Введение в дискретную математику и математическую логику

Лекция №1
Введение. Графы и графовые модели. Базовая терминология и типы графов.

- Апанович Зинаида Владимировна

План

- Классификация графов
- Графовые модели
- Терминология графов
- Специальные типы графов
- Операции над графами

Литература

1. В.А. Емеличев, О.И. Мельников , О.И. Сарванов , В.И. Тышкевич Лекции по теории графов
2. Kenneth H. Rosen Discrete Mathematics and Its Applications Seventh Edition
3. J.A. Bondy and U.S.R. Murty Graph theory with applications
4. T. H. CORMEN, C. E. LEISERSON, R. L. RIVEST, C. STEIN Introduction to algorithms
5. Cesar O. Aguilar An Introduction to Algebraic Graph Theory

ОПРЕДЕЛЕНИЕ 1 Граф $G = (V, E)$ состоит из V, непустого множества вершин (или узлов), и E, множества ребер.

Каждое ребро имеет одну или две вершины, называемые его конечными точками (концами).

Считается, что ребро соединяет свои конечные точки.

Замечание: Множество вершин V графа G может быть бесконечным.

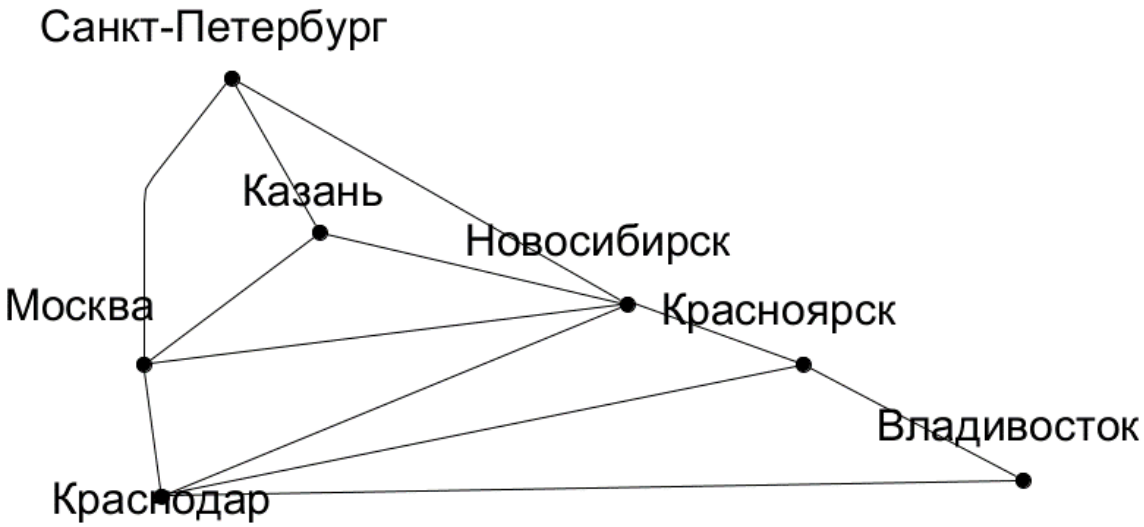
Граф с бесконечным множеством вершин или бесконечным множеством ребер называется бесконечным графом.

Для сравнения, граф с конечным множеством вершин и конечным множеством ребер называется конечным графом.

Предположим, что имеется сеть, состоящая из центров обработки данных и линий связи между компьютерами.

Мы можем представить расположение каждого центра обработки данных точкой, а каждого канала связи - отрезком прямой, как показано ниже.

Эту компьютерную сеть можно смоделировать с помощью графа, в котором вершины графа представляют центры обработки данных, а ребра - линии связи.



Простые графы

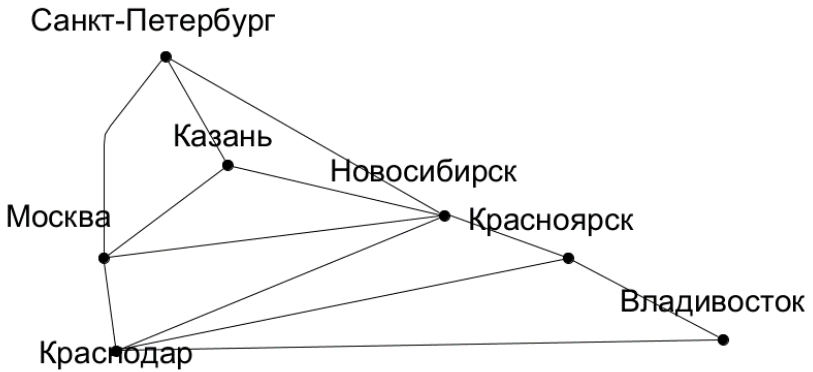
В данном случае каждое ребро графа, представляющего эту компьютерную сеть, соединяет 2 разные вершины.

То есть ни одно ребро не соединяет вершину с самой собой.

Кроме того, никакие два разных ребра не соединяют одну и ту же пару вершин (нет кратных ребер).

Граф, в котором каждое ребро соединяет 2 разные вершины и в котором никакие 2 ребра не соединяют одну и ту же пару вершин, называется простым графом.

Заметим, что в простом графе каждое ребро ассоциируется с неупорядоченной парой вершин $\{u, v\}$, и никакое другое ребро не ассоциируется с этим же ребром. То есть, когда есть ребро простого графа, связанное с неупорядоченной парой $\{u, v\}$, мы можем сказать, без возможной путаницы, что $\{u, v\}$ - это ребро графа.



Мультиграфы

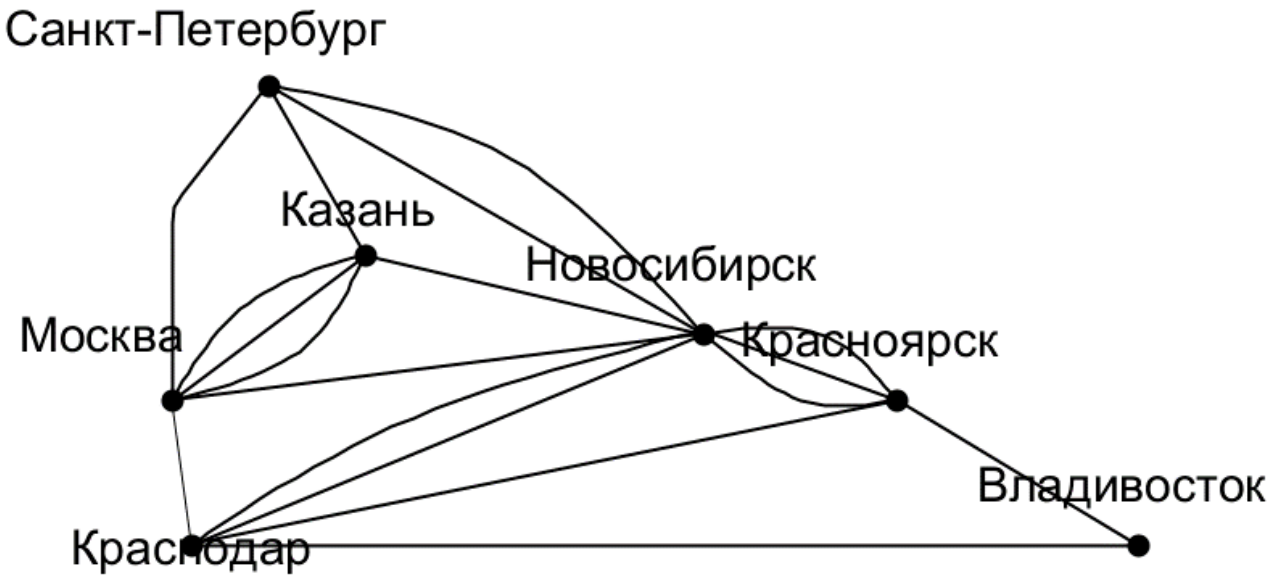
Компьютерная сеть может содержать несколько связей между центрами обработки данных, как показано ниже.

Для моделирования таких сетей нам нужны графы, которые имеют несколько ребер, соединяющих одну и ту же пару вершин.

Графы, которые могут иметь несколько ребер (кратные ребра), соединяющих одни и те же вершины, называются мультиграфами.

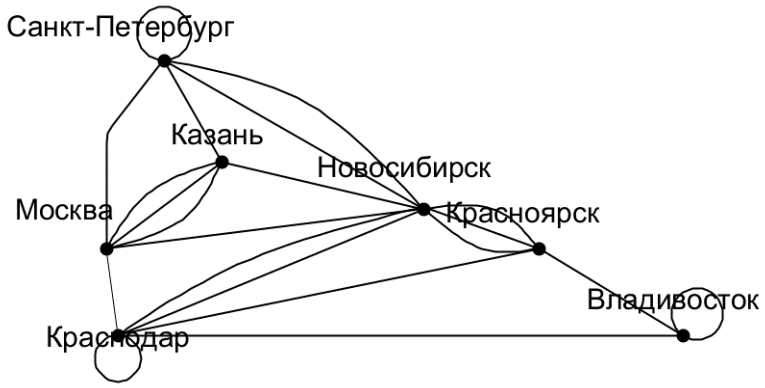
Когда есть m различных ребер, связанных с одной и той же неупорядоченной парой вершин $\{u, v\}$, мы также говорим, что m является кратностью ребра $\{u, v\}$.

То есть мы можем рассматривать это множество ребер как m различных копий ребра $\{u, v\}$.



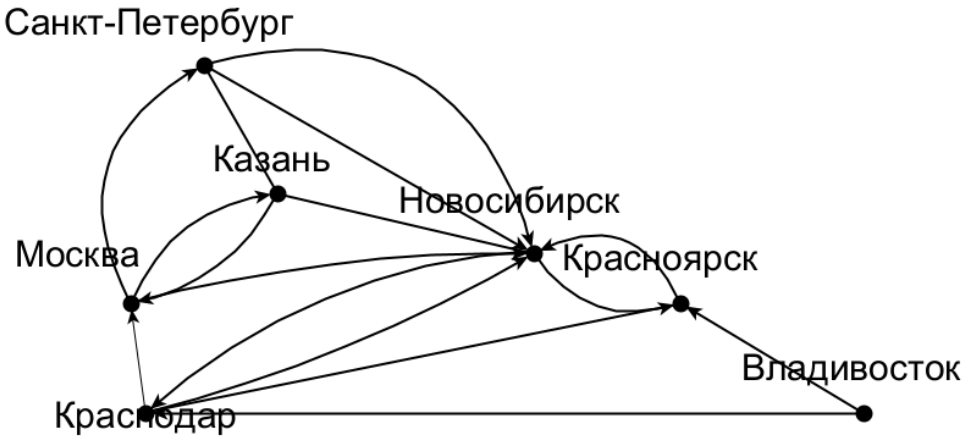
Псевдографы

Иногда канал связи соединяет центр обработки данных с самим собой, возможно, это обратная связь для диагностических целей.
Такая сеть показана ниже.
Для моделирования этой сети нам нужно добавить ребра, которые соединяют вершину с самой собой.
Такие ребра называются петлями, и иногда у нас может быть даже более одной петли в вершине.
Графы, которые могут содержать петли и, возможно, несколько ребер, соединяющих одну и ту же пару вершин или вершину с самой собой, иногда называются псевдографами.



Неориентированные и ориентированные графы

До сих пор мы рассматривали неориентированные графы.
Их ребра также называются неориентированными.
Однако для построения модели графа нам может потребоваться присвоить направления ребрам графа.
Например, в компьютерной сети некоторые соединения могут работать только в одном направлении (такие соединения называются одиночными дуплексными линиями).
Это может иметь место, если в некоторые центры обработки данных отправляется большой объем трафика, а в обратном направлении трафика мало или вообще нет. Такая сеть показана ниже.



Для моделирования такой компьютерной сети мы используем ориентированный граф.

Каждое ребро ориентированного графа связано с упорядоченной парой вершин.

ОПРЕДЕЛЕНИЕ 2 Ориентированный граф (или орграф) $G = (V, E)$ состоит из непустого множества вершин V и множества ориентированных ребер (или дуг) E.

Говорят, что ориентированное ребро (u, v), начинается в вершине u и заканчивается в вершине v.

Простой ориентированный граф

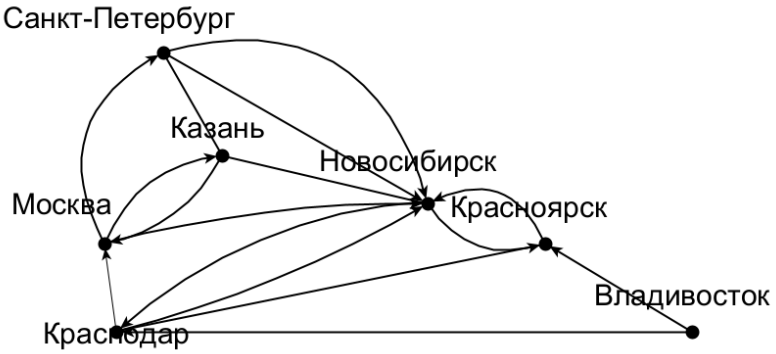
Ориентированный граф может содержать петли и может содержать несколько ориентированных ребер (кратные ориентированные ребра), которые начинаются и заканчиваются в одних и тех же вершинах.

Ориентированный граф может также содержать ориентированные ребра, которые соединяют вершины *u* и *v* в обоих направлениях; то есть, когда орграф содержит ребро из *u* в *v*, он также может содержать одно или несколько ребер из *v* в *u*.

Обратите внимание, что мы получаем ориентированный граф, когда назначаем направление каждому ребру в неориентированном графе.

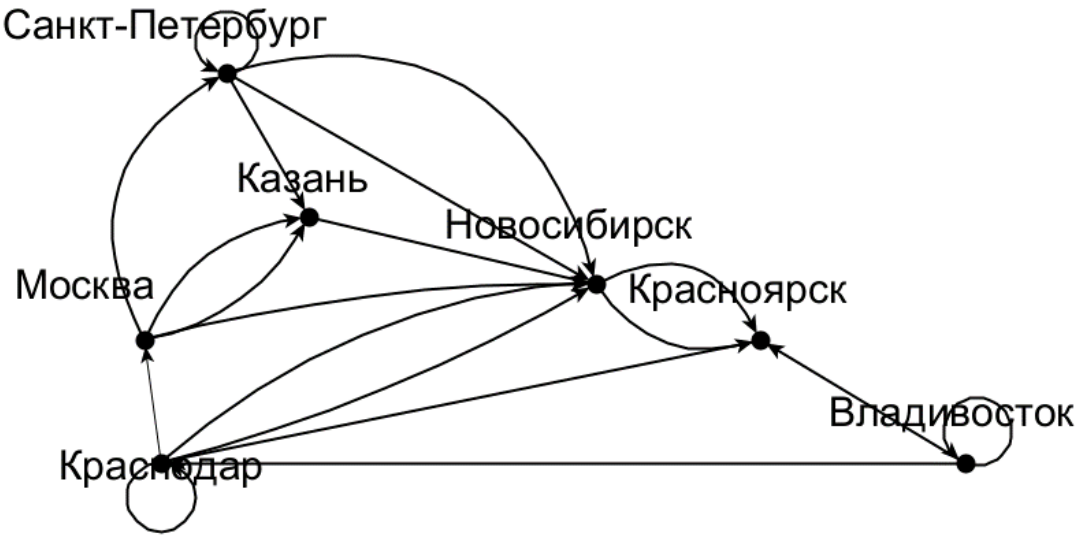
Когда ориентированный граф не имеет петель и не имеет кратных ориентированных ребер, он называется простым ориентированным графом.

Поскольку простой ориентированный граф имеет не более одного ребра, связанного с каждой упорядоченной парой вершин (*u, v*), мы называем упорядоченную пару (*u, v*) ребром, если в графе с ней связано ребро.



Ориентированные мультиграфы

В некоторых компьютерных сетях может присутствовать несколько каналов связи между двумя центрами обработки данных, как показано ниже.
Для моделирования таких сетей используются ориентированные графы, которые могут иметь несколько ориентированных ребер из одной вершины во вторую (возможно, ту же самую) вершину.
Мы называем такие графы ориентированными мультиграфами.
Когда имеется *m* ориентированных ребер, каждое из которых связано с упорядоченной парой вершин (*u, v*), мы говорим, что (*u, v*) является ребром кратности *m*.



Смешанный граф

Для некоторых моделей нам может понадобиться граф, в котором некоторые ребра неориентированные, а другие - ориентированные.

Граф с ориентированными и неориентированными ребрами называется смешанным графом.

Например, смешанный граф может использоваться для моделирования компьютерной сети, содержащей связи, которые работают в обоих направлениях, а также связи, которые работают только в одном направлении.

Из-за относительно недавнего интереса к теории графов и из-за того, что она имеет приложения к широкому спектру дисциплин, было введено много различных терминологий теории графов.

Терминология, используемая математиками для описания графов, все больше стандартизируется, но терминология, используемая для обсуждения графов, когда они используются в других дисциплинах, по-прежнему довольно разнообразна.

Хотя терминология, используемая для описания графов, может различаться, три ключевых вопроса помогут понять структуру графа

1) Являются ли ребра графа неориентированными или ориентированными (или есть и те и другие)?

2) Если граф неориентированный, присутствуют ли кратные ребра? Если граф ориентированный, присутствуют ли кратные ориентированные ребра?

3) Присутствуют ли петли?

Ответы на такие вопросы помогают нам определить тип графа.

Типы графов

Тип	Ребра	Разрешены ли кратные ребра?	Разрешены ли петли?
Простой граф	Неориентированные	Нет	Нет
Мультиграф	Неориентированные	Да	Нет
Псевдограф	Неориентированные	Да	Да
Простой ориентированный граф	Ориентированные	Нет	Нет
Ориентированный мультграф	Ориентированные	Да	Да
Смешанный граф	Ориентированные и неориентированные	да	Да

Графы используются для моделирования ВСЕГО

Социальные сети;

Сети связи;

Приложения для проектирования программного обеспечения;

Транспортные сети;

Биологические сети;

Нейронные сети;

Спортивные соревнования

Графы знаний…

Социальные сети

Графы широко используются для моделирования социальных структур, основанных на различных видах отношений между людьми или группами людей.

Эти социальные структуры и графы, которые их представляют, известны как социальные сети.

В этих графовых моделях отдельные лица или организации представлены вершинами; отношения между отдельными лицами или организациями представлены ребрами.

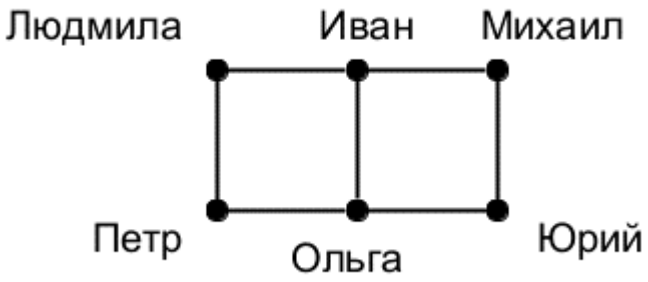
Изучение социальных сетей является чрезвычайно активной междисциплинарной областью, и с их помощью изучалось множество различных типов отношений между людьми.

Мы представим здесь некоторые из наиболее часто изучаемых социальных сетей.

Графы знакомств.

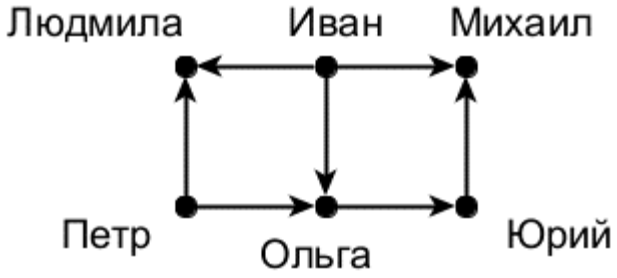
Мы можем использовать простой граф, чтобы представить, знают ли 2 человека друг друга, то есть, знакомы ли они или являются ли они друзьями (как в реальном мире, так и в виртуальном мире через сайт какой-нибудь социальной сети).
Каждый человек в определенной группе людей представлен вершиной.
Неориентированное ребро используется для соединения 2 человек, когда эти люди знают друг друга, или являются друзьями.
Никаких кратных ребер и обычно никаких петель не используется.
Небольшой граф знакомств показан ниже.
ЭТО ПРОСТОЙ НЕОРИЕНТИРОВАННЫЙ ГРАФ.

Граф знакомств всех людей в мире имеет более 8 миллиардов вершин и, вероятно, более 1 триллиона ребер!



Графы влияния

В исследованиях группового поведения наблюдается, что определенные люди могут влиять на других.
Для моделирования такого поведения можно использовать ориентированный граф, называемый графом влияния. Каждый человек в группе представлен вершиной. Существует ориентированное ребро из вершины *a* в вершину *b*, если человек, представленный вершиной *a*, может влиять на человека, представленного вершиной *b*.
Этот граф не содержит петель и не содержит ориентированных ребер.
ЭТО ПРОСТОЙ ОРИЕНТИРОВАННЫЙ ГРАФ.
Пример графа влияния для членов группы показан ниже.
В группе, смоделированной этим графом влияния, Иван не может быть подвержен чужому влиянию, но он может влиять на Людмилу, Михаила и Ольгу. Кроме того, Петр и Иван могут влиять Людмилу, а она ни на кого не влияет.



Граф сотрудничества

- Граф сотрудничества используется для моделирования социальных сетей, в которых 2 человека связаны совместной работой определенным образом.
- Графы сотрудничества — это простые графы, поскольку ребра в этих графах неориентированные, и нет кратных ребер или петель.
- Вершины в этих графах представляют людей;
- 2 человека соединены неориентированным ребром, когда люди сотрудничали.
- Голливудский граф — это граф сотрудничества, который представляет актеров вершинами и соединяет двух актеров ребром, если они работали вместе над фильмом или телешоу.
- Голливудский граф был графом с более чем 1,5 миллиона вершин (по состоянию на начало 2011 года).

Граф сотрудничества

В графе академического сотрудничества вершины представляют людей (возможно, ограниченных членами определенного академического сообщества), а ребра связывают 2 человек, если они совместно опубликовали статью.

В 2004 году было обнаружено, что граф сотрудничества для людей, опубликовавших исследовательские работы по математике, имел более 400 000 вершин и 675 000 ребер, и с тех пор эти цифры значительно выросли.

Графы сотрудничества также использовались в спорте, где два профессиональных спортсмена считаются сотрудничавшими, если они когда-либо играли в одной команде в течение регулярного сезона своего вида спорта.

Сети коммуникаций

Мы можем моделировать различные сети связи, используя вершины для представления устройств и ребра для представления конкретного типа интересующих нас каналов связи.

Графы звонков (call graphs) можно использовать для моделирования телефонных звонков, совершаемых в сети, например, в междугородной телефонной сети.

В частности, ориентированный мультиграф можно использовать для моделирования звонков, где каждый номер телефона представлен вершиной, а каждый телефонный вызов представлен ориентированным ребром.

Ребро, представляющее вызов, начинается с номера телефона, с которого был совершен вызов, и заканчивается на номере телефона, на который был совершен вызов.

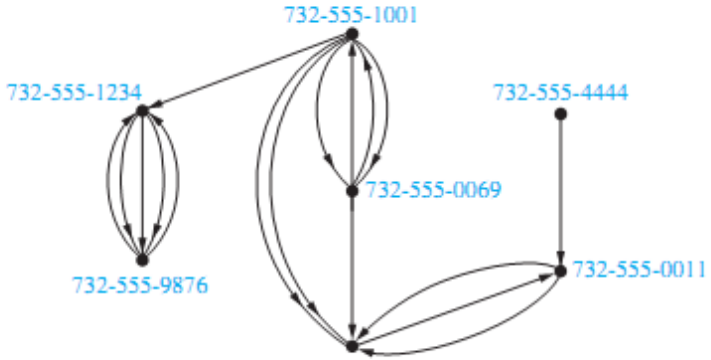
Нам нужны ориентированные ребра, поскольку направление, в котором совершается вызов, имеет значение.

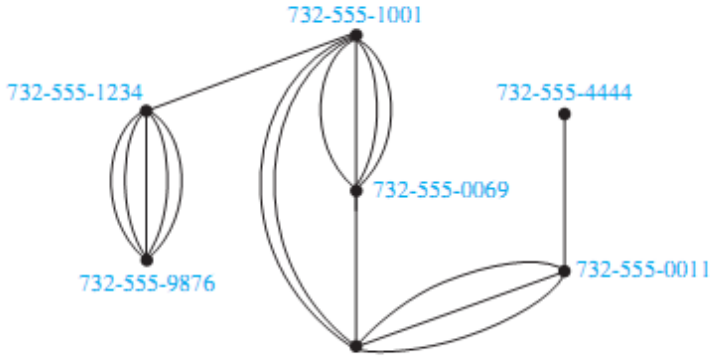
Нам нужны кратные ориентированные ребра, если мы хотим представить каждый звонок, совершаемый с определенного номера телефона на второй номер.

Мы можем моделировать различные сети связи, используя вершины для представления устройств и ребер для представления конкретного типа интересующих нас каналов связи.

Небольшой граф телефонных звонков, представляющий семь телефонных номеров, показан ниже. Когда нас интересует только, был ли звонок, соединяющий 2 телефонных номера, мы используем неориентированный граф с ребром, соединяющим телефонные номера, когда был звонок между этими номерами.

Эта версия графа звонков показана на рисунке справа. Графы звонков, которые моделируют фактическую деятельность телефонов, могут быть огромными. Например, один из графов вызовов, изученный в AT&T, который моделирует вызовы в течение 20 дней, имеет около 290 миллионов вершин и 4 миллиарда ребер.





Транспортные сети

Мы можем использовать графы для моделирования множества различных типов транспортных сетей, включая дорожные, воздушные и железнодорожные сети, а также сети судоходства

ПРИМЕР Авиарейсы

Мы можем моделировать авиационные сети, представляя каждый аэропорт вершиной

В частности, мы можем моделировать все рейсы определенной авиакомпани, совершаемые каждый день, используя ориентированное ребро для представления каждого рейса, идущее от вершины, представляющей аэропорт отправления, к вершине, представляющей аэропорт назначения.

Полученный граф, как правило, будет ориентированным мультиграфом, поскольку в течение одного дня может быть несколько рейсов из одного аэропорта в другой аэропорт.

Графы взаимодействия белков

Взаимодействие белков в живой клетке происходит, когда 2 или более белков в этой клетке связываются для выполнения биологической функции.

Поскольку взаимодействие белков имеет решающее значение для большинства биологических функций, многие ученые работают над открытием новых белков и пониманием взаимодействий между белками.

Взаимодействие белков внутри клетки можно смоделировать с помощью графа взаимодействия белков (также называемого сетью взаимодействия белок-белок), неориентированного графа, в котором каждый белок представлен вершиной, а ребро соединяет вершины, представляющие каждую пару взаимодействующих белков.

Определение подлинных взаимодействий белков в клетке является сложной задачей, поскольку эксперименты часто дают ложноположительные результаты, которые заключают, что 2 белка взаимодействуют, когда на самом деле это не так.

Графы взаимодействия белков

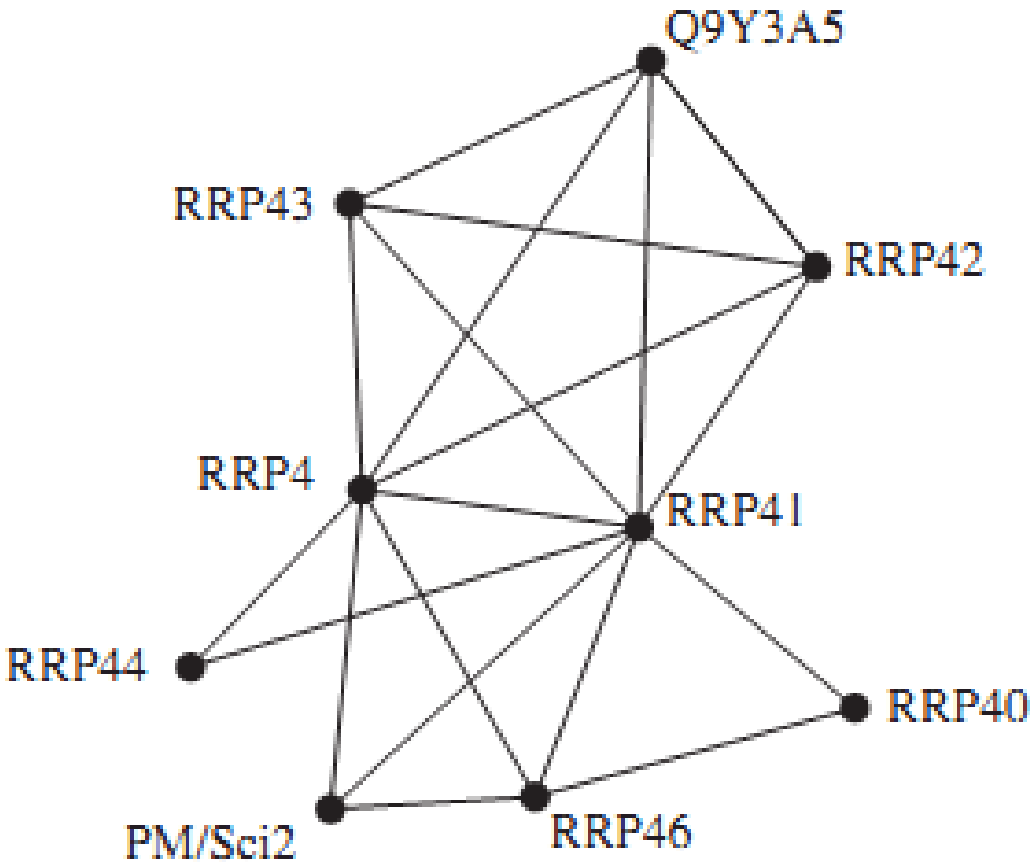
Графы взаимодействия белков можно использовать для получения важной биологической информации, например, путем определения наиболее важных белков для различных функций недавно обнаруженных белков.

Поскольку в типичной клетке есть тысячи различных белков, граф взаимодействия белков клетки чрезвычайно большой и сложный.

Например, клетки дрожжей имеют более 6000 белков, и известно более 80 000 взаимодействий между ними, а клетки человека имеют более 100 000 белков, и, возможно, до 1 000 000 взаимодействий между ними.

Дополнительные вершины и ребра добавляются к графу взаимодействия белков, когда обнаруживаются новые белки и взаимодействия между белками.

Один из модулей графа взаимодействия белков



Графы предшествования и параллельное исполнение программ

Компьютерные программы могут выполняться быстрее, исполняя определенные операторы одновременно

Важно не выполнять оператор, требующий результатов от операторов, которые еще не выполнены.

Зависимость операторов от предыдущих операторов может быть представлена ориентированным графом

Each statement is represented by a vertex, and there is an edge from one statement to a second statement if the second statement cannot be executed before the first statement.

Каждый оператор представляется вершиной, и существует ребро от одного оператора к другому оператору, если другой оператор не может быть выполнен раньше первого оператора

Этот результирующий граф называется precedence graph (граф предшествований).

Компьютерная программа и ее граф предшествований показаны ниже.

Например, граф показывает, что оператор S_5 не может быть выполнен до выполнения операторов S_1, S_2 и S_4

S_1 $a := 0$

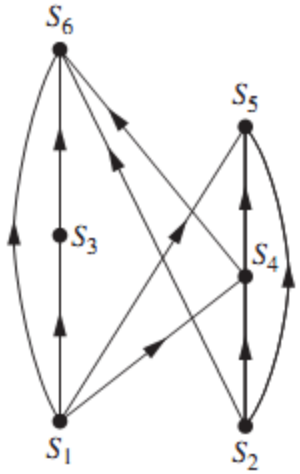
S_2 $b := 1$

S_3 $c := a + 1$

S_4 $d := b + a$

S_5 $e := d + 1$

S_6 $e := c + d$



Графы Знаний (Semantic Web)

Представим предложение «Человек по имени Сергей Бондарчук снял фильм «Война и мир», снимался в главной роли в фильме «Судьба человека» и был женат на Инне Макаровой и Ирине Скобцевой» в виде множества триплет.

Триплеты имеют форму: (Subject, Predicate, Object) или

(Подлежащее, Сказуемое, Дополнение).

dbr:Sergei_Bondarchuk dbo:spouse dbr:Inna Makarova.

dbr:Sergei_Bondarchuk dbo:spouse dbr:Irina Skobtseva.
dbr:Sergei_Bondarchuk dbo:starring dbr:Fate_of_a Man.
dbr:Sergei_Bondarchuk dbo:director dbr:War_and_Peace.

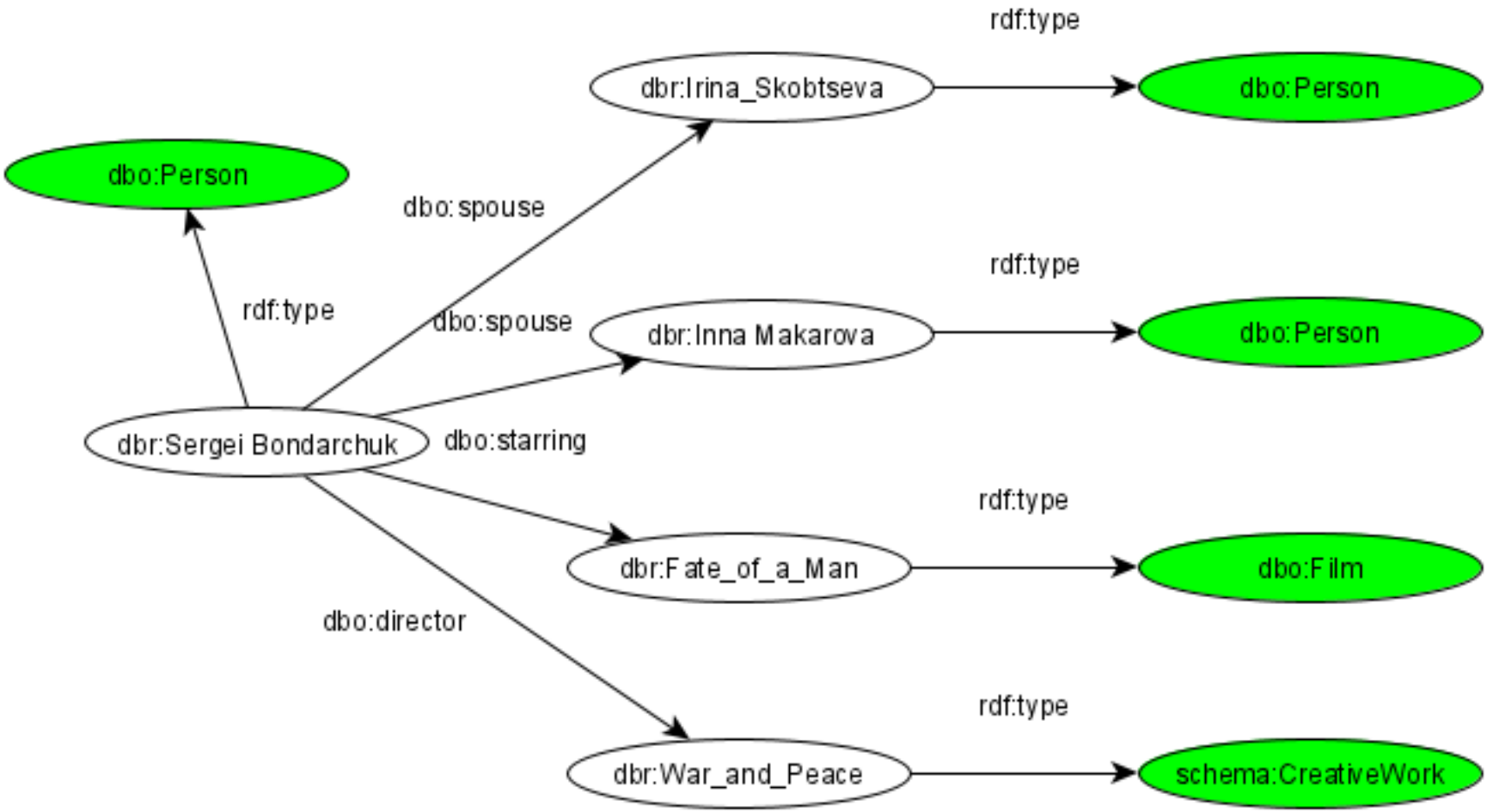
dbr:Sergei_Bondarchuk rdf:type dbo:Person.
dbr:Inna Makarova rdf:type dbo:Person.

dbr:Irina Skobtseva. rdf:type dbo:Person.

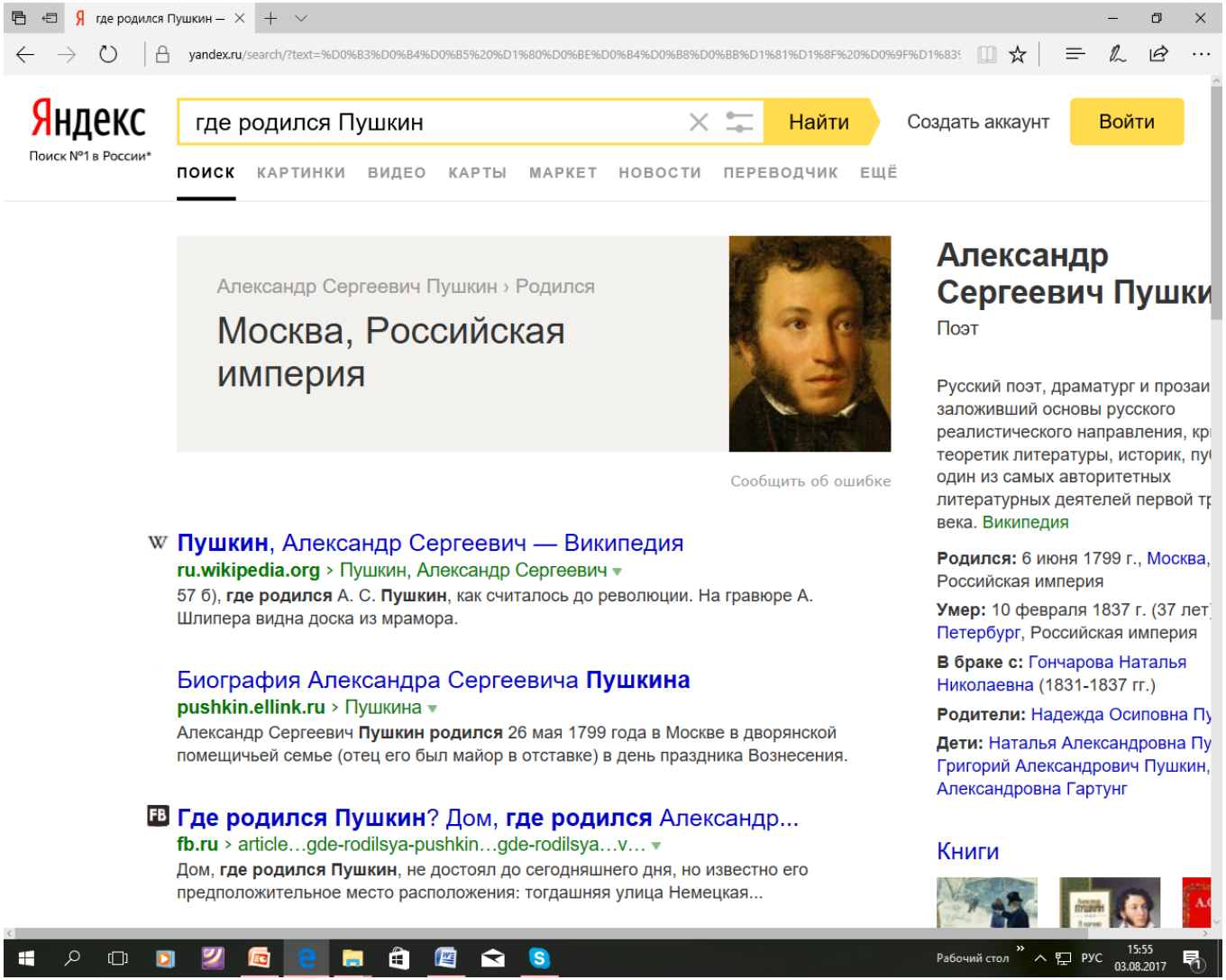
dbr:Fate_of_a Man rdf:type dbo:Film.

dbr:War_and_Peace rdf:type schema:CreativeWork.

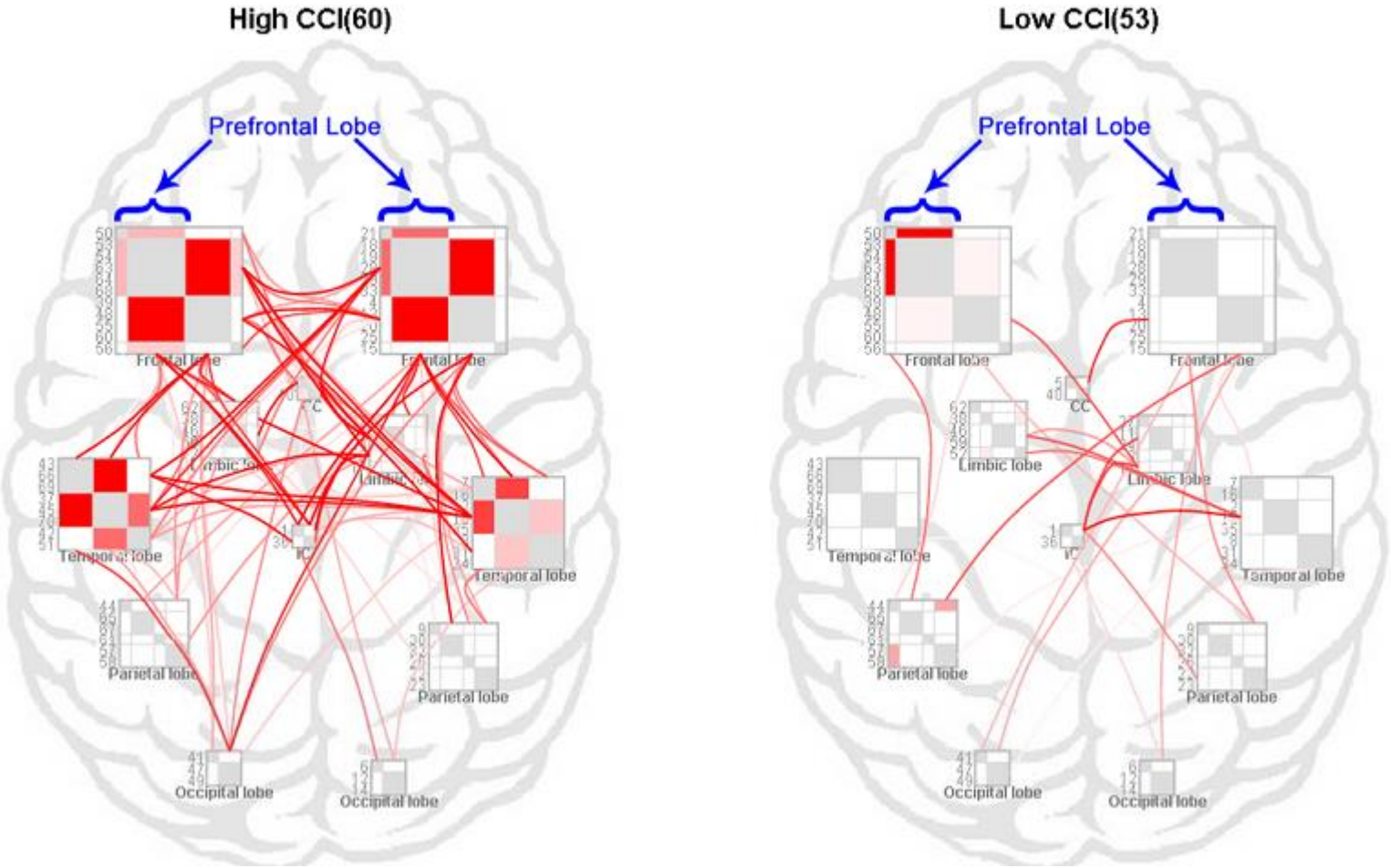
Граф знаний(Semantic Web)



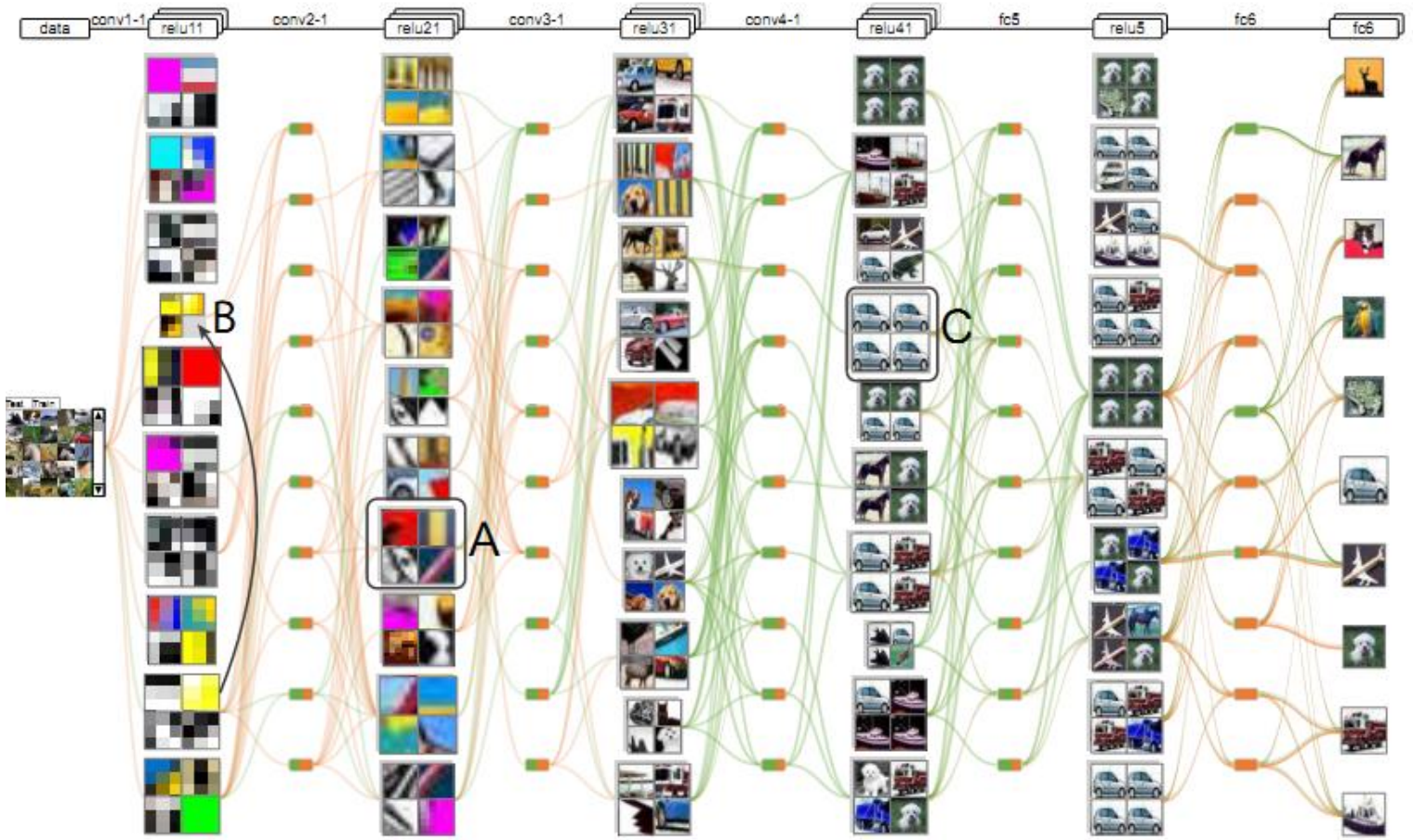
Yandex: где родился Пушкин?



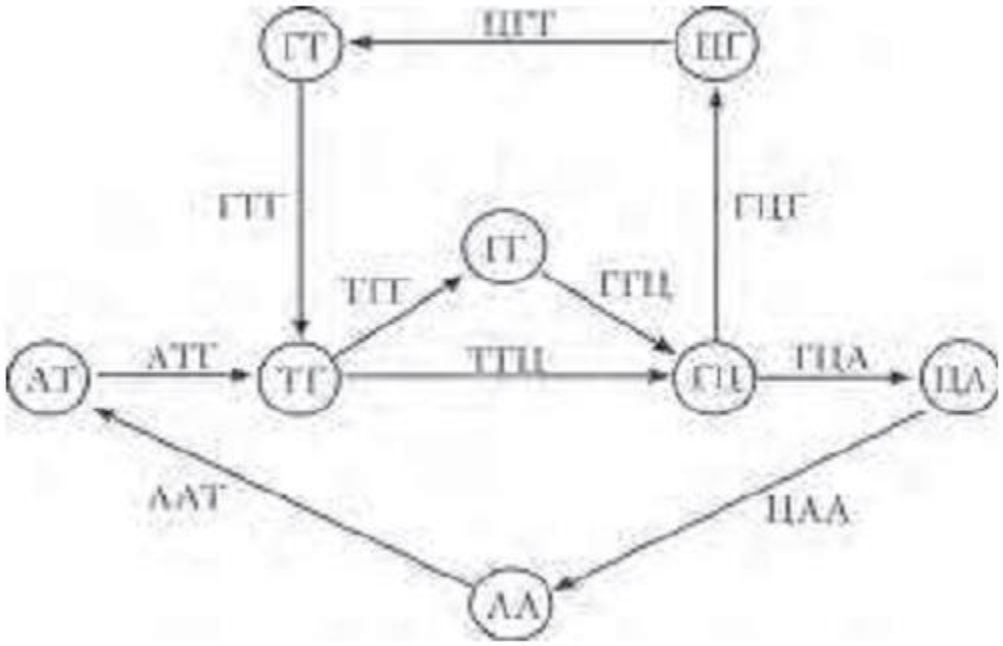
Связи человеческого мозга



Нейронные сети (RNN)



Сборка последовательностей ДНК и РНК из фрагментов (графы de Bruijn)



Введение в дискретную математику и математическую логику

Лекция №1, Часть 2
Базовая терминология и специальные типы графов

- Апанович Зинаида Владимировна

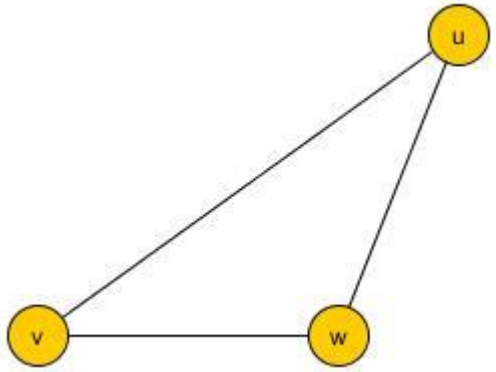
2024 г.

Смежность и инцидентность

ОПРЕДЕЛЕНИЕ 1 Две вершины *u* и *v* в неориентированном графе *G* называются смежными (или соседними) в *G*, если *u* и *v* являются конечными точками ребра *e* графа *G*.

Такое ребро *e* называется инцидентным вершинам *u* и *v*.

Так же говорят, что ребро *e* соединяет вершины *u* и *v*.

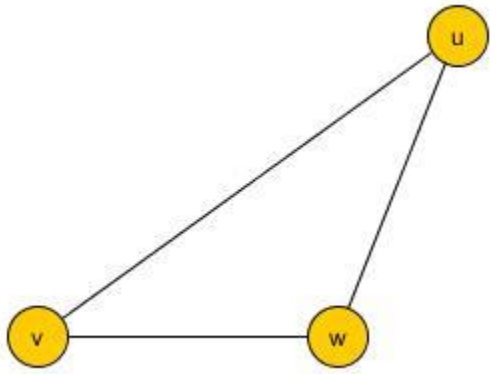


Окружение

ОПРЕДЕЛЕНИЕ 2 Множество всех соседей вершины *v* в графе *G* = (*V*, *E*), обозначаемое *N*(*v*), называется окружением *v*. Если *A* является подмножеством множества *V*, то мы обозначаем *N*(*A*) множество всех вершин в *G*, которые смежны с более чем одной вершиной из *A*. Таким образом,

,

$$N(A) = \bigcup_{v \in A} N(v)$$



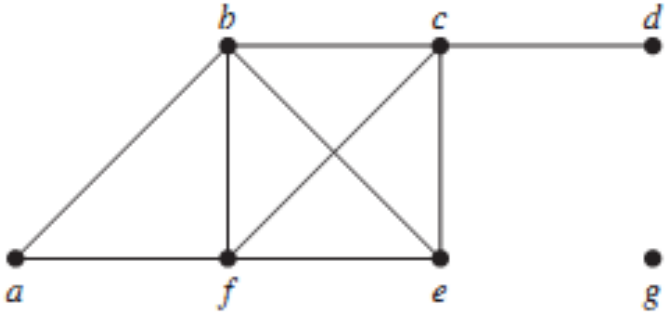
Степень вершины

ОПРЕДЕЛЕНИЕ 3 Степень вершины в неориентированном графе — это количество инцидентных ей ребер.

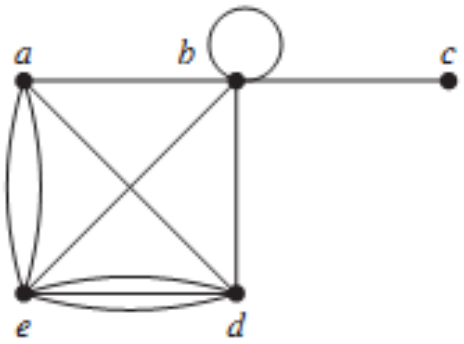
При этом петля в вершине вносит двойной вклад в степень этой вершины.

Степень вершины v обозначается как $\deg(v)$.

ПРИМЕР Чему равны степени и окружения вершин в графах G и H?



G



H

Решение: В графе G, $\deg(a) = 2$, $\deg(b) = \deg(c) = \deg(f) = 4$, $\deg(d) = 1$, $\deg(e) = 3$, $\deg(g) = 0$.
Окружения этих вершин $N(a) = \{b, f\}$, $N(b) = \{a, c, e, f\}$,
$N(c) = \{b, d, e, f\}$, $N(d) = \{c\}$, $N(e) = \{b, c, f\}$, $N(f) = \{a, b, c, e\}$, and $N(g) = \emptyset$.

В графе H, $\deg(a) = 4$, $\deg(b) = \deg(e) = 6$, $\deg(c) = 1$, and $\deg(d) = 5$.
Окружения этих вершин $N(a) = \{b, d, e\}$, $N(b) = \{a, b, c, d, e\}$, $N(c) = \{b\}$, $N(d) = \{a, b, e\}$, and $N(e) = \{a, b, d\}$.

Висячие и изолированные вершины

Вершина степени ноль называется изолированной.

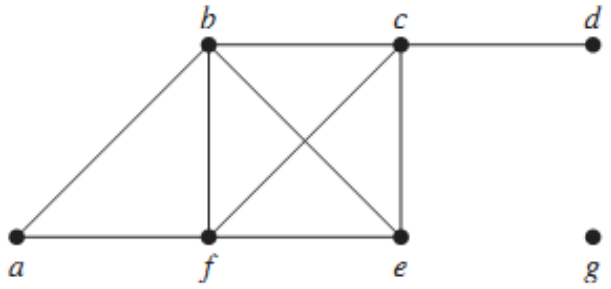
Из этого следует, что изолированная вершина не смежна ни с одной другой вершиной.

Вершина *g* в графе *G* является изолированной.

Вершина степени 1 называется висячей.

Следовательно, висячая вершина смежна ровно с одной другой вершиной.

Вершина *d* в графе *G,* показанном ниже является висячей.



G

Сумма степеней вершин

Что мы получим, если сложим степени всех вершин графа $G = (V, E)$?

Каждое ребро добавляет 2 к сумме степеней вершин, поскольку ребро инцидентно ровно двум (возможно, совпадающим) вершинам.

Это означает, что сумма степеней вершин в два раза больше количества ребер.

Теорема о рукопожатии

ТЕОРЕМА 1 (Теорема о рукопожатии) Пусть $G = (V, E)$ — неориентированный граф с m ребрами. Тогда

$$2m = \sum_{v \in V} \deg(v).$$

(Обратите внимание, что теорема верна даже при наличии кратных ребер и петель.)

EXAMPLE

Сколько ребер в графе с десятью вершинами степени 6 каждая?

Решение: Поскольку сумма степеней вершин равна $6 \cdot 10 = 60$, то $2m = 60$, где m — количество ребер.

Следовательно, $m = 30$.

Теорема 1 показывает, что сумма степеней вершин неориентированного графа четна.

Этот простой факт имеет много следствий, одно из которых дано в виде Теоремы 2.

ТЕОРЕМА 2 Неориентированный граф имеет четное число вершин нечетной степени.

Доказательство: Пусть V_1 — множество вершин четной степени, а V_2 — множество вершин нечетной степени в неориентированном графе $G = (V, E)$ с m ребрами. Тогда

$$2m = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v).$$

Поскольку deg(v) четно для $v \in V_1$, первый член в правой части последнего равенства четный.

Более того, сумма двух членов в правой части последнего равенства четная, поскольку эта сумма равна $2m$.

Следовательно, второй член в сумме также четный.

Поскольку все члены в этой сумме нечетные, таких членов должно быть четное количество.

Таким образом, имеется четное количество вершин нечетной степени.

Терминология для ориентированных графов

ОПРЕДЕЛЕНИЕ 4 Когда (*u, v*) является ребром ориентированного графа G, говорят, что вершина *u* является начальной вершиной ребра (*u, v*), а вершина *v* является конечной или концевой вершиной ребра(*u, v*).

Начальная и конечная вершины петли совпадают. Поскольку ребра в ориентированных графах являются упорядоченными парами, определение степени вершины можно уточнить, чтобы отразить количество ребер, имеющих данную вершину в качестве начальной или конечной вершины.

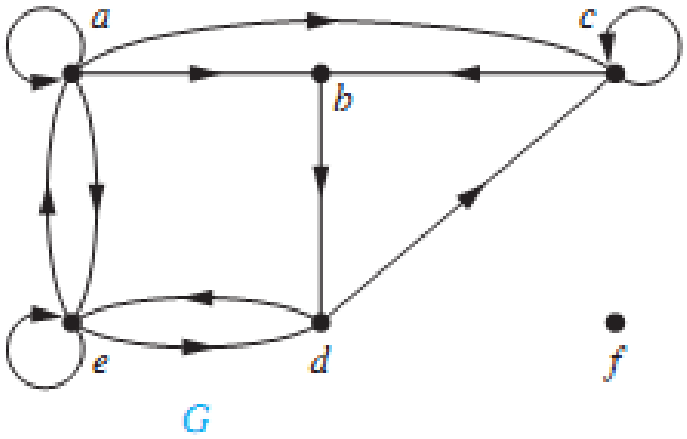
Полустепень исхода и полустепень захода

ОПРЕДЕЛЕНИЕ 5 В ориентированном графе с полустепень захода вершины *v*, обозначаемая как $\deg^-(v)$, — это количество ребер, имеющих вершину *v* в качестве конечной вершины.

Полустепень исхода вершины *v*, обозначаемая как $\deg^+(v)$, — это количество ребер, имеющих вершину *v* в качестве начальной вершины.

Обратите внимание, что петля в вершине добавляет единицу как к полустепени захода, так и к полустепени исхода этой вершины.

ПРИМЕР 4 Найдите полустепень захода и полустепень исхода каждой вершины в графе *G* с ориентированными ребрами.



Решение: Полустепени захода в графе *G:* $\deg^-(a) = 2$, $\deg^-(b) = 2$, $\deg^-(c) = 3$, $\deg^-(d) = 2$, $\deg^-(e) = 3$, and $\deg^-(f) = 0$.
Полустепени исхода в графе *G:* $\deg^+(a) = 4$, $\deg^+(b) = 1$, $\deg^+(c) = 2$, $\deg^+(d) = 2$, $\deg^+(e) = 3$, and $\deg^+(f) = 0$.

ТЕОРЕМА 3 Дан ориентированный граф $G = (V, E)$. Тогда

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|.$$

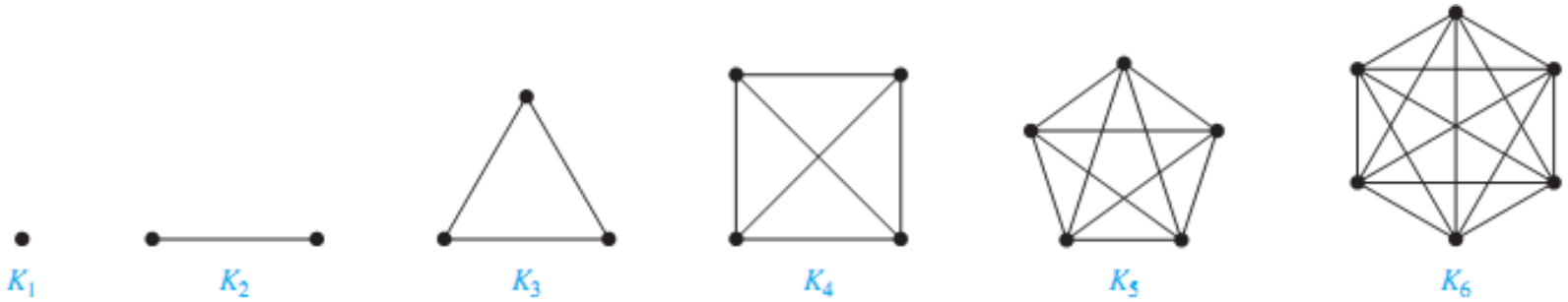
- Существует много свойств ориентированного графа, которые не зависят от направления его ребер. Следовательно, часто бывает полезно игнорировать эти направления.
- Неориентированный граф, который получается в результате игнорирования направлений ребер, называется основанием.
- Ориентированный граф с и его неориентированное основание имеют одинаковое количество ребер.

Специальные типы простых графов

Теперь мы представим несколько классов простых графов.

Эти графы часто используются в качестве примеров и возникают во многих приложениях.

Полные графы (K_n)

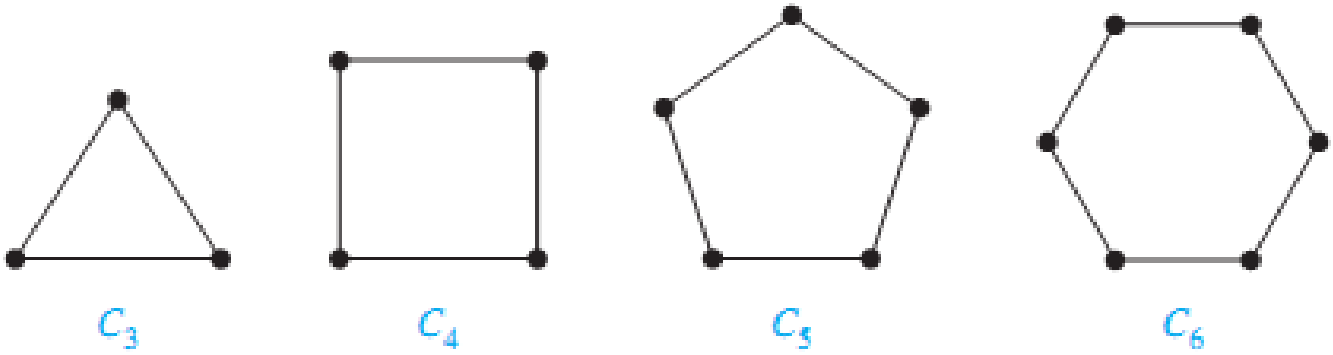


Полный граф на n вершинах, обозначаемый K_n, — это простой граф, содержащий ровно одно ребро между каждой парой различных вершин.

Выше изображены графы K_n для n = 1, 2, 3, 4, 5, 6.

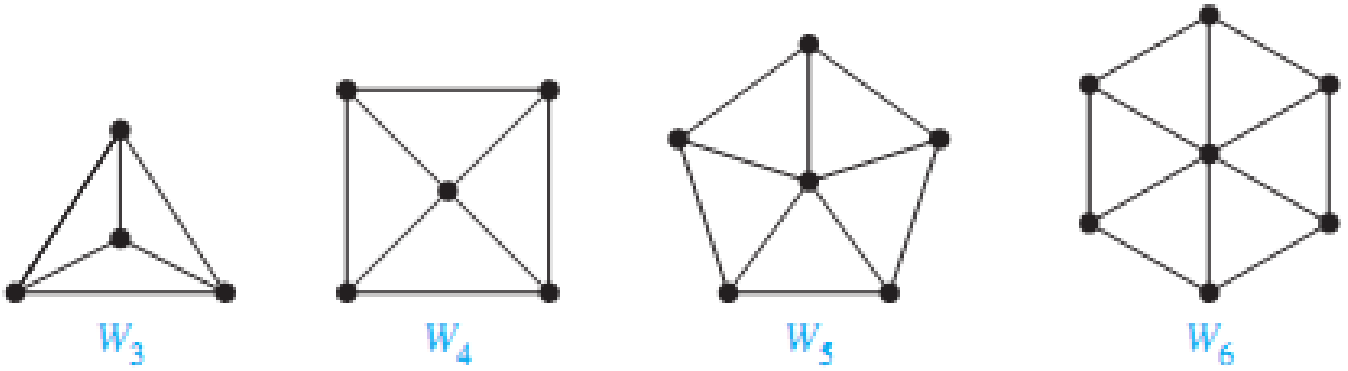
Простой граф, для которого существует по крайней мере одна пара различных вершин, не соединенных ребром, называется неполным.

Простые циклы (C_n)



Цикл C_n, n ≥ 3, состоит из n вершин v_1, v_2,..., v_n и ребер $\{v_1, v_2\}$, $\{v_2, v_3\}$,..., $\{v_{n-1}, v_n\}$ и $\{v_n, v_1\}$.
На рисунке выше показаны циклы C_3, C_4, C_5 и C_6.

Колеса (W_n)

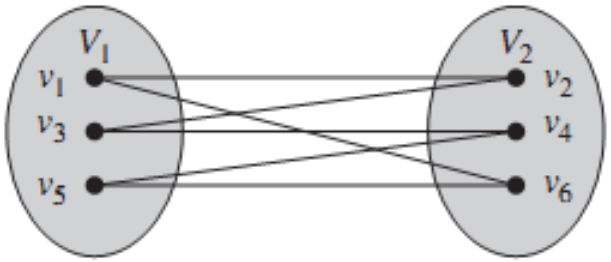


Колесо W_n получаем, когда к циклу C_n добавляется дополнительная вершина, для $n \geq 3$, и эта новая вершина соединяется с каждой из n вершин цикла C_n новыми ребрами. Колеса W_3, W_4, W_5 и W_6 показаны на рисунке выше.

Двудольные графы

Простой граф G называется двудольным, если множество его вершин V можно разбить на два непересекающихся множества V_1 и V_2 так, что каждое ребро в графе соединяет вершину в V_1 и вершину в V_2 (так что ни одно ребро в G не соединяет никакие две вершины в V_1, и никакие две вершины в V_2).

Когда это условие выполняется, мы называем пару (V_1, V_2) двудольным разбиением множества вершин V графа G.



$V_1 = \{v_1, v_3, v_5\}$ и $V_2 = \{v_2, v_4, v_6\}$, и каждое ребро C_6 соединяет вершину в V_1 и вершину в V_2.

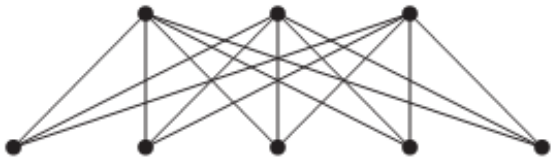
Полные двудольные графы ($K_{m,n}$)

Полный двудольный граф $K_{m,n}$ — это граф, множество вершин которого разделено на 2 подмножества по *m* и *n* вершин соответственно, так что один конец каждого ребра находится в первом подмножестве, а второй конец— во втором подмножестве.

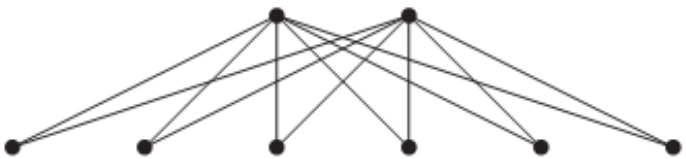
Полные двудольные графы $K_{2,3}$, $K_{3,3}$, $K_{3,5}$ и $K_{2,6}$ показаны ниже.

$K_{2,3}$

$K_{3,3}$

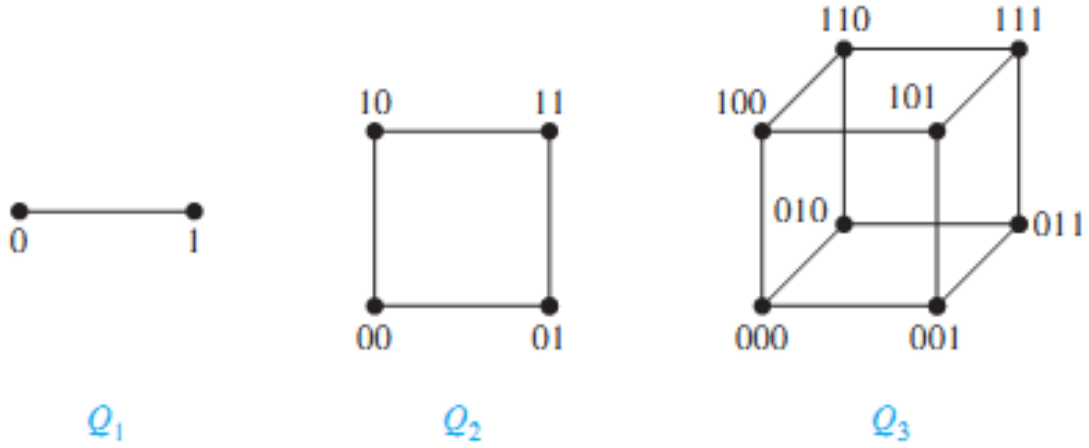


$K_{3,5}$



$K_{2,6}$

n- мерные кубы (Q_n)

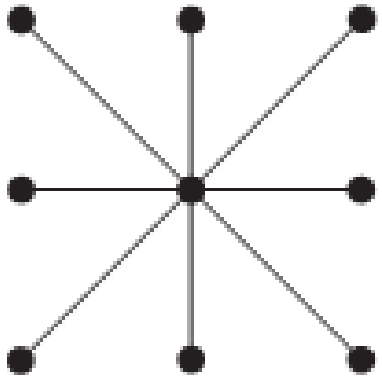


n-мерный гиперкуб, или n-куб, обозначаемый Q_n, — это граф, вершины которого представляют 2^n битовых строк длины *n*.

Вершины Q_n смежны ⇔ битовые строки, которые они представляют, отличаются ровно на одну битовую позицию.

Показаны Q_1, Q_2 и Q_3.

Локальные вычислительные сети (LAN)



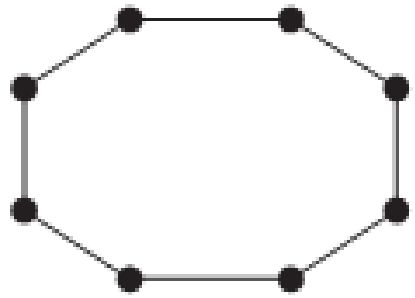
Различные компьютеры в здании, такие как персональные компьютеры, а также периферийные устройства, такие как принтеры и плоттеры, могут быть связаны с помощью локальной вычислительной сети.

Некоторые из этих сетей основаны на топологии звезды, где все устройства подключены к центральному устройству управления.

Локальная вычислительная сеть может быть представлена с помощью полного двудольного графа $K_{1,n}$.

Сообщения отправляются от устройства к устройству через центральное устройство управления.

Кольцевая топология

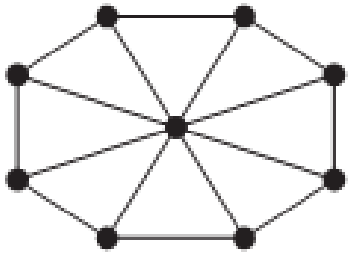


Другие локальные сети основаны на кольцевой топологии, где каждое устройство подключено ровно к двум другим.

Локальные сети с кольцевой топологией моделируются с использованием циклов C_n.

Сообщения отправляются от устройства к устройству по циклу, пока не будет достигнут предполагаемый получатель сообщения.

Топология колесо



Некоторые локальные сети используют гибрид этих двух топологий.

Сообщения могут отправляться либо по кольцу или через центральное устройство.

Такая избыточность делает сеть более надежной.

Локальные сети с такой избыточностью можно моделировать с помощью колес W_n.

Сети для параллельных вычислений

В течение многих лет компьютеры выполняли программы по одной операции за раз.

Алгоритмы, написанные для решения задач, были разработаны для выполнения одного шага за раз; такие алгоритмы называются последовательными.

Однако многие вычислительно интенсивные задачи, такие как моделирование погоды, медицинская визуализация и криптоанализ, не могут быть решены за разумное время с помощью последовательных операций даже на суперкомпьютере.

Кроме того, существует физическое ограничение на то, насколько быстро компьютер может выполнять основные операции, поэтому всегда будут проблемы, которые не могут быть решены за разумное время с помощью последовательных операций.

Параллельная обработка, которая использует компьютеры, состоящие из множества отдельных процессоров, каждый из которых имеет свою собственную память, помогает преодолеть ограничения компьютеров с одним процессором.

Параллельные алгоритмы разбивают задачу на ряд подзадач, которые могут быть решены одновременно. Они могут быть разработаны для быстрого решения задач на компьютере с несколькими процессорами.

В параллельном алгоритме один поток инструкций управляет выполнением алгоритма, отправляя подзадачи разным процессорам и направляя ввод и вывод этих подзадач соответствующим процессорам.

При использовании параллельной обработки одному процессору может потребоваться вывод, сгенерированный другим процессором. Следовательно, эти процессоры должны быть взаимосвязаны. Мы можем использовать соответствующий тип графа для представления сети процессоров в компьютере с несколькими процессорами.

Далее мы опишем наиболее часто используемые типы сетей для параллельных процессоров. Тип сети, используемый для реализации конкретного параллельного алгоритма, зависит от требований к обмену данными между процессорами, желаемой скорости и, конечно же, доступного оборудования.

Полный граф с n вершинами

Простейшие, но и самые дорогие процессоры сетевого взаимодействия содержат двустороннюю связь между каждой парой процессоров.

Эта сеть может быть представлена Kn, полным графом на n вершинах, когда есть n процессоров.

Однако существуют серьезные проблемы с этим типом сети, поскольку требуемое количество соединений очень велико.

В действительности количество прямых соединений с процессором ограничено, поэтому при большом количестве процессоров каждый процессор не может быть напрямую связан со всеми остальными.

Например, при наличии 64 процессоров потребуется C(64, 2) = 2016 соединений, и каждый процессор должен быть напрямую связан с 63 другими.

Линейный массив процессоров

.

С другой стороны, возможно, самый простой способ соединить n процессоров — это использовать схему, известную как линейный массив.

Каждый процессор P_i, кроме P_1 и P_n, связывается своим соседями P_{i-1} и P_{i+1} через 2-стороннюю связь.

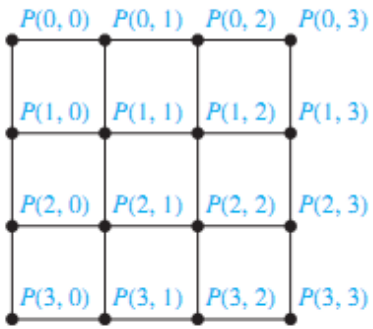
P_1 связан только с P_2, а P_n связан только с P_{n-1}.

Выше показан линейный массив из 6 процессоров.

Преимущество линейного массива заключается в том, что каждый процессор имеет не более 2 прямых соединений с другими процессорами.

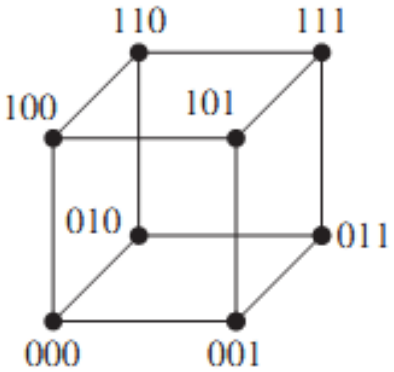
Недостаток заключается в том, что иногда необходимо использовать большое количество промежуточных связей, называемых переходами, для обмена информацией между процессорами.

Двумерный массив (GRID)



Двумерный массив — это часто используемая сеть взаимосвязей. В такой сети число процессоров является полным квадратом, скажем, $n = m^2$.
N процессоров обозначены как P(i,j), $0 \le i \le m - 1$, $0 \le j \le m - 1$.
Двусторонние связи соединяют процессор P(i,j) с его четырьмя соседями, процессорами $P(i \pm 1,j)$ и $P(i,j \pm 1)$.
Заметим, что 4 процессора в углах сетки имеют только 2 смежных процессора, а другие процессоры на границах имеют только 3 соседа.
Иногда используется вариант сети, в которой каждый процессор имеет ровно 4 соединения. Двумерный массив ограничивает количество связей для каждого процессора.
Связь между парами процессоров требует $O(\sqrt{n}) = O(m)$ промежуточных связей

Сеть-гиперкуб



Одним из важных типов процессорных сетей является гиперкуб.

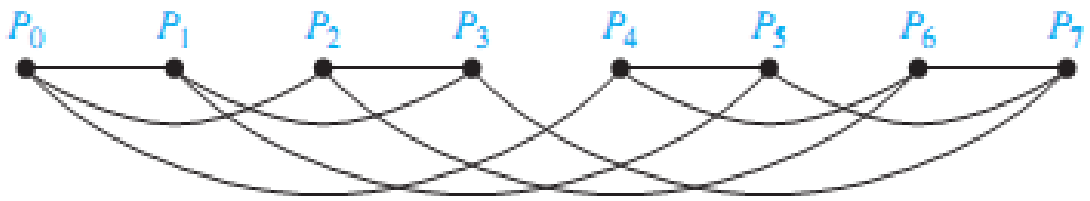
Для такой сети количество процессоров равно степени 2, $n = 2^m$.

N процессоров обозначены как P_0, P_1, ..., P_{n-1}.

Каждый процессор имеет двусторонние соединения с *m* другими процессорами.

Процессор P_i связан с процессорами с индексами, двоичные представления которых отличаются от двоичного представления *i* ровно на 1 бит.

Сеть-гиперкуб

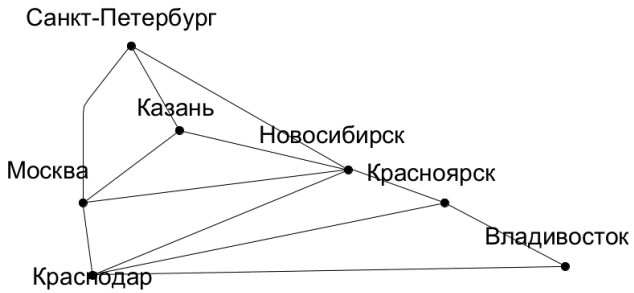


Сеть- гиперкуб уравновешивает количество прямых соединений для каждого процессора и количество промежуточных соединений, необходимых для того, чтобы процессоры могли взаимодействовать.

Многие компьютеры были построены с использованием сети- гиперкуба, и было разработано много параллельных алгоритмов, которые используют сеть-гиперкуб.

Граф Q_m, *m*-куб, представляет сеть-гиперкуб с $n = 2^m$ процессорами.

Подграфы



Иногда нам нужна только часть графа для решения задачи. Например, нас может интересовать только часть большой компьютерной сети, которая включает в себя компьютерные центры в Москве, Новосибирске и Владивостоке.

Тогда мы можем игнорировать другие компьютерные центры и все телефонные линии, не связывающие два из этих трех конкретных компьютерных центров.

В графовой модели для большой сети мы можем удалить вершины, соответствующие компьютерным центрам, отличным от трех интересующих нас, и мы можем удалить все ребра, инцидентные вершине, которая была удалена.

Когда ребра и вершины удаляются из графа, не удаляя конечные вершины каких-либо оставшихся ребер, получается меньший граф.

Такой граф называется подграфом исходного графа.

Подграф графа

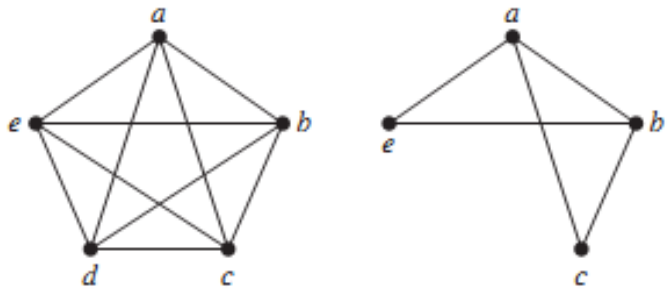
ОПРЕДЕЛЕНИЕ 7 Подграф графа $G = (V, E)$ — это граф $H = (W, F)$, где $W \subseteq V$ и $F \subseteq E$.

Подграф H графа G является собственным подграфом графа G, если $H \neq G$.

Для заданного набора вершин графа мы можем образовать подграф этого графа с этими вершинами и ребрами графа, которые их соединяют.

Граф G является подграфом K_5

.



УДАЛЕНИЕ РЕБЕР ГРАФА

Дан граф $G = (V, E)$ и ребро $e \in E$, мы можем создать подграф графа G, удалив ребро e.

Результирующий подграф, обозначаемый $G - e$, имеет то же множество вершин V, что и граф G.

Его множество ребер равно $E - e$.

Следовательно, $G - e = (V, E - \{e\})$.

Аналогично, если E' является подмножеством E, мы можем создать подграф графа G, удалив ребра множества E' из графа G.

Результирующий подграф имеет то же множество вершин V, что и G.

Его множество ребер равно $E - E'$.

Подграф, полученный путем удаления ребер, называется остовным подграфом графа G.

Добавление ребер к графу

Мы также можем добавить ребро e к графу, чтобы получить новый больший граф, если это ребро соединяет две вершины, уже находящиеся в G.

Мы обозначаем через $G + e$ новый граф, полученный путем добавления нового ребра e, соединяющего две ранее несмежных вершины, к графу G.

Следовательно, $G + e = (V, E \cup \{e\})$.

Множество вершин графа $G + e$ совпадает с множеством вершин G, а множество ребер является объединением множества ребер G и множества $\{e\}$.

УДАЛЕНИЕ ВЕРШИН ИЗ ГРАФА.

Когда мы удаляем вершину v и все инцидентные ей ребра из графа $G = (V, E)$, мы создаем подграф, обозначаемый как

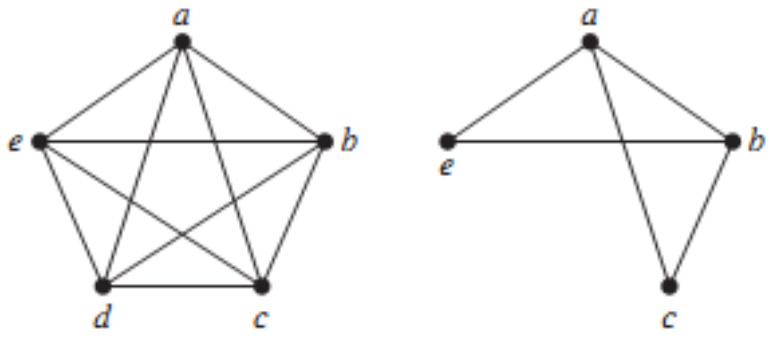
$G - v$.

Заметим, что $G - v = (V - v, E')$, где E' — множество ребер G, не инцидентных v.

Аналогично, если V' — подмножество V, то граф $G - V'$ является подграфом $(V - V', E')$, где E' — множество ребер G, не инцидентных вершине в V'.

Индуцированный граф

Определение 8 Пусть $G = (V, E)$ — простой граф. Подграф, индуцированный (индуцированный подграф) подмножеством W множества вершин V, — это граф (W, F), где множество ребер F содержит ребро из E: обе концевые вершины этого ребра находятся в W.



Если мы добавим ребро $\{c, e\}$ к графу G, мы получим подграф, индуцированный множеством вершин $W = \{a, b, c, e\}$.

Объединение графов($G_1 \cup G_2$)

Два или более графов можно объединить различными способами.

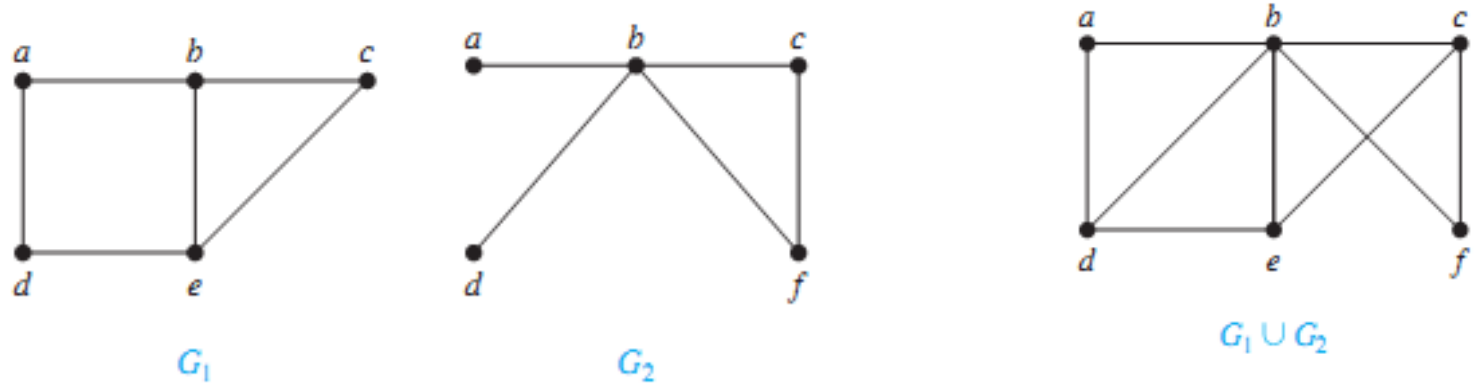
Новый граф, содержащий все вершины и ребра этих графов, называется объединением графов.

Объединение двух простых графов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ — это простой граф с множеством вершин $V_1 \cup V_2$ и множеством ребер $E_1 \cup E_2$.

Объединение G_1 и G_2 обозначается как $G_1 \cup G_2$.

ПРИМЕР Найти объединение графов G_1 и G_2

Решение: Множество вершин объединения $G_1 \cup G_2$ является объединением двух множеств вершин, а именно, {*a, b, c, d, e, f* }. Множество ребер объединения $G_1 \cup G_2$ является объединением двух множеств ребер.



(a) Простые графы G_1 и G_2;

(b) Их объединение $G_1 \cup G_2$.

Пересечение графов ($G_1 \cap G_2$)

Пересечение двух простых графов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ — это простой граф с множеством вершин $V_1 \cap V_2$ и множеством ребер $E_1 \cap E_2$.

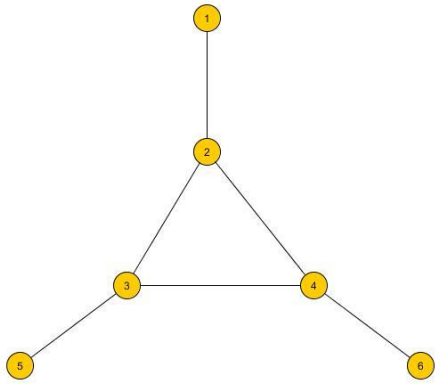
Пересечение G_1 и G_2 обозначается как $G_1 \cap G_2$.

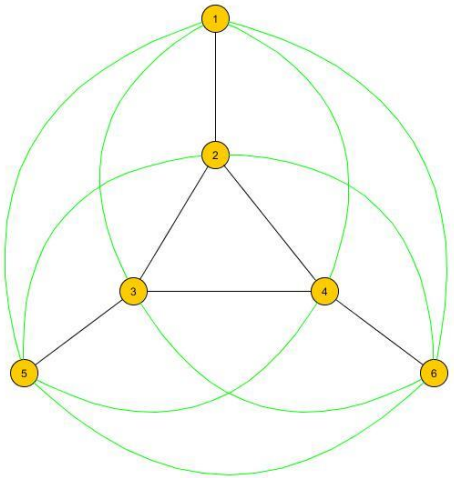
$\overline{G}$

Дополнение графа

Дополнение простого графа *G* граф *G*’ имеет те же вершины, что и *G*.

Две вершины смежны в *G*’ ⇔ они не смежны в *G*.





- Ваши вопросы?
- Контакты лектора:
  apanovich_09@mail.ru