l$, $i \neq l$, $k \neq j$. $|V| = n$.

Monochromatic class (set) — одноцветный класс.

Monotone transitive graph — монотонно транзитивный граф.

See *Chordal graph*.

Monotonicity property — свойство монотонности.

Multi-coloring — мультираскраска.

For a weighted undirected simple graph $G = (V, E)$ with n vertices, let the **length** of a vertex v be a positive integer denoted by $x(v)$ and called the **color requirement** of v. A **multi-coloring** of the vertices of G is a mapping into the power set of the positive integers, $\Psi : V \to 2^N$, such that $|\Psi(v)| = x(v)$ and adjacent vertices receive non-intersecting sets of colors. The traditional optimization goal is to minimize the total numbers of colors assigned to G.

Multicrown — мультикорона.

See *Crown of graphs.*

Multidimensional search tree — многомерное дерево сортировки.

Multidimensional search trees (or $K-d$**-trees**) are a generalization of the well known *binary search trees*, that handles records with keys of K attributes. In what follows and without loss of generality, we identify a record with its corresponding key as $x = (x^{(1)}, x^{(2)}, \ldots, x^{(K)})$, where each $x^{(i)}$, $1 \leq i \leq K$, refers to the value of the i-th attribute of the key x.

A **multidimensional search tree** for a set of keys is a *binary tree* in which:

1. Each node contains a K-dimensional key and has an associated discriminant $j \in \{1, 2, \ldots, K\}$.
2. For every node with a key x and discriminant j, any key y in the left subtree satisfies $y^{(j)} < x^{(j)}$ and any key y in the right subtree satisfies $y^{(j)} > x^{(j)}$.
3. The root node has depth 0 and discriminant 1. All nodes at the depth d have the discriminant $(d \pmod K) + 1$.

Note that, if $K = 1$, then **multidimensional search tree** is a binary search tree.

Multidimensional B-tree — многомерное B-дерево.

Multigraph — мультиграф.

A **multigraph** $G = (V, E)$ is a graph in which the edges may occur several times. Edges joining the same pair of vertices are called **multiple edges**.

Multigraph of strength s — мультиграф мощности s.

Multientry zone — многовходовая зона.

Multiple arcs — кратные дуги.

Multiple domination — кратное доминирование.

See *Double domination set.*

Multiple edges — кратные рёбра.

See *Multigraph.*

Multiplicity — кратность.

See *Petri net.*

Multiplicity of a covering — кратность покрытия.

Multiplicity of an edge — кратность ребра.

Multiway tree — многоходовое дерево.

A **multiway tree** of order m $(m \geq 2)$ is a tree such that the

properties P1, P2, P3 and P4 hold.
(P1) Every node, if it is not a leaf, has at most m sons.
(P2) Every node contains at most $m-1$ keys.
Let the generic node contain j keys, $1 \leq j \leq m-1$. The structure of such a node will be represented as:

$$[p_1(k_1, \alpha_1), p_2(k_2, \alpha_2), \ldots, (k_j, \alpha_j)p_{j+1})$$

where p_i is the i-th pointer and the pair (k_i, α_i) is the i-th key (k_i) with associated information (α_i). The following properties hold for every node of a multiway tree:
(P3) $k_1 < k_2 < \ldots < k_j$.
(P4) For each $p_i \neq 0$, letting $P(p_i)$ be the node pointed to p_i, and $K(p_i)$ be the set of keys contained in the subtree of which $P(p_i)$ is a *root*, we have:
(a) $\forall y \in K(p_i) \Rightarrow y < k_i$,
(b) $\forall y \in K(p_i) \Rightarrow k_{i-1} < y < k_i,\ 2 \leq i \leq j$,
(c) $\forall y \in K(p_{j+1}) \Rightarrow y > k_i$.

Mutual matchings — взаимные паросочетания.

Mutually connected vertices — бисвязные (взаимно связные, сильно связные) вершины.

Mutually eccentric vertices — взаимно эксцентричные вершины.
See *Eccentric sequence.*

Mutually graceful trees — взаимно грациозные деревья.
Let T_p and θ_p be two trees with vertices t_i and u_i $(i = 1, 2, \ldots, p)$, respectively; then a labeling f will be called **mutually graceful** if it satisfies the following conditions:

$$\{f(t_i)\} \cup \{f(u_i)\} = \{1, 2, \ldots, 2q\} \text{ for } i = 1, 2, \ldots, q(= p-1); \quad (1)$$

$$f(t_p) = 2q + 1, f(u_p) = 2q + 2; \quad (2)$$

and the vertex labels of each of the two trees — with exception of the highest ones defined by (2) — are at the same time the induced edge labels of the other tree.
Here the "induced edge labels" are defined as usual:

$$|f(x) - f(y)| \text{ for the edge } (x, y).$$

N

Naked vertex — голая вершина.

NCE graph grammar — графовая грамматика типа NCE.

An **NCE graph grammar** (or **neighborhood controlled embedding graph grammar**) is a system $\mathcal{G} = (\Sigma, \Delta, S, P)$, where
(1) Δ and Σ are alphabets with $\Delta \subseteq \Sigma$ as the set of terminal node labels, and $\Sigma - \Delta$ as the set of nonterminal node labels, respectively.
(2) S is a graph over Σ, the axiom of $\mathcal{G}$.
(3) P is a finite set of productions. Each production is a triple (A, R, C), where A is a nonterminal node label from $\Sigma - \Delta$ (the left-hand side), R is a graph over Σ (the right-hand side), and C is an embedding relation for R.

There are the following types of NCE graph grammars: *confluent* (C-NCE), *boundary* (B-NCE), and *linear* (L-NCE).

Near perfect matching — почти совершенное паросочетание.

See *Perfect matching.*

Nearest common ancestor — ближайший общий предок.

See *Directed tree.*

Nearest common dominator — ближайший общий доминатор.

See *Dominator tree.*

Nearly regular graph — почти однородный граф.

Neighbour transition — сосед-переход.

Neighbourhood matrix — матрица соседства, матрица смежности.

The *Adjacency matrix.*

Neighbourhood tree — дерево соседства, H-дерево.

Neighbourhood of a vertex — окрестность вершины.

For each vertex v the set $N(v)$ of vertices which are *adjacent* to v. The other name is **open neighbourhood**. The **closed neighbourhood** is $N[v] = N(v) \cup \{v\}$.

For disjoint subsets A and B of V, we define $[A, B]$ to be the set of all edges that join a vertex of A and a vertex of B. Furthermore, for $a \in A$, we define the **private neighbourhood** $pn(a, A, B)$ of a in B to be the set of vertices in B that are adjacent to a but to no other vertex of A; that is, $pn(a, A, B) = \{b \in B | N(b) \cap A = \{a\}\}$.

Given a digraph D, let x, y be distinct vertices in D. If there is an arc from x to y, then we say that x dominates y and write $x \to y$ and call y (respectively, x) an **out-neighbour (out-neighborhood)** (respectively, an **in-neighbour (in-neighborhood)**) of x (respectively, y). We let $N^+(x), N^-(x)$ denote the set of out-neighbours, respectively, the set of in-neighbours of x in D. Define $N(x)$ to be $N(x) = N^+(x) \cup N^-(x)$.

D is an **out-semicomplete digraph** (**in-semicomplete digraph**) if D has no pair of non-adjacent vertices with a common in-neighbour or a common out-neighbour. D is a **locally semicomplete digraph** if D is both out-semicomplete and in-semicomplete.

***k*-th Neighborhood of a vertex** — окрестность вершины k-го порядка.

The ***k*-th neighborhood of a vertex** v of G is the set of all vertices of *distance* k to v, i.e.

$$N^k(v) = \{u \in V : \ d_G(u, v) = k\}.$$

See also *Disc*.

Neighbouring vertices — соседние вершины.

Nested set of alts — иерархия вложенных альтов.

See *Alt*.

Nested set of zones — иерархия вложенных зон.

A set of zones A of a cf-graph G forms a **nested set of zones** of G if the following two properties hold: $S_1 \cap S_2 = \varnothing$ or $S_1 \cap S_2$ for any $S_1, S_2 \in A$; and for any zone S_1 of G there is such a zone $S_2 \in A$ that $S_1 \subseteq S_2$ and the zones S_1 and S_2 have a common initial node.

Net — сеть.

Net formula — формула сети.

Network — сеть.

Node — узел, вершина.

The same as *Vertex*.

***N*-node** — N-вершина.

See *T-numbering*.

Node bisector — вершинный бисектор.

Node listing — укладка уграфа.

Noncovered vertex — свободная вершина.

Nondecidable problem — (алгоритмически) неразрешимая проблема.

See *Decision problem*.

Nondeterministic finite automaton — недетерминированный конечный автомат.

See *Model of computation*.

Nondeterministic pushdown automaton — недетерминированный автомат с магазинной памятью.

See *Model of computation.*

Nondeterministic Turing machine — недетерминированная машина Тьюринга.

See *Model of computation.*

Nonstrong argument — необязательный аргумент (оператора).

See *Large-block schema.*

Nonstrong input — необязательный вход (оператора).

See *Large-block schema.*

Nonstrong output — необязательный выход (оператора).

See *Large-block schema.*

Nonstrong result — необязательный результат (оператора).

See *Large-block schema.*

Nonterminal alphabet — нетерминальный алфавит, алфавит нетерминальных символов, алфавит нетерминалов.

See *Grammar.*

Nonterminal symbol — нетерминальный символ.

See *Grammar.*

Non-circular grammar — ациклическая атрибутная грамматика.

Non-edge — неребро, отсутствие ребра.

This is a pair of nonadjacent vertices.

Non-interpreted schemata — неинтерпретированная схема.

Non-separable graph — неразделимый граф, неразложимый граф, несепарабельный граф.

Normal approximate (point) spectrum — нормально аппроксимирующий (точечно) спектр.

See *Spectrum.*

Normally symmetric graph — нормально симметричный граф.

A graph $G = (V, A)$ is called **normally symmetric graph** if the number of all *common servers* $d^+(u, v)$ is equal to the number of all *common receivers* $d^-(u, v)$ for any $u, v \in V$.

Normed weighted graph — нормированно-взвешенный граф.

See *Weighted graph.1.*

Nowhere-zero k-flow — нигде не нулевой k-поток, везде ненулевой k-поток.

A graph admits a **nowhere-zero k-flow** (k is an integer ≥ 2) if its edges can be oriented and labeled by numbers from $\{\pm 1, \ldots, \pm(k -$

1)} so that for every vertex the sum of the incoming values equals the sum of the outcoming ones. A graph without **nowhere-zero k-flow** is called **k-snark**. Note that if a graph is a k-snarks then it is a k'-snark for any integer $2 \leq k' \leq k$. Very famous is the **5-flow conjecture** of W.T.Tutte which says that there are no bridgeless 5-snarks.

Null graph — нуль-граф.

This is a graph with no vertices.

k-Null graph — k-нуль граф.

See *Clique graph.*

Number of noncongruence of a numbering — число несоответствия нумерации.

See *Numbering of cf-graph.*

Numbering — нумерация (вершин графа).

A bijection $f : V \rightarrow \{1, 2, \ldots, n\}$ is called a **numbering** of the vertices of G. Then $f(v)$ is referred to as the **number** associated with the vertex v, or simply the number of v with respect to the numbering f.

K-Numbering — K-нумерация.

L-Numbering — L-нумерация.

M-Numbering — M-нумерация.

See *Basic numberings.*

N-Numbering — N-нумерация.

See *Basic numberings.*

T-Numbering — T-нумерация.

Given a cf-graph G and its inverse numbering N, a node P is called a **binode** (or **N-node**) if $p \notin N < i >$ for all $i < N(p)$.

A **T-numbering** is such a numbering of G that the following two properties hold:

(1) $N < p >= T[T(p), T < p > + |N < p > | - 1]$ for any N-node p,

(2) for any two N-nodes p and q, $T(p) < T(q)$ if and only if $T(p) < T(q)$.

A fragment S of G is its strongly connected component if and only if $S = N < p >$ for an N-node p.

An N-node r is called the **cutpoint** of G if there is no arc (p, q) such that $T(p) < T(r) < T(q)$.

A fragment H of G is its linear component with the initial node p and the terminal node q if and only if, for some T-numbering of

G, the nodes p and q are cutnodes, $H = T[T(p), T(q) - 1]$, and $T[T(p)+1, T(q)-1]$ contains no cutnodes.

Numbering of cf-graph — нумерация уграфа.

Let G be a *cf-graph* with a set of nodes X, where $n = |X|$. A bijection $F:\ X \to [1, n]$ is called a **numbering** of G. $F(p)$ is called F-number of the node p, and $F^{-1}(k)$ denotes the node p having F-number k. By $F[i, j]$ we denote a set of nodes $\{p \in X : F(p) \in [i, j]\}$ and the subgraph of G induced by the set.

An arc $u = (p, q)$ is called an F**-direct arc** (or F**-arc**) if $F(p) < F(q)$ and an F**-inverse arc** if $F(p) \geq F(q)$.

The **depth** (or the **number of noncongruence**) of the numbering F of the graph G is defined as the greatest number of F-inverse arcs that belongs to a simple path in G.

If a path P from a node p to a node q does not contain any F-inverse arcs, then it is called an F**-path** and the node q is called F**-reachable** from p.

Let p be a node of G with F-number i, i.e. $F(p) = i$. A subgraph of G that consists of all those nodes from which p is reachable in $F[i, n]$ is called F**-region** and denoted by $F < p >$ or $F < i >$.

O

Oberwolfach problem — проблема Обервольфаха.

The problem of determining whether there exists an $(m_1, m_2, \ldots, m_t)$-2-factorization of K_n when n is odd, or $K_n - F$ when n is even, is the **Oberwolfach problem**, denoted $OP(m_1, m_2, \ldots, m_t)$.

The Oberwolfach problem was formulated by Ringle in 1967.

Oblique graph — скошенный граф.

A k-gon α of a *polyhedral graph* $G = (V, E, F)$ with the face set F is of **type** $\langle b_1, \ldots, b_k \rangle$ if the vertices incident with α in a cyclic order have degrees $b_1, \ldots, b_k$ and $\langle b_1, \ldots, b_k \rangle$ is the lexicographic minimum of all such sequences available for α. A polyhedral graph G is **oblique** if it has no two faces of the same type.

G is **superoblique** if both G and its *dual* G^* are oblique and they have no common face type. Let z be any given natural number. A polyhedral graph G is **z-oblique** if $F(G)$ contains at most z faces of the same type for any type of faces. Obviously, a 1-oblique graph is oblique and vice versa.

z-Oblique graph — z-скошенный граф.

See *Oblique graph.*

Obstruction set — препятствующее множество.

See *Minor of a graph.*

Occurence (of a graph H in G) — вхождение (графа H в граф G).

See *Labeled graph.*

Occurrence process net — параллельная сеть-процесс.

ODC — ортогональное двойное покрытие.

See *Orthogonal double cover.*

Odd component — нечетная компонента.

A component of G is called **odd** or **even** according to its *order* is odd or even.

Odd component number — число нечётных компонент.

See *Component of a graph.*

Odd graph — нечётный граф.

Odd-signable graph — нечётно-знаковый граф.

See *Signed labeled graph.*

Odd-signed graph — нечётно-знаковый граф.

See *Signed labeled graph.*

One-chromatic number — число один-хроматическое.

One-way infinite path — одно-лучевой бесконечный путь.
See *Ray.*

One-way infinite sequence — односторонне-бесконечный маршрут.

One-way pushdown automaton — односторонний магазинный автомат.

One-sided balanced tree — одностороннее балансированное дерево.
See *Height balanced tree.*

Open neighbourhood — открытая окрестность (вершины).
See *Neighbourhood.*

Open sequence — открытый маршрут.

Operation — операция.

Operation of a Petri net — функционирование сети Петри.
See *Petri net.*

Operation of formation of a set of merged places — операция формирования мест.

Operation of merging of places — операция слияния мест.

Operator — оператор.

Optimal 1-edge hamiltonian graph — оптимальный 1-рёберный гамильтонов граф.
See *1-hamiltonian graph.*

Optimal 1-hamiltonian graph — оптимальный 1-гамильтонов граф.
See *1-hamiltonian graph.*

Optimal 1-node hamiltonian graph — оптимальный 1-вершинный гамильтонов граф.
See *1-hamiltonian graph.*

Optimal numbering — оптимальная нумерация.

Optimal ordering for trees — оптимальное упорядочение деревьев.

Order of an automorphism group — порядок группы графа, число симметрии графа.

Order of a graph — порядок графа.
The **order of a graph** G is the number of vertices in G.

Order of a hypergraph — порядок гиперграфа.

Order of a tree — порядок дерева.
Given a tree, its **order** is the number of vertices in the tree.

Order relation — отношение упорядочения (порядка).

Ordered chromatic number — упорядоченное хроматическое число.

Ordered coloring of vertices — упорядоченная раскраска вершин.

Ordered edge chromatic number — упорядоченное реберное хроматическое число.

Ordered graph — упорядоченный граф.

Ordered labelled tree — упорядоченное помеченное дерево.
See *Labeled tree*.

Ordered tree — упорядоченное дерево.
An **ordered tree** is a rooted tree in which the order of the subtrees is significant. There is a one-to-one correspondence between ordered forests with n nodes and *binary trees* with n nodes.

k**-Ordered Hamiltonian graph** — k-упорядоченый гамильтонов граф.
See *Hamiltonian graph*.

Ordinary Petri net — ординарная сеть Петри.
See *Petri net*.

Orientation distance graph — граф расстояний ориентаций.
The **orientation distance graph** $\mathcal{D}_o(G)$ of a graph $G = (V, E)$ has a vertex set $\mathcal{O}(G)$, the collection of pair-wise nonisomorphic *orientations* of G. Adjacency is defined between two orientations iff the reversal of one arc in one orientation generates (an orientation isomorphic to) the other.

Orientation number — число ориентации.
See *Orientation of a graph*.

Orientation of a graph — ориентация графа.
Let $G = (V, E)$ be a finite undirected graph. Then $G' = (V, E')$ is an **orientation of a graph** G if for all $(x, y) \in E$ E' contains the arc (x, y) or (y, x). G' is a **transitive orientation** of G if E' is transitive as a *binary relation* on V. An **acyclic orientation** of a digraph $G = (V, E)$ is an acyclic digraph $\vec{G} = (v, \vec{E})$ such that $\vec{E} \subseteq E$.
An orientation D of G is **strong** if any pair of vertices in D are mutually reachable in D. Given a *2-edge-connected graph*, let $\mathcal{D}(G)$ be the set of all strong orientations of G. The **orientation number** of G is defined to be $\vec{d}(G) = \min\{d(D) \mid D \in \mathcal{D}(G)\}$. The problem of evaluating the orientation number of an arbitrary connected graph is very difficult.

Oriented edge — ориентированное ребро.
The same as *Arc*.

Oriented graph — ориентированный граф.
A digraph G is called an **oriented graph** if G does not contain a

cycle of two arcs. A complete oriented graph is called a **tournament**.

Oriented tree — ориентированное дерево.

See *Oriented Graph*, *Rooted Tree*.

Orthogonal double cover — ортогональное двойное покрытие.

An **orthogonal double cover** of a complete graph K by a graph G is a set of subgraphs of K each isomorphic to G, such that every edge of K is contained in exactly two subgraphs and each two subgraphs have exactly one edge in common.

See also *Suborthogonal double covers*.

Orthogonal (g, f)-factorization — ортогональная (g, f)-факторизация.

See *k-Factor of a graph*.

$\mathcal{F}$-Orthogonal subgraph — $\mathcal{F}$-ортогональный подграф.

Let be $\mathcal{F} = \{F_1, \ldots, F_t\}$ is *1-factorization* of G. A subgraph H of G is **suborthogonal** to $\mathcal{F}$ if $|E(H) \cap E(F_i)| \leq 1$ for $1 \leq i \leq t$, and **orthogonal** if $|E(H) \cap E(F_i)| = 1$ for $1 \leq i \leq t$.

Oscillation of a graph — осцилляция графа.

An **edge-ordering** of the finite simple graph $G = (V, E)$ is 1-1 function f from E to the set of positive integers. The set of all edge-orderings of G is denoted by $\mathcal{F}$. For $f \in \mathcal{F}$, a path with the edge sequence $e_1, e_2, \ldots, e_t$ is called an $f - zpath$ if for each $i = 1, \ldots, t-2$

$$f(e_i) - f(e_{i+1}) > 0 \text{ if and only if } f(e_{i+1}) - F(e_{i+2}) > 0.$$

The **vibration** $k(f)$ of f is the maximum length of an $f - zpath$ and the **oscillation** $\eta(G)$ of G is defined by

$$\eta(G) = \min_{f \in} k(f).$$

Observe that $\eta(G)$ is the greatest integer t such that G has $f - zpath$ for each $f \in \mathcal{F}$.

Outcenter — внешний центр.

Outcoming arc — исходящая дуга.

Outdegree, out-degree — полустепень исхода вершины.

The **outdegree** of the vertex v in a digraph G is the number of distinct arcs with the *source* v and it is denoted by $out(v, G)$.

Outdegree matrix — матрица полустепеней исхода.

Outerplanar graph — внешнепланарный граф.

A graph G is **outerplanar** if there is a crossing-free embedding of G in the plane such that all vertices are on the same *face*. G is outerplanar iff G contains no subgraph homeomorphic to K_4 or $K_{2,3}$ by a homeomorphism that deletes degree-2 vertices but does not add them.

G is k**-outerplanar** if for $k = 1$ G is an **outerplanar graph** and for $k > 1$ G has a planar embedding such that if all vertices on the *exterior face* are deleted, the connected components of the remaining graph are all $(k-1)$-outerplanar. See also *Halin graph*.

k**-Outerplanar graph** — k-внешнепланарный графа.

See *Outerplanar graph*.

Outerplane graph — внешнеплоский граф.

An **outerplane graph** is a particular embedding of an *outerplanar graph*.

Out-neighbour — исходящий сосед.

See *Neighborhood of a vertex*.

Out-neighbourhood — исходящая окрестность.

See *Neighbourhood of a vertex*.

Outpath — выходящий путь.

An **outpath** of a vertex x (an arc (x, y), respectively) in a digraph is a path starting at x ((x, y), respectively) such that x dominates the endvertex of a path only if the endvertex also dominates x. An outpath of length k is called a k**-outpath**.

k**-Outpath** — выходящий k-путь.

See *Outpath*.

Output — выход.

1. See *Control flow graph*.

2. See *Fragment*.

Output dependence — выходная зависимость, зависимость по выходу.

See *Data dependence*.

Output directed spanning tree — выходящий оркаркас.

Output node of fragment — выходная вершина фрагмента.

See *Fragment*.

Output place — выходное место.

Output tree — выходящее дерево.

See *Out-tree*.

Output vertex of subgraph — выходная вершина подграфа.

Outradius — внешний радиус.

Outseparation number — число внешнего разделения.

Outset — выходящее множество.

The **outset** $N^+(x)$ of a vertex x is the set of vertices dominated by x.

Out-semicomplete digraph — выходящий полуполный орграф.

See *Neighborhood of a vertex.*

Out-tree — выходящее ордерево.

An **out-tree** is a directed tree in which precisely one vertex has zero *in-degree.*

P

Pack of a graph — колода графа.

Packing of graphs — упаковка графов.

See *Embedding of a graph.*

2-Packing of a graph — 2-упаковка графа.

A subset A of G is called a **2-packing** of G if the closed *neighborhoods* of any two distinct vertices of A are disjoint. The **2-packing number** of G is the maximum cardinality, $\rho(G)$, of 2-packing of G.

2-Packing number — 2-упаковочное число.

See *2-Packing of a graph.*

Pair of connectivities — пара связностей.

Paired-domination number — число парно-доминирования.

See *Paired-dominating set.*

Paired-dominating set — парно-доминирующее множество.

A paired-dominating set S with a matching M is a dominating set $S = \{v_1, v_2, \ldots, v_{2t-1}, v_{2t}\}$ with an independent edge set $M = \{e_1, \ldots, e_t\}$, where each edge e_i joins two elements of S, that is, M is a *perfect matching* (not necessarily induced) in the subgraph $\langle S \rangle$ induced by S. A set S is called a **paired-dominating set** if it dominates V and $\langle S \rangle$ contains at least one perfect matching. The **paired-domination number** $\gamma_p(G)$ is the minimum cardinality of a paired-dominating set S in G.

k**-Pan** — k-сковорода.

A k**-pan** is a graph consisting of a cycle C_k of length k and an edge outside.

Pan-bicentral graph — пан-бицентральный граф.

See *Pancentral graph.*

Pancentral graph — панцентральный граф.

A graph G is called **pan-unicentral** if, given a vertex v in G, there exists a spanning tree T such that $C(T) = \{v\}$, where $C(T)$ is the *center* of T. G is called **pan-bicentral** if, given adjacent vertices u and v in G, there exists a spanning tree T such that $C(T) = \{u, v\}$. A graph G with both properties is called **pancentral**.

(a, b)**-Panconnected graph** — (a, b)-пансвязный граф.

Let a, b be integers and $a \leq i \leq b$. G is called (a, b)**-panconnected**, if there exists a path $P_i[u, v]$ between each pair of distinct vertices u, v of G.

Pancyclic graph — панциклический граф.

A graph G on n vertices is said to be a **pancyclic graph** if it contains cycles on every length from 3 to n.

A graph G of order n is said to be $[a, b]$**-pancyclic**, if for every integer i ($a \le i \le b$) there exists a cycle C_i of length i in G. Similarly, G is said to be $[a, b]$**-vertex-pancyclic** (resp. $[a, b]$**-edge-pancyclic**), if for every vertex v (resp. edge e) and every i there is a cycle C_i containing v (resp. e). G is said to be $[a, b]$**-panconnected**, if for every pair of distinct vertices u, v and every i there exists a path $P_i[u, v]$ of i vertices connecting u and v.

Obviously, if G is $[a, b]$-panconnected, then G is $[a, b]$-edge-pancyclic; if G is $[a, b]$-edge-pancyclic, then G is $[a, b]$-vertex-pancyclic and if G is $[a, b]$-vertex-pancyclic, then G is $[a, b]$-pancyclic.

See also *Uniquely pancyclic graph*, *Weakly pancyclic graph*.

j**-Pancyclic graph** — j-панциклический граф.

Panpropositionable Hamiltonian graph — панпропозицируемый гамильтонов граф.

A Hamiltonian graph G is **panpropositionable** if for any two different vertices x and y of G and any integer k with $d_G(x, y) \le k < |V(G)|/2$, there exists a Hamiltonian cycle C of G with $d_G(x, y) = k$.

Pan-unicentral graph — пан-уницентральный граф.

See *Pancentral graph*.

Parallel Random Access Machine (PRAM) — параллельная равнодоступная адресная машина (ПРАМ).

A **Parallel Random Access Machine** (or **PRAM**) is an abstract model of parallel computation which can be used by parallel algorithms designers to estimate the inherent parellelism of a given problem. PRAM neglects such issues as synchronization and communication, but provides any (problem size-dependent) number of processors.

An (n, m)**-PRAM** consists of n processors running synchronously and m memory locations, where each processor is a random-access machine. All processors share the memory, and hence are commutative via it. During a given cycle each processor may read an element from the shared memory into its local memory, write an element from its local memory to the shared memory, or perform any RAM operation on the data which it already has in its local memory. It is a synchronous model, that is no processor will proceed with instruction

$i + 1$ until all have finished instruction i.
The read/write conflicts in accessing the same shared memory location simultaneously can be resolved by different strategies. There is a family of PRAM models, each of which differs in its characteristics on this point. The members of the family are:
(1) the **Exclusive Read Exclusive Write PRAM** (or **EREW PRAM**), where every memory location can be read or written to by only one processor at a time,
(2) the **Concurrent Read Exclusive Write PRAM** (or **CREW PRAM**), where multiple processors may read a particular memory location, but at most one processor may write to a particular memory location at a time,
(3) the **Concurrent Read Concurrent Write PRAM** (or **CRCW PRAM**), where multiple processors may read or write to any memory location.
The Exclusive Read Concurrent Write PRAMs are not considered, since a machine with enough power to support concurrent writes should be able to support concurrent reads.
The read causes no discrepancies while the concurrent write is further defined as follows:
(1) the **Common CRCW PRAM**, where all values written concurrently must be identical,
(2) the **Arbitrary CRCW PRAM**, where the processor that succeeds in its concurrent write is chosen arbitrary from the writing processors,
(3)the **Priority CRCW PRAM**, where the processor that succeeds in its concurrent write is the processor with the highest priority, e.g., the smallest processor index,
(4) the **Combining CRCW PRAM**, where the value written is a linear combination of all values which where concurrently written, e.g. a sum of the values. The values may be combined with any associative and commutative operation which is computable in constant time on a RAM.

Parikh mapping — отображение Парика.

Parse tree — синтаксическое дерево.

The same as *Derivation tree*.

Partial edge — частичное ребро.

A **partial edge** of a hypergraph $\mathcal{H}$ is any nonempty subset of some edge of $\mathcal{H}$. If (u, v) is a partial edge of $\mathcal{H}$, then the vertices u and v are said to be **adjacent** in $\mathcal{H}$.

A partial edge of $\mathcal{H}$ which is a *separator* is called a **partial-edge separator** of $\mathcal{H}$.

Partial-edge separator — частично-рёберный сепаратор.

See *Partial edge*.

Partial graph morphism — частичный морфизм графов.

Given two graphs G and H with colors in L, *a pair of partial mappings* $h = (h_V : G_V \to H_V, h_E : G_E \to H_E)$ is called **partial graph morphism** if

(1) whenever h_E is defined for $e \in G_E$, h_V is defined for $s_G(e)$ and $t_G(e)$, and $h_V \circ s_G(e) = s_H \circ h_E(e)$ and $h_V \circ t_G(e) = t_H \circ h_E(e)$;

(2) whenever h_V, respectively h_E, is defined for o, $vl_G(o) = vl_H \circ h_V(o)$, respectively $el_G(o) = el_H \circ h_E(o)$.

Here two mappings $s, t : E \to V$ provide the source and target vertices for each edge, and two mappings $vl : V \to L_V$, respectively $el : E \to L_E$, attach a color to every vertex, respectively edge.

Partial hypergraph — частичный гиперграф.

For a given hypergraph $\mathcal{H}$, a hypergraph $\mathcal{H}'$ with $V(\mathcal{H}') \subseteq V(\mathcal{H})$, $E(\mathcal{H}') \subseteq E(\mathcal{H})$.

See also *Hypergraph, Subhypergraph*.

Partial order relation — отношение частичного упорядочения (порядка).

The *binary relation* R is a **partial order relation** (or simply a **partial order**) on V, if R is a *reflexive, transitive and antisymmetric* relation. Partial orders are often denoted by $\leq$ instead of R:

$$x \leq y \text{ if } (x, y) \in R, \text{ and } x < y \text{ and } x \neq y.$$

$P = (V, \leq)$ is then called a **poset** (partially ordered set). A poset $(V, \leq)$ is finite if V is finite.

A poset $P = (V, \leq)$ is a **linear order** if for all $u, v \in V$ $u \leq v$ or $v \leq u$ holds.

Two posets $(V_1, \leq_1)$, $(V_2, \leq_2)$ are **isomorphic** (denoted $(V_1, \leq_1) \cong (V_2, \leq_2)$) if there is a bijective function f from V_1 onto V_2 such that $u \leq_1 v$ iff $f(u) \leq_2 f(v)$.

Partial signed domination number — частично знаковое число доминирования.

Let $G = (V, E)$ be a simple graph. For any real valued function

$f : V \to R$ and $S \subseteq V$, let $f(S) = \sum_{v \in S} f(v)$. Let c, d be positive integers such that $gcd(c, d) = 1$ and $0 < \frac{c}{d} \leq 1$. A $\frac{c}{d}$**-dominating function (partial signed dominating function)** is a function $f : V \to \{-1, 1\}$ such that $f(N[v]) \geq 1$ for at least $\frac{c}{d}$ of the vertices $v \in V$. The $\frac{c}{d}$**-domination number (partial signed domination number)** of G is

$$\gamma_{\frac{c}{d}}(G) = \min\{f(V) | f \text{ is a } \frac{c}{d}\text{-dominating function on } G\}.$$

Partial k-tree — частичное k-дерево.

A **partial k-tree** is a subgraph of a *k-tree*. The class of partial k-trees is exactly the class of graphs with a *treewidth* at most k. Note that, for each constant k, the class of partial k-trees is *minor-closed.*

Partially decidable problem — частично разрешимая задача.

See *Decision problem.*

Partially ordered set — частично упорядоченное множество.

See *Partial order relation.*

Partially square graph — частично квадратный граф.

Given a graph G, its **partially square graph** G^* is the graph obtained by adding an edge (u, v) for each pair of vertices of G at distance 2, whenever the vertices u and v have a common neighbor x satisfying the condition

$$N_G(x) \subseteq N_G[u] \cup N_G[v]$$

where N(x) is an *open neighborhood* and N[x] is a *closed neighborhood* of a vertex x. In the case where G is a *claw-free* graph, G^* is equal to G^2.

k-Partite graph — k-дольный граф.

Partition of a graph — разбиение графа.

Partition of a set — разбиение множества.

A **partition** of a nonempty set S is a collection of pairwise disjoint nonempty subsets, whose union is S. If two partitions $\{A_i\}$ and $\{B_j\}$ of the same set are such that each A_i is a subset of some B_j, then we say that the partition $\{A_i\}$ is finer than the partition B_j, and that $\{B_j\}$ is coarser than $\{A_i\}$.

Partitioning problem — задача о разбиении.

Passive state of compound transition — пассивное состояние составного перехода.

Path — путь.

1. Given a digraph $G = (V, A)$, a **path** is a sequence of vertices $(v_0, \dots, v_k)$ such that $(v_i, v_{i+1}) \in A$ for $i = 0, \dots, k-1$; its **length** is k. The **path** is **simple** if all its vertices are pairwise distinct. A **path** $(v_0, \dots, v_s)$ is a **cycle** if $s > 1$ and $v_0 = v_s$, and a **simple cycle** if in addition $v_1, \dots, v_{s-1}$ are pairwise distinct.

2. Given a hypergraph $\mathcal{H}$, a **path** from a vertex u to a vertex v is a sequence of edges $(e_1, \dots, e_k)$, $k \geq 1$, such that $u \in e_1,\ v \in e_k$ and , if $k > 1,\ e_h \cap e_{h+1} \neq emptyset$ for $h = 1, \dots, k-1$; furthermore, we say that this path passes through a subset X of $V(\mathcal{H})$, if $e_h \cap e_{h+1}$ is a subset of X for some $h < k$.

F**-Path** — F-путь.

See *Numbering of cf-grahp.*

H**-path** — H-путь.

See *H-distance.*

Path coloring — путевая раскраска.

A coloring such that a subset V_i induces a subgraph whose connected components are paths is called a **path coloring**.

Path covering — путевое покрытие.

Path-decomposition — путевая декомпозиция, разбиение на пути.

This is a *tree-decomposition* (S, T) such that T is a *path.*

Path-Hamiltonian edge — гамильтоново-путевое ребро.

An edge e in G is called **path-Hamiltonian** if there is a *Hamiltonian path* in G that contains e.

k**-Path graph** — граф k-путей.

The k**-path graph** $\mathcal{P}_k(H)$ of a graph H has all length-k paths of H as vertices; two such vertices are adjacent in the new graph if their union forms a path or cycle of length $k+1$ in H and if the edge-intersection of both paths forms a path of length $k-1$. It is known that, given a graph $G = (V, E)$, there is an $\mathcal{O}(|V|^4)$-time algorithm that decides whether there is some graph H of minimum degree at least $k+1$ with $G = \mathcal{P}_k(H)$.

Path layer matrix — матрица путевых слоёв.

The path layer matrix was introduced for simple graphs with the standard metric. Denote by $p(G)$ the order of a graph G (the number of vertices). The **path layer matrix** of a graph G is the matrix

$$\tau(G) = [\tau_{ij}],\ i = 1, 2, \dots, p(G),\ j = 1, 2, \dots, p(G) - 1,$$

where τ_{ij} is the number of paths with the initial vertex v_i that have the length j. By ordering the rows of $\tau(G)$, first by decreasing the length (the number of the last nonzero element) and then with rows of the same length arranged lexicographically, one obtains a canonical form for $\tau(G)$.

Let G be a weighted graph and let $l_1, \ldots, l_n$ be the possible lengths of paths in G. The **path layer matrix** of a weighted graph G is the matrix $\tau w(G) = [\tau w_{ij}],\ i = 1, 2, \ldots, p(G),\ j = 1, 2, \ldots, n$, where τw_{ij} is the number of paths with the initial vertex v_j that have the length l_j.

Path pile — путевая куча.

A set of nontrivial paths in a graph G is called a **path pile** of G, if every edge is on exactly one path and the paths are internally disjoint.

The least number which is the cardinality of a path pile of G is called the **path pile number** $\eta(G)$ of G.

Path pile number — число путевой кучи.

See *Path pile.*

Pathwidth of a graph — путевая ширина графа.

The minimum value k for which the graph is a *partial k-path*. The **pathwidth of a graph** G equals the minimum width over all *path-decompositions* of G.

Pebbling number — фишечное число.

The **pebbling number** of a graph G, $f(G)$, is the least m such that, however m are placed on the vertices of G, we can move a pebble to any vertex by a sequence of moves, each move taking two pebbles off one vertex and placing one on an adjacent vertex.

We say a graph satisfies the **2-pebbling property**, if two pebbles can be moved to any specified vertex, when the total starting number of pebbles is $2f(G) - q + 1$, where q is the number of vertices with at least one pebble.

A graph G without the 2-pebbling property is called a **Lemke graph**.

2-Pebbling property — свойстов 2-фишечности.

See *Pebbling number.*

Pendant edge — висячее ребро.

Pendant vertex — висячая вершина.

A **pendant vertex** is a vertex with degree 1 (in an unoriented graph) or with *in-degree* 1 and *out-degree* 0 (in a directed graph).

Peninsula — полуостров.

t**-Perfect code** — t-совершенный код.

See *t-Code (in a graph).*

Perfect elimination graph — граф совершенного исключения.

See *Chordal graph, Perfect elimination scheme.*

Perfect elimination scheme — совершенная схема удаления.

Let $G = (V, E)$ be a graph. A simplicial vertex of G is a vertex of which the *neighborhood* induces a *clique*. An ordering of the vertices $\sigma = (v_1, \ldots, v_n)$ is called a **perfect elimination scheme** if for every $1 \leq i \leq n$, v_i is a simplicial vertex in $G[v_i, \ldots, v_n]$.

A graph G is a **perfect elimination graph** (or *chordal graph*) if and only if there exists a **perfect elimination scheme** for G.

Perfect fractional matching — совершенное дробное паросочетание.

Let us associate a variable x_{ij} with each edge (i, j) of a graph $G = (V, E)$. A **perfect fractional matching** of G is a vector $\vec{x} \in \Re^{|E|}$, where $\Re$ is the set of real numbers, such that:

$$\sum_{j \in N(i)} x_{ij} = 1, \text{ for all } i = 1, \ldots, |V|,$$

$$x_{ij} \geq 0, \text{ for all } (i, j) \in E.$$

It is not difficult to see that G admits a perfect fractional matching if and only if G can be covered by pairwise disjoint edges and odd cycles.

Perfect graph — совершенный граф.

A graph $G = (V, E)$ is called a **perfect graph** if the following two conditions are both satisfied: first, the *clique number* and the *chromatic number* must be equal for all induced subgraphs, (i.e. $\omega(G[A]) = \chi(G[A])$ for all $A \subseteq V$), and second, the *stability number* must equal the *clique cover number* for all induced subgraphs of G (i.e., $\alpha(G[A]) = k(G[A])$ for all $A \subseteq V$). Notice that the two conditions are dual in the sense that a graph satisfies the first condition if and only if its *complement* satisfies the second. The remarkable fact that a graph satisfies the first equality if and only if it satisfies the second equality was conjectured by C.Berge and proven by L.Lovasz. This is known as the **perfect graph theorem**.

A **near perfect matching** in a graph G is a matching saturating all but one vertex in G.

See also *Minimal imperfect graph.*

Perfect graph theorem — теорема о совершенных графах.

See *Perfect graph.*

Perfect matching — совершенное паросочетание.

See *Matching.*

Perfect k-matching — совершенное k-паросочетание.

See *k-Matching.*

Perfect one-factorization — совершенная один-факторизация.

See *One-factorization.*

Perfect sequence — совершенный маршрут.

Perfectly contractile graph — совершенно стягиваемый граф.

See *Contraction of an even pair.*

Period — период.

See *Primitive directed graph.*2.

Periodicity of a graph — периодичность графа.

Let ϕ be a graph operator defined on the class C_f of all finite undirected graphs. For every positive integer r we define the power ϕ^r so that $\phi^1 = \phi$ and for $r \geq 2$ the operator ϕ^r is such that $\phi(\phi^{r-1}(G))$ for each $G \in C_f$. A graph $G \in C_f$ is called ϕ-periodic, if there exists a positive integer r such that $\phi^r(G) \cong G$. The minimum number r with this property is the **periodicity** of the graph G in the operator ϕ.

Peripheral vertex — периферийная вершина.

See *Periphery.*

q-Peripheral vertex — q-периферийная вершина.

Periphery — периферия.

The **periphery** $P(G)$ is a set of vertices of maximum *eccentricity*, $e(v) = diam(G)$, and those vertices are called **peripheral**.

Permutation graph — перестановочный граф, граф перестановки

If π is a permutation of the numbers $1, \ldots, n$, we can construct an undirected graph $G[\pi] = (V, E)$ with a vertex set $V = \{1, \ldots, n\}$ and an edge set E:

$$(i, j) \in E \Leftrightarrow (i - j)(\pi_i^{-1} - \pi_j^{-1}) < 0.$$

An undirected graph is called a **permutation graph** if there exists a permutation π such that $G \cong G[\pi]$. It is known that the *complement* of a **permutation graph** is also a **permutation graph** and a **permutation graph** is a *comparability graph.*

α-Permutation graph — α-перестановочный граф.

Persistence problem — проблема устойчивости.

Persistent Petri net — устойчивая сеть Петри.

Persistent transition — устойчивый переход.

Petal of a flower — лепесток цветка (граф).

See *Flower.*

Petersen graph — граф Петерсена.

A **generalized Petersen graph** $P(n, m)$, $1 \leq m \leq \frac{n}{2}$, consists of an outer n-cycle $y_1, Y_2, \ldots, y_n$, a set of n spokes $y_i x_i$, $1 \leq i \leq n$, and n inner edges $x_i x_{i+m}$, $1 \leq i \leq n$, with indices taken modulo n.

The standard **Petersen graph** is the instance $P(5, 2)$. It is possible to form the **Petersen graph** by constructing a vertex for each 2-element subset of a 5-element set, and connecting two vertices by an edge if the corresponding 2-element subsets are disjoint from each other.

The **Petersen graph** is a small graph that serves as a useful example and counterexample for many problems in graph theory. It is named for Julius Petersen, who in 1898 constructed it to be the smallest bridgeless cubic graph with no edge 3-coloring.

Petersen hypernet — гиперсеть Петерсена.

Petri graph — граф Петри.

Petri net — сеть Петри.

A **Petri net** is a finite directed graph with two types of nodes, referred to as **places** and **transitions**. It is a *bipartite graph*: every arc goes either from a place to a transition or from a transition to a place. Consider a transition t. Every place p (respectively, q) such that there is an arc from t to p (respectively, from q to t) is called an **input** (respectivelyб an **output**) **place** of t. The same place can be both an input and an output place of t.

A **marking** of a Petri net is a mapping m of the set of places into the set of nonnegative integers. The fact that $m(p) = k$ is usually visualized by saying that there are k **tokens** in the place p. A specific **initial marking** m_0 is usually given in the definition of a Petri net. Thus, formally, a (marked) Petri net is a quadruple $N = (P, T, A, m_0)$, where P and T are nonempty finite disjoint sets of places and transitions, A is a subset of $P \times T \bigcup T \times P$. (It is often also required that the union of the domain and codomain of A equals $P \bigcup T$; that is, every place and transition is either the beginning or the end of some arc.) Finally, m_0 (initial marking) is a mapping of P into the set of nonnegative integers.

We now define the **operation** (or **execution**) of a Petri net. A transition is **enabled** (at a marking) iff all its input places have at least one token. An enabled transition may **fire** by removing one token from each of its input places and adding one token to each of its output places.

The operation of a Petri net starts with the **initial marking** m_0. Whenever some transition is enabled, it may fire. This leads to a new marking. If more than one transition is enabled, the firing of such transitions is viewed in an asynchronous fashion: They may fire simultaneously or at different times, one after another. If two transitions have common input places, they are said to be in **conflict**. This means that only one of them can fire at any marking.

A Petri net defined above is an **ordinary Petri net**. A general definition of a Petri net is obtained by introducing **multiplicities** for arcs. Multiplicities means that there is an integer greater than or equal to 1 associated to each arc. The multiplicity of an arc indicates the number of tokens to be subtracted from the input place, as well as the number of tokens to be added to the output place. A transition is not enabled if there are not sufficiently many tokens in each of its input places.

Petri net with place capacities — сеть Петри с емкостью мест.

A Petri net with place capacities is a pair (N, C), where N is a Petri net and C is a mapping of the set of places of N into the set of positive integers. For a place p, the value $C(p)$ is called the **capacity** of the place p.

A transition in a Petri net with place capacities is not enabled if there are not sufficiently many tokens in each of its input places or if the capacity of some of its output places will be exceeded.

Petri net with priorities — сеть Петри с приоритетами.

Petri net with waiting — сеть Петри с ожиданием.

Pfafian orientation of a graph — пфафианова ориентация графа.

Let G be a graph, and H be a subgraph of G. We say that H is central if $G \setminus V(H)$ has a *perfect matching*. Let D be an orientation of G, and let C be a *circuit* of G of even length. We say that C is oddly oriented (in D), if C contains an odd number of edges that are directed (in D) in the direction of each orientation of C. We say that D is a **Pfafian orientation** of G, if every central circuit of G of even length is oddly oriented in D.

It is known that a bipartite graph admits a Pfafian orientation if and only if it does not contain $K_{3,3}$.

Phrase-structure grammar — грамматика с фразовой структурой.

The same as *Grammar*.

Phylogeny digraph — филогенный орграф.

Given a graph $G = (V, E)$, the acyclic digraph D is a **phylogeny digraph** for G if G is an induced subgraph of a *phylogeny graph* $P(D)$ and D has no arcs from vertices outside of G to vertices in G. The **phylogeny number** $p(G)$ is defined to be the smallest r such that G has a phylogeny digraph D with $|V(D) - V(G)| = r$.

Phylogeny graph — филогенный граф.

Given an acyclic digraph $D = (V, A)$, its **phylogeny graph** $P(D)$ is the undirected graph (V, E) with the same vertex set as D and with the following properties for $x \neq y$:

$$xy \in E \Leftrightarrow (\exists a \in V)[(x, a) \in A \& (y, a) \in A]$$

$$or\ [(x, y) \in A]\ or[(y, x) \in A].$$

Phylogeny number — филогенное число.

See *Phylogeny digraph*.

Place — место.

See *Petri net*.

k**-Placement** — k-размещение.

Planar embedding of a graph — плоское вложение графа.

See *Planar graph*.

Planar graph — планарный граф, плоский граф.

A crossing-free embedding of a graph in the plane is given by drawing a graph G in the plane with points representing vertices and curves representing edges such that no two curves for edges intersect except at common endvertices. G is a **planar graph** if there is a crossing-free embedding of G in the plane.

In other words, a graph (digraph) G is called a **planar graph**, if there is a projection Π of the vertices and edges of G into the plane such that the intersections of the projections of edges occur at the projections of vertices, and $\Pi((u, v))$ is a Jordan-curve from $\Pi(u)$

to $\Pi(v)$. The projection Π is called a **planar embedding** of G. It divides the plane into a number of connected regions, called **faces**, each bounded by the projection of edges of the graph. There is always a face with an infinite area, which is called the **exterior face**.

See also *Plane graph*.

(a, b)**-Planar graph** — (a, b)-плоский граф.

Planar matroid — планарный матроид.

Planar tree — плоское дерево.

Planar triangulation — плоская триангуляция.

A planar map in which each *face* is a triangle.

Planarity criteria — критерии планарности.

The following three planarity criteria are classical.

1. Kuratowski's criterion. A graph G is planar if and only if it does not contain a subdivision of K_5 or $K_{3,3}$.

Another name is **Pontrjagin-Kuratowski's criterion**.

2. Whitney's criterion. A graph G is planar if and only if it has a combinatorial dual graph G^*.

3. MacLane's criterion. A graph G is planar if and only if it has a cycle basis such that each edge of G belongs to at most two circuits of the basis.

Plane graph — плоский граф.

A **plane graph** is a *planar graph* with a fixed embedding in the Euclidean plane.

A graph is said to be **plane** if it is drawn on the Euclidean plane in such a way that edges do not cross each other except at vertices of the graph.

Plane map — плоская карта.

Plane numbering — плоская нумерация.

Plane triangulation — плоская триангуляция.

A plane graph is a **plane triangulation** if all its faces are bounded by 3-cycles.

Plex — сплетение (сеть).

Point — точка, вершина.

Point-covering number — число вершинного покрытия.

Point spectrum — точечный спектр.

See *Spectrum*.

Point-tree hypergraph — дерево-точечный гиперграф.

A hypergraph H is called a **point-tree hypergraph** if it is obtained from a *bipartite* graph by replacing, in each edge, the point in one side of the graph (the same side for all edges) by a tree. More formally, H is **point-tree**, if for some set X and a tree, whose vertex set is disjoint with X, each edge $e \in H$ is of the form $\{x\} \cup V(t)$, where $x = x(e) \in X$and $t = t(e)$ is a subtree of T. For such a hypergraph we denote by $\sigma(H)$ the number $w(H, F)$, where $F = F(H) = \{\{x(e)\} : e \in H\} \cup \{V(t(e)) : \ e \in H\}$.

Polar graph — полярный граф.
The same as *Split graph*.

Pole — полюс.

Polynomial algorithm — полиномиальный алгоритм.

Polynomial expression of the stability function — полиномиальное выражение функции устойчивости.
See *Stability function*.

Polynomial graph inclusion problem — проблема включения графов полиномов.

Polynomial transformation — полиномиальная сводимость (трансформируемость).

Polygonal tree — многоугольное дерево.
A graph G is called a **polygonal tree**, if it consists of finitely many regular polygons (we assume any two distinct polygons be not coplanar) and has the following two properties:
(1) any two distinct polygons are disjoint or have exactly one edge in common (such an edge can be a common edge of several polygons),
(2) the diagram obtained by joining the centroids of the polygons to the mid-point of the common edge has no closed curve.
If all polygons of a polygonal tree G are the same, say s-gons, then G is called an **s-gonal tree**. 6-gonal tree is called **hexagonal tree**. Consider the diagram defined in condition 2. If we set the centroids and the mid-points of common edges of some polygons as “red” vertices and “green” vertices, respectively, and the straight line segments as edges of a bipartite graph, then this graph is a tree.

Polytop graph — граф многогранника.

Polyhedral graph — полиэдральный граф.
A **polyhedral graph** $G = (V, E, F)$ with a vertex set V, an edge set E and a face set F is a planar and 3-connected graph. A **polyhedral graph** $G = (V, E, F)$ is called **face transitive**, if for each pair of faces α, $\beta \in F$ there is an automorphism $\phi_{\alpha,\beta}\alpha = \beta$.

Polyhedron graph — граф многогранника.

Pontrjagin-Kuratowski's criterion — критерий Понтрягина-Куратовского.

See *Planarity criteria.*

Poset — чу-множество.

See *Partially order relation.*

Post-condition — постусловие.

Position tree — дерево позиций.

Postdomination — постдоминирование.

Postdominator — обязательный преемник, постдоминатор.

Postdominator tree — постдоминаторное дерево.

Potential liveness of transitions problem — проблема живости переходов.

Potentially dead transition — потенциально мертвый переход.

Potentially live transition — потенциально живой переход.

k**-th Power of a graph** — k-я степень графа.

See *Dually chordal graph.*

Power-chordal graph — степенно-хордальный граф.

This is a graph G such that all of its *kth powers* are chordal.

PRAM — параллельная равнодоступная машина.

See *Parallel Random Access Machine.*

Pre-condition — предусловие.

Predecessor of a vertex — предок вершины.

See *Flow graph.*

Predicate term — логическое выражение, слово применимости.

See *Large-block schema.*

Prefix — префикс.

See *String.*

Prefix graph — префиксный граф.

For all $n \in N$, a **prefix graph** of width n is a *directed acyclic graph* $G = (V, E)$ with n distinguished input vertices $x_1, \ldots, x_n$ of indegree zero and n distinguished output vertices $y_1, \ldots, y_n$ of outdegree zero and with the following properties, where the span of a vertex $v \in V$, $span(v)$, is defined as $\{i : 1 \leq i \leq n$ and G contains a path from x_i to $v\}$:

(1) For $i = 1, \ldots, n$, $span(y_i) = \{1, \ldots, i-1\}$ (for $i = 1$ this is $\emptyset$).

(2) For all $v \in V$, $span(v)$ is either empty or an “interval” of the form $\{s, \ldots, t\}$, for some integers s and t with $1 \leq s \leq t \leq n$.

(3) Any two vertices in V with a common successor have disjoint spans.

Prefix graph of width n — префиксный граф ширины n.

Prefix language — префиксный язык.

Prefix tree — префиксное дерево, нагруженное дерево.

The same as *Trie*.

Preorder — предпорядок.

A binary relation on $\{1, 2, \ldots, n\}$ is a **preorder**, if it is reflexive and transitive.

Prependant vertex — предвисячая вершина.

A vertex is **prependant**, if it is adjacent to a *pendant* vertex.

Prescribed chromatic number — предписанное хроматическое число.

See *List chromatic number*.

Prime hammock — простой гамак.

See *Hammock*.

Prime graph — примитивный граф, элементарный граф.

See *Prime labeling*.

Prime labeling — примитивная разметка, элементарная разметка.

A graph with a vertex set V is said to have a **prime labeling**, if its vertices are labelled with distinct integers from $\{1, 2, \ldots, |V|\}$ such that, for each edge xy, the labels assigned to x and y are relatively prime.

Primitive cycle — примитивный цикл.

See *Cycle*.

Primitive directed graph — примитивный орграф.

1. A digraph D is **primitive** if there exists an integer k such that there is a *walk* $u \to v$ of length k for every pair $u, v \in V$. The least such k is called the **exponent** of D, denoted $\gamma(D)$. The **local exponent** of D at a vertex $u \in V$, denoted by $\exp_D(u)$, is the least integer k such that $u \to v[k]$ (a walk of length k) for each $v \in V$.

2. The **index** and **period** of a given digraph D are the minimum nonnegative integer $k = k(D)$ and the minimum positive integer $p = p(D)$ such that for any ordered pair of vertices x and y there is a *walk* of length k from x to y if and only if there is a walk of length $k + p$ from x to y in D. A digraph D is **primitive** if D is strongly connected and $p(D) = 1$.

Prism — призма.

A **prism** D_n, $n \geq 3$, is a trivalent graph which can be defined as the

Cartesian product $P_2 \times C_n$ of a path on two vertices with a cycle on n vertices. The prism can also be defined as the *Cayley graph* of the dihedral group of order $2n$.
See also *Antiprism*.

Private neighbour — приватный сосед.
See *Private neighbor set*.

Private neighbor set — приватное соседнее множество.
The **private neighbour set** of a vertex v in S is denoted by

$$PN[v, S] = N[v] - N[S - \{v\}].$$

If $PN[v, S] \neq \emptyset$, then every vertex of $PN[v, S]$ is called a **private neighbor** of v with respect to S, or just an S-pn.
See also *Private neighbourhood*.

Private neighbourhood — приватное соседство.
See *Neighbourhood of a vertex*.

Primitive net formula — примитивная формула.

Primitive Petri net — примитивная сеть Петри.

Print operator — оператор печати символа.

Priority — приоритет.

Problem — проблема.

Problem of finite-state automaton minimization — проблема минимизации конечного автомата.

Problem size — размер проблемы.
See *Time complexity*.

Process — процесс.

Process net — сеть-процесс.

Process net with competition — сеть-процесс с конкуренцией.

Product of two graphs — произведение двух графов.
There are some kinds of products.
1. (Weak) direct product $G_1 \times G_2$ of G_1 and G_2, defined by

$$V(G_1 \times G_2) = V(G_1) \times V(G_2),$$

$$E(G_1 \times G_2) = \{((x_1, x_2), (y_1, y_2)) | (x_1, y_1) \in E(G_1), (x_2, y_2) \in E(G_2)\}.$$

Other names are **Cardinal product**, **Cross product**, **Tensor product**, **Kronecker product**, **Categorical product**, **Graph conjunction**.
2. Strong direct product $G_1 \cdot G_2$ of G_1 and G_2, defined by

$$V(G_1 \cdot G_2) = V(G_1) \times V(G_2),$$

$$E(G_1 \cdot G_2) = \{((x_1, x_2), (y_1, y_2))|$$
$$(x_1, y_1) \in E(G_1) \text{ and } (x_2, y_2) \in E(G_2), \text{ or } x_1 = y_1 \text{ and}$$
$$(x_2, y_2) \in E(G_2), \text{ or } (x_1, y_1) \in E(G_1), \text{ and } x_2 = y_2\}.$$

3.1. Cartesian product $G_1 \oplus G_2$, defined by

$$V(G_1 \oplus G_2) = V(G_1) \times V(G_2),$$
$$E(G_1 \oplus G_2) = \{((x_1, x_2), (y_1, y_2))|$$
$$x_1 = y_1 \text{ and } (x_2, y_2) \in E(G_2); \text{ or } (x_1, y_1) \in E(G_1) \text{ and } x_2 = y_2)\}.$$

Thus, $G_1 \cdot G_2 = (G_1 \times G_2) \cup (G_1 \oplus G_2)$.

3.2. For given graphs $G_i = (V_i, E_i)$, $i \in [1, n]$, the vertex set of the product graph $G = (V, E)$ is the **Cartesian product** of the vertex sets of the factors G_i, i.e. $V = \{(a_1, \ldots, a_n)\}$, where $a_i \in V_i$. Two vertices $\bar{a} = (a_1, \ldots, a_n)$ and $\bar{b} = (b_1, \ldots, b_n)$ are *adjacent* in n-fold **C.p.** if and only if $a_i \neq b_i$ for precisely one i, $1 \leq i \leq n$, and for this i (a_i, b_i) is an edge in G_i.

4. Categorical product of graphs is defined as follows. For given graphs $G_i = (V_i, E_i)$, $i \in [1, n]$, the vertex set of the product graph $G = (V, E)$ is the Cartesian product of the vertex sets of the factors G_i, i.e. $V = \{(a_1, \ldots, a_n)\}$, where $a_i \in V_i$. Two vertices $\bar{a} = (a_1, \ldots, a_n)$ and $\bar{b} = (b_1, \ldots, b_n)$ are *adjacent* in n-fold **Cartesian product** if and only if for all i, $1 \leq i \leq n$, (a_i, b_i) is an edge in G_i.

5. Square product is the same as *Cartesian product.*

6. Semi-strong product $G = G_1 \bullet G_2$ has the vertex set $V(G) = V(G_1) \times V(G_2)$ and the edge set $E(G) = \{(u_1, u_2)(v_1, v_2)|\ u_1 = v_1$ and $u_2v_2 \in E_2$ or $u_1u_1 \in E(G_1)$ and $u_2v_2 \in E(G_2)\}$.

7. Lexicographic product $G = G_1[G_2]$ has the vertex set $V(G) = V(G_1 \times V(G_2)$ and the edge set $E(G) = \{(u_1, u_2)(v_1, v_2)|\ u_1 = v_1$ and $u_2v_2 \in E(G_2)$ or $u_1v_1 \in E(G_1)\}$.

8. Special product $G_1 \oslash G_2$ has the vertex set $V(G) = V(G_1 \times V(G_2)$ and the edge set

$$E(G) = \{(u_1, u_2)(v_1, v_2)|\ u_1v_1 \in E(G_1) \text{ or } u_2v_2 \in E(G_2)\}.$$

9. Product of two hypergraphs $\mathcal{H}_1$ and $\mathcal{H}_2$ is a hypergraph $\mathcal{H}_1 \times \mathcal{H}_2$, defined by

$$V(\mathcal{H}_1 \times \mathcal{H}_2) = V(\mathcal{H}_1) \times V(\mathcal{H}_2),$$
$$E(\mathcal{H}_1 \times \mathcal{H}_2) = \{E_1 \times E_2 | E_1 \in E(\mathcal{H}_1), E_2 \in E(\mathcal{H}_2)\}.$$

Product of two languages — произведение двух языков, конкатенация двух языков.

See *Formal language.*

Production — продукция.

See *Grammar.*

Production grammar — порождающая грамматика.

The same as *Grammar.*

Profile numbering — профильная нумерация.

For a *proper numbering* f, the **profile width** of a vertex v is defined as

$$w_f(v) = f(v) - \min_{x \in N[v]} f(x),$$

where $N[v]$ is the *closed neighborhood* of v. The **profile of numbering** f for G is defined as

$$P_f(G) = \sum_{v \in V} w_f(v).$$

The **profile of** G is the minimum value

$$P(G) = \min_f (P_f(G)),$$

where f is taken over all proper numberings of G. A proper numbering f that attains the minimum value is called a **profile numbering**.

Profile of a graph — профиль графа.

See *Profile numbering.*

Profile of numbering — профиль нумерации.

See *Profile numbering.*

Profile width of a vertex — профильная ширина вершины.

See *Profile numbering.*

Program — программа.

A **computer program** (or a **program**) is an algorithm for a computer. A program can be either in an executable form (an **executable program**) or a source code (or a **source program**) from which an equivalent executable program is derived (e.g., compiled); the source program may also be used to describe an algorithm to a reader.

Program dependence graph — граф программных зависимостей.

Program dependences — программные зависимости.

Program equivalence — эквивалентность программ.

When a *program* calculates some function (as is usually the case), there is a natural and most general definition of equivalence: two programs, which have a common set of arguments are **functionally equivalent**, if their functions are the same.

However, an unavoidable obstacle in developing a general theory is the following negative result in the theory of algorithms. Some property of a program is said to be internal, if it takes place for all programs functionally equivalent to it. Rice has proved that for any internal property of a program, there is no algorithm which would recognize those programs which possess the given property (naturally, the class of programs should be reasonably meaningful, for instance, it should compute any recursive function).

The principal way of narrowing the concept of equivalence of programs is to compare not only the values of the functions evaluated by programs, but also some history of computation during the execution. Formally the concept of history is introduced as follows. In addition to the universal algorithm of program execution, another algorithm is introduced which, in accordance with the program and a set of its input data, constructs some object. The latter is called the **history of the program realization** and contains some information about its execution. The history may comprise any number of details, but the result of the program execution has to be recovered by it in a single-valued way. Hence, programs with coincident histories automatically have coincident results. A special case of a history is the program itself (an identical tracing algorithm). This history, naturally, is the most detailed one, because we can get any information from the program by applying to it the universal algorithm of execution. The equivalence based on this history appears to be the narrowest: the program is equivalent only to itself.

Program of automaton — программа автомата.

Program optimization — оптимизация программ.

Program schemata — схемы программ.

Program schemata are a class of abstract *programs* with an *equivalence relation* between them. They retain many structural properties of programs, in particular, splitting into statements with indications of the information and control flow between them. This makes it possible for program schemata to construct many of the characteristics typical for concrete programs, for example, histories of program execution. In program schemata, variables, operations and predicates are represented by formal symbols without any internal properties. These formal objects keep only the information needed for constructing histories of their realizations, for example, for formal operations, only the number of formal arguments and the names of formal variables substituted for them are indicated; for a formal statement of control transfer, only those statements to which control may be transferred are labelled, and so on.

Every program scheme α describes (**models**) a set P_α of concrete programs, and two program schemata α and β are considered as **functionally equivalent** only if any concrete programs $p_1 \in P_\alpha$ and $p_2 \in P_\beta$ having the same structure are functionally equivalent (i.e. computes the same function).

Concrete programs can be obtained from schemata by means of **interpretation** which consists in bringing some concrete variables and operations into correspondence with formal variables and operations. Very important is the concept of the set Ω of all interpretations of program schemata. The theory is developed in such a way that a fact ascertained for some schema should be true for any interpreting program. In particular, in this way the notion of equivalence of two program schemata is introduced: two program schemata S_1 and S_2 are **equivalent** in the sense of the history H, if for any interpretation $I \in \Omega$ concrete programs obtained from S_1 and S_2 are equivalent in the sense of this history.

Progressive bounded graph — прогрессивно ограниченный граф.

Progressive finite graph — прогрессивно конечный граф.

Proper control flow graph — правильный уграф.

Proper coloring — собственная, правильная раскраска.

See *Coloring.*

Proper dominator — собственный доминатор, собственный обязательный предшественник.

See *Dominator.*

Proper interval graph — собственный интервальный граф.

Proper labeling — правильная нумерация.

See *Proper numbering.*

Proper matching — правильное паросочетание.

Proper numbering — правильная нумерация.

For a simple graph $G = (V, E)$ with n vertices, a bijection (1-1, onto mapping) $f:\ V \to [1, n]$ is called a **proper numbering** of G.

Another name is **Proper labeling**.

Proper (vertex) colouring — правильная раскраска (вершин).

A **proper colouring** of G is an assignment of colors to the vertices so that adjacent vertices obtain distinct colors. The **chromatic number** $\chi(G)$ is the minimum number of colors required among all proper colorings of G.

Proper substring — собственная подцепочка.

See *String*.

Provable problem — частично разрешимая задача.

See *Decision problem*.

Pruned tree — сокращенное дерево.

If a *leaf* v (together with the unique edge e incident with v) of a nontrivial tree T is removed from T, we say that v has been pruned from T. We refer to the removal of all the leaves of a tree T of order $n \geq 3$ as a pruning of T. The graph that results from a pruning of T is a (possibly trivial) tree T^*, called the **pruned tree** of T.

Pseudograceful graph — псевдограциозный граф.

A graph $G = (V, E)$ such that $|V| \leq |E| + 1$ is said to be **pseudo-graceful**, if there exists an injective function called pseudograceful labelling $f : V \to \{0, 1, \ldots, |E| - 1, |E| + 1\}$ such that the induced function

$$f^* : E(G) \to \{1, 2, \ldots, |E(G)|\}$$

defined by

$$f^*(xy) = |f(x) - f(y)| \text{ for all } xy \in E(G)$$

is an injection.

See also *Graceful graph*.

Pseudograph — псевдограф.

Let $G = (V, E)$ be a digraph on n vertices. G is called **a pseudograph**, if it permits *loops* but no multiple arcs in D.

Pseudo-hamiltonian graph — псевдогамильтонов граф.

See *Pseudo-h-hamiltonian graph*.

Pseudo-hamiltonicity number — число псевдогамильтоновости.

See *Pseudo-h-hamiltonian graph*.

Pseudo-h-hamiltonian cycle — псевдо-h-гамильтонов цикл.

See *Pseudo-h-hamiltonian graph*.

Pseudo-h-hamiltonian graph — псевдо-h-гамильтонов граф.

For an integer $h \geq 1$, an undirected graph $G = (V, E)$ is a **pseudo-**

h-**hamiltonian graph**, if there exists a circular sequence of $h \cdot |V|$ vertices such that the following properties hold:
(1) every vertex of G appears precisely h times in the sequence, and
(2) any two consecutive vertices in the sequence are *adjacent* in G.
A sequence with these properties will be termed a **pseudo-h-hamiltonian cycle**. In this sense, **pseudo-1-hamiltonian** corresponds to the standard notion *hamiltonian*, and a **pseudo-1-hamiltonian cycle** is just a *hamiltonian cycle*. The **pseudo-hamiltonicity number** $ph(G)$ of the graph G is the smallest integer $h \geq 1$ for which G is pseudo-h-hamiltonian; in case no such h exists, $ph(G) = \infty$. A graph G with finite $ph(G)$ is called **pseudo-hamiltonian**. Pseudo-h-hamiltonicity is a non-trivial graph property. E.g., for every $h \geq 2$, the graph G_h that results from gluing together h triangles at one of their vertices is pseudo-h-hamiltonian but it is not pseudo-$(h-1)$-hamiltonian.

Pseudo-polynomial algorithm — псевдополиномиальный алгоритм.

A numeric algorithm runs in pseudo-polynomial time, if its running time is polynomial in the numeric value of the input (which is exponential in the length of the input - its number of digits).
An NP-complete problem with known pseudo-polynomial time algorithms is called **weakly *NP*-complete**. An NP-complete problem is called **strongly *NP*-complete**, if it is proven that it cannot be solved by a pseudo-polynomial time algorithm. The **strong/weak** kinds of ***NP*-hardness** are defined analogously.

Pseudo-product — псевдопроизведение.

Let G and G' be simple graphs on the same set of vertices $V(G) = V(G') = V$, where $|V| = n \geq 1$. Define the **pseudo product** of G and G' to be the simple graph $G * G'$ on the vertex set V with the edge set $E(G * G') = E(G) \cup E(G') \cup E^*$, where

$$E^* = \{\{u, v\} : \exists w \in V : \{u, w\} \in E(G), \{w, v\} \in E(G'),$$

$$\text{and } \exists w' \in V : \{u, w'\} \in E(G'), \{w', v\} \in E(G)\}.$$

For a simple graph G and nonnegative integers s and t, we have

$$G^s * G^t = G^{s+t}.$$

In particular, the pseudo product is an associative operation on the set $\{G^k : k \in \{0, 1, 2, \ldots, \}\}$ for any fixed simple graph G.

Pseudosimilar vertices — псевдоподобные вершины.

Pseudosymmetric digraph — псевдосимметричный орграф.

See *Symmetric graph.*

Pseudovertex — псевдовершина.

Pseudo-wheel — псевдо-колесо.

The **pseudo-wheel** consists of a cycle graph on $2n$ vertices with n additional edges connecting vertices on the opposite sides of the cycle.

Pumping lemmas — леммы о возрастании.

Pure synthesized grammar — чисто синтезированные грамматики.

Pushdown automaton — автомат с магазинной памятью.

See *Model of computation.*

Q

Quad cycle — квадрат.

A **quad cycle** in a bigraph is a p-cycle, where p is divisible by 4.

Quadrilateral — четырехсторонник, четырехугольник.

A cycle of length 4 is called a **quadrilateral**.

Quadtree — кваддерево.

A **quadtree** is a ternary tree representing a hierarchical decomposition of the plane, originally proposed for representing point sets. Each node of the **quadtree** corresponds to a square region, called a box. The root usually corresponds to the smallest enclosing square of the given set of objects. A node of the **quadtree** acquires four *children*, when its associated box is split into its four quadrants.

Quasi-bipartite mixed graph — квазидвудольный смешанный граф.

A *mixed graph* is called **quasi-bipartite**, if it does not contain a nonsingular cycle, i.e., a cycle containing an odd number of unoriented edges.

Quasibipyramid — квазибипирамида.

The plane dual graph A_n^* of the *antiprism* A_n is the graph of a **quasibipyramid**.

Quasi-diameter — квазидиаметр.

Let $\rho(x, y)$ be a distance function on the vertex set V of a directed graph without loops and let $\rho_m(x, y)$ be a function defined by

$$\rho_m(x, y) = \min\{\rho(x, y), \rho(y, x)\}.$$

Then the **quasi-diameter** $d_m(G) = \max_{x,y \in V} \rho_m(x, y)$ and the **quasi-radius** $r_m(G) = \min_{x \in V} \max_{y \in V} \rho_m(x, y)$

Quasi-radius — квазирадиус.

See *Quasi-diameter.*

Quasistrongly connected graph — квазисильно связный граф.

Quasi-transitive tournament — квазитранзитивный турнир.

See *Transitive tournament.*

R

Radial path — радиальный путь.

A **radial path** in a graph G is a path of length $r(G)$, where $r(G)$ is the *radius* of G that joins a central vertex to one of its eccentric vertices.

Radially critical graph — радиально критический граф.

Radius of a graph — радиус графа.

See *Eccentricity of a vertex.*

p-**Radius** — p-радиус.

Let $G = (V, E)$ be a graph and $w: V \to R^+ \cup \{0\}$ be a nonnegative weight function defined on V. We define the **radius** $r(S)$ of a set $S \subseteq V$ as $\max\{w(u)d(u, S): u \in V\}$, where $d(u, S) = \min\{d(u, v): v|inS\}$. For a given positive integer $p \leq |V|$, we define the p-**radius** of G as

$$r_p(G) = \min\{r(C);\ C \subseteq V,\ |C| = p\}.$$

See also *Radius of a graph, p-center.*

Radius-essential edge — радиус-существенное ребро.

An edge e is **radius-essential** if $rad(G/e) < rad(G)$. The number of radius-essential edges in a graph G is denoted by $c_r(G)$.

Ramanujan graph — граф Рамануджана.

1. A finite regular graph of degree k is said to be a **Ramanujan graph** if, apart from the trivial eigenvalues $\pm k$, its *spectrum* is contained not only in $[-k, k]$ as Perron–Frobenius guarantees, but in the smaller range $[-2\sqrt{k-1}, 2\sqrt{k-1}]$. This range is in some asymptotic sense the smallest possible.

2. A k-regular graph X is a **Ramanujan graph** if and only if its Ihara zeta function $Z_X(s)$ satisfies the "Riemann hypothesis", i.e., all poles of $Z_X(s)$ in $0 < \Re s < 1$ lie on the line $\Re s = \frac{1}{2}$.

Ramsey graph game — рамсеевская игра на графах.

The board of the game is the complete graph K_s with s vertices. The players alternately occupy the edges of K_s, and that player wins who first occupies all the $\binom{n}{2}$ edges of some complete subgraph K_n.

The Ramsey Graph Game is denoted by $R(s, n)$.

Random access machine — равнодоступная адресная машина.

See *Model of computation.*

Random graph — случайный граф.

Rank function — ранговая функция.
See *Matroid*.
Rank of a graph — ранг графа.
Rank of a graph group — ранг группы графа.
Rank of a hypergraph — ранг гиперграфа.
See *Hypergraph*.
Rank of a matroid — ранг матроида.
See *Matroid*.
Ranking number — число ранжирования.
See *k-ranking*.
k**-Ranking** — k-ранжирование.
Given an undirected graph G, a **(vertex)** k**-ranking** of G is a mapping (coloring) $f: V(G) \to \{1, 2, \ldots, k\}$ such that every path connecting two vertices u, v of the same rank $f(u) = f(v)$ contains a vertex w with a higher rank, $f(w) > f(u)$. The **ranking number** $\chi_r(G)$ is the minimum integer k for which there exists a k-ranking. It is well known and easy to see that, for the path P_L of length $L-1$ on L vertices, we have

$$\chi_r(P_L) = \lfloor \log L \rfloor + 1$$

and that the longest k-rankable path $P_{2^k-1} = x_1 x_2 \ldots x_{2^k-1}$ admits the unique optimal ranking f with

$$f(x_i) = \max\{j : \ 2^j | i\} + 1$$

for all $1 \leq i < 2^k$.
Analogously, k**-ranking** for directed graphs is defined.
Ray — луч.
1. A **ray** $\langle x_0, x_1, \ldots \rangle$ is an infinite path (or chain) in an infinite graph. The other name is **one-way infinite path**. A **double ray** $\langle \ldots, x_{-1}, x_0, x_1, \ldots \rangle$ is an infinite path (or chain) which contains the vertex x_0. The other name is **two-way infinite path**.
2. See *Basic block*.
F**-Ray** — F-луч.
Let F be a *numbering* of a *cf-graph* G and A be a ray of G. A is called F**-ray** if it contains only *F-arcs*.
Reachability — достижимость.
Reachability graph — граф достижимости, граф разметок.
The **reachability graph** of a *Petri net* N is a (not necessary finite)

directed graph whose nodes are labeled by markings reacheable for N. The arcs of the reachability graph are labeled by the transitions of N in such a way that there is an arc labeled by t from a node labeled by m_1 to a node labeled by m_2 iff the firing of t changes the marking m_1 to m_2.

Reachability matrix — матрица достижимости.

Reachability problem — проблема достижимости (разметки).

The **reachability problem** for *Petri nets* consists in finding an algorithm for deciding about a Petri net N and a marking n of N whether or not m is reachable for N.

The reachability problem was for a long time the most-celebrated open problem in the theory of Petri nets. Before a proof for its decidability was found, many equivalent versions for the statement were known. The equivalent versions dealt with various aspects of Petri nets, language theory and vector-additive systems. For example, the reachable problem for Petri nets is decidable iff the empty marking problem for Petri nets is decidable.

Reachability relation — отношение достижимости.

F**-Reachable (from** p**) node** — F-достижимая (из p) вершина.

See *Numbering of cf-graph.*

Reachable (from a**) vertex** — достижимая (из a) вершина.

Given a digraph $G = (V, A)$, a vertex $w \in V$ is called **reachable** from $v \in V$ iff there exists a path from v to w.

Reachable marking — достижимая разметка.

Let N be a *Petri net*. A marking m is called **reachable** for N iff there is a finite sequences of firings of the transitions of N leading N from the *initial marking* to m. All reachable markings of a Petri net N are denoted by $R(N)$.

Reaching matrix — матрица контрадостижимостей, матрица обратных достижимостей.

Reaching set — контрадостижимое множество.

Reach-preservable graph — сохраняющий достижимость граф.

Given a *spanning tree* T of a graph G, a vertex $v \in V(G)$ is called **reach-preserving** if the distance $d_T(v, w) = d_G(v, w)$ for all w in G. A graph is called **reach-preservable graph** if each of its spanning trees has a reach-preserving vertex.

By definition, it is clear that all trees are reach-preservable, and any cycle is reach-preservable. Furthermore, we can deduce that connected *unicyclic* graphs are reach-presevable.

Reach-preserving vertex — сохраняющая достижимость вершина. See *Reach-preservable graph.*

Realizable admissible sequence — реализуемая допустимая последовательность.

See *Admissible sequence.*

Realization of a hypergraph — реализация гиперграфа.

Realizer of P — реализёр P.

See *Linear extenson of a poset.*

Reasonable numbering — разумная нумерация.

A numbering of a *cf-graph* G is called **reasonable** if the following two properties hold:

(1) for any two distinct nodes p and q, if p is a *dominator* of q, then $F(p) < F(q)$.

(2) if G is an *arrangeable graph*, then F is its *arrangement.*

Receiver — приёмник.

See *Directed graph.*

Recognizer — распознаватель.

See *Large-block schema.*

k**-Recognizer** — k-распознаватель.

See *Large-block schema.*

Reconstructible graph — реконструируемый граф.

1. A graph G is **reconstructible**, if every graph *hypomorphic* to G is isomorphic to G.

2. An infinite *locally finite* connected graph G is **reconstructible**, if there exists a finite family $(\Omega_i)_{0\leq i<n}$ $(n \geq 2)$ of pairwise finitely separable subsets of its end set $\mathcal{E}(G)$ such that, for all $x, y, x', y' \in V(G)$ and every isomorphism f of $G - \{x, y\}$ onto $G - \{x', y'\}$, there is a permutation π of $\{0, \ldots, n-1\}$ such that $f(\Omega_i) = \Omega_{\pi(i)}$ for $0 \leq i < n$.

k**-reconstructible graph** — k-реконструируемый граф.

Let k be an integer $(k \geq 1)$ and $G = (V, E)$ a graph with more than k vertices, a graph $G' = (V, E')$ is a k**-reconstuction** of G if, for any subset W of V with k elements, the subgraph $G(W)$ and $G'(W)$ induced by W are isomorphic. The graph G is k**-reconstructible**, when each k-reconstruction of G is isomorphic to G. G. Lopez (1978) proved that any graph is 6-reconstructible.

Reconstruction of a graph — реконструкция графа.

k-reconstruction of a graph — k-реконструируемый граф.
See *k-reconstructible graph.*

Rectangular graph — прямоугольный граф.

Recursive set — рекурсивное множество.
See *Decision problem.*

Recursively enumerable language — рекурсивно-перечислимый язык.
See *Grammar.*

Recursively enumerable set — рекурсивно-перечислимое множество.
See *Decision problem.*

Recursive nonterminal symbol — рекурсивный нетерминальный символ.

Redex — редекс.
See *Graph transformation rule.*

Reduced graph — приведенный граф.

Reduced hypergraph — сокращенный гиперграф.
A *hypergraph* $\mathcal{H}$ is a **reduced hypergraph** if no edge $e \in \mathcal{H}$ is contained in its another edge.

Reduced path covering — приведенное путевое покрытие.

P_4-Reduced graph — P_4-сводимый граф.
This class was introduced by Jamison and Olariu (1989) as the class of graphs for which no vertex belongs to more than one induced P_4.
See *P_4-sparse graphs.*

Y-Reduced sequence — Y-сводимый маршрут.

Reducible additive hereditary graph property — свойство сводимой аддитивной наследуемости графов.
See *Additive hereditary graph property.*

Reducible (control) flow graph — сводимый управляющий граф.
Let G be a *cf-graph* and let $k \geq 0$. The **k-derived** cf-graph G_k of G, denoted $G_k = I_k(G)$, is defined by the following rules: $G_0 = G$, and for any $k > 0$ the cf-graph G_k is derived from the cf-graph G_{k-1} by reduction of its maximal interval into nodes. The **limit** cf-graph of G is defined as its k-derived cf-graph G_k such that $G_k = I_{k+1}(G)$. G is called **(interval) reducible** if its limit cf-graph is trivial and **(interval) irreducible** otherwise.

Reducible by Hecht and Ullman flow graph — сводимый по Хехту и Ульману управляющий граф.
See *Collapsible graph.*

P_4-Reducible graph — P_4-сводимый граф.
See *P_4-reduced graph.*

Reduction tree — дерево редукций.

Reflexive graph — рефлексивный граф.
Undirected graph which has loops in all vertices is called a **reflexive graph**.

Reflexive relation — отношение рефлексивности.
See *Binary relation.*

Reflexive-transitive closure of a graph — рефлексивно-транзитивное замыкание графа.

F-Region — F-область.
See *Numbering of cf-graph.*

Region of connectivity — область связности.

Region-interval presentation — зонно-интервальное представление.
The same as *Zone-interval reprezentation.*

Register-interference graph — граф межрегистровых связей.
For each procedure, a **register-interference graph** is constructed whose nodes are symbolic registers and an edge connects two nodes, if one is live at the point, where the other is defined.

Regressive bounded graph — регрессивно ограниченный граф.

Regressive finite graph — регрессивно конечный граф.

Regular basic subnet — регулярный базовый фрагмент сети.

Regular expression — регулярное выражение.
Assume that Σ and $\Sigma' = \{+,^*, \emptyset, (,)\}$ are disjoint alphabets. A string w over the alphabet $\Sigma \bigcup \Sigma'$ is a **regular expression over** Σ iff w is a symbol of Σ, or the symbol $\emptyset$, or w is of one of the forms $(w_1 + w_2)$, $(w_1 w_2)$, $(w_1)^*$, where w_1 and w_2 are regular expressions over Σ.
Each regular expression w over Σ **denotes** a language $L(w)$ over Σ according to following conventions:
(1) the language denoted by $\emptyset$ is the empty set,
(2) the language denoted by $a \in \Sigma$ consists of the string a,
(3) for all regular expressions w_1 and w_2 over Σ, we have
$L((w_1 + w_2))=L(w_1)) \bigcup L(w_2)$,
$L((w_1 w_2))=L(w_1)L(w_2)$,
$L((w)^*)=(L(w))^*$.
The following property holds:
$L = L(w)$ for a regular expression w over Σ iff L is a *regular language* over Σ.

Regular expression nonequivalence problem — задача о неэквивалентности регулярных выражений.

Regular graph — регулярный граф, однородный граф.

G is a **regular graph** of degree k, if every vertex is incident with k edges (i.e., every vertex has the degree equal to k).

A graph that is not regular is called **irregular**. It is well known that a *simple* graph must have at least two vertices of the same degree. If a graph has exactly two vertices of the same degree, we call it a **maximally irregular graph**.

If multiple edges and loops are allowed, the degree sequences in which all elements are distinct are realizable. If no two vertices of a graph have the same degree, we call it a **totally irregular graph**. It is obvious that a totaly irregular graph cannot be a simple graph.

A graph is (r, s)**-regular**, if the degree of each vertex is either r or s.

Regular group of a graph — регулярная группа графа.

Regular language — регулярный язык.

Regular loop — регулярный цикл.

Regular matroid — регулярный матроид.

See *Matrix matroid.*

Regular Petri net — регулярная сеть Петри.

Regular Petri net with finite marking — регулярная конечноразмеченная сеть Петри.

Regular set — регулярное множество.

Let Σ be an alphabet. **Regular sets over the alphabet** Σ are all sets that can be obtained by finitely many applications of the three following rules:

(1) the empty set $\emptyset$ is a regular set over the alphabet Σ,

(2) if a is in Σ, then the singleton set $\{a\}$ is a regular set over the alphabet Σ,

(3) if P and Q are regular sets over the alphabet Σ, then $P \cup Q$, PQ and P^* are also regular sets over the alphabet Σ.

The following property holds for any language L over Σ:

L is a regular set iff L is a *regular language.*

Regular tournament — регулярный турнир.

See *Tournament.*

(r, s)**-Regular graph** — (r, s)-регулярный граф.

See *Regular graph.*

d-Regular tree with boundary — d-регулярное дерево с границей.
See *Graph with boundary.*

Regularizable graph — регуляризуемый граф.
1. A graph $G = (V, E)$ is called **regularizable** (Berge), if for each edge $e \in E$ there is a positive integer $m(e)$ such that the multigraph which arises from G by replacing every edge e by $m(e)$ parallel edges is a *regular* graph.
2. A *cf-graph* G is called a **regularizable graph** (or **generalized reducible**) (Kasyanov), if there is a sequence of cf-graphs G_0, G_1, ..., G_r such that $G_0 = G$, G_r is trivial, and for all i, $0 < i \leq r$, graph G_i is obtained from G_{i-1} by reduction of a nonempty set of nontrivial disjoint intervals into nodes.
Many problems on program development and processing are significantly simplified if programs have a regular structure and admit a representation in the form of a nested fragments of a special form. For example, structured programming consider various types of statement composition as basic ones, but all of them are intervals. When allowing any intervals to be considered as basic ones, we get the class of programs with regularizable cf-graphs.
Control flow graphs of programs that occur in practice frequently fall into the class of regularizable graphs. Exclusive use of structured flow-of-control statements, such as if-then-else, while-do, continue, and break statements, produces programs whose flow graphs are always regularizable. Even programs written using goto statements by programmers with no prior knowledge of structured program design are almost always regularizable. Moreover, any program can be transformed via splitting statements to the equivalent one with a regularizable cf-graph.
Theorem. The following properties of a cf-graph G are equivalent: (1) G is a *regularizable* graph, (2) G is a *reducible* graph, (3) G is an *arrangeable* graph, (4) G is a *collapsible* graph,(5) G is a *single-entry* graph, (6) G has no *forbidden subgraph*, (7) G has a single *dag*.

Relation — отношение.

Relation precedence — отношение предшествования.

Reliable relations of execution frequency — достоверные отношения частоты исполнения.

Removal of an edge — удаление ребра.

Removal of a set of vertices — удаление множества вершин.

For a given graph (digraph or hypergraph) G, removal of vertices in X together with all edges incident to them. The resulting [hyper-, di-]graph is denoted by $G \setminus X$ or $G - X$. If $X = \{x\}$, it is simply $G - x$.

Removal of a vertex — удаление вершины.

Removal-similar vertices — подобные по удалению вершины.

Repeatedly executed region — участок повторяемости.

Repetion-free scheme — код, свободный от повторения.

A**-reprezentation of a cf-graph** — A-представление униграфа.

See *Alt*.

Reproduction graph — граф воспроизводства.

Restrained dominating set — ограниченное доминирующее множество.

A set $D \subseteq V(G)$ is a **restrained dominating set** of G, if each vertex of $V(G) - D$ has a *neighbour* in D, as well as another neighbour in $V(G) - D$.

Restricted block duplicate graph — ограниченный блоково дублированный граф.

A **restricted block duplicate (RBD) graph** is a graph obtained by adding zero or more true twins to each vertex of a block graph B, subject to the restriction that a cut-vertex belonging to three or more blocks of B receive at most one true twin.

Restricted domination number — число ограниченного доминирования.

Let U be a subset of vertices of a graph G. The **restricted domination number** $r(G, U, \gamma)$ of U is the minimum cardinality of a dominating set of G containing U. A smallest possible dominating set of G containing all the vertices in U is called a γ_U-set. The k**-restricted domination number** of G is the smallest integer $r_k(G, \gamma)$ such that $r_k(G, U, \gamma) \leq r_k G, \gamma$ for all subsets U of $V(G)$ of cardinality k. In the case $k = 0$, the k-restricted dominaiton number is the *domination number*. When $k = 1$, the k-restricted domination number is called the **domsaturation number** of a graph and is denoted by $ds(G)$.

k**-restricted domination number** — число k-ограниченного доминирования.

See *Restricted domination number*.

Restricted unimodular chordal graph — ограниченный унимодулярный хордальный граф.

A **restricted unimodular (RU) chordal graph** is a chordal graph G such that a *vertex-clique incidence bigraph* $VK(G)$ is ∞-chorded, or equivalently 4-chorded.

k**-Restricted total domination number** — число k-ограниченного тотального доминирования.

The k**-restricted total domination number** of a graph G is the smallest integer $r_k(G, \gamma_t)$ such that, given any subset U of k vertices of G, there exists a *total dominating set* of G of cardinality at most $r_k(G, \gamma_t)$ containing U. Hence, the k-restricted total domination number of a graph G measures how many vertices are necessary to totally dominate a graph if an arbitrary set of k vertices must be included in the total dominating set. When $k = 0$, the k-restricted total domination number is the **total domination number**.

Γ**-Restricted graph** — Γ-ограниченный граф.

Restriction of a hypergraph — сужение гиперграфа.

The **restriction of a hypergraph** $\mathcal{H}$ onto $X \subset V(\mathcal{H})$ is the hypergraph $\mathcal{H}_X$ on the set X, for which $E(\mathcal{H}_X)$ is the collection of sets $E \cap X$, $E \in E(\mathcal{H})$. If $X = V(\mathcal{H}) - Y$, then we adopt the notation $\mathcal{H}_X = \mathcal{H} \setminus Y$ and $\mathcal{H}_X = \mathcal{H} - y$, if $Y = \{y\}$.

Restriction method — метод сужения задачи.

Restriction of a graph — ограничение графа.

Result — результат (оператора).

See *Large-block schema.*

Retract — ретракт.

A **retraction** f from a graph $H = (V_H, E_H)$ to a subgraph $G = (V_G, E_G)$ is a mapping $f : V_H \to V_G$ such that for every edge $(u, v) \in E_H$ $(f(u), f(v)) \in E_G$ and $f(w) = w$ for all $w \in V_G$. Then G is a **retract** of H. G is an **absolute retract** if G is a retract of any graph H containing G as an *isometric subgraph*, provided that $\chi(G) = \chi(H)$. Note that a retract G of H is necessarily an isometric subgraph of H.

Retraction — ретракция [графа].

See *Retract.*

Retreating arc — возвращающая дуга.

See *Depth of a flow graph.*

Reverse arc — обратная дуга.

For a given arc (v, w), the arc (w, v) is called the **reverse arc** of (v, w).

Reverse digraph — обратный орграф.

For a given digraph G, the graph $G_r = (V, E_r)$ is said to be **reversal** or **reverse digraph**, where

$$E_r = \{(x, y)|\ (y, x) \in E\},$$

or in other words,

$$(x, y) \in E_r \Longleftrightarrow (y, x) \in E.$$

Reverse path — обратный путь.

Rewriting rule — правило переписывания.

Ridge graph — хребтовый граф.

See *Skeleton graph.*

Right-linear grammar — праволинейная грамматика.

Right-linear language — язык праволинейный.

Right-linear tree — правостороннее дерево.

Rightmost derivation — правый вывод.

Right-sided balanced tree — правостороннее балансированное дерево.

See *Height balanced tree.*

Rigid circuit graph — циклически жесткий граф, триангулированный граф, хордальный граф.

See *Chordal graph.*

Rigid graph — жесткий граф.

A graph that has no proper *endomorphism* is called a **rigid graph**.

Rim — обод (граф).

See *Wheel.*

Ring-sum — кольцевая сумма.

The **ring-sum** of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, written as $G_1 \oplus G_2$, is the graph

$$((V_1 \cup V_2), ((E_1 \cup E_2) \setminus (E_1 \cap E_2)).$$

In other words, the edge-set of $G_1 \oplus G_2$ consists of those edges which are either in G_1 or in G_2, but not in both. It is easy to see that the operation of ring-sum is both commutative and associative.

Roman domination — римское доминирование.

A **Roman dominating function** on a graph $G = (V, E)$ is a function $f : V \to \{0, 1, 2\}$ satisfying the condition that every vertex u for which $f(u) = 0$ is adjacent to at least one vertex v for which $f(v) = 2$. The weight of a Roman dominating function is the value $f(V) = \sum_{u \in V} f(u)$. The minimum weight of a Roman dominating function on a graph G is called the **Roman domination number** of G.

Root — корень.

See *Directed tree.*

Rooted balance — корневой баланс.

Rooted graph — корневой граф.

Rooted tree — корневое дерево.

A **rooted tree** is a tree in which one of its vertices is designated as a *root.*

Rooted product — корневое произведение.

Let $G = (V, E)$ be a simple graph of order n and let $\mathcal{H} = \{H_1, \ldots, H_n\}$ be a family of rooted graphs. The **rooted product** $G(\mathcal{H})$ is the graph obtained by identifying the root of H_i with i-th vertex of G. In particular, if $\mathcal{H}$ is the family of the paths $P_{k_1}, \ldots, P_{k_n}$ with the rooted vertices of *degree* one, the corresponding graph $G(\mathcal{H})$ is called the **sunlike graph** and is denoted by $G(k_1, \ldots, k_n)$.

Rotational Cayley digraph — вращательный граф Кэли.

See *Complete rotation.*

Round forest — лес обхода.

Route — обход, маршрут.

See *Routing.*

Routing — маршрутизация

A **routing** ρ in a graph or digraph G assigns to every pair of different vertices a path (a chain) $\rho(x, y)$ from x to y. The paths $\rho(x, y)$ are called **routes**. Given a graph G, it is assumed that all communications between vertices are done through the routes of a fixed routing. Two parameters have been proposed to measure the efficiency and fault tolerance of a fixed routing in a graph or a digraph: the forwarding index and the diameter of the surviving route digraph. The **vertex-forwarding index** of a routing ρ in a graph or a digraph G, $\xi(G, \rho)$, is the maximum number of routes passing through a vertex. The **edge-** or **arc-forwarding index**, $\pi(G, \rho)$, is defined analogously.

For a given set F of faulty vertices and/or arcs, the vertices of the **surviving route digraph** are the non-faulty vertices and there is an arc between two vertices if and only if there are no faults on the route between them. Fault-tolerant routings are such that the *diameter* of the surviving route digraph is small for any set of faults of a bounded size.

S

Safe Petri net — безопасная сеть Петри.

A *Petri net* is called **safe** if it is *1-bounded.*

Safe place — безопасное место.

Safeness problem — проблема безопасности.

Satisfiability problem — задача о выполнимости.

Saturated vertex — насыщенная вершина.

See *Deficiency of a graph*. See also *Unary node*.

k**-Saturated graph** — k-насыщенный граф.

A graph G is k**-saturated**, if G is K_k-free, $k \geq 3$, and adding any new edge e to G creates a copy K_k.

H**-Saturated graph** — H-насыщенный граф.

A graph G is H**-saturated**, if G is H-free and adding any new edge e to G creates a copy of H.

Schema with node number repetition — код с дублированием номеров вершин.

Schema with distributed memory — схема с распределенной памятью.

Schema simulation — схемное моделирование.

Let $\alpha = (G_\alpha, R_\alpha, \Omega_\alpha)$ be a *large-block schema.*

A **scale** Δ of α is such a pair (Δ_1, Δ_2) that Δ_1 is a partion of G_α into disjoint fragments and Δ_2 is a partion of X_α into disjoint subsets.

A schema β **simulates** α on the scale Δ, if the following property holds. For any $I_1 \in \Omega_\alpha$ there is $I_2 \in \Omega_\beta$ such that I_1 and I_2 are equal on Σ_α; the execution and memory state sequences of α under I_1 and β under I_2, as well as information connections between the statements, are equal in the sense of the above correspondence.

It is clear that the relation of schema simulation is transitive and that there is such a standard schema α and its scale Δ that any β simulating α on Δ is not a standard schema.

There is an algorithm that, for any large-block schema α and any scale Δ of α, constructs a schema β that simulates α on Δ. But the property of precise schema simulation is not partially decidable in the class of large-block schemas.

Scheme with indirect addressing — схема с косвенной адресацией.

Scheme with separators — код с использованием ограничителей.

Scorpion — скорпион.

See *Spider*.

Search forest — лес обхода.

Second Order formula — формула второго порядка.

See *Logic for expressing of graph properties*.

Section — сечение.

2-**Section graph** — 2-секционный граф.

The 2-**section graph** $2SEC(\mathcal{H})$ of the hypergraph $\mathcal{H}$ has the vertex set V of this hypergraph, and two distinct vertices are *adjacent* if and only if they are contained in a common edge of $\mathcal{H}$.

Segment — сегмент.

When M is a cycle, for i, j with $i \leq j < i+p$, we define the **segment** $M[v_i, v_j]$ of M by $M[v_i, v_j] = (v_i, v_{i+1}, \ldots, v_j)$ (indices are to be read modulo p); when M is a path, for i, j with $1 \leq i \leq j \leq p$, we define the **segment** $M[v_i, v_j]$ of M by $M[v_i, v_j] = (v_i, v_{i+1}, \ldots, v_j)$.

Seidel switching — переключатель Зейделя.

See *Vertex switch*.

Seidel spectrum — спектр Зейделя.

The **Seidel spectrum** consists of the eigenvalues

$$\lambda_1^* \geq \lambda_2^* \geq \cdots \lambda_n^*$$

of its $(0, -1, 1)$ adjacency matrix $A^* = A^*(G)$.

$P_G^*(\lambda) = |\lambda I - A^*|$ denotes the **Seidel characteristic polynomial**.

Seidel characteristic polynomial —характеристический полином Зейделя.

See *Seidel spectrum*.

Self-centered graph — самоцентрированный граф.

A graph G is **self-centered**, if every vertex is in the *center*, that is, $C(G) = V(G)$.

Self-complementary graph — самодополнительный граф.

A graph is said to be **self-complementary**, if it is isomorphic to its *complement*. One of self-complementary graphs is the *transitive tournament*.

Self-converse digraph — самообратный граф.

See *Converse digraph*.

Selfdual tournament — самодвойственный турнир.

See *Dual tournament*.

Self-loop — самопетля, петля.

The *Loop*.

Self-modified Petri net — самомодифицируемая сеть Петри.
Self-negational signed graph — самонегативный граф.
Self-opposite directed graph — самообратный граф.
Semantic net — семантическая сеть.
Semaphore — семафор.
Semicomplete multipartite digraph — полуполный c-дольный орграф.

See *Semicomplete c-partite digraph.*

Semicomplete c-partite digraph — полуполный c-дольный орграф.

A **semicomplete c-partite digraph** is a digraph obtained from a complete c-partite graph by substituting each edge with an arc, or pair of mutually opposite arcs with the same end vertices. A **semicomplete multipartite digraph (SMD)** is a semicomplete c-partite digraph with $c \geq 2$. Special cases of SMD's are semicomplete bipartite digraphs ($c = 2$) and semicomplete digraphs ($c = n$, the number of vertices). A c-partite tournament is a semicomplete c-partite digraph with no cycles of length 2, and, analogously, a multipartite tournament (MT) is a SMD wih no cycles of length 2.

Semicomplete digraph — полуполный орграф.

If each partite set in a *semicomplete c-partite digraph* consists of a single vertex, then such a digraph is called a **semicomplete digraph**.

Semicycle — полуконтур.
Semidecidable problem — частично разрешимая задача.

See *Decision problem.*

Semieuler graph — полуэйлеров граф.
Semigirth — полуобхват.

Let G be a (di)graph with its diameter D and the minimum degree δ, and let $\pi \geq 0$ be an integer. The **π-semigirth** $\ell^{\pi} = \ell^{\pi}(G)$, $1 \leq \ell^{\pi} \leq D$, is defined as the greatest integer such that, for any two vertices u, v,

a) if $d(u, v) < \ell^{\pi}$, the shortest $u \to v$ path is unique and there are at most π paths $u \to v$ of length $d(u, v) + 1$,

b) if $d(u, v) = \ell^{pi}$, there is only one shortest $u \to v$ path.

The 0-semigirth $\ell^0 = \ell$ is simply called a **semigirth**. Observe that if G is a graph with girth g, then the semigirth $\ell = \lfloor (g-1)/2 \rfloor$.

Semigraph — полуграф.

A **semigraph** G is a pair (V, X), where V is a nonempty set whose elements are called vertices of G and X is a set of n-tuples, called edges of G, of distinct vertices for various $n \geq 2$, satisfying the following conditions.

S.G.-1. Any two edges have at most one vertex in common.

S.G.-2. Two edges $(u_1, u_2, \ldots, u_n)$ and $(v_1, v_2, \ldots, v_m)$ are considered to be equal if and only if

(i) $m = n$ and (ii) either $u_i = v_i$ for $1 \leq i \leq n$ or $u_i = v_{n-i+1}$ for $1 \leq i \leq n$.

The **adjacent graph** or a graph G_a of G is the graph with the same vertex set as that of G, and two vertices are adjacent in G_a if and only if they are adjacent in G.

The **consecutive adjacent graph** G_{ca} of G is the graph with the same vertex set as that of G, and two vertices are adjacent in G_{ca} if and only if they are consecutive adjacent vertices in G.

Semigroup of a graph — полугруппа графа.

Semihamiltonian graph — полугамильтонов граф.

Semiirreducible graph — полунесводимый граф.

Semikernel — полуядро.

A **semikernel** of a digraph D is an *independent* set of vertices such that for every $z \in V(D) \setminus S$ for which there exists a Sz-arc there also exists a zS-arc. It is introduced by Newmann-Lara (1971). See *Semikernel modulo F*.

Semikernel modulo F — полуядро по модулю F.

Let F be a set of arcs of a digraph D (i.e., $F \subseteq A(D)$), a set $S \subseteq V(D)$ is called a **semikernel of** D **modulo** F, if S is an independent set of vertices such that for every $z \in V(D) \setminus S$ for which there exists an Sz-arc of $D \setminus F$ there also exists a zS-arc in D.

Semiorder — полупорядок.

The relation P is a **semiorder** if the following conditions hold:

(1) P is irreflexive;

(2) if $(x, y) \in P$ and $(z, w) \in P$, then $(x, w) \in P$ or $(z, y) \in P$;

(3) if $(x, y) \in P$ and $(y, z) \in P$, then $(x, w) \in P$ or $(w, z) \in P$.

semiorder is an interesting subclass of interval orders.

Semipath — полупуть.

(p, q) **Semiregular graph** — (p, q) полурегулярный граф.

A graph G is called (p, q) **semiregular**, if G is *bipartite* and the degrees of vertices in each bipartite partition of the vertex set are p and q, respectively.

Semiregular group of a graph — полурегулярная группа графа.

Semisymmetric graph — полусимметричный граф.

Let G be a subgroup of the full automorphism group of a graph X. We call X G**-semisymmetric graph**, if it is regular and G acts transitively on its edge set but not on its vertex set.

Semi-Strong Perfect Graph Conjecture — гипотеза о полустрогих совершенных графах.

See P_4*-isomorphic graphs.*

Sentence — предложение.

See *Grammar.*

Sentencial form — сентенциальная форма, выводимая цепочка грамматики.

See *Grammar.*

Separable graph — разделимый граф.

k**-Separability** — k-отделимость.

Separating set — разделяющее множество, разрез.

See *Cutset.*

Separating triangle — разбивающий треугольник.

Separation-width — ширина укладки.

Given a *layout* φ of G, $V^-_\varphi(i)$, $V^+_\varphi(i)$, and $V^*_\varphi(i)$ are defined as follows:

$$V^-_\varphi(i) = \{u \in V[E_\varphi(i)] : \varphi(u) \leq i\},$$

$$V^+_\varphi(i) = \{v \in V[E_\varphi(i)] : i < \varphi(v)\},$$

$$V^*_\varphi(i) = \begin{cases} V^-_\varphi(i) \text{ if } |V^-_\varphi(i)| \leq |V^+_\varphi(i)| \\ V^+_\varphi(i) \text{ otherwise.} \end{cases}$$

Then we define the **separation-width** of G with respect to φ, denoted by $sw_\varphi(G)$, by

$$sw_\varphi(G) = \max\{|V^*_\varphi(i)| : 1 \leq i \leq |V|\}.$$

We further define the **separation-width** $sw(G)$ of G by

$$sw(G) = \min_\varphi sw_\varphi(G),$$

where minimum is taken over all possible layouts of G.

Separator — сепаратор.

1. (Separator of a graph) In a connected graph G, a **separator** S is a subset of vertices whose removal separates G into at least two connected components. S is called an (a — b) *separator* iff it disconnects the vertices a and b. An (a — b) separator is said to be a **minimal separator** iff it does not contain any other (a — b) separator. A **minimum separator** is a separator of minimum size. Clearly, every minimum separator is a minimal separator. The *(vertex) connectivity* $\kappa(G)$ is defined to be the size of a minimum separator. The connectivity of a complete graph is $|V|-1$. It doesn't have any separator.

See also *Cutset, (a,b)-cut.*

2. (Separator of a hypergraph) Let $\mathcal{H}$ be a hypergraph and X be a subset of $V(\mathcal{H})$. A proper subset X of $V(\mathcal{H})$ is a **separator** of $\mathcal{H}$, if $\langle\bar{X}\rangle$ is not connected.

Special separators are given by borders defined as follows. Let $\mathcal{C}$ be a connected induced subhypergraph of $\mathcal{H}$. The **boundary of** $\mathcal{C}$ **in** $\mathcal{H}$, denoted $\partial_H\mathcal{C}$, is the set of vertices that do not belong to $V(\mathcal{C})$ and are adjacent to some vertex of $\mathcal{C}$; the **closure of** $\mathcal{C}$ **in** $\mathcal{H}$, denoted by $[\mathcal{C}]_H$, is the subhypergraph of $\mathcal{H}$ induced by $V(\mathcal{C}) \cup \partial_H\mathcal{C}$. A **border** is a separator of $\mathcal{H}$ that is the boundary of an e-component of $\mathcal{H}$ for some edge e of $\mathcal{H}$.

A partial edge of $\mathcal{H}$ which is a separator is called a **partial-edge separator** of $\mathcal{H}$. A partial-edge separator X of $\mathcal{H}$ is a **divider**, if there exist two vertices u and v of $\mathcal{H}$ that are separated by X but not by any proper subset of X.

We say that two vertices of $\mathcal{H}$ are tightly connected in $\mathcal{H}$, if they are separated in $\mathcal{H}$ but not in any partial edge of $\mathcal{H}$. Moreover, by a **compact** of $\mathcal{H}$ we mean a set of pairwise tightly connected vertices of $\mathcal{H}$.

(a,b)**-Separator** — (a,b)-сепаратор.

Let a and b be nonadjacent vertices. A set S of vertices is a minimal (a,b)**-separator** if a and b are in different connected components of $G-S$ and there is no proper subset of S with the same property.

A minimal **separator** is a set S of vertices for which there exist nonadjacent vertices a and b such that S is a minimal (a,b)-separaror.

Sequence — маршрут.

Sequence of length n — маршрут длины n.

Sequential control structure — последовательная структура управления.

Sequential-alternative process — последовательно-альтернативный процесс.

Sequential-alternative process net — последовательно-альтернативная сеть-процесс.

k-Sequentially additive graph — k-последоваельно аддитивный граф. See *k-sequentially additive graph.*

k-Sequentially additive labeling — k-последовательно аддитивная разметка.

Given a graph G with p vertices, q edges and a positive integer k, a **k-sequentially additive labeling** of G is an assignment of distinct numbers $k, k+1, k+2, \dots, k+p+q-1$ to the $p+q$ *elements* of G sush that every edge uv of G receives the sum of the numbers assigned to the vertices u and v. A graph which admits such an assignment to its elements is called a **k-sequentially additive graph**.

Sequential-parallel control structure — последовательно-параллельная структура управления.

Sequential process — последовательный процесс.

Series-parallel graph — параллельно-последовательный граф.

Series-parallel graphs are recursively defined as:
(1) A one-vertex graph with a loop is series-paralel.
(2) Subdividing an edge of a series-parallel graph G with a new vertex gives a series-parallel graph (the series operation).
(3) Creating a parallel edge for a non-loop edge of a series-parallel graph (the parallel operation).
(4) There are no further series-parallel graphs.

Note that these graphs can have loops and multiple edges. Since **series-parallel graphs** are the graphs which contain no subgraphs homeomorphic to K_4, the *outerplanar graphs* are series-parallel graphs.

See also *Transitive series-parallel graph.*

Series-parallel poset — параллельно-последовательное чу-множество.

Let $P_1 = (V_1, <_1)$ and $P_2 = (V_2, <_2)$ be finite posets with $V_1 \cap V_2 = \emptyset$. The parallel composition $P_1 + P_2 = (V, <_+)$ of P_1, P_2 is defined by $V = V_1 \cup V_2$, and $u < v$ iff ($u, v \in V_1$ and $u <_1 v$) or ($u, v \in V_2$ and $u <_2 v$). The series composition $P_1 * P_2 = (V, <_*)$ of P_1, P_2 is defined by $V = V_1 \cup V_2$, and $u < v$ iff ($u, v \in V_1$ and $u <_1 v$) or ($u, v \in V_2$ and $u <_2 v$) or $u \in V_1$ and $v \in V_2$. $P = (V, <)$ is **series-parallel poset** if $|V| = 1$ or P is obtained by $P_1 + P_2$ or $P_1 * P_2$ of smaller series-parallel posets P_1, P_2, i.e. the class of series-parallel posets is the smallest class which contains the one-element poset and is closed under $+$ and $*$.

Server — сервер.

See *Directed graph*.

Set — множество.

3-set exact cover problem — задача о точном покрытии 3-множествами.

Set of firing sequences — множество последовательностей срабатываний.

Set of priorities — множество приоритетов.

Set of reachable markings — множество достижимых разметок.

Shell — оболочка шара, сфера.

A **shell** of radius r centered at v is defined by

$$C_r(v) = \{x \in V : d(x,v) = r\}.$$

Shortest path — кратчайший путь.

See *Length of a path*.

Shortest-path distance — дистанция кратчайшего пути.

See *Length of a path*.

Shortest spanning tree — кратчайший остов, минимальный остов, кратчайшая связывающая сеть.

Shortest Steiner's tree — наикратчайшее дерево Штейнера.

Shredder — разделитель.

A subset S of $V(G)$ is called a **shredder**, if $G - S$ consists of three or more components. A shredder of cardinality k is referred to as a k-shredder.

Sierpinski graph — граф Серпинского.

The **Sierpinski graph** $S(n,k)$ $(n,k \geq 1)$ is defined on the vertex set $\{0,1,\ldots,n\}^n$, two different vertices $u = (i_1, i_2, \ldots, i_n)$ and $v = (j_1, j_2, \ldots, j_n)$ being adjacent if and only if $u \sim v$. The relation $\sim$ is defined as follows: $u \sim v$ if there exists an $h \in \{1,2,\ldots,n\}$ such that

(i) $\forall t, t < h \Rightarrow i_t = j_t$,

(ii) $i_h \neq j_h$,

(iii) $\forall t, t > h \Rightarrow i_t = j_h \& j_t = i_h$.

Signed dominating function — знаковая доминирующая функция.

See *Dominating function*.

Signed domination number — число знакового доминирования.

See *Dominating function*.

Signed labeled graph — знаковый помеченный граф.

A graph G is said to be **signed** if each edge of G is given an odd or even label. In a signed graph G, a subset of $E(G)$ is odd (resp., even) if it contains an odd (resp. even) number of odd-labeled edges. A graph is **odd-signable**, if it can be signed so that the edge set of every chordless cycle is odd. A signed graph is **odd-signed**, if the edge set of every chordless cycle is odd. A signed graph S is **balanced**, if every cycle in S is positive.

Signed total domination — знаковое тотальное доминирование.

A function $f: V(G) \to \{-1, 1\}$ defined on the vertices of a graph G is a **signed total domination funciton (STDF)**, if the sum of its values over any open neighborhood is at least 1. An STDF f is minimal, if there does not exist an STDF $g: V(G) \to \{-1, 1\}$, $f \neq g$, for which $g(v) \leq f(v)$ for every $v \in V(G)$. The weight of an STDF is the sum of its values over all vertices. The **signed total domination number** of G is the minimum weight of an STDF of G, while the upper signed total domination number of G is the maximum weight of a minimal STDF on G.

Signed total domination function — функция знакового тотального доминирования.

See *Signed total domination.*

Signed total domination number — число знакового тотального доминирования.

See *Signed total domination.*

Sign of a graph — знак графа.

Sigraph — знаковый граф.

The same as *Signed graph.*

Similar edges — подобные ребра.

Similar vertices — подобные вершины.

Simple chain — простая цепь.

Simple circuit — простой цикл.

Simple clique polynomial — простой кликовый полином.

See *Clique polynomial.*

Simple cutset — простой разрез.

Simple cycle — простой цикл.

See *Cycle, Path.*

Simple edge — простое ребро.

An edge of multiplicity one is called **simple**.

Simple eigenvalue — простое собственное значение.

An eigenvalue is **simple**, if its multiplicity is equal to 1.

Simple elimination ordering — простое упорядочение исключения.

See *Strongly chordal graph*.

Simple graph — простой граф, обыкновенный граф.

A graph without loops and multiple edges is called **simple**.

Simple hierarchical graph — иерархический граф.

See *Hierarchical graph*.

Simple hypergraph — простой гиперграф.

A hypergraph $\mathcal{H} = \{e_1, \ldots, e_m\}$ is **simple**, if the condition $e_i \subseteq e_j$ implies $i = j$.

Simple loop — простой цикл.

Simple path — простой путь.

See *Path*.

Simple rotation — простое вращение.

Simple transition — простой переход.

Simple vertex — простая вершина.

See *Strongly chordal graph*.

Simplicial clique — симплициальная клика.

A **simplicial clique** is a clique induced by a *simplicial vertex* and all its neighbors.

Simplicial vertex — симплициальная вершина.

A vertex $v \in V$ is **simplicial** in G, if a *closed neighborhood* $N[v]$ is a *clique* in G.

Simply related paths — взаимно простые пути.

Simply sequential numbering — простая последовательная нумерация.

For a graph G with p vertices and q edges, a labeling is a **simply sequential numbering** if each of the numbers $1, 2, \ldots, p+q$ is either a vertex label or an induced edge label.

If h is the smallest vertex label in a **simply sequential numbering** (and so the numbers $1, 2, \ldots, h-1$ are induced edge labels, if $h < 1$), then h is called the **height** of the **simply sequential numbering**.

Single-entry graph — одновходовый граф.

A *control flow graph* is called **single-entry**, if it contains no *multientry zones*.

Single-entry zone — одновходовая зона.

See *Strongly connected region*.

Singular edge exchange — сингулярная реберная замена.

Singularly related graphs — сингулярно связанные графы.

Sink — выход, сток.

A **sink** of a directed graph D is a vertex $v \in V(D)$ such that $N^+(v) = \emptyset$.

Sink-tree — дерево источников.

See *Directed tree*.

Size of a graph — размер графа.

The **size of a graph** G is the number of edges in G.

Size of a directed hypergraph — размер ориентированного гиперграфа.

See *Directed hypergraph*.

Skein — моток.

Skeleton graph — остов выпуклого конуса.

The **skeleton graph** of the convex cone C is the graph G_C whose vertices are the extreme rays (a face of dimension 1) of C and there is an edge between two vertices if they are adjacent on C (two extreme rays of C are said to be adjacent on C, if they span a two-dimensional face of C).

The **ridge graph** of C is the graph G_C^* whose vertices are the facets of C and there is an edge between two nodes if they are adjacent on C. So, the ridge graph of a convex cone is the skeleton of its dual.

Skewed tree — дерево со скосом.

Skewness of a graph — искаженность графа.

Skirting cycle — крайний цикл.

See *Halin graph*.

Skolem-graceful graph — грациозный по Сколему граф.

A graph G with p vertices and q edges is **Skolem-graceful** if it admits a **Skolem-labeling** defined as follows: the vertex labels are $1, 2, \ldots, p$ and the edge labels are $1, 2, \ldots, q$.

Skolem-labeling — разметка Сколема.

See *Skolem-graceful graph*.

Slater number — число Слатера.

Let G be a connected graph with $n \geq 2$ vertices and let v be a vertex of G. The **Slater number** of a vertex v, denoted by $f(v)$, is

$$f(v) = \min\{f(v, w) : \ w \in V(G) \setminus \{v\}\},$$

where $f(v, w)$ is the number of vertices of G that are closer to v than to w minus the number of vertices of G that are closer to w than to v.

Smith graph — граф Смита.

Smith graph is a connected graph whose *index* is equal to 2.

Snark — снарк.

A **snark** is a *connected*, *bridgeless cubic* graph with *chromatic index* equal to 4.

P. G. Tait initiated the study of **snarks** in 1880, when he proved that the *four color theorem* is equivalent to the statement that no **snark** is *planar*. The first known **snark** was the *Petersen graph*, discovered in 1898. **Snarks** were so named by the American mathematician Martin Gardner in 1976, after the mysterious and elusive object of the poem The Hunting of the Snark by Lewis Carroll.

In 2001, Robertson, Sanders, Seymour, and Thomas announced a proof of W. T. Tutte's conjecture that every **snark** has the Petersen graph as a *minor*.

k**-Snark** — k-снарк.

See *Nowhere-zero k-flow*.

Solvable problem — частично разрешимая задача.

See *Decision problem*.

Solution of a digraph — решение орграфа.

See *Independent set*.

Son of a vertex — сын вершины.

See *Directed tree*.

Sorting tree — дерево сортировки.

Source — источник, начало дуги.

1. If $e = (v, w)$ is the arc of a digraph G then v is the **source**, $s(e) = v$, of e.

2. The same as *Input*.

Space complexity of an algorithm — емкостная сложность алгоритма.

Span of f — Span f.

See *Span-labeling*.

Span-labeling — Span-разметка.

An $L(j, k)$-labeling of a graph G, where $j \geq k$, is defined as a function $f : V(G) \to Z^+ \cup \{0\}$ such that if u and v are adjacent vertices in G, then $|f(u) - f(v)| \geq j$, while if u and v are vertices such that $d(u, v) = 2$, then $|f(u) - f(v)| \geq k$. The largest label used by f is the

span of f, denoted $span(f)$. The smallest span among all $L(j,k)$-labelings of G, denoted $\lambda_{j,k}(G)$, is called the **span** of G. An $L(j,k)$-labeling of G that has a span of $\lambda_{j,k}(G)$ is called a **span-labeling** of G.

t**-Spanner** — t-стягиватель.

For any real valued parameter $t \geq 1$, a spanning subgraph $S = (V, E'; w)$ with $E' \subseteq E$ is a t**-spanner** of an edge-weighted graph $G = (V, E; w)$, if

$$d_S(u,v) \leq t \cdot d_G(u,v)$$

for all $u, v \in V$. The parameter t is called a **stretch factor**.

The t-spanner is called **tree** t**-spanner**, if the subgraph S is a tree.

Obviously, for an unweighted graph, the only 1-spanner is the graph itself.

Spanning cotree — коостов.

Spanning forest — лес-каркас.

Spanning hypertree — гиперкаркас.

A **spanning hypertree** of H is an undirected *hypertree*, $T_R = (V, E_T)$, such that:

1. $E_T \subseteq E$;
2. $(T_e \cup \{h_e\}) \not\subseteq R$, $\forall e \in (E \setminus E_T)$.

Recall that R is the root set.

Spanning sequence — остовный маршрут.

Spanning subgraph — суграф.

Spanning tree — каркас, остов, остовное дерево, скелет, стягивающее дерево.

Spanning tree vector — вектор-каркас.

P_4**-Sparse graph** — P_4-разреженный граф.

The class of P_4**-sparse graphs** was introduced by Hoang (1985), as the class of graphs for which every set of five vertices induces at most one P_4 (i.e., a path of length 4). This class contains the class of P_4-reducible graphs. These two classes contain the class of *cographs*. They have practical applications in the areas such as scheduling, clustering and computational semantics. It is known that linear $\mathcal{O}(|V| + |E|)$ time algorithms are proposed for solving five optimization problems on the class of P_4-sparse graphs: maximum size clique, maximum size stable set, minimum coloring, minimum covering by cliques, and *minimum fill-in*.

Babel and Olariu (1995) introduced the class of (q,t) graphs which, for $t = q - 4$, extends the class of P_4-sparse graphs. In such a graph no set of at most q vertices is allowed to induce more than t distinct P_4's. $(4, 0)$ graphs are exactly the *cographs*.

Spectral radius — радиус спектра.

See *Characteristic polynomial of a graph*.

Spectrum of a graph — спектр графа.

Given a graph G, the **spectrum of the graph** G is the spectrum (collection of eigenvalues) of the *adjacency matrix* A_G of G. Since A_G is symmetric, the eigenvalues of G (the elements of its spectrum) are real.

For an infinite digraph, a complex number λ is a spectrum of an *operator* A, if $A - \lambda$ has no bounded inverse, and the set $\sigma(A)$ of all spectra of A is called the spectrum of A. In particular, $\pi_0(A)$ denotes the point spectrum, that is, the set of all proper values of A. If there is a sequence $\{x_n\}$ of unit vectors in a Hilbert space $\mathcal{H}$ with $\|(T-\lambda)^* x_n\| \to 0$, then λ is called an approximate proper value, and all approximate proper values form the approximate point spectrum $\pi(A)$. If $\|(T - \lambda)x_n\| \to 0$ and $\|(T - \lambda)^* x_n\| \to 0$ for some unit vectors $\{x_n\}$, then λ is called a normal approximate spectrum of A, all of which form the normal approximate (point) spectrum $\pi_n(A)$. Obviously we have

$$\pi_n(A) \subseteq \pi(A) \subseteq \sigma(A).$$

For a directed graph G, the **spectrum** $\sigma(G)$, the **point spectrum** $\pi_0(G)$, the **approximate point spectrum** $\pi(G)$ and the **normal approximate (point) spectrum** $\pi_n(G)$ are defined by

$\sigma(G) = \sigma(A(G))$, $\pi_0(G) = \pi_0(A(G))$, $\pi(G) = \pi(A(G))$ and $\pi_n(G) = \pi_n(A(G))$.

Sperner's Lemma — Лемма Шпернера.

Lemma. Let T be a triangulation of Δ_n and let χ be a coloring of the points of T by $n + 1$ colors, which satisfies the following conditions:

1. Each vertex of Δ_n is colored by a different color.
2. The points of T on a face τ of Δ_n are colored by the vertices of τ.

Then there exists a simplex in the triangulation, whose vertices receive all $n + 1$ colors.

Sperner property — свойство Шпернера.

A poset P is said to have the **Sperner property** if the maximum size of an *antichain* in P equals the size of the largest rank of P.

Spider — паук.

A **spider** is a tree having at most one vertex with its degree being greater than 2. This vertex is called the **body of a spider**. A path connecting the body and a leaf is called a **leg**. If all but, possibly, one of legs have length at most 2, then the spider is called a **scorpion**. See also *Caterpillar*.

A **wounded spider** is a tree with a single vertex of degree p, p pendant vertices, and at most $p-1$ vertices of degree 2, each of which is adjacent to a pendant vertex and the vertex of degree p, where $p \geq 1$.

Split dominating set — расщепляемое доминирующее множество.

See *Split domination number*.

Split domination number — расщепляемое доминирующее число.

A *dominating set* D of G is a **split dominating set** of G, if the induced subgraph $\langle V(G) - D\rangle_G$ is disconnected. The **split domination number** $\gamma_s(G)$ of G is the minimum cardinality of a split dominating set of G.

Split graph — расщепляемый граф, граф расщеплений.

Split graph is a graph (G, A, B) for which there exists a partition of the vertex set into a *clique* $G(A)$ and a *stable set* $G(B)$. If the clique and the stable set both have the same number of vertices k, this graph is called a k-**sun**.

The split graphs were introduced independently by Foldes and Hammer (1976 – 1977) and R.Tyshkevich (1980) (as **polar graphs**).

Split isomorphism — расщепленный изоморфизм.

Let (G, A, B) and (H, C, D) be *split graphs* and let f be an isomorphism of G onto H. f is called a **split isomorphism**, if $f(A) = C$ and $f(B) = D$.

Split sequence — расщепляемая последовательность.

Split tree — расщепляемое дерево.

See *Binary split tree*.

Splitoid — сплитоид.

Let S be the class of all *split* graphs. A **splitoid** is a *hereditary* S-*well-covered graph*. It is known that all split graphs are splitoid.

Splitting of a vertice — расщепление вершины.

For **splitting of a vertice** x of a graph G into vertices $x_1, \ldots, x_k$, one needs to remove x and replace each (x, y)-edge $(y \in V(G) - \{x\})$ by an (x, y)-edge for exactly one i, $1 \leq i \leq k$.

Splitting off — расщепление (пары рёбер).

Splitting off a pair of edges su, sv in a graph G means replacing these two edges by a new edge uv.

Square — квадрат.

See *Chordless cycle*.

Square of a graph — квадрат графа.

The **square of a graph** G^2 is the graph with a vertex set $V(G^2) = V(G)$, for which (u, v) is an edge in G^2 if and only if the *distance* $d_G(u, v)$ between u and v is 1 or 2. It is known that the **square** of every *2-connected* graph is *hamiltonian* and the **square** of a connected graph is *pseudo-hamiltonian*.

Square radical from a graph — квадратный корень из графа.

Squared graph — отквадрированный граф.

A graph G is a **squared graph**, if $G \cong H^2$ for some graph H.

SSSP problem — задача о кратчайшем пути.

This is the **single-source shortest path problem**. Given a digraph with non-negative arc weights, Dijkstra's algorithm takes $\mathcal{O}(n^2)$ time. An implementation of Dijkstra's algorithm that uses Fibonacci heaps reduced the time to $\mathcal{O}(n \log n + m)$.

k**-Stability** — k-устойчивость.

A property P defined on all graphs of order n is said to be k**-stable**, if for any graph of order n that does not satisfy P the fact that uv is not an edge of G and that $G+uv$ satisfies P implies $d_G(u)+d_G(v) < k$. Every property is $(2n-3)$-stable and every k-stable property is $(k+1)$-stable. We denote by $s(P)$ the smallest integer k such that P is k-stable and call it the **stability** of P. This number usually depends on n and is at most $2n-3$.

Stability function — функция независимости.

The function $\alpha_G : \{0,1\}^n \to N$ such that for each $x \in \{0,1\}^n$ $\alpha_G(x)$ is the *stability number* of a subgraph induced by x. It can be shown that this function can be expressed uniquely in the form

$$\sum_{t \in \Delta} a_t \prod_{i \in t} x_i,$$

where Δ is a collection of subsets of $\{1, \ldots, n\}$ and the a_t's are real coefficients. This expression is called the **polynomial expression of the stability function**.

Stability number — вершинное число независимости.

The maximum cardinality of an independent set is called a **stability number**.

The other name is **Independent number**.

f**-Stability number** — число f-стабильности.

See *f-stable set*.

Stable set — стабильное, устойчивое множество.

See *Independent set*.

f**-Stable set** — f-устойчивое множество.

A set of vertices $S \subset V(G)$ is said to be an f**-stable set**, if $d_G(u, v) \geq f(u) + f(v)$ holds for each pair of distinct vertices $u, v \in S$. If we take a constant function taking the value 1 as f, an f-stable set is an ordinary *stable set* (also called an *independent set*). The f**-stability number**, denoted by $\alpha_f(G) = \max\{|S| : S \text{ is an } f\text{-stable set}\}$.

Stable vertex set — стабильное множество вершин.

It is the same as *Stable set* and *Independent set*.

Stamen of a flower — тычинка цветка (граф).

See *Flower*.

Standard form of a net — стандартная форма сети.

Standard schemata — стандартные схемы.

They are an important subclass of **large-block schemas** which includes all schemas α such that the following properties hold. Ω_α is the set of all possible interpretations. The set operands of α consists of only nonstrong inputs and strong outputs. Every transformer has a single strong output, and every recognizer has no outputs.

Star — звезда.

1. A tree with one vertex connected to all other vertices is a **star**.

2. A **star** is a complete bipartite graph $K_{1,n}$.

3. A **star** is either a tree of order 2 or a tree of order $n \geq 3$ whose *pruned tree* S^* is a trivial tree.

Another name is *Starred graph*.

Star-chromatic number — звездное хроматическое число.

The **star-chromatic number** of a graph G (denoted $\chi^\star(G)$) is defined as

$$\chi^\star(G) = \inf\{k/d : G \text{ has a } (k, d) - \text{coloring}\}.$$

It is known that

$$\chi(G) - 1 < \chi^\star(G) < \chi(G), \text{ i.e., } \chi(G) = \lceil \chi^\star(G) \rceil,$$

where $\chi(G)$ is the *chromatic number* of G.

Thus the chromatic number of a graph is determinated by its star-chromatic number. On the other hand, two graphs with the same chromatic number could have different star-chromatic numbers. In this sense, the star-chromatic number of a graph captures its structure more precisely than the ordinary chromatic number.

Star coloring — звёздная раскраска.

See *Circular coloring of a graph.*

Star-extremal graph — звездно-экстремальный граф.

A graph G, for which the *star-chromatic number* is equal to the *fractional-chromatic number*, i.e., $\chi^\star(G) = chi_f(G)$, is a **star-extremal graph** G. It has a very interesting property, namely, if G is a **star-extremal graph** and H is an arbitrary graph, then the star-chromatic number $\chi^\star(G[H])$ of the *lexicographic product* $G[H]$ is equal to the product of $\chi^\star(G)$ and the *chromatic number* $\chi(H)$ of the graph H.

n**-Star graph** — n-звездный граф.

The n**-star graph** S_n is an undirected graph consisting of $n!$ nodes labeled with $n!$ permutations on symbols $1, 2, \ldots, n$. There is an edge between two nodes u and v in S_n if and only if there is a transposition $\pi[1, i]$, $2 \leq i \leq n$, such that $\pi[1, i](u) = v$. The n**-star graph** is an $(n-1)$-connected vertex-symmetric *Cayley graph* (with the generating set $\{\pi[1, 2], \pi[1, 3], \ldots, \pi[1, n]\}$ for the symmetric group of order n).

Starlike tree — звёздоподобный граф, звёздоподобное дерево.

A tree is called **starlike**, if it has exactly one vertex of degree greater than two.

See also *Star.*

Starred graph — звезда.

The same as *Star.*

Starred polygon — звездный многоугольник.

Start operator — оператор старт.

Start vertex — начальная вершина.

Starting node of a fragment — стартовая вершина фрагмента.

See *Fragment.*

State — состояние.

State of compound transition — состояние составного перехода.

State transition diagram — конечно-автоматная диаграмма.

State-machine Petri net — автоматная сеть Петри.

Status of a vertex — статус вершины.

The **status** $s(v)$ **of a vertex** v in G is the sum of the *distances* from v to all other vertices. Let $\chi(G) = r$, so that G is r-chromatic. Let $S = \{v_1 < v_2, \ldots, v_r\}$ be a set of r vertices having distinct colors in a *proper r-coloring* of G. The **total status** of S is defined as the sum of distances

$$\sum d(S) = \sum_{1 \le i < j \le r} d(v_i, v_j).$$

The **Chromatic status** $\sum \chi(G)$ of G is the minimum value of the total status $\sum d(S)$ of all sets S among all proper r-coloring of G.

Steiner distance — расстояние Штейнера.

See *Steiner n-center.*

Steiner n-center — n-центр Штейнера.

The **Steiner distance** of a set S of vertices in a connected graph G is the minimum number of edges in a connected subgraph containing S. The **Steiner n-eccentricity** $e_n(v)$ is the maximum Steiner distance among sets of order n containing v. The **Steiner n-center** is the set of vertices with the minimum Steiner n-eccentricity. It is easy to verify that the Steiner 2-eccentricity and 2-center correspond to the *eccentricity* and *center*, respectively.

Steiner n-eccentricity — n-эксцентриситет Штейнера.

See *Steiner n-center.*

Steiner minimal tree — минимальное дерево Штейнера.

See *Steiner's problem in Euclid plane.*

Steiner point — точка Штейнера.

See *Steiner's problem in Euclid plane.*

Steiner trade — трейд Штейнера.

See *G-trade.*

Steiner's problem in Euclid plane — евклидова задача Штейнера.

Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of n given points (called **terminal points**) in the plane with a distance function d. A minimum cost network interconnecting these terminal points is called a **Steiner minimal tree** (SMT). The cost of a network is the sum of its edge costs and the cost of an edge is the distance (measured by the distance function d) between its end points. Therefore, an SMT is always a tree network whose *leaf vertices* are some of the terminal points and whose internal vertices are either terminal points or some **Steiner points** which are introduced to reduce the network cost. The **Steiner's problem in Euclid plane** or the **Steiner tree problem** asks for an SMT for a given set of terminal points with a given distance function. The Steiner tree problem with Euclidean and rectilinear distances attracted much attention due to their applications in telecommunications and in VLSI design of printed circuits.

Steiner's problem in graphs — задача Штейнера на графах.

Stem — ствол, стебель.

See *Leaf density*.

Stochastic Petri nets — стохастические сети Петри.

See *High-level Petri nets*.

Stop operator — оператор стоп.

Stratified net formula — расслоенная формула сети.

Stretcher — носилки.

A **stretcher** is a graph whose edge set may be partitioned into two triangles and three vertex-disjoint chordless paths, each with endpoints in both triangles In a graph with no odd *hole*, every induced **stretcher** has three vertex-disjoint chordless paths of the same parity; therefore, a stretcher is called odd or even depending on the parity of the paths (odd or, respectively, even).

Strict partial order relation — отношение строгого частичного упорядочения (порядка).

String — цепочка.

A **string** (or **word**) α over an alphabet $\sum$ is a finite sequence of symbols from $\sum$ placed one after another. α consists of zero or more symbols of $\sum$, and the same symbol of $\sum$ may occur in α several times. The string consisting of zero symbols is called the **empty string**, written e. The set of all words over an alphabet $\sum$ is usually denoted by $\sum^*$ (using the **Kleene star**), and the set of all nonempty strings over an alphabet $\sum$ is usually denoted by $\sum^+$. The sets $\sum^*$ and $\sum^+$ are infinite for any $\sum$. In terms of algebra, $\sum^*$ and $\sum^+$ are a free monoid (with the identity e) and a free semigroup generated by $\sum$.

For strings α and β, the sequence $\alpha\beta$ is called the **concatenation** (or **catenation**) of α and β. The empty string is an identity with respect to concatenation: $\alpha e = e\alpha = \alpha$ holds for all strings α. Because concatenation is associate, the notion α^i, where i is a positive integer, is used in a customary sense. By definition, α^0 is the empty string e.

The **length** of a string α, denoted by $|\alpha|$, is the number of symbols α, when each letter is counted as many times as it occurs.

A string α is a **substring** (or a factor) of a string β, if there are strings γ and δ such that $\beta = \gamma\alpha\delta$. Furthermore, if $\gamma = e$ (respectively, $\delta = e$), then α is called an **initial substring**, or a **prefix**, of β (respectively, a **final substring** or a **suffix** of β). A substring α of a string β is called a **proper** substring of β, if $\alpha \neq \beta$.

Strong argument — обязательный аргумент (оператора).

See *Large-block schema.*

Strong chromatic index — сильный хроматический индекс.

The **strong chromatic index** is the minimum size of a partition of the edges of a graph into induced *matchings.*

Strong closure of a graph — сильное замыкание графа.

The **strong closure** of a graph G is the graph obtained from G by recursively joining pairs of nonadjacent vertices, whose degree sum is at least $n+1$ (n is the number of vertices of G), until no such pair remains. The **strong closure** of G is denoted by $sc(G)$. The notion of a **strong closure** is useful in the study of Hamiltonian graphs.

Strong component of a digraph — бикомпонента.

This is a maximal strongly connected subgraph.

(Strong) equivalence of schemas — (функциональная) эквивалентность схем.

See *Large-block schema.*

Strong degree of a graph — сильная степень графа.

Strong dominating set — строго доминирующее множество.

A subset D of G is a **strong (weak) dominating set** of G, if for any vertex $y \in V(G) - D$ there exists a vertex $x \in D$ adjacent to y in G and such that $deg_G(x) \geq deg_G(y)$ ($deg_G(x) \leq deg_G(y)$), respectively.

Strong domination number — число строгого доминирования.

The **strong domination number** is defined as the minimum cardinality of a strong dominating set of G.

Strong Helly property — сильное свойство Хелли.

See *Helly hypergraph.*

Strong input — обязательный вход (оператора).

See *Large-block schema.*

Strong matching — строгое паросочетание.

A **strong matching** is a matching $M = \{e_1, e_2, \ldots, e_k\}$ where no end of e_i is adjacent to an end of e_j, $1 \leq i \neq j \leq k$.

Strong orientation — сильная ориентация.

See *Orientation of a graph.*

Strong output — обязательный выход (оператора).

See *Large-block schema.*

Strong perfect graph conjecture — строгая гипотеза о совершенных графах.

See *Minimal imperfect graph, Semi-Strong Perfect Graph Conjecture.*

Strong product of graphs — сильное произведение графов.

For given graphs $G_i = (V_i, E_i)$, $i = 1, \ldots, n$, the vertex set of the product graph $G = (V, E)$ is the Cartesian product of the vertex sets of the factors G_i, i.e. $V = \{(a_1, \ldots, a_n)\}$, where $a_i \in V_i$. Two vertices $\bar{a} = (a_1, \ldots, a_n)$ and $\bar{b} = (b_1, \ldots, b_n)$ are *adjacent* in n-fold **strong product** if and only if for each i, $1 \leq i \leq n$, either (a_i, b_i) is an edge in G_i or $a_i = b_i$.

Strong result — обязательный результат (оператора).

See *Large-block schema.*

Strong support vertex — строго опорная вершина.

See *Support vertex.*

Strong unique independence graph — строго единственный граф независимости.

A graph G is a **strong unique independence graph**, if G is *bipartite* and has a unique $\beta(G)$-set. ($\beta(G)$ is the independence number).

Theorem(G. Hopkins, W. Staton). A tree is a strong unique independence tree if and only if the distance between any pair of its leaves is even.

Strong B-tree — сильное B-дерево.

Strongly chordal graph — строго хордальный граф.

A vertex v of a graph G is **simple**, if the set $\{N[u] : u \in N[v]\}$ is totally ordered by inclusion. A linear ordering $(v_1, \ldots, v_n)$ of V is a **simple elimination ordering** of G, if for all $i \in \{1, \ldots, n\}$ v_i is simple in G_i, where G_i is a graph induced by $\{v_i, \ldots, v_n\}$. A graph is **strongly chordal graph** if it admits a simple elimination ordering.

A strongly chordal graph is a *trampoline*-free chordal graph.

Strongly circuit closed graph — сильно циклически замкнутый граф, двусвязный граф.

Strongly circuit connected edges — сильно циклически связные ребра.

Strongly circuit connected vertices — сильно циклически связные вершины.

Strongly coadjoint vertices — сильно косопряженные вершины.
See *Coadjoint pair.*

Strongly *NP*-complete problem — сильно *NP*-полная задача.
See *Pseudo-polynomial algorithm.*

Strongly connected component — бикомпонента, сильная компонента, компонента сильной связности.
The relation of strong connection (see *Strongly connected vertices*) is an equivalence relation on the vertex set of a *digraph.* Strong connection partitions the vertices of this graph into subsets, each of which induces a **strongly connected component** (or **bicomponent**).
In other words, the **strongly connected components** of a directed graph G are its maximal strongly connected subgraphs. If each strongly connected component is contracted to a single vertex, the resulting graph is a directed acyclic graph called the **condensation** (or **factor-graph**) of G.

Strongly connected graph — сильно связный граф.
A digraph G is **strongly connected**, if for any pair of vertices v, w there is a path from v to w and vice versa.

Strongly connected region — зона, сильно связная область.
A nontrivial strongly connected subgraph of a *cf-graph* is called a **strongly connected region** (or **zone**).
A zone with a single entry node is called **single-entry**.
A zone is called **multientry**, if it has no less than two entry nodes.
A multientry zone S is **maximal**, if there is no such multientry zone Z that S is a proper subfragment of Z.
The following properties of maximal multientry zones hold. Maximal multientry zones are not pairwise intersected. A node p is an *initial node* of maximal multientry zone S iff p is an *entry node* of S.

Strongly connected vertices — сильно связные вершины.
Two vertices, v_1 and v_2, are said to be **strongly connected**, if there is a directed path from v_1 to v_2 and(!) a directed path from v_2 to v_1.

Strongly cyclic edge connected graph — сильно ориентированно-циклически-реберно связный граф.

Strongly cyclically closed graph — сильно ориентированно-циклически замкнутый граф.

Strongly dense m-ary tree — сильно плотное m-арное дерево.

See *r-dense tree.*

Strongly NP-hard problem — сильно NP-трудная задача.

See *Pseudo-polynomial algorithm.*

Strongly equistable graph — сильно эквистабильный граф.

See *Equistable graph.*

Strongly geodetic graph — строго геодезический граф.

Strongly non-circular grammar — сильно ациклическая грамматика.

Strongly perfect graph — строго совершенный граф.

G is a **strongly perfect graph** if each *induced subgraph* H of G has a *stable set* meeting all maximal *cliques* in H: for all $V' \subseteq V$ there is $S \subseteq V'$ stable in $G[V']$ such that for all maximal cliques C in $G[V']$ $|S \cap C| = 1$. The class of **strongly perfect graph** is the proper subclass of *perfect* graphs.

G is **very strongly perfect**, if for each induced subgraph H of G each vertex of H belongs to a stable set of H meeting all maximal cliques of H. Very strongly perfect graphs are also strongly perfect.

Strongly quasibiconnected graph — строго квазибисвязный граф.

Strongly transitive graph — сильно транзитивный граф.

Strongly unilateral digraph — строго односторонний орграф.

Strongly weak digraph — строго слабый орграф.

Structured net formula — структурная формула сети.

Subchain — подцепь.

Subchromatic number — подхроматическое число.

Subdegrees of a graph group — подстепени группы графа.

Subdivided edge — подразбитое ребро.

Subdivision graph — граф подразбиений.

A graph G' is a direct subdivision of a graph G, if G' is obtained from G by subdividing an edge of G into two edges by inserting a new vertex: there is an edge $(u, v) \in E$ with $E' = (E \setminus \{(u, v)\} \cup \{(u, x), (x, v)\}$ and $V' = V \cup \{x\}, x \notin V$ (**subdivision of an edge**). A graph G' is a **subdivision** of G, if it is obtained from G by a sequence of direct subdivisions.

Subdivision of an edge — подразбиение ребра.

See *Subdivision graph.*

Subgraph — подграф, часть графа, частичный граф.

1.(**Subgraph** in a weak sense) For a graph $G = (V, E)$ this is a graph $H = (V_H, E_H)$ with $V_H \subseteq V$ and $E_H \subseteq E$. Another name is *Part of* *a graph.*

2. (**Subgraph** in a strong sense) If $W \subseteq V$ then $G[W]$ denotes the **subgraph** induced by W, i.e. $G[W]$ is the subgraph with the vertex set W in which two vertices are *adjacent* whenever they are adjacent in G. Another name is *Induced* [*with vertices*] *subgraph.*

Subgraph derivable graph — граф, порождённый подграфами.

See *Subgraph derivation.*

Subgraph derivation — вывод подграфа (подграфовый вывод).

A graph H' is directly **subgraph derivable** from a graph G, denoted by $G \mapsto H'$, if there is a graph H, such that $G \Longrightarrow H$ and H' is a *subgraph* of H.

We say $G \overset{*}{\mapsto} H$ is a **subgraph derivation**, where $\overset{*}{\mapsto}$ is the transitive and reflexive *closure* of $\mapsto$; in that case, we also say that H is **subgraph derivable** from G.

Subgraph isomorphism problem — проблема изоморфного подграфа.

Subhypergraph — подгиперграф.

A **subhypergraph** induced by a set $A \subseteq V$ is the hypergraph $\mathcal{H}_A$ defined on A by the edge set $\mathcal{H}_A = \{e \cap A : e \in \mathcal{H}\}$.

Submodular function (of a matroid) — субмодулярная функция (матроида).

Submodular inequality — субмодулярное неравенство.

Subnet — подсеть.

Suborthogonal double cover — субортогональное двойное покрытие.

A **suborthogonal double cover** (or **SODC**) of K_n by a simple graph G is a set $S = (G_1, \ldots, G_s)$ of subgraphs of K_n, called pages, isomorphic to G such that

- every edge of K_n is contained in exactly two pages,
- $|E(G_i) \cap E(G_j)| \leq 1, \ \forall i \neq j$, i.e. two different pages have at most one edge in common.

An SODC differs from an *ODC* in the second condition, where for ODCs the edge sets of different pages are required to have exactly one edge in common.

Suborthogonal subgraph — субортогональный подграф.

See *Orthogonal subgraph.*

Substitution of a graph — подстановка графа.

1. A **substitution of a graph** G for a vertex x of a graph H (supposing G and H are vertex-disjoint) is the following operation: we remove x from H and replace each edge (x, y) $(y \in V(G) - \{x\})$

by the $|V(G)|$ edges connecting y to the vertices of G.

2. Let G and R be two graphs over Σ (a finite set of node labels). Let $C \subseteq \Sigma \times V_R$ be an embedding relation for R, and let u be a node from G. The graph $G[u/_C R]$ is obtained by replacing the node u by R with respect to C as follows:

1. Let J be the union of G (without the node u and its incident edges) and a copy of R which is disjoint with G.
2. For each edge (v, u) from G, add an edge (v, w) to the set of edges of J if and only if $(\phi(v), w) \in C$. The resulting graph is $G[u/_C R]$.

Thus $G[u/_C R]$ is obtained **substituting** u **by** R and establishing connections according to C.

Substitutional closure — подстановочное замыкание.

For a class $\mathcal{P}$ of graphs, the **substitutional closure** $\mathcal{P}^*$ consists of all graphs that can be obtained from $\mathcal{P}$ by repeated substitutions; that is, $\mathcal{P}^*$ is generated by the following rules:

(S1) $\mathcal{P} \subseteq \mathcal{P}^*$, and

(S2) if $G, H \in \mathcal{P}^*$ and $v \in V(G)$, then $G(v \to H) \in \mathcal{P}^*$.

Subword — подслово.

See *String*.

Substring — подцепочка.

See *String*.

Subtree — поддерево.

Subtree with the root r — поддерево с корнем r.

Succession relation — отношение следования.

Successive coloring — последовательная раскраска.

Successor of a vertex — преемник вершины.

See *Flow graph*.

Suffix — суффикс.

See *String*.

Sum graph — граф сумм.

A graph $G(V, E)$ is called a **sum graph**, if there is an injective labeling called **sum labeling** L from V to a set of distinct positive integers S, such that $xy \in E$ if and only if there is a vertex w in V such that $L(w) = L(x) + L(y) \in S$. In this case w is called a working vertex. Every graph can be made into a sum graph by adding some isolated vertices, if necessary.

Sum graph labeling offers a new method for defining graphs and for storing them digitally.

Sum hypergraph — суммарный гиперграф.

A hypergraph $\mathcal{H}$ is a **sum hypergraph** iff there are a finite $S \subseteq N^+$ and $\underline{d}, \bar{d} \in N^+$ such that $calH$ is isomorphic to the hypergraph $\mathcal{H}_{\underline{d},\bar{d}}(S) = (V, \mathcal{E})$, where $V = S$ and $\mathcal{E} = \{e \subseteq S : \ \underline{d}\ leq|e| \leq \bar{d} \wedge \sum_{v \in e} v \in S\}$. For an arbitrary hypergraph $\mathcal{H}$, the **sum number** $\sigma = \sigma(\mathcal{H})$ is defined as the minimum number of isolated vertices $w_1, \ldots, w_\sigma \notin V$ such that $\mathcal{H} \cup \{w_1, \ldots, w_\sigma\}$ is a sum hypergraph.

Sum labeling — суммарная разметка.

See *Sum graph.*

Sum number — суммарное число.

See *Sum hypergraph.*

Sum of graphs — сумма графов.

The **sum** $G_1 + G_2$ of graphs G_1 and G_2 is the graph with the vertex set $V(G_1) \times V(G_2)$ in which two vertices (u_1, u_2) and (v_1, v_2) are adjacent if and only if $u_1 = v_1$ and $(u_2, v_2) \in E_2$ or $u_2 = v_2$ and $(u_1, v_1) \in E_1$.

See also *Product of graphs.*

k**-Sun** — k-солнце.

See *Split graph.*

Sunlike graph — солнцеподобный граф.

See *Rooted product.*

Superconnected graph — суперсвязный граф.

A graph G is said to be **superconnected** if, for every minimum vertex cut (cut set of vertices) C of G, $G \setminus C$ has *isolated* vertices.

See also *Connected graph.*

Supercritical graph — суперкритический граф.

See *Total domination edge critical graph.*

Supereulerian graph — суперэйлеров граф.

A graph is **supereulerian** if it has a spanning eulerian subgraph (a spanning closed trail).

Supereulerian index — суперэйлеров индекс.

The **supereulerian index** is defined as

$$s(G) = \min\{\ m : L^m(G) \text{— supereulerian}\}.$$

Supergraph — надграф, накрывающий граф.

A graph G' is a **supergraph** of the graph G, if $V(G') = V(G)$ and $E(G) \subseteq E(G')$.

Supermagic graph — супермагический граф.

If G is a (p,q)-graph in which the edges are labeled by $1, 2, \ldots, q$ so that the vertex sums defined by $f^+(u) = \sum\{f(u,v) : (u,v) \in E\}$ are constant, then G is called **supermagic**.

Superoblique graph — суперкосой граф.

See *Oblique graph.*

Superperfect graph — суперсовершенный граф.

See *Interval coloring.*

Super point-connected graph — супер точечно-связный граф.

See *Connected graph.*

Superposition of graphs — суперпозиция графов.

Super edge-connected graph — суперрёберно-связный граф.

A connected graph is said to be **super edge-connected**, if every minimum *edge-cut* isolates a vertex.

Super (a,d)-edge-antimagic total graph — супер (a,d)-рёберно-антимагический тотальный граф.

See *Super (a,d)-edge-antimagic total labeling.*

Super (a,d)-edge-antimagic total labeling — супер (a,d)-рёберно-антимагическая тотальная раскраска.

The **edge-weight** of an edge u under a labeling is the sum of labels (if present) carried by that edge and the vertices x, y incident with the edge u.

An **(a,d)-edge-antimagic total labeling** is defined as a bijection from $V(G) \bigcup E(G)$ into the set $\{1, 2, \ldots, |V(G)| + |E(G)|\}$ such that the set of edge-weights of all edges in G is equal to $\{a, a+d, \ldots, a + (|E|-1)d\}$, for two integers $a > 0$ and $d \geq 0$. An (a,d)-edge-antimagic total labeling g is called **super** if $g(V(G)) = \{1, 2, ..., |V(G)|\}$ and $g(E(G)) = \{|V(G)|+1, |V(G)|+2, \ldots, |V(G)|+|E(G)|\}$.

A graph G is called **(a,d)-edge-antimagic total** or **super (a,d)-edge-antimagic total** if there exists an (a,d)-edge-antimagic total or a super (a,d)-edge-antimagic total labeling of G.

Support vertex — поддерживающая вершина.

Any vertex which is adjacent to a *pendant vertex* (*leaf*), while a **strong support vertex** is adjacent to at least two leaves.

Surviving route digraph — орграф выживаемых маршрутов.

See *Routing.*

Switch operation — операция переключения.

See *2-Switch.*

Switch equivalent graphs — графы, эквивалентные по переключению.
See *2-Switch*.

2-Switch — 2-переключение.
A **2-switch** in a simple graph G is the replacement of a pair of edges xy and zw in G by the edges yz and wx, given that ez and wx were not present in G originally.
The **switch operation** is addition or deletion of an edge whose endpoints have the same degree. Graphs H and H' are **switch equivalent**, if there is a sequence of switches transforming H to H'.

Switching — переключение.
Switching G^{σ} of G on a proper subset σ of $V(G)$ is the graph obtained from G by deleting all edges between σ and σ^c, the complement of σ in $V(G)$, and introducing new edges between σ and σ^c whenever they were nonadjacent in G.

Symbol — символ.
See *Alphabet*.

Symmetric binary tree — симметричное бинарное дерево.

Symmetric directed graph — симметричный орграф.
This is a digraph $G = (V, A)$ containing, for every arc $(v, w) \in A$, the *antiparallel* arc $(w, v) \in A$. A **pseudosymmetric digraph** is a digraph such that $deg^+(x) = deg^-(x)$ at every vertex.
See also *Balanced digraph*.

Symmetric edge — симметричное ребро.

Symmetric relation — симметричное отношение.

Symmetric traversal — симметричный обход.

Symmetrical difference of graphs — симметрическая разность графов.

Symmetrical group of a graph — симметрическая группа графа.

Synchrograph — синхрограф.

Synchronization graph — синхронизационный граф.

Syntactical diagram — синтаксическая диаграмма.

Syntax analysis — синтаксический анализ.

Syntax diagram — синтаксическая диаграмма.

Syntax tree — синтаксическое дерево.
1. The same as *Derivation tree*.
2. The same as *Abstract syntax tree*.

System of disjoint representatives — система различных представителей.

The same as *System of distinct representatives.*

System of distinct representatives — система различных представителей.

Given a hypergraph $\mathcal{H}$, a **system of distinct representatives** is a one-to-one mapping $\varrho :\ E(\mathcal{H}) \rightarrow V(\mathcal{H})$ such that $\varrho(E) \in E$ for each $E \in E(\mathcal{H})$. If no confusion can arise, we also call the range $\varrho(E(\mathcal{H}))$ a system of distinct representatives.

T

Tail of a hyperarc — конец гипердуги.
See *Directed hypergraph.*

Tail place — хвостовое место.

Target — сток, конец дуги.
1. If $e = (v, w)$ is the arc of a digraph G, then w is the **target** of e, denoted $t(e) = v$.
2. The same as *Output.*

Tensor product — тензорное произведение.
See *Product of two graphs.*

Term — терм.

Terminal alphabet — терминальный алфавит, алфавит терминальных символов, алфавит терминалов.
See *Grammar.*

Terminal edge — висячее ребро.

Terminal language — терминальный язык.

Terminal marking — заключительная разметка.

Terminal marking — терминальная разметка.

Terminal state — заключительное состояние.

Terminal symbol — терминальный символ.
See *Grammar.*

Termination of a compound transition — завершение составного перехода.

Terminal node — конечная вершина.
See *Control flow graph.*

Terminal node of a fragment — конечная вершина фрагмента.
See *Fragment.*

Terminal vertex — концевая вершина; (иногда) висячая вершина.

Term-rewriting system — система переписывания термов.

Test and decrement operator — оператор условного вычитания единицы.

Theta-graph — тэта-граф.

Thickness of a graph — толщина графа.
The **thickness** $T(G)$ of a graph G is the minimum number of planar subgraphs of G whose *union* is G. It is known that the thickness T of

a simple graph with n vertices and $|E|$ edges satisfies the following:

$$T \geq \left\lceil \frac{|E|}{3n-6} \right\rceil.$$

For a complete graph K_n

$$T \geq \left\lceil \frac{n(n-1)}{6(n-2)} \right\rceil = \left\lfloor \frac{n(n-1)+(6n-14)}{6(n-2)} \right\rfloor = \lfloor \frac{1}{6}(n+7) \rfloor.$$

Threshold graph — пороговый граф.

A graph is called **threshold**, if there is a non-negative weight function on its vertices such that each *stable* (*independent*) set of vertices has the total weight at most 1, and each non-stable set of vertices has a total weight exceeding 1.

Tightened graph — стягиваемый граф.

Tightly connected vertices — плотно связанные вершины.

See *Separator*.

Time complexity — временная сложность, вычислительная сложность.

The **time complexity**, or simply **complexity**, of an *algorithm* is the number of computational steps that it takes to transform the input data to the result of computation. Generally this is a function of the quantity of the input data, commonly called the **problem size**. For graph algorithms the problem size is determined by one or perhaps both of the variables n $(= |V|)$ and m $(= |E|)$.

For a problem size s, we denote the complexity of a graph algorithm A by $C_A(s)$, dropping the subscript A when ambiguity arise. $C_A(s)$ may vary significantly, if the algorithm A is applied to structurally different graphs but which are nevertheless of the same size. We therefore need to be more accurate in our definition. In this text we take $C_A(s)$ to mean the **worst-case complexity**. Namely, this is the maximum number, over all input sizes s, of computational steps required for execution of the algorithm A. Other definitions can also be used. For example, the **expected time-complexity** is the average, over all input sizes s, of the number of computational steps required.

The complexities of two algorithms for the same problem will in general differ. Let A_1 and A_2 be such algorithms and suppose that $C_{A_1}(n) = \frac{1}{2}n^2$ and $C_{A_2}(n) = 5n$. Then A_2 is faster than A_1 for all problem sizes $n > 10$. In fact, whatever had been the (finite and positive) coefficients of n^2 and of n in these expressions, A_2 would be faster than A_1 for all n greater than some value, say n_0. The reason, of course, is that the asymptotic growth, as the problem size tends to infinity, of n^2 is greater than that of A_1. The complexity of A_2 is said to be of lower order than that of A_1.

Given two functions F and G whose domain is the natural numbers, we say that the order of F is lower than or equal to the order of G, provided that:

$$F(n) \leq K \cdot G(n)$$

for all $n > n_0$, where K and n_0 are two positive constants. If the order of F is lower than or equal to the order of G, then we write $F = \mathcal{O}(G)$ or we say that F is $\mathcal{O}(G)$. F and G are of the same order provided that $F = \mathcal{O}(G)$ and $G = \mathcal{O}(F)$. It is occasionally convenient to write $\theta(G)$ to specify the set of all functions which are of the same order as G. Although $\theta(G)$ is defined to be a set, we conventionally write $F = \theta(G)$ to mean $F \in \theta(G)$. Illustrating these definitions, we see that $5n$ is $\mathcal{O}(\frac{1}{2}n^2)$ but $5n \neq \theta(\frac{1}{2}n^2)$, because $\frac{1}{2}n^2$ is not $\mathcal{O}(5n)$. Note also that low order terms of a function can be ignored in determining the overall order. Thus the polynomial $(3n^3 + 6n^2 + n + 6)$ is $\mathcal{O}(3n^3)$. It is obviously convenient when specifying the order of a function to describe it in terms of the simplest representative function. Thus, $(3n^3 + 6n^2)$ is $\mathcal{O}(n^3)$ and $\frac{1}{2}n^2$ is $\mathcal{O}(n^2)$.

When comparing two functions in terms of order, it is often convenient to take the following alternative definition.

Letting $\lim_{n\to\infty} F(n)/G(n) = L$, we see that:

(i) If $L =$ a finite positive constant, then $f = \theta(G)$.

(ii) If $L = 0$, then F is of lower order than G.

(iii) If $L = \infty$, then G is of lower order than F.

The complexity of an algorithm is important to the computer scientist. One reason for this is that the existence of an algorithm does not guarantee in practical terms that the problem can be solved. The algorithm may be so inefficient that, even with computation speed vastly increased over those of the present day, it would not be possible to obtain a result within a useful period of time. We need then to characterize those algorithms which are efficient enough to make their implementation useful so that they can be distinguished from those which may have to be disregarded for practical purposes. Fortunately, computer scientists are able to make use of a rather simple characterizing distinction which, for most occasions, satisfies the need. The yardstick is any $\mathcal{O}(P)$-algorithm, where P is a polynomial in the problem size, which is called an **efficient algorithm**. Many algorithms have complexities which are exponential, or even factorial, in the problem size.

Notwithstanding our earlier warnings, we call any problem for which no polynomial time algorithm is known and for which it is conjectured that no such algorithm exists, an **intractable problem**.

Timed Petri nets — временные сети Петри.

See *High-level Petri nets.*

Token — фишка.

See *Petri net.*

Tolerance graph — толерантный граф.

A graph $G = (V, E)$ is a **tolerance graph** (an **interval tolerance graph**), if there exists a finite collection $I = \{I_x : x \in V\}$ of closed intervals on a line and a set $t = \{t_x : x \in V\}$ of positive numbers satisfying $(x, y) \in E$ iff $|I_x \cap I_y| \geq \min\{t_x, t_y\}$ (where $|I|$ denotes the length of I).

The pair (I, t) is called a **tolerance representation** of G. A tolerance representation (I, t) is **bounded** if $t_x \geq |I_x|$ for all $x \in V$. G is a **bounded tolerance graph**, if G is a **tolerance graph** which admits a bounded tolerance representation.

The class of **tolerance graph** is the subclass of *weakly chordal graphs.*

ϕ**-Tolerance competition graph** — ϕ-толерантный граф конкуренции.

See *Generalized competition graphs.*

ϕ**-Tolerance competition number** — ϕ-толератное число конкуренции.

See *Generalized competition graphs.*

Tolerance representation — представление толерантности.

See *Tolerance graph.*

Topological graph — топологический граф.

Topological measures of program complexity — топологические меры сложности программ.

Topological representation of a graph — топологическое представление графа.

Topological sorting — топологическая сортировка.

S**-Topological graph** — S-топологический граф.

Toroidal graph — тороидальный граф.

A graph G is **toroidal** if it is crossing-free embeddable in the torus just as *planar graphs* in the plane.

Toroidal thickness — толщина тороидальная.

Total chromatic number — тотальное хроматическое число.

See *Total coloring.*

Total coloring — тотальная раскраска.

The **total coloring** of a graph G is a coloring of its vertices and edges in which any two adjacent or incident elements of $V(G) \cup E(G)$ are colored with different colors. The minimal number of colors required for this coloring is called the **total chromatic number** and is denoted by $\chi_t(G)$.

L**-Total coloring** — L-тотальная раскраска.

See *List total coloring.*

Total connectivity — тотальная связность.

See *Connectivity.*

Total dominating function — тотально доминирующая функция.

As a fractional generalization of the *total dominating set*, a **total dominating function** (TDF) of a graph $G = (V, E)$ is defined as a function $f : V \to [0, 1]$ such that $\sum_{u \in N(v)} f(u) \geq 1$ for each $v \in V$. (Here $N(v)$ is the open neighborhood of v.) A TDF f is minimal (MTDF) if no function $g : V \to [0, 1]$ with $g < f$ is also a TDF of G, where $g < f$ means that $g(v) \leq f(v)$ for each $v \in V$ and $f \neq g$. Obviously, an integer-valued (minimal) TDF is exactly the characteristic function of a (minimal) *total dominating set.*

Total dominating set — тотально доминирующее множество.

See *Dominating set.*

Total dominating number — число тотального доминирования.

See *Dominating set.*

Total domination edge critical graph — реберно-критический граф тотального доминирования.

A graph G is defined to be **total domination edge critical**, or simply k_t-critical, if

$$\gamma_t(G + e) < \gamma_t(G) = k$$

for any edge $e \in E(\bar{G})$.

A graph G is **supercritical**, if $\gamma_t(G+e) = \gamma_t(G)-2$ for any $e \in E(\bar{G})$, where $E(\bar{G}) \neq emptyset$.

Total domination number — число тотального доминирования.
The **total domination number** $\gamma_t(G)$ is the minimum cardinality of a total dominating set.

Total domination subdivision number — число подразбиений тотального доминирования.
The **total domination subdivision number** $sd_{\gamma_t}(G)$ is the minimum number of edges that must be subdivided (where each edge in G can be subdivided at most once) in order to increase the *total domination number*.

Total edge length of a graph — тотальная реберная длина графа.

Total graph — тотальный граф.
Given a graph $G = (V, E)$, a **total graph** is the graph $T(G) = (V \cup E, E'')$, where:

$$E'' = E \cup \{(e_1, e_2) | e_1, e_2 \in E \text{ and } e_1, e_2 \text{ are adjacent in } G\}$$

$$\cup \{(v, e) | v \in V, e \in E \text{ and } v \text{ is one of the ends of } e \text{ in } G\}.$$

Total labeling — тотальная разметка.
See *Labeling.*

Total restrained dominating set — тотальное ограниченное доминирующее множество.
For a graph $G = (V, E)$, a set $D \subseteq V(G)$ is a **total restrained dominating set**, if it is a *dominating set* and both $\langle D \rangle$ and $\langle V(G) - D \rangle$ are *isolate* free.

Total status — тотальный статус.
See *Status of a vertex.*

Total k-subdominating function — тотальная k-субдоминирущая функция.
Let $G = (V, E)$ be a simple graph. For any real-valued function $f : V \to R$, the weight of f is defined as $f(V) = \sum f(v)$, over all vertices $v \in V$. For a positive integer k, a **total k-subdominating function** (TkSF) is a function $f : V \to \{1, -1\}$ such that $f(N(v)) \geq 1$ for at least k vertices v of G. The **total k-subdomination number** $\gamma^t_{ks}(G)$ of a graph G equals the minimum weight of a TkSF on G. In the special case for $k = |V|$, γ^t_{ks} is the **signed total domination number**.

Total k-subdomination number — тотальное k-субдоминирующее число.
See *Total k-subdominating function.*

Total Z-transformation graph — тотальный Z-трансформационный граф.

See *Z-transformation graph*.

Totally adjacent vertex set — множество тотально-смежных вершин.

Totally balanced hypergraph — тотально сбалансированный гиперграф.

A *hypergraph* is **totally balanced** if every cycle of length greater than two has an edge containing at least three vertices of the cycle. It is known that a hypergraph $\mathcal{H}$ is **totally balanced** if and only if every *subhypergraph* of $\mathcal{H}$ is a *hypertree*. See also *Balanced hypergraph*.

Totally balanced matrix — тотально сбалансированная матрица.

See *Incidence matrix*.

Totally irregular graph — тотально иррегулярный граф.

See *Regular graph*.

Totally stratified net formula — тотально-расслоенная формула сети.

t-Tough graph — t-жесткий граф.

A graph is **t-tough**, if the number of components of $G \setminus S$ is at most $|S|/t$ for every cutset $S \subseteq V(G)$.

In particular, a graph G is called **1-tough**, if $\omega(G - S) \leq |S|$ for every set S of some vertices of G satisfying $\omega(G - S) > 1$, where $\omega(G - S)$ denotes the number of components of $G - S$.

Toughness of a graph — жесткость графа.

The **toughness** $t(G)$ of a graph G (where G is not a complete graph) is defined (Chvatál, 1973) by

$$t(G) = \min_{W} \frac{|W|}{c(G - W)},$$

where W is a *cutset* of G and $c(G - W)$ denotes the number of connected components of the graph $G - W$. It is well known that a hamiltonian graph has **toughness** at least 1 and *pseudo-h-hamiltonian* graph has **toughness** at least $\frac{1}{h}$.

Tournament — турнир.

An oriented complete graph, i.e. a (simple) digraph T without loops in which exactly one of (x, y) or (y, x) is an arc for every pair $x \neq y$, $x, y \in T$. The vertex v of a **tournament** T has a positive (negative) valence k, if there are k arcs from (into) v. A **tournament** T is *regular* of degree t, if the positive valence of each of its vertices is t. A **tournament** is *doubly regular with a subdegree t*, if all pairs of vertices jointly dominate precisely t vertices. A tournament T is called **almost regular**, when $\Delta = \delta$.

See also *Transitive tournament*, *Quasi-transitive tournament.*

Tournament matrix — матрица обходов.

Traceable digraph — вычерчиваемый орграф.

A digraph is said to be **traceable**, if it contains a *hamiltonian path.*

Traceable graph — вычерчиваемый граф.

A graph is **traceable**, if it contains a *spanning path.*

Trail — след, маршрут.

See *Walk.*

G**-Trade** — G-трейд.

Given a simple graph G, let T_1 and T_2 be two different decompositions of some graph H on v vertices into s edge-disjoint copies of G, with the property that the copies of G in T_1 are distinct from the copies of G in T_2, that is, $T_1 \cap T_2 = \emptyset$. Then the pair $\{T_1, T_2\}$ is a G**-trade** of **volume** s and **foundation** v denoted by $T_G(s; v)$. The copies of G in T_1 and T_2 are referred to as **blocks**. The trade is a **Steiner trade** provided that H is simple. Such a G-trade is called a **graphical trade** to distinguish it from trades based on other combinatorial objects, such as blocks design and latin squares.

Trampoline of order p — трамполин порядка p.

A **trampoline of order** p ($p \geq 3$) is a graph obtained from a p-cycle C by adding enough chords to make it *chordal*, and adding for each edge (not chord) of C a new vertex adjacent only to the two ends of that edge.

Transducer — преобразователь.

See *Large-block schema.*

Transformer — преобразователь.

See *Large-block schema.*

Transformation graph — граф преобразований.

A **transformation graph** (**TRAG**, for short) $\mathcal{G}$ is given by a (not necessarily finite) set $V_{\mathcal{G}}$ of vertices and a finite set $\Lambda_{\mathcal{G}}$ of (not necessarily total) transformations of $V_{\mathcal{G}}$. The elements of $\Lambda_{\mathcal{G}}$ specify labelled arcs in $\mathcal{G}$, as follows. For each $v \in V_{\mathcal{G}}$ and each $\lambda \in \Lambda_{\mathcal{G}}$ which is defined at v, there is an arc labelled by λ from the vertex v to the vertex $v\lambda$ (the image of v under λ).

Another name is **Data graph**.

Transition — переход.

See *Petri net*.

Transition firing — срабатывание перехода.

See *Petri net*.

Transitivable graph — транзитируемый граф.

Transitive closure of a directed graph — транзитивное замыкание орграфа.

Given a digraph $G = (V, A)$, the **transitive closure of** G is the digraph $G^+ = (V, A^+)$ such that an arc (x, y) belongs to A^+ iff there exists a path $P(x, y)$ in G.

Warshall's algorithm computes transitive closures in $\mathcal{O}(n^3)$ time.

Transitive closure of a relation — транзитивное замыкание отношения.

Transitive directed graph — транзитивный орграф.

A directed graph is **transitive directed graph** if, whenever the arcs (x, y) and (y, z) are in G, the arc (x, z) is also in G.

Transitive group of a graph — транзитивная группа графа.

Transitive orientation — транзитивная ориентация.

See *Comparability graph*.

Transitive reduction of a digraph — транзитивная редукция орграфа.

A **transitive reduction** TR of a digraph $G = (V, A)$ is a digraph $G^- = (V, A^-)$ having the minimum number of arcs and the same *transitive closure* of G. It is known that, for any *dag* G, the **transitive reduction** G^- is unique and is a subgraph (partial graph) of G.

Transitive relation — транзитивное отношение.

See *Binary relation*.

Transitive series-parallel digraph — транзитивный параллельно-последовательный орграф.

Transitive series-parallel digraphs are recursively defined as:
(1) A digraph on a single node is TSP (transitive series-parallel).
(2) If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are TSP digraphs and $V_1 \cap V_2 = \emptyset$, then
(2.1) $G_1 \parallel G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ is a TSP digraph (the parallel composition).
(2.2) $G_1 \to G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup (V_1 \times V_2))$ is a TSP digraph (the series composition).
(3) There are no further TSP digraphs.

Transitive tournament — транзитивный турнир.

A *tournament* T such that $(x, y) \in E(T)$ and $(y, z) \in E(T)$ imply $(x, y) \in E(T)$ is called **transitive**. The vertices of a **transitive tournament** have an ordering $(x_1, \ldots, x_n)$ such that $(x_i, x_j) \in E(G)$ $\leftrightarrow i < j$.

A tournament T is **quasi-transitive**, if $v_k \Rightarrow v_{k+1}$ and $v_j \Rightarrow v_i$, whenever $1 \leq k \leq n-1$ and $1 \leq i < j-1 \leq n-1$. ($u \Rightarrow v$ means that (u, v) is an arc in T.)

k**-Transitive graph** — k-транзитивный граф.

k**-Transitive group of a graph** — k-транзитивная группа графа.

Transitively orientable graph — транзитивно-ориентируемый граф.

See *Comparability graph.*

Transportation network — транспортная сеть.

A **(transportaton) network** is a finite connected digraph in which:
(a) one vertex s, with $deg^+(s) > 0$, is called the *source* of the network, and
(b) one vertex t, with $deg^-(t) > 0$, is called the *sink* of the network.
See also *Flow*, *Network.*

Transposition symmetry permutation — симметричная перестановка транспозиций.

See *Coadjoint pair.*

Transversal (of a family S**)** — трансверсаль (семейства S).

The set T is called a **transversal** of a family of sets $S = \{A_i\}$, if
1. $T \cap A_i \neq \emptyset$, for any i;
2. for any $a_j \in T$ there exists A_{i_j} such that $A_{i_j} \cap T = a_j$.

The maximum number of pairwise disjoint sets from the class S is called the **independence number** of S and denoted as $\alpha(G)$. $\tau(G)$ denotes the **transversal number** of S, that is the size of the minimum transversal of S.

See also *System of distinct representatives.*

For a *hypergraph* $\mathcal{H} = (V, \mathcal{E})$, a **transversal** is a transversal of $\mathcal{E}$.

Transversal number — трансверсальное число.

See *Transversal.*

Transversal set — вершинное покрытие.

Transversal set of a hypergraph — трансверсальное множество гиперграфа.

Trap — ловушка.

Trapezoid graph — трапециедальный граф.

The **trapezoid graph** is the intersection graph of a collection of trapezoids with corner points lying on two parallel lines. Note that **trapezoid graphs** are *co-comparability graphs* and hence they are *perfect graphs*. The *coloring problems* on **trapezoid graphs**, *Hamiltonian cycle problem* and others are solvable in a polynomial time.

Traveling salesman problem — задача коммивояжера.

Traveling tourist problem — задача о туристе.

Given a graph $G = (V, E)$, find the shortest walk visiting a subset of vertices, such that each vertex is either visited, or has at least one of its neighbors visited. The vertices of the graph correspond to monuments the tourist would like to see, and an edge between two vertices denotes visibility of one monument from another. The shortest such walk would guarantee that the tourist sees all monuments of interest.

Traversal of a graph — обход графа.

Tree — дерево.

Tree is a special type of a graph, which is connected (i.e., every two vertices are connected by a *chain*) and does not contain any *cycle*. The vertices in a tree usually are called *nodes* or *points*. A tree with one vertex is called **trivial, degenerate** or **empty**.

Trees are a typical example for recursive definitions of graphs:

(1) A one-vertex graph is a tree.

(2) If $T = (V, E)$ is a tree and $x \notin V, y \in V$, then $T' = (V \cup \{x\}, E \cup \{(x, y)\})$ is also a tree.

(3) There are no other trees.

It may also be defined as a connected graph such that removing any edge disconnects it; or as a *circuit-free* graph in which the introduction of any new edge will produce a circuit. A tree on n points has exactly $n-1$ edges, and it always has at least two points of *degree* 1, provided $|V(G)| \geq 2$.

See also *Forest*.

B**-Tree** — B-дерево.

BB**-Tree** — BB-дерево, балансированное по весу дерево.

H**-Tree** — H-дерево, дерево соседства.

HB**-Tree** — HB-дерево, дерево братства.

See *Brother tree*.

HS**-Tree** — HS-дерево.

I**-Tree** — I-дерево.

k**-Tree** — k-дерево.

1. k**-Tree** is a graph which can be recursively defined as follows. A *clique* with $k+1$ vertices is a k-tree, and a k-tree with $n+1$ vertices can be obtained from k-tree with n vertices by making a new vertex *adjacent* to exactly all vertices of a k-clique. Subgraphs of k-trees are called *partial* k*-trees*. If a partial k-tree G is a subgraph of a k-tree H, we call H a k-tree embedding for G.

The minimum value k for which a graph is a subgraph of a k-tree is called a **treewidth** of a graph. It is not difficult to see that k-trees are partial l-trees for every $l \geq k$. Hence, every graph of treewidth k is a partial l-tree for every $l \geq k$, i.e., the class of partial k-trees is exactly the class of graphs of treewidth at most k.

It is clear that every graph $G = (V, E)$ with $|V| = n$ is a partial n-tree. The problem to determine the smallest k such that G is a partial k-tree, however, is NP-complete.

2. k**-Tree** is a spanning tree in which every vertex has degree at most k.

k**-Tree with small height** — k-дерево малой высоты.

kB**-Tree** — kB-дерево, многомерное B-дерево.

$K - d$**-Tree** — многомерное дерево поиска, $K - d$-дерево.

See *Multidimensional search tree.*

1**-Tree** — 1-дерево, унициклическое дерево.

$2 - 3$**-Tree** — $2 - 3$-дерево.

Tree arc — древесная дуга.

See *Basic numberings.*

Tree automaton — автомат над деревьями.

A **tree automaton** over an alphabet Σ_k is a quadruple (S, S_0, S_A, f), where S is a finite set of states; $S_0 \in S$ is the initial state; $S_A \subseteq S$ is the set of accepting states; and $f: S \times S \times \Sigma_k \to S$ is the transition function. Often the alphabet Σ_k is the set of graphs on $k+1$ or fewer (labeled) vertices.

Tree-decomposition — декомпозиция дерева.

See *Treewidth of a graph.*

Tree dominating set — древесное доминирующее множество.

A *dominating set* S is called a **connected (acyclic) dominating set** if the induced subgraph $\langle S \rangle$ is connected (acyclic). The **connected (acyclic) domination number** is the minimum cardinality taken over all minimal connected (acyclic) dominating sets of G.

If $\langle S\rangle$ is both connected and acyclic, then $\langle S\rangle$ is a tree. A dominating set S is called a **tree dominating set**, if the induced subgraph $\langle S\rangle$ is a tree. The **tree domination number** $\gamma_{tr}(G)$ of G is the minimum cardinality taken over all minimal tree dominating sets of G.

Tree domination number — число древесного доминирования.

See *Tree dominating set.*

Tree grammar — древовидная грамматика.

Tree graph — граф каркасов.

Tree language — древовидный язык.

Tree model — древесная модель.

For a given graph $G = (V, E)$, a **tree model** of G is a pair $(T, \mathcal{T})$, where T is a tree, and $\mathcal{T}$ is a set of subtrees of T, $\mathcal{T} = \{T_v : v \in V\}$, such that $T_u \cap T_v \neq \emptyset$ iff $(u, v) \in E$. It is well known that a graph has a tree model iff it is *chordal.*

A tree model for a chordal graph G is called a **clique model**, if the node set of T is the set Cl of cliques in G and $c \in T_v$ is equivalent to $v \in c$ for all $c \in Cl$ and $v \in V$. It is known that every chordal graph has a clique model.

See also *Tree-decomposition.*

Tree packing — укладка дерева.

Tree polynom of a graph — многочлен деревьев графа.

Tree t-spanner — древесный t-спаннер.

See *t-Spanner.*

Tree symmetry number — число симметрий дерева.

Tree traversal inorder — внутренний порядок обхода дерева, инфиксный порядок.

Treewidth of a graph — древесная ширина дерева.

The minimum value k for which a graph is a subgraph of a *k-tree.* One can also define the **treewidth of a graph** by a concept called the tree-decomposition of a graph.

A **tree-decomposition** of a graph $G = (V, E)$ is a pair $D = (S, T)$, where $S = \{X_i | i \in I\}$ is a collection of subsets of vertices of G and $T = (I, F)$ a tree, with one node for each subset of S, such that the following three conditions are satisfied:

(1) $\cup_{i \in I} X_i = V$,

(2) for all edges $(v, w) \in E$, there is a subset $X_i \in S$ such that both v and w are contained in X_i,

(3) for each vertex x, the set of nodes $\{i|x \in X_i\}$ forms a subtree of T.

The **width** of a tree-decomposition $(\{X_i|i \in I\}, T = (I, F))$ is $\max_{i \in I}(|X_i| - 1)$. The **treewidth of a graph** G equals the minimum width over all tree-decompositions of G.

TREEWIDTH problem — проблема ширины дерева.

See *Triangulation of a graph.*

Tree-perfect graph — дерево-совершенный граф.

The class of **tree-perfect graphs** contains all trees and their complements, and graphs without P_4 (or *cographs*).

Triad — триада.

A **triad** is a set of three edges incident to a vertex of degree 3.

Triangle — треугольник.

See *Chordless cycle.*

Triangle-free graph — граф без треугольников.

Triangular graph — триангулированный граф.

See *Triangular vertex.*

Triangular vertex — триангулированная вершина.

A vertex u is a **triangular vertex**, if every vertex in the open neighborhood $N(u)$ is in a triangle with u. Stated equivalently, a vertex is triangular, if the induced subgraph $[G(N(u)]$ contains no isolated vertices. Notice if a vertex u is triangular, then $deg(u) \geq 2$. We say that a graph G is **triangular**, if it contains at least one triangular vertex, and is **completely triangular**, if every vertex in G is triangular.

Triangulated graph — триангулированный граф.

See *Chordal graph.*

Triangulated-perfect graph — триангулировано-совершенный граф.

Triangulated triangle — триангулированный треугольник.

The **triangulated triangle** T_l is the graph whose vertices are the triples of nonnegative integers summing to l, with an edge connecting two triples, if they agree in one coordinate and differ by 1 in other two coordinates.

Triangulation of a circuit — триангуляция цикла.

This is a graph consisting of this circuit C and $n - 3$ non-crossing "interior" diagonals (n is the length of C).

Triangulation of a graph — триангуляция графа.

Given a graph G, **triangulation** of G is a graph H with the same set of vertices such that G is a subgraph of H and such that H is triangulated. One says that G is triangulated into H.

There are two problems which have drawn much attention because of a large number of applications: MINIMUM FILL-IN problem and TREEWIDTH problem. The first problem is to triangulate a graph so that the number of added edges is minimum and the second one is to triangulate a graph so that the maximum *clique size* in the triangulated graph is minimum. These problems are both *NP*-hard.

See also *Minimal triangulation, Planar triangulation.*

Triconnected graph — трисвязный граф.

See *k-connected graph.*

Trie — префексное дерево, нагруженное дерево.

A **trie** is an ordered tree data structure that is used to store an associative array, where the keys are usually strings. Unlike a binary search tree, no node in the tree stores the key associated with that node; instead, its position in the tree shows what key it is associated with. All the descendants of a node have a common prefix of the string associated with that node, and the root is associated with the empty string. Values are normally not associated with every node, only with leaves and some inner nodes that correspond to keys of interest.

Another name is **Prefix tree**.

Trivial graph — тривиальный граф.

A graph with one vertex is called **trivial**.

Trivial interval — тривиальный интервал.

See *Critical tournament.*

Trivial deadend — тривиальный тупик.

Trivial deadlok — тривиальный тупик.

Trivial tree — тривиальное дерево.

See *Tree, Degenerate tree, Empty tree.*

True dependence — истинная зависимость, информационная связь.

See *Data dependence.*

True twins — истинные близнецы.

Two vertices of a graph are called **true twins**, when they are adjacent and every other vertex is adjacent to both or to none of them.

h**-tuple domination** — n-кратное доминирование.

See *Double dominating set.*

Turing machine — машина Тьюринга.
See *Model of computation.*

Tournament — турнир.
See *Oriented graph.*

Tutte polynomial — полином Татта.
If G has an empty edge set, then we set the **Tutte polynomial** $t(G; x, y)$ or $t(G)$ of G to be 1. Otherwise we have for any $e \in E(G)$
(R1) $t(G) = t(G \setminus e) + t(G/e)$, if e is not a *loop* or a *bridge*,
(R2) $t(G) = xt(G \setminus e)$, if e is a bridge,
(R3) $t(G) = yt(G \setminus e)$, if e is a loop.
Here (G/e) is the *contraction* of the edge e.

Two edge merging — слияние двух ребер.

Two-terminal DAG — двухполюсный бесконтурный орграф.
A **two-terminal DAG** (st-dag) G is a directed graph without any cycle, having a unique *source* s and a unique *target* t. This implies that an st-dag is *weakly connected*, namely, there is a path from s to any vertex and from any vertex to t.

Two-way infinite path — двусторонне-конечный маршрут.
See *Ray.*

Two-way infinite sequence — двусторонне-бесконечный маршрут.

Two-way pushdown automaton — двусторонний магазинный автомат.

Type-0 grammar — грамматика типа 0.
See *Chomsky hierarchy.*

Type-1 grammar — грамматика типа 1.
See *Chomsky hierarchy.*

Type-2 grammar — грамматика типа 2.
See *Chomsky hierarchy.*

Type-3 grammar — грамматика типа 3.
See *Chomsky hierarchy.*

U

Ultracenter — ультрацентр.

The subgraph of the *center* $C(G)$ of G induced by the vertices v with the central distance $c(v) = m$ is called the **ultracenter** of G, which we denote by $UC(G)$. The number m is referred to as the **ultraradius** of G which we denote by $urad(G)$. In a certain sense, the vertices of $UC(G)$ are the "most central" vertices of G.

See also *Central fringe*.

Ultraradius — ультрарадиус.

See *Ultracenter*.

Unary vertex — унарная вершина.

For a given m-ary tree, a node with only one son is called an **unary node**. A node with two sons is called **binary**; a node which has the maximal allowable number m sons or which is a *leaf* is said to be **saturated**, otherwise it is said to be **unsaturated**.

n**-Unavoidable graph** — n-неизбежный граф.

A digraph is said to be n**-unavoidable**, if every *tournament* of order n contains it as a subgraph.

Unbalanced tree — несбалансированное дерево.

Unbounded face — бесконечная грань плоского графа, внешняя грань.

Unbounded Petri net — неограниченная сеть Петри.

Unbounded place — неограниченное место.

Unconnected directed graph — несвязный орграф.

Undecidable problem — (алгоритмически) неразрешимая задача.

See *Decision problem*.

Undensity — неплотность графа, число независимости, число внутренней устойчивости.

Underlying graph — важный граф, основной граф.

1. See *Directed graph*.

2. See *Hierarchical graph*.

Underlying hyperedge tree — важное гиперрёберное дерево.

See *Hypertree*.

Underlying vertex tree — важное вершинное дерево.

See *Hypertree*.

Undirected graph — неориентированный граф.

See *Graph*.

Undirected hyperpath — неориентированный гиперпуть.

See *Directed hyperpath*.

Unfold process net — развернутая сеть-процесс.

Unicursal graph — уникурсальный граф, эйлеров граф.

The same as *Eulerian graph*.

Unicyclic graph — одноциклический граф, унициклический граф.

A connected graph with n vertices and n edges. Notice that a **unicyclic graph** has a *treewidth* at most 2.

Unification problem — задача унификации.

Uniform central graph — однородно-центральный граф.

A **uniform central graph** is a graph for which every central vertex has the same set of *eccentric* vertices.

Uniform star-factor — униформный стар-фактор.

A **star-factor** of a graph G is a *spanning subgraph* of G each component of which is a nontrivial *star*.

Uniform hypergraph — униформный гиперграф, однородный гиперграф.

A hypergraph $\mathcal{H}$ is a **uniform hypergraph** if $\min\{|e|;\ e \in \mathcal{E}\} = \max\{|e| :\ e \in \mathcal{E}\}$ for all hyperedges $e \in \mathcal{E}$.

h**-Uniform hypergraph** — h-униформный гиперграф.

A hypergraph $\mathcal{H}$, where $|e| = h$ for all edges of the hypergraph is called h**-uniform**.

Uniform inflation — униформная инфляция.

See *Inflation*.

Unigraph — униграф.

A graph G is called **unigraph** if G is determined by its *degree sequence* up to *isomorphism*, i.e. if a graph H has the same degree sequence as G, then H is isomorphic to G.

Unigraphical (degree) sequence — униграфическая (степенная) последовательность.

Unilateral connectivity — односторонняя связность.

Unilaterally connected component — односторонняя компонента.

Unilaterally connected digraph — односторонне-связный граф, односторонний граф.

Union of graphs — объединение графов.

The **union** of two graphs (not necessarily disjoint) G and H, denoted by $G \cup H$, is the graph with the point set $V(G \cup H) = V(G) \cup V(H)$ and the edge set $E(G \cup H) = E(G) \cup E(H)$.

Unique eccentric point graph — граф с единственной эксцентрической точкой.

See *Eccentric sequence.*

Uniquely coloured graph — однозначно раскрашиваемый граф.

Uniquely pancyclic graph — уникально панциклический граф.

A graph G on n vertices is said to be a **uniquely pancyclic graph**, abbreviated UPC, if it contains exactly one cycle of every length from 3 to n. It is known that there exist only 7 graphs with less than 57 vertices.

Uniquely transitively orientable graph — единственно транзитивно ориентируемый граф.

Unit interval graph — единичный интервальный граф.

Unitary graph — унитарный граф.

k**-Unitransitive graph** — k-унитранзитивный граф.

Universal graph — универсальный граф.

Among a family of graphs $\mathcal{H}$, a graph G is called **universal**, if any graph in $\mathcal{H}$ is isomorphic to an induced subgraph of G, and is called w**-universal**, if any graph in $\mathcal{H}$ is isomorphic to a subgraph of G.

w**-Universal graph** — w-универсальный граф.

See *Universal graph.*

Unordered labeled tree — неупорядоченное помеченое дерево.

See *Labeled tree.*

Unrestricted grammar — грамматика без ограничений.

The same as *Grammar.*

Unsaturated vertex — свободная вершина.

1. See *Unary node.*

2. A vertex v is **unsaturated** by a *matching* M, if there is no edge of M incident with v. A matching M is called **1-factor** (or a *perfect matching*), if there is no vertex of the graph unsaturated by M.

The **deficiency** $def(G)$ of G is the number of vertices unsaturated by a maximum matching of G. Observe that $def(G) = |V| - 2|M|$ for any maximum matching M in G.

UPC-graphs — уникальный панциклический граф.

See *Uniquely pancyclic graph.*

Upper n**-domination number** — верхнее число n-доминирования.

See *n-Dominating set.*

Upper-level transition — переход верхнего уровня.

V

Valency of a vertex — валентность вершины, степень вершины.

See *Degree of a vertex.*

Valuation — оценивание.

See *Labeling.*

Value of a cut — величина разреза, пропускная способность разреза.

Value of a flow — величина потока.

Value of a schema under interpretation — значение схемы при интерпретации.

Let $\alpha = (G_\alpha, R_\alpha, \Omega_\alpha)$ be a *large-block schema.* For any $I \in \Omega_\alpha$ the schema α can be run. The run consists in the execution instructions that results in transformations of the memory state. Every **memory state** W defines for any variable $x \in X_\alpha$ an element $W(x) \in D_I$ called the value of x at W.

Let Φ be a term and W be a memory state. The value of Φ at W, denoted by $W(\Phi)$, is defined by the following rules:

if Φ is constant, then $W(\Phi) = I(\Phi)$;

if Φ is an operand, then $W(\Phi) = W(R_\alpha(\Phi))$;

if $\Phi = f(\Phi_1, \ldots, \Phi_n$, then $W(\Phi) = I(f)(W(\Phi_1), \ldots, W(\Phi_n))$.

The run of α under I starts with the START instruction whose execution defines the memory state W_1, such that $W_1(x) = I(x)$ for any $x \in X_\alpha$, proceeds sequentially by executing the instructions in the order they occur in some path (called an **execution sequence**) through G_α, terminates abnormally with such S_i that $W_i(P) =$ **false** for the predicate term P of S_i and the current memory state W_i, and terminates normally with a STOP instruction. Only in the former case the **value of** α **under** I, denoted $val(\alpha, I)$, is defined; it is the tuple of current values of the arguments of the STOP instruction.

Let START, $S_1, S_2, \ldots, S_i, \ldots$ and $W_1, W_2, \ldots, W_i, \ldots$ be execution and memory sequences of α under I. The execution of S_i at W_i, in a general case, consists in processing the current values of its arguments, replacing some parts of the current values of its results and defining the next executed instruction (if S_i is a recognizer with a case term Φ and $W_i(\Phi) = r$, S_{i+1} is the r-th successor of S_i). Let an output d of $S - i$ have a variable x and a data term Φ. If d is a strong output, then the newly computed value $W_i(\Phi)$ is assigned to x. If d is a nonstrong output with an access term $g(\Phi_1, \ldots, \Phi_n, \Phi, d)$, then a (possible empty) part of the current value $W_i(x)$ of x is replaced with $W_i(\Phi)$ in such a way that $W_{i+1} = W_i(g(\Phi_1, \ldots, \Phi_n, \Phi, d))$; the interpretation of g and the values $W_i(g(\Phi_1), \ldots, W_i(\Phi_n)$ specify the part of x which is replaced with $W_i(\Phi)$. Thus, the execution $S-i$ can use the value assigned to x by S_j, $j < i$ (i.e. there is an **information flow** between S_j and S_i through x) only if there is no such S_k, $j < k < i$, that either x is a strong result of S_k or S_k redefines the part of $W_k(x)$ which is assigned to x by S_j.

Variable vertex — переменная вершина.

Variance of a graph — дисперсия графа.

Vector matroid — матроид векторный.

Consider an $r \times n$ matrix A over a field F with its columns labeled by $\{1, 2, \ldots, n\}$. Define E as the set of column labels and $\mathcal{I}$ as subsets of column labels that correspond to linearly independent sets of columns in the vector space $V(r, F)$. Then $\mathcal{I}$ satisfies the three postulates and the resulting *matroid*, denoted by $M[A]$, is called the **vector matroid** of A.

P=NP problem, P versus NP problem — P=NP проблема.

See *Complexity theory*.

Vertex (of a graph) — вершина (графа).

See *Graph*.

Vertex-antimagic total labeling — вершинно-антимагическая тотальная разметка.

A bijection $\lambda: V \cup E \to \{1, 2, \ldots, ||V| + |E|\}$ is called a **vertex-antimagic total labeling** of $G = (V, E)$, if the *weights* of vertices $wt(x)$, $x \in V$, are distinct. A bijection $\lambda: V \cup E \to \{1, 2, \ldots, ||V| + |E|\}$ is called an (a, d)**-vertex-antimagic total labeling** of $G = (V, E)$ if the set of vertex *weights* $W = \{wt(x)| \ x \in V\} = \{a, a + d, \ldots, a + (|V| - 1)d\}$ for some integers a and d.

See also *Magic labeling*.

Vertex-arboricity — вершинная древесность.

Vertex of attachment — соединяющая вершина.

Vertex clique cover — покрытие вершин кликами.

A **vertex clique cover** is a collection of cliques that covers all vertices of G.

The minimum number of cliques in a vertex clique cover is called the **vertex clique cover number** and denoted by $\theta(G)$.

Vertex clique cover number — число вершинного покрытия кликами.
See *Vertex clique cover*.

Vertex-clique incidence bigraph — двудольный граф вершины-клики.
The **vertex-clique incidence bigraph** of a graph G is a *bigraph* $VK(G)$ with the vertices of G on one side and the *cliques* of G on the other side, such that a vertex v of G is adjacent to a clique K of G in $VK(G)$ if and only if v is a member of K in G.

Vertex coloring — раскраска вершин, вершинная раскраска.
See *Coloring*.

Vertex connectivity — вершинная связность.
For an arbitrary graph G, we define its **vertex-connectivity** or simply **connectivity**, written $K_v(G)$ or simply K_v, to be the minimum number of vertices whose removal will disconnect G. Also we say that G is h-**connected** for any positive integer h satisfying $h \leq K_v(G)$. Any subset of vertices whose removal will disconnect G is called a **vertex-cut**.
See also *Edge connectivity*.

Vertex-connectivity number — число вершинной связности.

Vertex cover, vertex covering — вершинное покрытие.
A subset $V' \subseteq V$ of a graph $G = (V, E)$ such that for all edges $e = (u, v) \in E$ we have $u \in V'$ or $v \in V'$.
See also *Vertex clique cover*.

Vertex-cover polynomial — многочлен вершинных покрытий.
Let $\mathcal{CV}(G, r)$ be the set of r-vertex covers in G, and $cv(G, r) = |\mathcal{CV}(G, r)|$. We define the following generating function:

$$\Psi(G, \tau) = \sum_{v=0}^{v=|V(G)|} cv(G, r)\tau^r.$$

It is natural to call $\Psi(G, \tau)$ the **vertex-cover polynomial** of G.

Vertex covering number — число вершинного покрытия.
The **vertex covering number** is the minimum cardinality of a *vertex cover* in G, denoted by $\tau(G)$.

Vertex covering problem — задача о вершинном покрытии.

Vertex critical graph — вершинно-критический граф.

Vertex cut — вершинное сечение.
A set S of vertices of a graph G is called a **vertex cut** of G, if $G \setminus S$ has more connected components than G. See also *Vertex connectivity*.

Vertex disjoint graphs — вершинно непересекающиеся графы.

Vertex-edge incidence matching — вершинно-реберное инцидентное паросочетание.

Vertex-edge incidence matrix — матрица инциденций.
The same as *Incidence matrix*.

Vertex-forwarding index — See *Routing*.

Vertex of a hypergraph — вершина гиперграфа.
See *Hypergraph*.

Vertex incidence matrix — матрица смежности.

Vertex incident to an edge — вершина, инцидентная ребру.

Vertex involving — втягивание вершины, слияние двух вершин.

Vertex kernel — вершинное ядро.

Vertex-labeling — разметка вершин.
See *Labeling*.

Vertex level — уровень вершины.

Vertex linear arboricity — вершинно линейная древесность.
The **vertex linear arboricity** $vla(G)$ of a graph G is the minimum number of subsets into which the vertex set $V(G)$ can be partitioned so that each subset induces a subgraph whose *connected components* are paths.

Vertex-magic graph — вершинно-магический граф.
A graph G is **vertex-magic** if a *vertex-magic total labelling* of G exists.

Vertex-magic labeling — вершинно-магическая разметка.
See *Magic labeling*.

Vertex-magic total labeling — вершинно-магическая тотальная разметка.
A one-to-one map λ from $E \cup V$ onto integers $\{1, 2, \ldots, e + v\}$ is a **vertex-magic total labeling**, if there is a constant k such that for every vertex x,

$$\lambda(x) + \sum \lambda(xy) = k,$$

where the sum is over all vertices y adjacent to x. Let us call the sum of labels at the vertex x the weight of the vertex; we require $wt(x) = k$ for all x. The constant k is called the magic constant for λ. The edge labels are all distinct.

Vertex minimal dominating graph — вершинно минимальный доминирующий граф.
A **vertex minimal dominating graph** $M_vD(G)$ was introduced as the graph having $V(M_v(D(G)) = V(G) \cup S(G)$, where $S(G)$ is the

set of all *minimal dominating sets* of G, and two vertices u and v are adjacent, if they are adjacent in G or $v = D$ is a minimal dominating set containing u.

Vertex pancyclic graph — вершшинно-панциклический граф.

See $[a, b]$-*Vertex pancyclic.*

Vertex path cover — вершинное путевое покрытие.

Let $\mathcal{P} = \{P_1, \ldots, P_k\}$ be a set of paths in a digraph D. $\mathcal{P}$ is a **vertex path cover** of D iff $\{V(P_1), \ldots, V(P_k)\}$ is a partition of $V(D)$.

$$pn_v(D) = \min\{|\mathcal{P}| : \mathcal{P} \text{ is a vertex path cover of } D\}$$

is the **vertex path number** of D.

Vertex path number — вершинное число путей.

See *Vertex path cover.*

(a, b)**-Vertex pancyclic graph** — (a, b)-вершинно панциклический граф.

Let a, b be integers and $a \leq i \leq b$. G is called (a, b)**-vertex pancyclic**, if for any $u \in V$ there exists a cycle containing u with i vertices. In particular, G is **vertex pancyclic**, if $a = 3$ and $b = |V|$.

Vertex t-ranking — вершинное t-ранжирование.

Let $G = (V, E)$ be a graph and let t be a positive integer. A **vertex t-ranking**, called **ranking** for short if there is no ambiguity, is a *coloring* $c : V \to \{1, \ldots, t\}$ such that, for every pair of vertices x and y with $c(x) = c(y)$ and for every path between x and y, there is a vertex z on this path with $c(z) > c(x)$. The **vertex ranking number** of G, $\chi_r(G)$, is the smallest t for which the graph G admits a t-ranking.

See also *Edge t-ranking.*

Vertex t-ranking number — число вершинного t-ранжирования.

See *Vertex t-ranking.*

Vertex regular graph — вершинно-регулярный граф.

Let G be a subgroup of the *full automorphism group* of a graph X. We call X a G**-vertex regular graph**, if the action of G on $V(X)$ is regular. When $G = Aut(X)$, we remove the prefix "$G-$" and call X **vertex regular**.

G**-Vertex regular graph** — G-вершинно-регулярный граф.

See *Vertex regular graph.*

Vertex separator — вершинный сепаратор.

Given a graph $G = (V, E)$, a subset $S \subset V$ is called a **vertex separator** for nonadjacent vertices a and b in $V \setminus S$, if a and b

are in different *connected components* of $G[V \setminus S]$. If S is a **v.s.** for a and b but no proper subset of S separates a and b in this way, then S is called a **minimal vertex separator for** a **and** b. A subset $S \subseteq V$ is called a **vertex separator**, if there exists a pair of nonadjacent vertices for which S is a minimal vertex separator.

Vertex space — пространство вершин.

The **vertex space** $\mathcal{V}(G)$ is the power set of the vertices $V(G)$ viewed as a vector space over F_2.

Vertex splitting — расщепление вершины.

A **vertex splitting** of a digraph G is achieved by replacing a vertex v_0 of G by two new vertices v_i and v_j, replacing each arc $v_0 \to v_x$ in G by $v_j \to v_x$, replacing each arc $v_x \to v_0$ in G by $v_x \to v_i$, and adding the arc $v_i \to v_j$.

Vertex switch — переключатель вершин.

A **vertex switch** of a simple, undirected graph $G = (V, E)$ at a vertex v is obtained by deleting the edges incident to v ad adding to G all edges that are incident to v in $\bar{G}$. Vertex switching was first introduced by van Lint and Seidel and is often referred to as **Seidel switching**. Switching has been studied in the context of *pan-cyclic graphs*, *acyclic graphs*, *isomorfism*, and reconstruction.

k**-Vertex connected graph** — k-вершинно-связный граф.

Vertex-symmetric graph — вершинно-симметричный граф.

A graph $G = (V, E)$ is a **vertex-symmetric graph** if the group of graph *automorphisms* $A(G)$ acts transitively on V, i.e. for any two vertices $v, w \in V$ there is a graph automorphism $\alpha \in A(G)$ with $\alpha(v) = w$. *Cayley graphs* are vertex-symmetric.

Vertex transitive graph — вершинно-транзитивный граф.

A digraph or graph is **vertex transitive**, if for every pair of vertices i and j there is an *automorphism* that maps i to j.

Vertex star — вершиная звезда.

See *Edge cut*.

Very strongly perfect graph — очень строго совершенный граф.

See *Strongly perfect graph*.

Vibration — вибрация, колебания.

See *Oscillation of a graph*.

Visibility graph — граф видимости.

The **visibility graph** $VG(S)$ of a set S of n disjoint line segments in a plane has a vertex for every endpoint of a segment in S, two of them being adjacent when the corresponding points see each other, i.e., the segment they define does not cross any segment in S.

The **contracted visibility graph** $CVG(S)$ of a set S of n disjoint line segments has a vertex for every segment in S, two of them, s_1 and s_2, being adjacent when *some* endpoint of s_1 sees *some* endpoint of s_2.

Let S be a set of n disjoint horizontal line segments in the plane without two endpoints having equal abscissa. The **bar-visibility graph** $BVG(S)$ has a vertex for every segment of S, and two segments s and t are adjacent, if there is a vertical segment joining s and t and touching no other segment of S.

Vizing's conjecture — гипотеза Визинга.

In 1963 V.Vizing conjectured that

$$\gamma(G)\gamma(H) \leq \gamma(G\Box H)$$

for all graphs G and H, where $\gamma(G)$ denotes the *domination number* of G and $G\Box H$ is the *Cartesian product* of G and H.

Volume of a graph — объём графа.

See *Geometric realization.*

Volume of G-trade — объём G-трейда.

See *G-trade.*

Voronoi diagram — диаграмма Вороного.

The standard **Voronoi diagram** of a set of n given points (called sites) is a subdivision of the plane into regions, every one associated with each site. Each site's region consists of all points in the plane closer to it than to any of the other sites. One application that frequently occurs is what Knuth called the "post office" problem.

W

Walk — маршрут.

An alternating sequence of not necessarily distinct vertices and edges, starting and ending with a vertex, in which every edge is incident with two vertices immediately preceding and following it. If all vertices of a **walk** are distinct (and hence also all edges in the **walk** are distinct), the **walk** is called a *simple chain* or sometimes a *path*. The **walk** $(x_1, \ldots, x_{k+1})$ is **open** [**closed**] iff $x_{k+1} \neq x_1$ [$x_{k+1} = x_1$]. The **length** of the **walk** is k above. A **walk** is a **trail**, if no edge is used twice.

The other name is **Sequence**.

Walk-matrix — матрица маршрутов.

The **walk-matrix** of a graph G is defined by $W(G) = (w_{ij})$, where w_{ij} is the number of walks in G of length j that start at v_i, $1 \leq i \leq n$, $1 \leq j \leq n - 1$.

k**-Walk** — k-маршрут.

A k**-walk** in a graph is a *spanning closed walk* using each vertex at most k times. When $k = 1$, a 1-walk is a Hamilton cycle.

Weak *NP*-complete problem — слабо NP-полная задача.

See *Pseudo-polynomial algorithm.*

Weak clique-covering cycle — слабо кликово-покрывающий цикл.

A cycle of a graph C is called **weak clique-covering**, if each component of $G - V(C)$ is a *clique.*

Weak clique-covering path — слабо кликово-покрывающий путь.

A path P of G is called a **weak clique-covering**, if each component of $G - V(P)$ is a *clique.*

Weak computation — слабая вычисляемость.

Weak k**-covering cycle** — слабо k-покрывающий цикл.

A **weak** k**-covering cycle** C of a graph G is a cycle C such that each component of $G - V(C)$ has fewer than k vertices.

The other name is k**-dominating cycle**.

Weak direct product — слабое прямое произведение.

See *Product of two graphs.*

Weak dominating set — слабое доминирующее множество.

See *Strong dominating set.*

Weak dual graph — слабо двойственный граф.

The **weak dual graph** of a plane graph G is the graph G^* with (finite) faces of G as its vertices, two vertices being adjacent if the corresponding faces of G share a boundary edge.

Weak NP-hard problem — слабо NP-трудная задача.

See *Pseudo-polynomial algorithm.*

Weak isomorphism — слабый изоморфизм.

See *Cycle isomorphism.*

Weak order — слабый порядок.

Binary relation on $\{1, 2, \ldots, n\}$ is a **weak order** if it is a complete *preorder.*

Weak Perfect Graph Conjecture — слабая гипотеза о совершенных графах.

Weak Perfect Graph Conjecture (or **WPGC**) is formulated as follows: A graph is perfect if and only if its complement is perfect.

See also *Strong Perfect Graph Conjecture.*

Weakly arithmetic vertex function — слабо арифметическая вершинная функция.

See *Weakly (k, d)-arithmetic graph.*

Weakly (k, d)-arithmetic graph — слабо (k, d)-арифметический граф.

A **weakly arithmetic vertex function** of a graph $G = (V, E)$ is a vertex function $f : V(G) \to \{0, 1, 2, \ldots\}$, such that, for specified positive integers k and d, $\{k, k + d, k + 2d, \ldots\}$ is the set of values of the induced edge function f^* defined by $f^*(uv) = f(u) + f(v)$ for each edge $uv \in E(G)$. If a graph admits such a vertex function, f, then G is said to be **weakly (k, d)-arithmetic**.

In the above definition, if we impose the condition that the vertex function f is injective, then f is called a **(k, d)-arithmetic numbering** of the graph G and if a graph G admits such a numbering, then the graph G is called a **(k, d)-arithmetic graph**.

Weakly chordal graph — слабо хордальный граф.

A graph G is called a **weakly chordal graph** if G and $\bar{G}$ contain no induced $C_k, k \geq 5$. It is known that the class of chordal graphs is contained in the class of weakly chordal graphs and the class of weakly chordal graphs is contained in the class of perfect graphs.

Weakly-connected dominating number — слабо связное доминирующее число.

See *Weakly-connected dominating set.*

Weakly-connected dominating set — слабо связное доминирующее множество.

A **weakly-connected dominating set**, $\mathcal{W}$, of a graph G is a dominating set such that the subgraph consisting of $V(G)$ and all edges incident with vertices in $\mathcal{W}$ is connected. Define the minimum cardinality of all weakly-connected dominating sets of G as the **weakly-connected domination number** of G and denote this by $\gamma_w(G)$.

Weakly connected graph — слабо связный граф, слабый орграф.

A digraph not representable as $G_1 \cup G_2$, where G_1, G_2 are vertex-disjoint non-empty digraphs.

Weakly connected vertices — слабо связанные вершины.

If vertices v_1 and v_2 are not *strongly connected* but are connected in the corresponding undirected graph, then v_1 and v_2 are said to be **weakly connected**.

Weakly dense m-ary tree — слабо плотное m-арное дерево.

See *r-dense tree.*

Weakly geodetic graph — слабо геодезический граф.

G is a **weakly geodetic graph** if for every pair of vertices whose distance is at most 2 there is a unique path of the minimum length between them.

Weakly pancyclic graph — слабо панциклический граф.

A graph G with n vertices is called a **weakly pancyclic graph**, if it has cycles of all lengths from the *girth* to the *circumference.*

Weakly triangulated graph — слабо триангулируемый граф.

These are graphs without induced cycles of length ≥ 5 or complement of such cycles. It is known that all *triangulated-perfect graphs* are weakly triangulated.

Weight (of a vertex) — вес вершины.

See *Magic labeling.*

Weighted degree of a vertex — взвешенная степень вершины.

Weighted degree $d^w(v)$ **of a vertex** v is the sum of the weights of the edges incident with v.

Weighted domination number — взвешенное число доминирования.

The **weighted domination number** $\gamma_w(G)$ of a weighted graph (G, w) is the minimum weight $w(D) = \sum_{w \in D} w(v)$ of a set $D \subseteq V(G)$ such that every vertex $x \in V(G) - D$ has a neighbor in D.

Weighted girth problem — задача о взвешенном обхвате.

Given a weighted undirected graph G, the **weighted girth problem (WGP)** is to find a *cycle* having minimal weight. This problem is in general *NP*-hard but it can be solved in polynomial time, when G

does not contain any cycle of a negative weight.

If we force the length of the cycle in the WGP to be less than some value k, we obtain the **cardinality constrained circuit problem (CCCP)**. If the cycle in WGP is forced to go through all vertices of $V(G)$, we obtain the well-known *traveling salesman problem.*

Weighted graph — взвешенный граф.

1. A **weighted graph** is a pair (G, w), where G is a graph and w is a weight function which associates with every vertex x a non-negative weight $w(x)$. For a subset S of the vertices we define the weight of S, denoted by $w(S)$, as the sum of weights of the vertices in S. If $\sum_{v \in V(G)} w(v) = |V(G)|$, then we speak of a **normed weighted graph**.

2. In some applications it is natural to assign a number (non-negative real) to each edge of a graph. For any edge e, this number is written as $w(e)$ and is called its **weight**. Naturally, the graph in question is called a **weighted graph**. The **weight of a (sub)graph** is equal to the sum of weights of its edges.

An unweighted graph can be regarded as a weighted graph in which the weight $w(e) = 1$ e is assigned to each edge.

3. The graph G is called a **weighted graph** if there exist a vertex-weight function $w^V : V(G) \to R^+$ and an edge-weight function $w^E(G) : E(G) \to R_0^+$. For a subgraph H of G, the vertex-weight and the edge-weight of H are defined by

$$w^V(H) = \sum_{v \in V(H)} w^V(v); \; w^E(H) = \sum_{e \in E(H)} w^E(e).$$

Well-covered graph — хорошо покрытый граф.

Let β, respectively i, denote the maximum, respectively minimum, cardinality of a maximal *independent set* of G. A graph is called **well-covered** if for this graph $i = \beta$ and $\beta + \Delta = \lceil 2\sqrt{n} - 1 \rceil$. The problem of determining whether or not a graph is *not* well-covered is NP-complite.

P**-well-covered graph** — P-хорошо покрытый граф.

See *Hereditary P-well-covered graph.*

Well-located graph — хорошо размещённый граф.

A *dominating set*, say D, it said to be located if, for every pair of vertices not in the set D, their *neighbours* that are in D differ in at least one vertex. A graph is called **well-located**, if it has the property that every independent dominating set is located.

Wheel — колесо.

A **wheel** is a graph (denoted W_n) obtained from a *cycle* of length n (**rim** of the wheel) by adding a new vertex v_0 (**hub** of the wheel) that is *adjacent* with all vertices of the cycle.

See also *Pseudo-wheel.*

k**-Wide diameter** — k-широкий диаметр.

The k**-wide diameter** $d_k(G)$ of G is the maximum value of the k*-Wide distance* between two vertices of G.

k**-Wide distance** — k-широкое расстояние.

For two distinct vertices $x, y \in V(G)$, the k**-wide distance** $d_k(x, y)$ between x and y is the minimum integer l such that there exist k vertex-disjoint (x, y)-chains whose lengths are at most l. We define $d_k(x, x) = 0$.

Width — ширина.

See F*-width.*

Width of a layout — ширина укладки.

See *Layout.*

Width of a tree-decomposition — ширина древесной декомпозиции.

See *Treewidth of a graph.*

F**-Width (of a hypergraph)** — F-ширина.

Let H and F be two hypergraphs on the same vertex set. The F**-width** $w(H, F)$ of H is the minimal size of an H*-covering set* of edges from F. The F**-matching width** $mw(H, F)$ of H is the maximum, over all *matchings* M in H, of $w(H, F)$. We write $w(H)$ and $mw(H)$ for $w(H, H)$ and $mw(H, H)$, respectively, and call these parameters the **width** and **matching width** of H.

The **independent** F**-width** $iw(H, F)$ of H is the minimal size of a covering matching in F. The **independent** F**-matching width** $imv(H, F)$ of H is the maximum, over all matchings M of H, of $im(M, F)$. Again, we write $iw(H)$ for $iw(H, H)$ and $imw(H, H)$, and call them the **independent width** and **independent matching width** of H, respectively.

Windmill — ветряная мельница.

A **windmill**, denoted by $pS.K_r$, is the graph obtained by *coalescing* a complete graph K_r, $r \geq 2$, with p disjoint copies of a graph $S(v)$, rooted at $v \in V(S)$, $0 \leq p \leq r$. The root vertex, v, in the labelled graph S, is identified with one vertex of K_r so that p distinct vertices of K_r are cut vertices of the windmill.

A **complex windmill** is the graph obtained by coalescing a complete graph K_r, $r \geq 2$, with disjoint graphs $S_1, S_2, \ldots, S_p$ at p distinct vertices of K_r, $0 \leq p \leq r$, so that these vertices are cut vertices of the complex windmill.

Word — слово.

The same as *String*.

Wounded spider — искалеченный паук.

See *Spider*.

Wreath product of graphs — кольцевое произведение графов, лексикографическое произведение.

The same as *Lexicographic product*.

Y

Yanov schemata — схемы Янова.

Yanov schemata were introduced by A.A. Lyapunov and Yu.I. Yanov in 1956. A complete presentation of results was described in paper "Yu.I. Yanov. On logical algorithm schemata, Cybernetics Problems, 1, Fizmatgiz, Moscow, 1958". It became a classical work owing to its completeness: all the basic components of the theory of program transformations were explicitly formulated and, within the constructed system of concepts, were completely studied. Among these components are: formalization of the concepts of program schemata, assignment of the equivalence relation, determination of the algorithm recognizing schemata equivalence and, finally, construction of the system of equivalent transformations which is **complete** in the sense that any pair of equivalent schemas can be transformed into each other by successive applications of these transformations retaining the equivalence.

Z

Zone — зона, сильно связная область.

The same as *Strongly connected region.*

Zone-interval reprezentation — зонно-интервальное представление.

A sequence of different cf-graphs $G_0, G_1, \ldots, G_r$ is called a **zone-interval representation** of the cf-graph G, if $G_0 = G$, G_r contains no zones and for all i, $0 < i \leq r$, the graph G_i is obtained from G_{i-1} by reduction of mutually disjoint intervals, being zones, into nodes.

Z**-transformation graph** — Z-преобразованный граф.

Z**-transformation graph**, $Z_F(G)$, of G with respect to a specific set F of faces is a graph on the perfect matchings of G, such that two perfect matchings M_1 and M_2 are adjacent provided M_1 and M_2 differ only in a cycle that is the boundary of a face in F. If F is a set of all interior faces, $Z_F(G)$ is a usual Z-transformation graph; if F contains all faces of G it is a novel graph called the **total** Z**-transformation graph**.

В.А. Евстигнеев, В.Н. Касьянов

СЛОВАРЬ ПО ГРАФАМ В ИНФОРМАТИКЕ

Под редакцией
проф. Виктора Николаевича Касьянова

Рукопись поступила в редакцию 15. 10. 2009
Ответственный за выпуск Г. П. Несговорова
Редактор А. А. Шелухина

Подписано в печать 11. 12. 2009
Формат бумаги 60 × 84 1/16 Объем 17,0 уч.-изд.л., 18,6 п.л.
Тираж 75 экз.

Центр оперативной печати “Оригинал 2”,
г.Бердск, ул. Островского, 55, оф. 02, тел. (383) 214-45-35