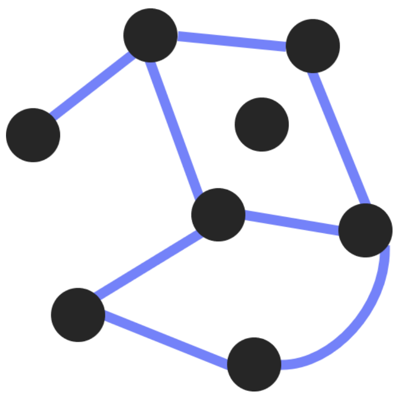
Специальная математика и основы статистики

Графы и деревья

Вопрос 1. Основные виды графов

Граф – это топологическая модель, которая состоит из множества вершин и множества соединяющих их рёбер. При этом значение имеет только сам факт, какая вершина с какой соединена.



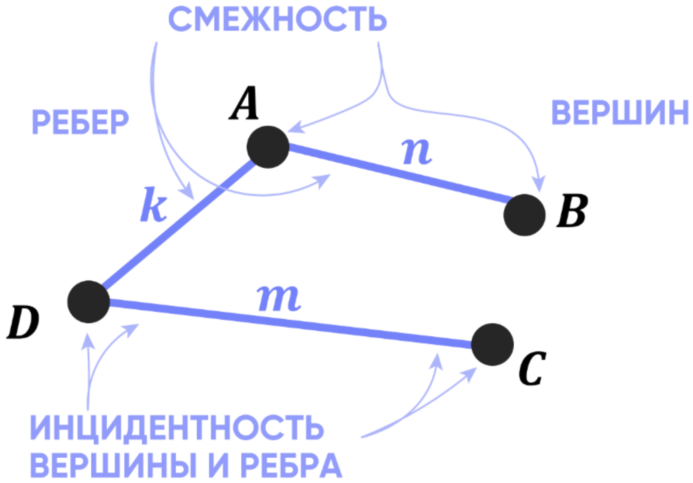
ГРАФ ВЕРШИНА

$$G = \{v, E\}$$

РЕБРА

Вершина – точка в графе, отдельный объект, для топологической модели графа не имеет значения координата вершины, её расположение, цвет, вкус, размер; однако при решении некоторых задачах вершины могут раскрашиваться в разные цвета или сохранять числовые значения.
Ребро – неупорядоченная пара двух вершин, которые связаны друг с другом. Эти вершины называются ***концевыми точками*** или концами ребра. При этом важен сам факт наличия связи, каким именно образом осуществляется эта связь и по какой дороге – не имеет значения; однако ребрк может быть присвоен «вес», что позволит говорить о «нагруженном графе» и решать задачи оптимизации.
Инцидентность – вершина и ребро называются инцидентными, если вершина является для этого ребра концевой.
Смежность вершин – две вершины называются смежными, если они инцидентны одному ребру.
Смежность рёбер – два ребра называются смежными, если они инцедентны одной вершине.

Говоря проще: две вершины смежные, если они соединены ребром; два ребра смежные, если они соединены вершиной.



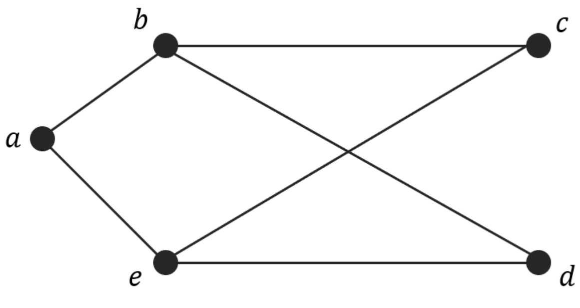
Таким образом, мы можем представлять себе множество E ребер как множество пар смежных вершин, определяя тем самым нерефлексивное, симметричное отношение на множестве V.

Логическая матрица отношения на множестве вершин графа, которое задается его ребрами, называется ***матрицей смежности***.

Симметричность отношения в терминах матрицы смежности M означает, что M симметрична относительно главной диагонали. А из-за нерефлексивности этого отношения на главной диагонали матрицы M стоит символ «Л».

Пример: Нарисуйте граф $G(V, E)$ с множеством вершин V = {a, b, c, d, e} и множеством ребер E = {ab, ae, bc, bd, ce, de}. Выпишите его матрицу смежности.

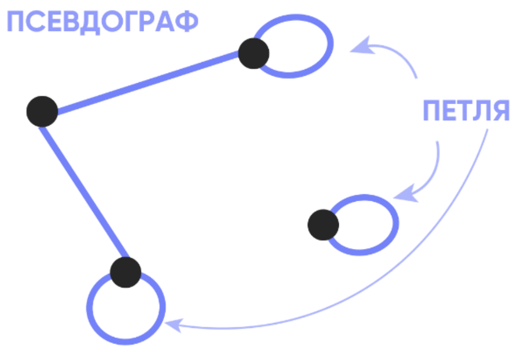
Решение. Граф G представлен на рисунке

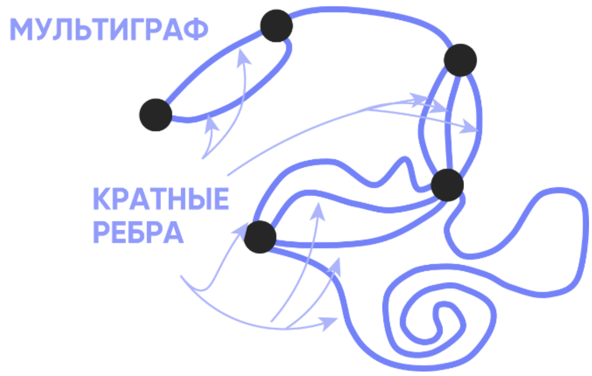


Его матрица смежности имеет вид: $\begin{array}{c c} & \begin{array}{ccccc} \boldsymbol{a} & \boldsymbol{b} & \boldsymbol{c} & \boldsymbol{d} & \boldsymbol{e} \end{array} \\ \begin{array}{c} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{c} \\ \boldsymbol{d} \\ \boldsymbol{e} \end{array} & \begin{pmatrix} Л & И & Л & Л & И \\ И & Л & И & И & Л \\ Л & И & Л & Л & И \\ Л & И & Л & Л & И \\ И & Л & И & И & Л \end{pmatrix} \end{array}$

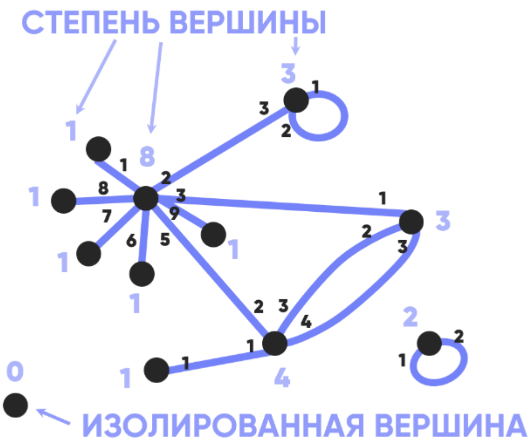
Петля – ребро, инцидентное одной вершине. Ребро, которое замыкается на одной вершине.
Псевдограф – граф с петлями. С такими графами не очень удобно работать, потому что переходя по петле мы остаёмся в той же самой вершине, поэтому у него есть своё название.
Кратные рёбра – рёбра, имеющие одинаковые концевые вершины, по-другому их называют ещё параллельными.
Мультиграф – граф с кратными рёбрами.
Псевдомультиграф – граф с петлями и кратными рёбрами.

Степень вершины – это количество рёбер, инцидентных указанной вершине. По-другому – количество рёбер, исходящих из вершины.

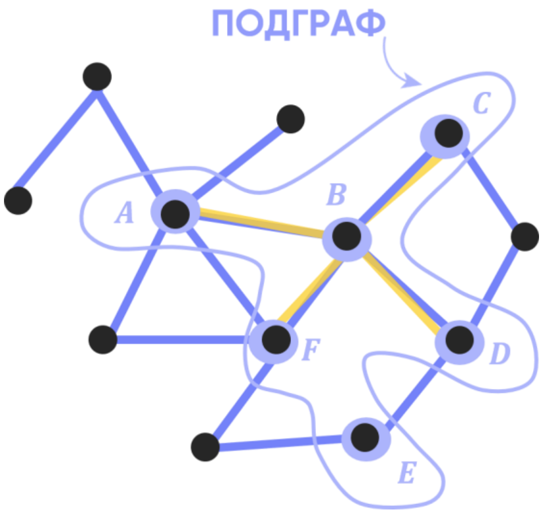




Петля увеличивает степень вершины на 2.
Изолированная вершина – вершина с нулевой степенью.
Висячая вершина – вершина со степенью 1.



Если в исходном графе выделить несколько вершин и несколько рёбер (между выбранными вершинами), то мы получим ***подграф*** исходного графа.

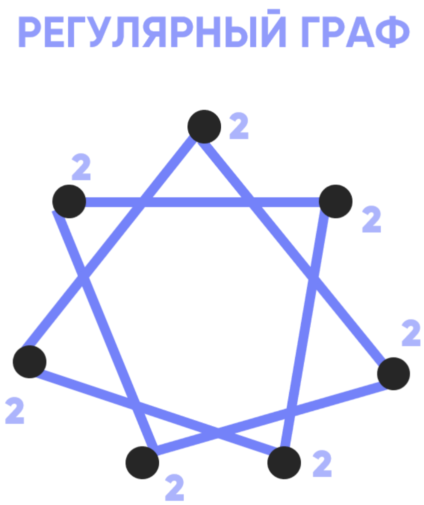


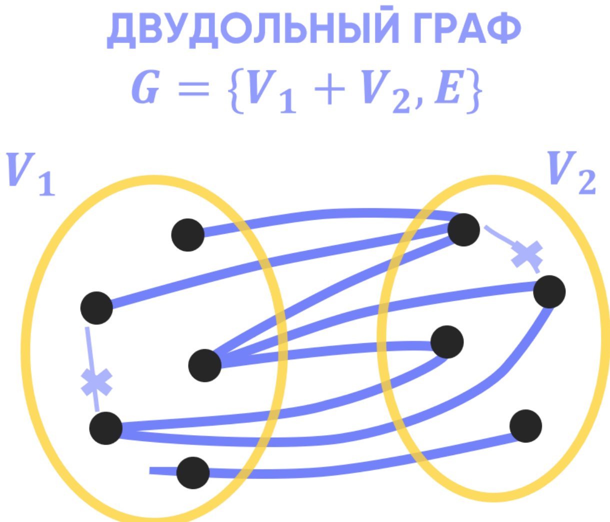
Полный граф – это граф, в котором каждые две вершины соединены одним ребром.
Пример (Задача о рукопожатиях): собралось N человек (вершин) и каждый с каждым обменялся рукопожатием (ребро), сколько всего было рукопожатий?
Решение. Вычисляется как сумма чисел от 1 до N – каждый новый участник должен пожать руку всем присутствующим, вычисляется по формуле: $\frac{N(N-1)}{2}$.

■ ■ ■

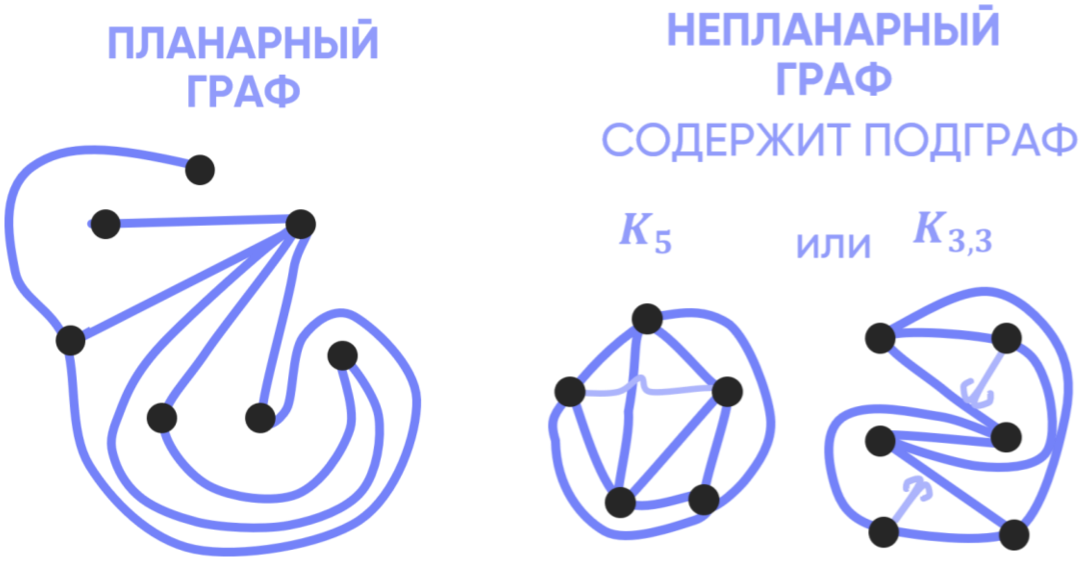
Регулярный граф – граф, в котором степени всех вершин одинаковые.
Двудольный граф – если все вершины графа можно разделить на два множества таким образом, что каждое ребро соединяет вершины из разных множеств, то такой граф называется двудольным.
Пример: клиент-серверное приложение содержит множество запросов (рёбер) между клиентом и сервером, но нет запросов внутри клиента или внутри сервера.

■ ■ ■

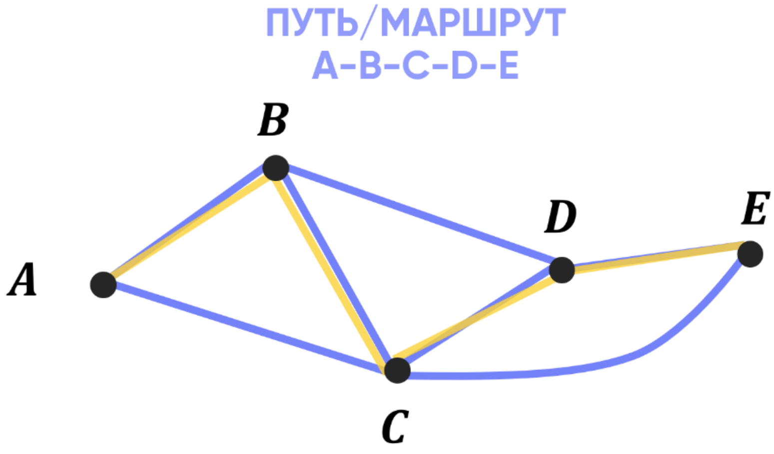




Если граф можно разместить на плоскости таким образом, чтобы рёбра не пересекались, то он называется ***планарным*** или ***плоским графом***. Если это невозможно сделать, то граф называется ***непланарным***. *Минимальные непланарные графы* – это полный граф К5 из 5 вершин и полный двудольный граф К3,3 из 3+3 вершин. Если какой-либо граф в качестве подграфа содержит К5 или К3,3, то он является непланарным.

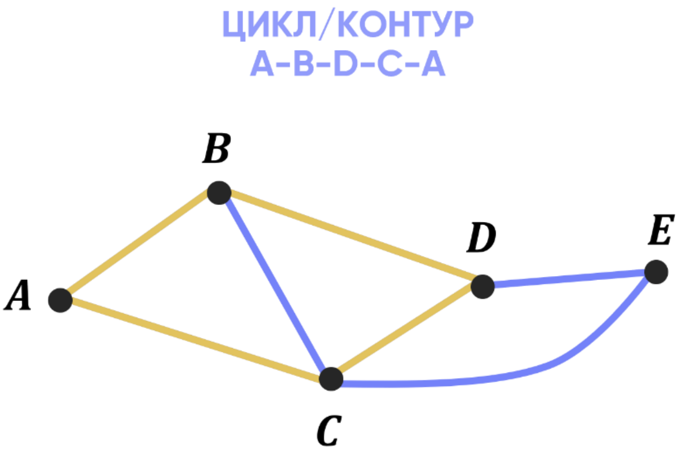


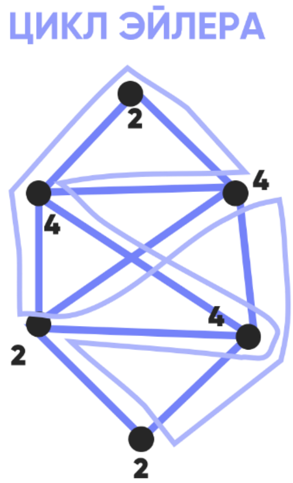
Путь или ***маршрут*** – это последовательность смежных рёбер. Обычно путь задаётся перечислением вершин, по которым он пролегает.
Длина пути – количество рёбер в пути.
Цепь – маршрут без повторяющихся рёбер.
Простая цепь – цепь без повторяющихся вершин.

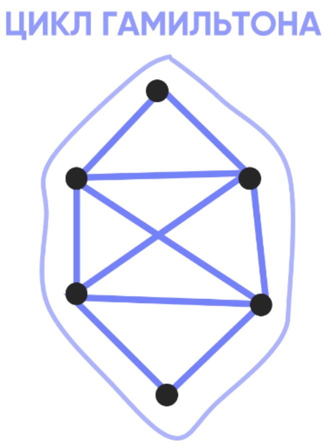


Цикл или ***контур*** – цепь, в котором последняя вершина совпадает с первой.
Длина цикла – количество рёбер в цикле.
Самый короткий цикл – это петля.

Цикл Эйлера – цикл, проходящий по каждому ребру ровно один раз. Эйлер доказал, что такой цикл существует тогда, и только тогда, когда все вершины в связанном графе имеют чётную степень.
Цикл Гамильтона – цикл, проходящий через все вершины графа по одному разу. Другими словами – это простой цикл, в который входят все вершины графа.



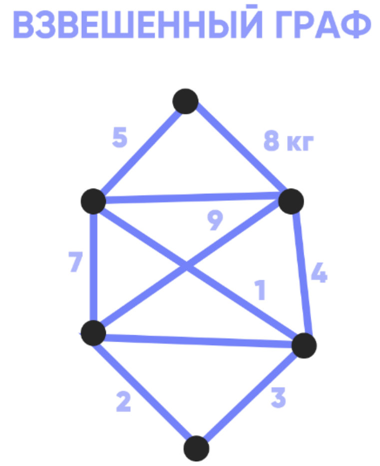




Гамильтоновы графы применяются для моделирования многих практических задач. Основой всех таких задач служит классическая **задача коммивояжёра:** коммивояжёр должен совершить поездку по городам и вернуться обратно, побывав в каждом городе ровно один раз, сведя при этом затраты на передвижения к минимуму.

■ ■ ■

Взвешенный граф – граф, в котором у каждого ребра и/или каждой вершины есть «вес» – некоторое число, которое может обозначать длину пути, его стоимость и т. п. Для взвешенного графа составляются различные алгоритмы оптимизации, например поиск кратчайшего пути.

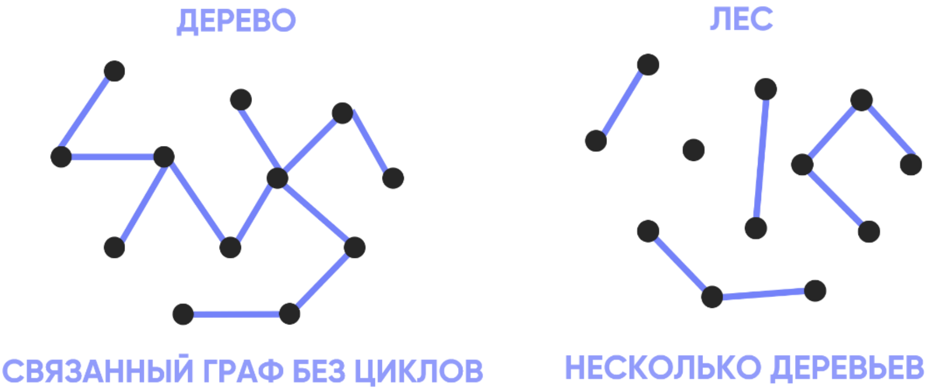


Связный граф – граф, в котором существует путь между любыми двумя вершинами.

Вопрос 2. Деревья

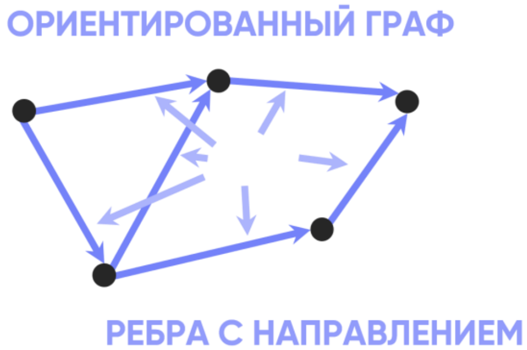
Дерево – связный граф без циклов.
Между любыми двумя вершинами дерева существует единственный путь.
Пусть *G* = (V, *E)* – граф с *n* вершинами и *m* ребрами. Можно сформулировать несколько необходимых и достаточных условий, при которых *G* является деревом:
• Любая пара вершин в *G* соединена единственным путем.
• *G* связен и *m*=*n*-*1*.
• *G* связен, а удаление хотя бы одного его ребра нарушает связность графа.
• *G* ацикличен, но если добавить хотя бы одно ребро, то в G появится цикл.
Несложно доказать, что в любом связном графе найдется подграф, являющийся деревом. Подграф в G, являющийся деревом и включающий в себя все вершины G, называется ***остовным деревом***.
Остовное дерево в графе *G* строится просто: выбираем произвольное его ребро и последовательно добавляем другие ребра, не создавая при этом циклов, до тех пор, пока нельзя будет добавить никакого ребра, не получив при этом цикла.
Задачи поиска кратчайшего соединения: нужно построить железнодорожную сеть, связывающую некоторое число городов. Известна стоимость строительства отрезка путей между любой парой городов. Требуется найти сеть минимальной стоимости.
На языке теории графов нам нужно в нагруженном графе найти остовное дерево наименьшего общего веса. Такое дерево принято называть ***минимальным остовным деревом*** или, сокращенно, МОД.

Деревья часто используются для организации иерархической структуры данных, например, при создании двоичных деревьев поиска или кучи, в этом случае одну вершину дерева называют корнем.
Лес – граф, в котором несколько деревьев.



Вопрос 3. Ориентированный граф

Ориентированный граф или ***орграф*** – граф, в котором рёбра имеют направления.
Другими словами, орграф представляет собой пару G=(F, E), где V – конечное множество вершин, а E – отношение на V.
Дуга – направленные рёбра в ориентированном графе.
Дугу, соединяющую пару (u, v) вершин *u* и *v* орграфа G, будем обозначать через *uv*. В простом орграфе отсутствуют петли и кратные дуги. Следовательно, для любой пары вершин *u* и *v* в орграфе найдется не более одной дуги *uv* из вершины *u* в *v*, и не более одной дуги *uv* из *v* в *u*. Если *uv* – дуга орграфа, то *u* называют ***антецедентом*** *v*.



Путем длины k в орграфе называют последовательность различных вершин $v_0, \dots v_k$, каждая пара $v_{i-1}v_i$ которой образует дугу ($i = \overline{1,k}$).
Контуром в орграфе G принято называть последовательность вершин $v_0, \dots v_k$, образующую путь, в которой первая вершина совпадает v_0 с последней v_k, а других повторяющихся вершин в ней нет.
Орграф G называют ***бесконтурным***, если в нем нет контуров.
Кратчайший путь – это путь минимального общего веса, соединяющий выбранные вершины. ***Общий вес***, по определению, равен сумме весов всех дуг, составляющ;их путь. Общий вес

кратчайшего пути, ведущего из вершины u в вершину *v*, называют ***расстоянием от u до v.***

Определим ***весовую матрицу*** *w*, чьи элементы *w{u, v)* задаются формулой $w(u,v)=\begin{cases}0, \text{если } u=v\\ \infty, \text{если } u \text{ и } v \text{ не соединены дугой}\\ d, \text{если } uv - \text{дуга веса } d\end{cases}$

В течение работы алгоритма каждой вершине *v* орграфа присваивается число *d[v]* равное расстоянию от вершины *A* до *v*.

Алгоритм Дейкстры может найти кратчайший путь между вершинами *u* и *v* в графе, только если существует хотя бы один путь между этими вершинами. Если это условие не выполняется, то алгоритм отработает корректно, вернув значение "бесконечность" для пары несвязанных вершин.

Алгоритм:

1) задаём множество X={u}, состоящее из исходной вершины.

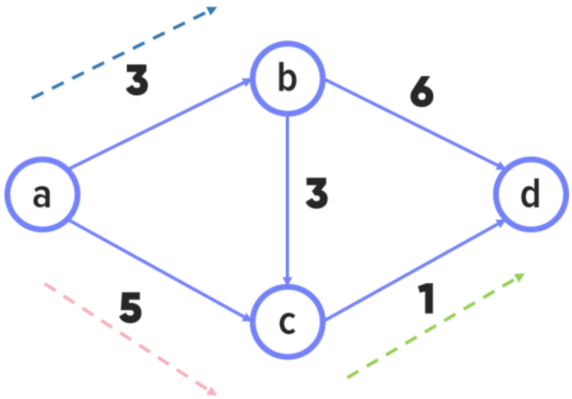
2) массив длин кратчайших путей А, в котором изначально есть А[u]=0, кратчайших путь от вершины до себя самой.

3) пока все вершины не исследованы (или формально X≠V), повторяем:

- среди всех рёбер в графе (*v*,*w*) таких, что $v \in X$, а $w \notin X$, выбираем одно, которое минимизирует сумму: $A[v]+d_{vw}$;
- добавляем эту вершину *w* в X;
- задаём А[*w*] равным $A[v]+d$

4) В итоге исполнения этого алгоритма, массив А будет содержать все оптимальные пути, исходящие из u.

Пример: Найти кратчайшие пути от вершины а до всех остальных



Решение.

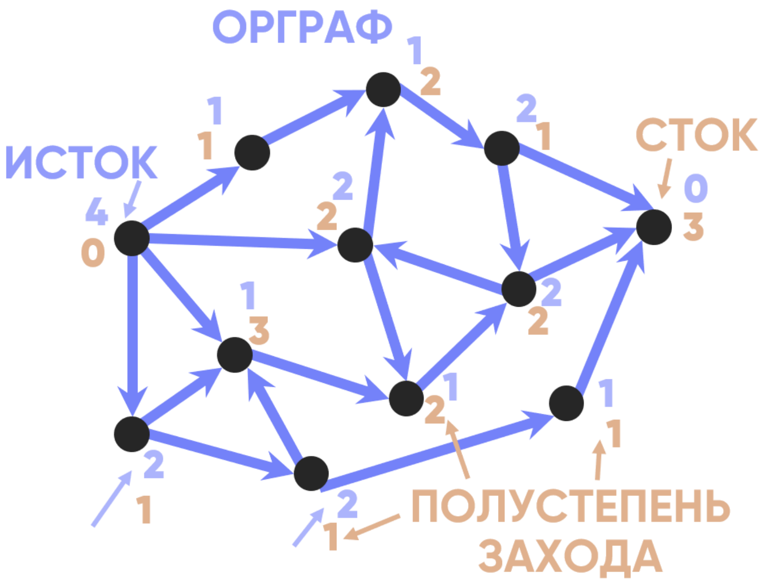
Первый шаг алгоритма определит, что кратчайший путь до b проходит по направлению синей стрелки и зафиксирует кратчайший путь.

Второй шаг рассмотрит, все возможные варианты $A[v]+d_{vw}$ и окажется, что оптимальный вариант двигаться вдоль красной стрелки, поскольку 5 меньше, чем 3+3=6 и 3+6=9. Добавляется длина кратчайшего пути до с.

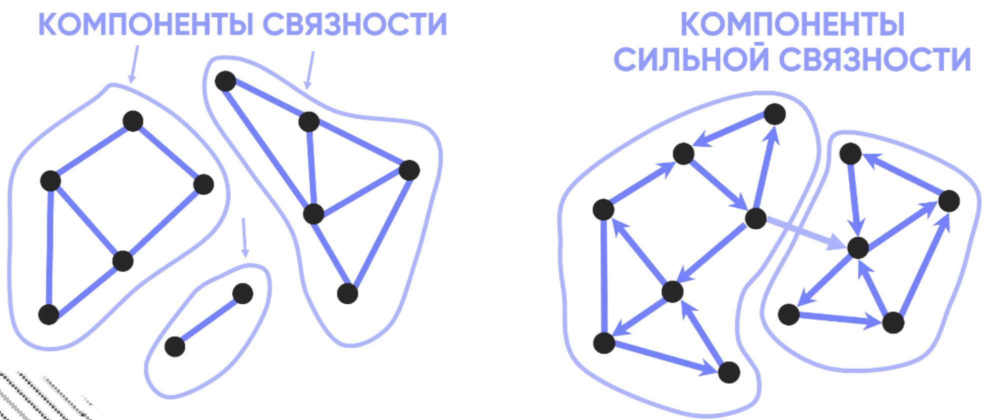
И наконец, третьим шагом, когда три вершины a,b,c уже лежат в X, остается рассмотреть только два ребра и выбрать, лежащее вдоль зеленой стрелки.

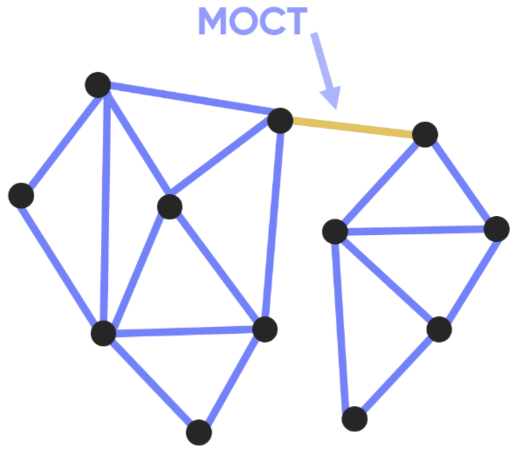
■■■

Полустепень захода вершины – количество дуг, заходящих в эту вершину.
Исток – вершина с нулевой полустепенью захода.
Полустепень исхода вершины – количество дуг, исходящих из этой вершины
Сток – вершина с нулевой полустепенью исхода.



Компонента связности – множество таких вершин графа, что между любыми двумя вершинами существует маршрут.
Компонента сильной связности – максимальное множество вершин орграфа, между любыми двумя вершинами которого существует путь по дугам.
Компонента слабой связности – максимальное множество вершин орграфа, между любыми двумя вершинами которого существует путь по дугам без учёта направления (по дугам можно двигаться в любом направлении).
Мост – ребро, при удалении которого, количество связанных компонент графа увеличивается.




МОСТ