# Теория графов: основные понятия и задачи. Графы как структура данных

* [**Что такое теория графов и что такое граф?**](https://function-x.ru/graphs1_relations.html#paragraph0)
* [**Основные понятия теории графов**](https://function-x.ru/graphs1_relations.html#paragraph1)
* [**Классические задачи теории графов и их решения**](https://function-x.ru/graphs1_relations.html#paragraph2)
* [**Задачи с графами для закрепления основных понятий**](https://function-x.ru/graphs1_relations.html#paragraph3)
* [**Теория графов и важнейшие современные прикладные задачи**](https://function-x.ru/graphs1_relations.html#paragraph4)

## Что такое теория графов и что такое граф?

**Теория графов** - один из обширнейших разделов дискретной математики, широко применяется в решении экономических и управленческих задач, в программировании, химии, конструировании и изучении электрических цепей, коммуникации, психологии, психологии, социологии, лингвистике, других областях знаний. **Теория графов** систематически и последовательно изучает свойства графов, о которых можно сказать, что они состоят из множеств точек и множеств линий, отображающих связи между этими точками. Основателем теории графов считается Леонард Эйлер (1707-1882), решивший в 1736 году известную в то время задачу о кёнигсбергских мостах.

**Графы строят** для того, чтобы отобразить отношения на [множествах](https://function-x.ru/sets1.html). Пусть, например, множество A = {a1, a2, ... an} - множество людей, а каждый элемент будет отображён в виде точки. Множество B = {b1, b2, ... bm} - множество связок (прямых, дуг, отрезков - пока не важно). На множестве A задано отношение знакомства между людьми из этого множества. **Строим граф** из точек и связок. Связки будут связывать пары людей, знакомых между собой. Естественно, число знакомых у одних людей может отличаться от числа знакомых у других людей, а некоторые вполне могут и не быть ни с кем знакомы (такие элементы будут точками, не соединёнными ни с одной другой). Вот и получился граф!



То, что мы сначала назвали "точками", следует называть вершинами графа, а то, что называли "связками" - рёбрами графа.

Теория графов не учитывает конкретную природу множеств A и B. Существует большое количество самых разных конкретных задач, при решении которых можно временно забыть о специфическом содержании множеств и их элементов. Эта специфика никак не сказывается на ходе решения задачи, независимо от её трудности! Например, при решении вопроса о том, можно ли из точки a добраться до точки e, двигаясь только по соединяющим точки линиям, неважно, имеем ли мы дело с людьми, городами, числами и т.д. Но, когда задача решена, мы получаем решение, верное для любого содержания, которое было смоделировано в виде графа. Не удивительно поэтому, что теория графов - один из самых востребованных инструментов при создании искусственного интеллекта: ведь искусственный интеллект может обсудить с собеседником и вопросы любви, и вопросы музыки или спорта, и вопросы решения различных задач, причем делает это без всякого перехода (переключения), без которого в подобных случаях не обойтись человеку.

А теперь строгие математические определения графа.

**Определение 1. Графом называется** система объектов произвольной природы (вершин) и связок (рёбер), соединяющих некоторые пары этих объектов.

**Определение 2.**Пусть V – (непустое) множество вершин, элементы v∈V – вершины. Граф G = G(V) с множеством вершин V есть некоторое cемейство пар вида: e = (a, b), где a,b∈V, указывающих, какие вершины остаются соединёнными. Каждая пара e = (a, b) - ребро графа. Множество U - множество рёбер e графа. Вершины a и b – концевые точки ребра e.

**Графы как структура данных.**Широким применением теории графов в компьютерных науках и информационных технологиях обусловлено добавлением к вышеизложенным определениям понятия графа как структуры данных. В компьютерных науках и информационных технологиях граф определяется как нелинейная структура данных. Что же тогда - линейная структура данных и чем от них отличаются графы? Линейные структуры данных характеризуются тем, что связывают элементы отношениями типа "простого соседства". Линейными структурами данных являются, например, массивы, таблицы, списки, очереди, стеки, строки. В противоположность им нелинейные структуры данных - такие, в которых элементы располагаются на различных уровнях иерархии и подразделяются на три вида: исходные, порождённые и подобные. Итак, граф - нелинейная структура данных.

Слово граф греческого происхождения, от слов "пишу", "описываю". Из начала этой статьи известно, что именно описывает граф: описывает он отношения. То есть, любой граф описывает отношения. И наоборот: любое отношение можно описать в виде графа.

## Основные понятия теории графов

Этот урок - вводный в теорию графов, поэтому лишь перечислим её основные понятия, заодно анонсируя другие уроки темы. Одно из центральных понятий теории графов, опираясь на которое строятся другие понятия - понятие инцидентности. Друг другу инцидентны две вершины графа, если они соединены ребром; вершина и ребро графа инцидентны, если вершина является началом или концом ребра. Более подробно [**виды вершин и рёбер графа исходя из понятия инцидентности представлены в отдельном уроке**](https://function-x.ru/graphs5.html).

А многие [**виды графов**](https://function-x.ru/graphs2_definitions_classes.html) определяются тем, какие виды рёбер и/или вершин эти графы содержат. На этом основаны и понятия ориентированного и неориентированного графов, которыми обязан владеть каждый освоивший дискретную математику вообще и теорию графов. Есть также графы, которые определяются некоторыми специфическими принципами построения, например, двудольные графы, которые разобраны на этом уроке в параграфе с задачами, а также на всё том же уроке о [**видах графов**](https://function-x.ru/graphs2_definitions_classes.html).

Понятие инцидентности необходимо и при составлении алгоритмов решения многих практических задач с графами. Например, можно ознакомиться с программной реализацией [**обхода в глубину графа, представленного матрицей инцидентности**](https://function-x.ru/cpp_graph1.html). Идея проста: можно двигаться лишь через вершины, соединённые рёбрами. А уж если рёбрам приписаны какие-то значения ("весы", чаще всего в виде чисел, такие графы называются взвешенными или помеченными), то можно решать сложные прикладные задачи, некоторые из которых упомянуты в завершающем параграфе этого урока.

## Классические задачи теории графов и их решения

Один из первых опубликованных примеров работ по теории графов и применения графов - работа о "задаче с Кёнигсбергскими мостами" (1736 г.), автором которой является выдающийся математик 18-го века Леонард Эйлер. В задаче даны река, острова, которые омываются этой рекой, и несколько мостов. Вопрос задачи: возможно ли, выйдя из некоторого пункта, пройти каждый мост только по одному разу и вернуться в начальный пункт? (рисунок ниже)



Задачу можно смоделировать следующим образом: к каждому участку суши прикрепляется одна точка, а две точки соединяются линией тогда и только тогда, когда соответствующие участки суши соединены мостом (рисунок ниже, соединительные линии начерчены пунктиром). Таким образом, построен граф.



Ответ Эйлера на вопрос задачи состоит в следующем. Если бы у этой задачи было положительное решение, то в получившемся графе существовал бы замкнутый путь, проходящий по рёбрам и содержащий каждое ребро только один раз. Если существует такой путь, то у каждой вершины должно быть только чётное число рёбер. Но в получившемся графе есть вершины, у которых нечётное число рёбер. Поэтому задача не имеет положительного решения.

По устоявшейся традиции эйлеровым графом называется граф, в котором можно обойти все вершины и при этом пройти одно ребро только один раз. В нём каждая вершина должна иметь только чётное число рёбер. Задача средней трудности на эйлеровы графы - в материале "[Основные виды графов](https://function-x.ru/graphs2_definitions_classes.html)".

В 1847 г. Кирхгоф разработал теорию деревьев для решения совместной системы линейных алгебраических уравнений, позволяющую найти значение силы тока в каждом проводнике (дуге) и в каждом контуре электрической цепи. Абстрагируясь от электрических схем и цепей, которые содержат сопротивления, конденсаторы, индуктивности и т.д., он рассматривал соответствующие комбинаторные структуры, содержащие только вершины и связи (рёбра или дуги), причём для связей не нужно учитывать, каким типам электрических элементов они соответствуют. Таким образом, Кирхгоф заменил каждую электрическую цепь соответствующим графом и показал, что для решения системы уравнений необязательно рассматривать в отдельности каждый цикл графа электрической цепи.

Кэли в 1858 г., занимаясь чисто практическими задачами органической химии, открыл важный класс графов, называемых деревьями. Он стремился перечислить изомеры насыщенных углеводородов, с данным числом атомов углерода. Кэли прежде всего сформулировал задачу абстрактно: найти число всех деревьев с p вершинами, каждое из которых имеет вершины со степенями 1 и 4. Ему не удалось сразу решить эту задачу, и он стал изменять её формулировку таким образом, чтобы можно было решить новую задачу о перечислении:

* корневых деревьев (в которых выделена одна из вершин);
* всех деревьев;
* деревьев, у которых степени вершин не превышают 4;
* деревьев, у которых степени вершин равны 1 и 4 (постановка задачи из химии).

## Задачи с графами для закрепления основных понятий

**Пример 1.**Пусть A - множество чисел 1, 2, 3: A = {1, 2, 3}. Построить граф для отображения отношения "<" ("меньше") на этом множестве.

Решение. Очевидно, что числа 1, 2, 3 следует представить в виде вершин графа. Тогда каждую пару вершин должно соединять одно ребро. Решая эту задачу, мы пришли к таким основным понятиям теории графов, как [**ориентированные и неориентированные графы**](https://function-x.ru/graphs2_definitions_classes.html). Неориентированные графы - такие, рёбра которых не имели направления. Или, как говорят ещё чаще, порядок двух концов ребра не существенен. В самом деле, граф, построенный в самом начале этого урока и отображавший отношение знакомства между людьми, не нуждается в направлениях рёбер, так как можно утверждать, что "человек номер 1" знаком с "человеком номер 2" в той же мере, как и "человек номер 2" с "человеком номер 1". В нашем же нынешнем примере одно число меньше другого, но не наоборот. Поэтому соответствующее ребро графа должно иметь направление, показывающее, какое всё же число меньше другого. То есть, порядок концов ребра существенен. Такой граф (с рёбрами, имеющими направление) называется ориентированным графом или орграфом.

Итак, в нашем множестве A число 1 меньше числа 2 и числа 3, а число 2 меньше числа 3. Этот факт отображаем рёбрами, имеющими направление, что показывается стрелками. Получаем следующий граф:



**Пример 2.**Пусть A - множество чисел 2, 4, 6, 14: A = {2, 4, 6, 14}. Постоить граф для отображения отношения "делится нацело на" на этом множестве.

Решение. В этом примере часть рёбер будут иметь направление, а некоторые не будут, то есть строим [**смешанный граф**](https://function-x.ru/graphs2_definitions_classes.html). Перечислим отношения на множестве: 4 делится нацело на 2, 6 делится нацело на 2, 14 делится нацело на 2, и ещё каждое число из этого множества делится нацело на само себя. Это отношение, то есть когда число делится нацело на само себя, будем отображать в виде рёбер, которые соединяют вершину саму с собой. Такие рёбра называются [**петлями**](https://function-x.ru/graphs5.html). В данном случае нет необходимости давать направление петле. Таким образом, в нашем примере три обычных направленных ребра и четыре петли. Получаем следующий граф:



**Пример 3.**Пусть даны множества A = {α, β, γ} и B = {a, b, c}. Построить граф для отображения отношения "декартово произведение множеств".

Решение. Как известно из определения [**декартова произведения множеств**](https://function-x.ru/sets1.html#paragraph4), в нём нет упорядоченных наборов из элементов одного и того же множества. То есть в нашем примере нельзя соединять греческие буквы с греческими и латинские с латинскими. Этот факт отображается в виде [**двудольного графа**](https://function-x.ru/graphs2_definitions_classes.html), то есть такого, в котором вершины разделены на две части так, что вершины, принадлежащие одной и той же части, не соединены между собой. Получаем следующий граф:



[**Нет времени вникать в решение? Можно заказать работу!**](https://function-x.ru/homework.html)

**Пример 4.**В агентстве по недвижимости работают менеджеры Игорь, Сергей и Пётр. Обслуживаются объекты О1, О2, О3, О4, О5, О6, О7, О8. Построить граф для отображения отношений "Игорь работает с объектами О4, О7", "Сергей работает с объектами О1, О2, О3, О5, О6", "Пётр работает с объектом О8".

Решение. Граф, отображающий данные отношения, будет так же двудольным, так как менеджер не работает с менеджером и объект не работает с объектом. Однако, в отличии от предыдущего примера, граф будет ориентированным. В самом деле, например, Игорь работает с объектом О4, но не объект О4 работает с Игорем. Часто, когда такое свойство отношений очевидно, необходимость давать рёбрам направления может показаться "математической тупостью". Но всё же, и это вытекает из строгого характера математики, если отношение носит односторонний характер, то давать направления рёбрам нужно. В приложениях отношений эта строгость окупается, например, в программах, предназначенных для планирования, где тоже применяются графы и маршрут по вершинам и рёбрам должен проходить строго в заданном направлении. Итак, получаем следующий ориентированный двудольный граф:



И вновь к примерам с числами.

**Пример 5.**Пусть задано множество C = {2, 3, 5, 6, 15, 18}. Построить граф, реализующий отношение, определяющее все пары чисел a и b из множества C, у которых при делении второго элемента на первый получаем частное, которое является целым числом больше 1.

Решение. Граф, отображающий данные отношения, будет ориентированным, так как в условии есть упоминание о втором и первом элементе, то есть, ребро будет направлено от первого элемента ко второму. Из этого однозначно понятно, какой элемент является перым, а какой вторым. Ещё добавим терминологии: ориентированные рёбра принято называть дугами. В нашем графе будет 7 дуг: e1 = (3, 15), e2 = (3, 18), e3 = (5, 15), e4 = (3, 6), e5 = (2, 18), e6 = (6, 18), e7 = (2, 6). В этом примере рёбра (дуги) графа просто пронумерованы, но порядковые номера - не единственное, что можно приписать дуге. Дуге можно приписать также весы означающие, например, стоимость пересылки груза из одного пункта в другой. Но с весами дуг мы познакомимся позже и подробнее. Итак, получаем следующий ориентированный граф:



Как мы уже знаем из теоретической вступительной части, теория графов не учитывает специфическую природу множеств и с помощью одного и того же графа можно задать отношения на множествах с самым разным содержанием. То есть, от этого самого содержания при моделировании задачи можно абстрагироваться. Перейдём к примерам, иллюстрирующим это замечательное свойство теории графов.

**Пример 6.**На кусочке шахматной доски размером 3 Х 3 размещены два белых коня и два чёрных коня так, как показано на рисунке ниже.



Можно ли переместить коней в состояние, которое изображено на следующем рисунке, не забывая, что две фигуры не могут находиться на одной клетке?



Решение. В конструируемом графе пары вершин будут связаны отношением "ход коня". То есть, одна вершина - та, из которой конь ушёл, а другая - та, в которую пришёл, а промежуточная клетка буквы "г" будет за пределами этого отношения. Получаем следующий граф:



И всё же конструкция получилась громозкой. В ней видны клетки шахматной доски, а многие рёбра графа пересекаются. Нельзя ли абстрагироваться от физического вида шахматной доски и вообразить отношения проще? Оказывается, можно. В новом графе соседними вершинами будут те, которые связаны отношением "ход коня", а не соседние по шахматной доске (рисунок ниже).



Теперь легко увидеть, что ответ на вопрос этой задачи - отрицательный. В начальном состоянии между двумя белыми конями нет чёрного коня, а в конечном состоянии этот чёрный конь должен быть. Рёбра графа размещены так, что два находящихся рядом коня не могут перепрыгнуть друг через друга.

**Пример 7.**Задача о волке, козе и капусте. На одном берегу реки находятся человек (Ч), лодка, волк (В), коза (Кз) и капуста (Кп). В лодке одновременно могут находиться человек и не более одного из перевозимых объектов. Человек должен перевезти на другой берег все объекты, соблюдая условие: нельзя оставлять без присмотра волка вместе с козой и козу вместе с капустой.

Решение. В конструируемом графе вершины - конфигурации, а рёбра - отношение "связь одним плаваньем лодки" между конфигурациями. Конфигурация означает расположение объектов на первоначальном берегу и на противоположном берегу. Каждая конфигурация отображается в виде (A|B), где A - объекты, находящиеся на первоначальном берегу, а B - объекты, находящиеся на противоположном берегу. Первоначальная конфигурация, таким образом, - (ЧВКпКз|). Например, после переправки на другой берег козы конфигурация будет (ВКп|ЧКз). Конечная конфигурация всегда (|ЧВКпКз). Теперь можем построить граф, зная уже, что означают вершины и рёбра:



Разместим вершины графа так, чтобы рёбра не пересекались, а соседними были вершины, которые связаны отношением на графе. Тогда увидеть отношения будет намного проще (для увеличения рисунка щёлкните по нему левой кнопкой мыши):



Как видим, существуют два различных непрерывных маршрута из начальной конфигурации в конечную. Поэтому задача имеет два различных решения (и оба правильные).

## Теория графов и важнейшие современные прикладные задачи

На основе теории графов разработаны методы решения прикладных задач, в которых в виде графов моделируются весьма сложные системы. В этих моделях узлы содержат отдельные компоненты, а рёбра отображают связи между компонентами. Обычно для моделирования транспортных сетей, систем массового обслуживания, в сетевом планировании используются взвешенные графы. Мы о них уже говорили, это графы, в которым дугам присвоены весы.

Графы-деревья применяются, например, для построения [**деревьев решений**](https://function-x.ru/graphs4_modeling_decision_tree_game_tree.html) (служат для анализа рисков, анализа возможных приобретений и убытков в условиях неопределённостей). С применением теории графов разработаны и [**другие многочисленные математические модели**](https://function-x.ru/graphs4_modeling_decision_tree_game_tree.html#paragraph5) для решения задач в конкретных предметных областях.

### Графы и задача о потоках

Постановка задачи. Имеется система водопроводных труб, представленная графом на рисунке ниже.



Каждая дуга графа отображает трубу. Числа над дугами (весы) - пропускная способность труб. Узлы - места соединения труб. Вода течёт по трубам только в одном направлении. Узел S - источник воды, узел T - сток. Требуется максимизировать объём воды, протекающей от источника к стоку.

Для решения задачи о потоках можно воспользоваться методом Форда-Фулкерсона. Идея метода: поиск максимального потока производится по шагам. В начале работы алгоритма поток полагается равным нулю. На каждом последующем шаге значение потока увеличивается, для чего ищется дополняющий путь, по которому поступает дополнительный поток. Эти шаги повторяются до тех пор, пока существуют дополнительные пути. Задача успешно применяется в различных распределённых системах: система электоснабжения, коммуникационная сеть, система железных дорог и других.

### Графы и сетевое планирование

В задачах планирования сложных процессов, состоящих из множества работ, часть из которых выполняется параллельно, а часть последовательно, широкое применение получили взвешенные графы, известные под названием сети ПЕРТ (PERT).

PERT - Program (Project) Evaluation and Review Technique - техника оценки и анализа программ (проектов), которая используется при управлении проектами.

Сеть ПЕРТ - взвешенный ациклический ориентированный граф, в котором каждая дуга представляет работу (действие, операцию), а вес дуги - время, требуемое для её выполнения.

Если в сети есть дуги (a, b) и (b, c), то работа, представленная дугой (a, b), должна быть завершена до начала выполнения работы, представленной дугой (b, c). Каждая вершина (vi) представляет момент времени, к которому должны быть завершены все работы, задаваемые дугами, оканчивающимися в вершине (vi).

В таком графе:

* одна вершина, не имеющая предшественников, определяет момент времени начала выполнения работ;
* одна вершина, не имеющая последователей, соответствует моменту времени завершения комплекса работ.

Путь максимальной длины между этими вершинами графа (из начала в конец процесса выполнения работ), называется критическим путём. Для сокращения времени выполнения всего комплекса работ необходимо найти работы, лежащие на критическом пути, и уменьшить их продолжительность за счёт, например, привлечения дополнительных исполнителей, механизмов, новых технологий.