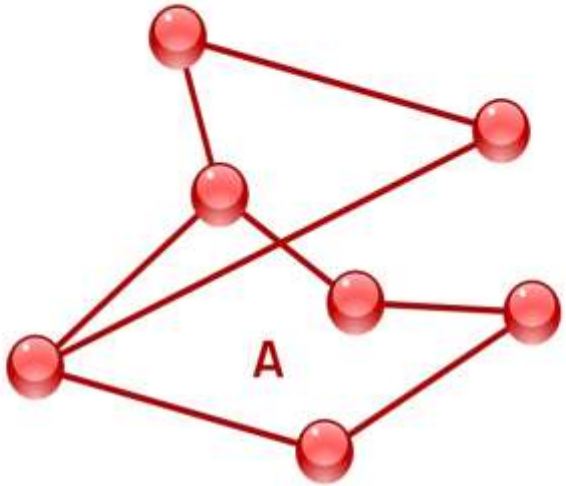
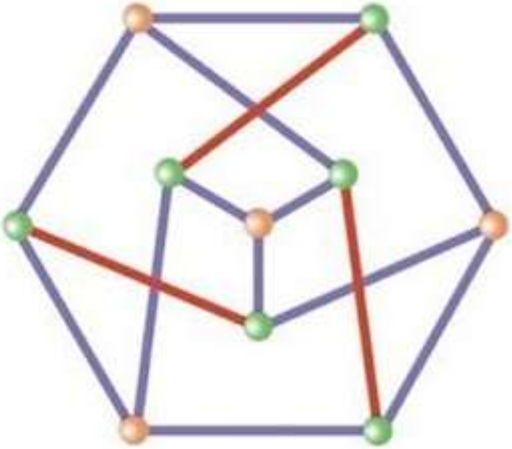


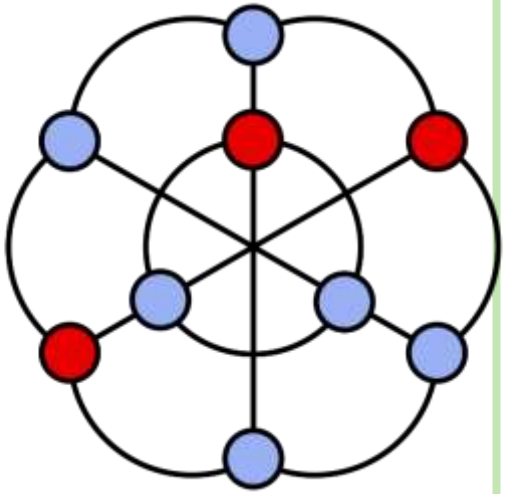
Теория графов



План лекции

1) Граф, его представление и элементы графа
2) Виды графов
3) Степень вершины графа
4) Матричное представление графов
   - матрица смежности
   - матрица инцидентности
   - весовая матрица
5) Задача о кёнигсбергских мостах и теоремы Эйлера
6) Эйлеровы графы
7) Гамильтоновы графы
8) Поиск кратчайшего пути на графе
9) Алгоритм Дейкстры
10) Задачи из теории графов

Теория графов – обширный раздел дискретной математики, изучающий свойства графов



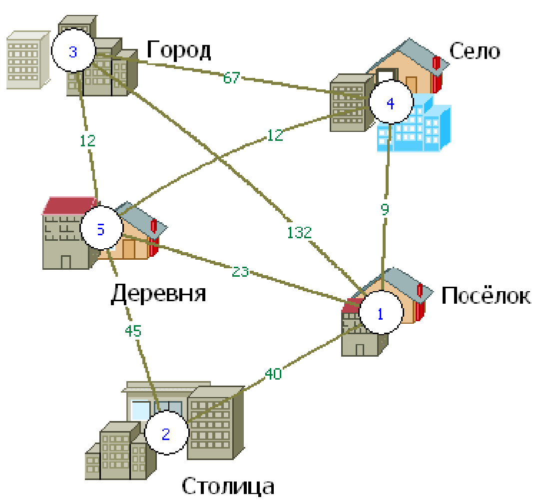
Теория графов применяется:

- в информатике для поиска кратчайших путей,
- в коммуникационных системах для маршрутизации данных,
- в экономике для решения задач оптимизации,
- в логистике для поиска оптимальных маршрутов,
- в химии для описания структур молекул и сложных реакций.

Применение теории графов:

Используется при проектировании компьютерных сетей, трубопроводов, строительстве дорог для минимизации затрат на прокладку коммуникаций.

Пример Существующие или вновь проектируемые сооружения рассматриваются как вершины, а соединяющие их дороги — как рёбра. Нужно найти кратчайший путь или спланировать оптимальный маршрут. Вычисления на таком графе, позволяет это сделать



Графом называется конечное множество вершин, некоторые из которых соединены ребрами.

Состав графа

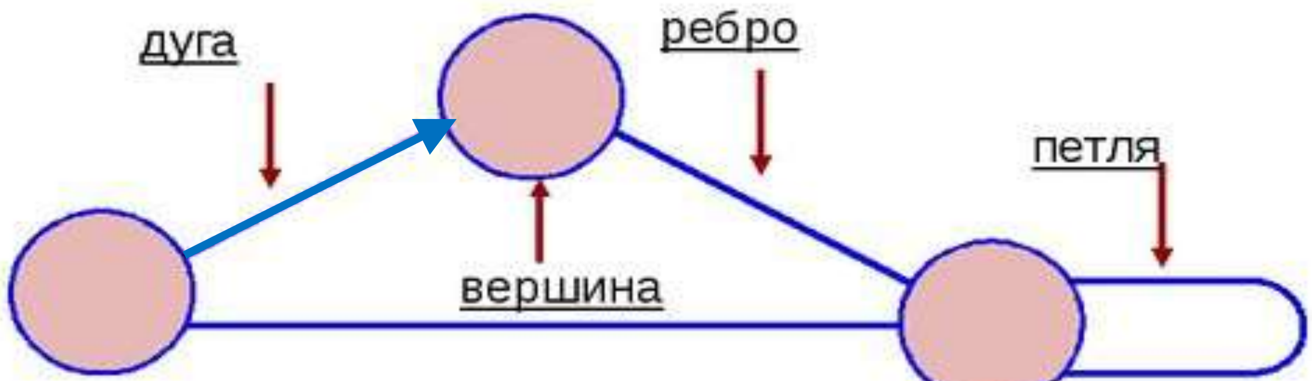
Вершина - 

Дуга - направленная линия (со стрелкой)

Ребром - линия ненаправленная (без стрелки)

Петля - линия, выходящая из некоторой вершины и входящая в неё же.

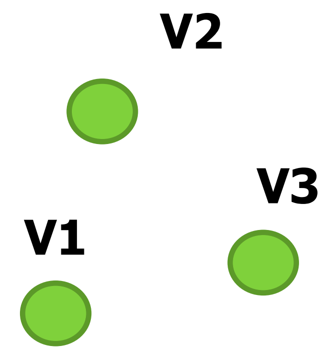
Укажите элементы графа на рисунке:



Проверь себя!

Описание графа

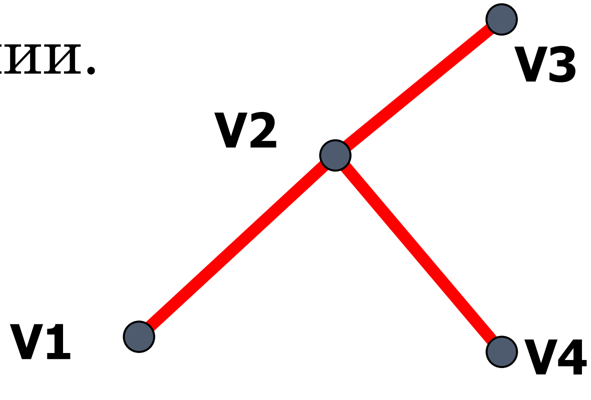
Граф описывается с помощью **конечного** множество вершин ***V*** и множества ребер ***R***, соединяющих пары вершин, **G=(V,R).**



- Множество ребер может быть пустым.

Примеры вершин – объекты любой природы (населенные пункты, горда, компьютеры в сети).

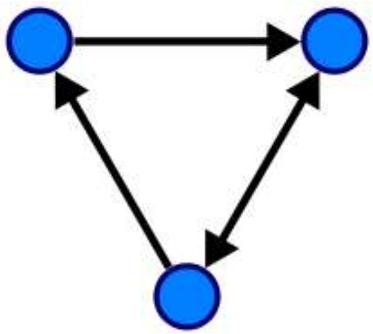
Примеры ребер – дороги, линии.



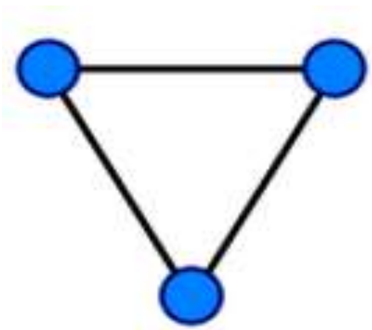

РЕЧНОЙ ВОКЗАЛ
ВОДНЫЙ СТАДИОН
ВОЙКОВСКАЯ
СОКОЛ
АЭРОПОРТ
ДИНАМО
САВЕЛОВСКАЯ
МЕНДЕЛЕЕВСКАЯ
МЕДВЕДКОВО
БАБУШКИНСКАЯ
СВИБЛОВО
БОТАНИЧЕСКИЙ САД
ВДНХ
ЩЕРБАКОВСКАЯ
РИЖСКАЯ
УЛИЦА ПОДБЕЛЬСКОГО
ЧЕРКИЗОВСКАЯ
ПРЕОБРАЖЕНСКАЯ ПЛОЩАДЬ
СОКОЛЬНИКИ
КРАСНОСЕЛЬСКАЯ
ПЛАНЕРНАЯ
СХОДНЕНСКАЯ
ТУШИНСКАЯ
ЩУКИНСКАЯ
ОКТЯБРЬСКОЕ ПОЛЕ
ПОЛЕЖАЕВСКАЯ
БЕГОВАЯ
УЛИЦА 1905 ГОДА
БЕЛОРУССКАЯ
НОВОСЛОБОДСКАЯ
ПРОСПЕКТ МИРА
ЩЕЛКОВСКАЯ
ПЕРВОМАЙСКАЯ
ИЗМАЙЛОВСКАЯ
ИЗМАЙЛОВСКИЙ ПАРК
СЕМЕНОВСКАЯ
ЭЛЕКТРОЗАВОДСКАЯ
БАУМАНСКАЯ
КРЫЛАТСКОЕ
МОЛОДЕЖНАЯ
КУНЦЕВСКАЯ
ПИОНЕРСКАЯ
ФИЛЕВСКИЙ ПАРК
БАГРАТИОНОВСКАЯ
ФИЛИ
КУТУЗОВСКАЯ
СТУДЕНЧЕСКАЯ
БАРРИКАДНАЯ
КРАСНОПРЕСНЕНСКАЯ
ЦВЕТНОЙ БУЛЬВАР
КОЛХОЗНАЯ
КОМСОМОЛЬСКАЯ
ТУРГЕНЕВСКАЯ
КИРОВСКАЯ
КРАСНЫЕ ВОРОТА
МАЯКОВСКАЯ
ДЗЕРЖИНСКАЯ
КУЗНЕЦКИЙ МОСТ
ПУШКИНСКАЯ
ЧЕХОВСКАЯ
ГОРЬКОВСКАЯ
КИЕВСКАЯ
КУРСКАЯ
НОВОГИРЕЕВО
ПЕРОВО
ШОССЕ ЭНТУЗИАСТОВ
АВИАМОТОРНАЯ
ПЛОЩАДЬ ИЛЬИЧА
СМОЛЕНСКАЯ
ПРОСПЕКТ МАРКСА
ПЛОЩАДЬ СВЕРДЛОВА
ПЛОЩАДЬ РЕВОЛЮЦИИ
ПЛОЩАДЬ НОГИНА
СМОЛЕНСКАЯ
АРБАТСКАЯ
КАЛИНИНСКАЯ
БОРОВИЦКАЯ
БИБЛИОТЕКА имени ЛЕНИНА
АРБАТСКАЯ
ТРЕТЬЯКОВСКАЯ
НОВОКУЗНЕЦКАЯ
ТАГАНСКАЯ
МАРКСИСТСКАЯ
КРОПОТКИНСКАЯ
ПАРК КУЛЬТУРЫ
ПОЛЯНКА
ПАВЕЛЕЦКАЯ
ФРУНЗЕНСКАЯ
СПОРТИВНАЯ
ЛЕНИНСКИЕ ГОРЫ
УНИВЕРСИТЕТ
ПРОСПЕКТ ВЕРНАДСКОГО
ЮГО-ЗАПАДНАЯ
ОКТЯБРЬСКАЯ
ДОБРЫНИНСКАЯ
СЕРПУХОВСКАЯ
ПРОЛЕТАРСКАЯ
ВОЛГОГРАДСКИЙ ПРОСПЕКТ
ТЕКСТИЛЬЩИКИ
КУЗЬМИНКИ
РЯЗАНСКИЙ ПРОСПЕКТ
ВЫХИНО
АВТОЗАВОДСКАЯ
КОЛОМЕНСКАЯ
КАШИРСКАЯ
ТУЛЬСКАЯ
НАГАТИНСКАЯ
НАГОРНАЯ
НАХИМОВСКИЙ ПРОСПЕКТ
ВАРШАВСКАЯ
КАХОВСКАЯ
ШАБОЛОВСКАЯ
ЛЕНИНСКИЙ ПРОСПЕКТ
АКАДЕМИЧЕСКАЯ
ПРОФСОЮЗНАЯ
НОВЫЕ ЧЕРЕМУШКИ
КАЛУЖСКАЯ
БЕЛЯЕВО
КОНЬКОВО
ТЕПЛЫЙ СТАН
ЯСЕНЕВО
БИТЦЕВСКИЙ ПАРК
СЕВАСТОПОЛЬСКАЯ
ЧЕРТАНОВСКАЯ
ЮЖНАЯ
ПРАЖСКАЯ
КАНТЕМИРОВСКАЯ
ЦАРИЦЫНО
ОРЕХОВО
ДОМОДЕДОВСКАЯ
КРАСНОГВАРДЕЙСКАЯ

Графы

ориентированные **неориентированные**



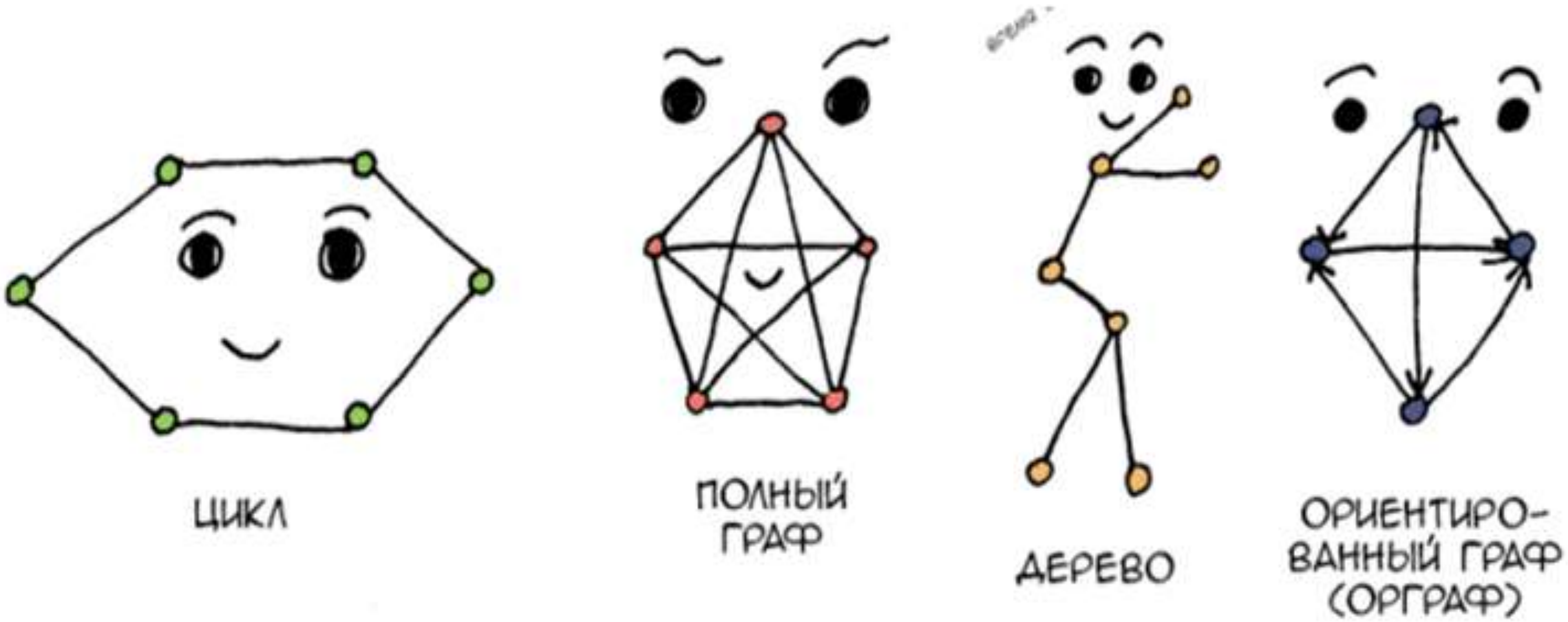
G_1



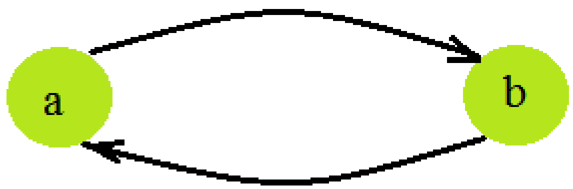
G_2

Ориентированный граф – все ребра являются упорядоченными парами вершин (v_s , v_t) , т.е. для каждого ребра определено направление из вершины v_s в вершину v_t .
Неориентированный граф – все ребра являются неупорядоченными парами вершин, (v_s , v_t) т.е. возможно прохождение из вершины в вершину в обоих направлениях.

Виды графов

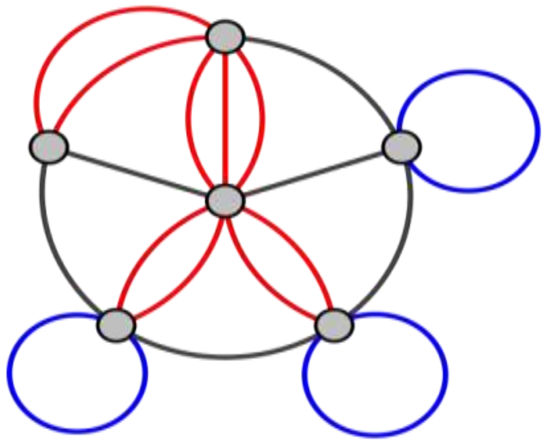


Мультиграф и псевдограф

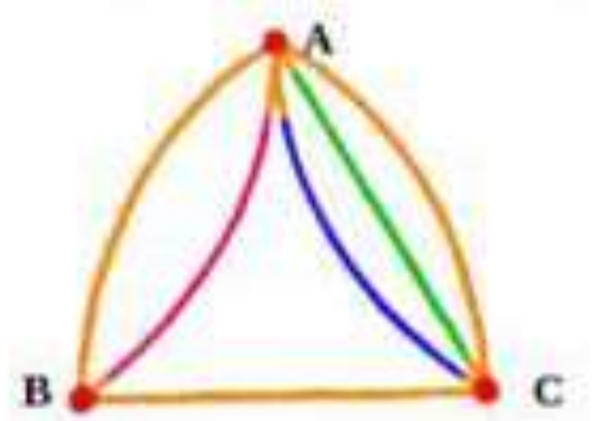


Кратные ребра – рёбра многократно соединяющие две вершины

петли



G_3 **- Псевдограф**

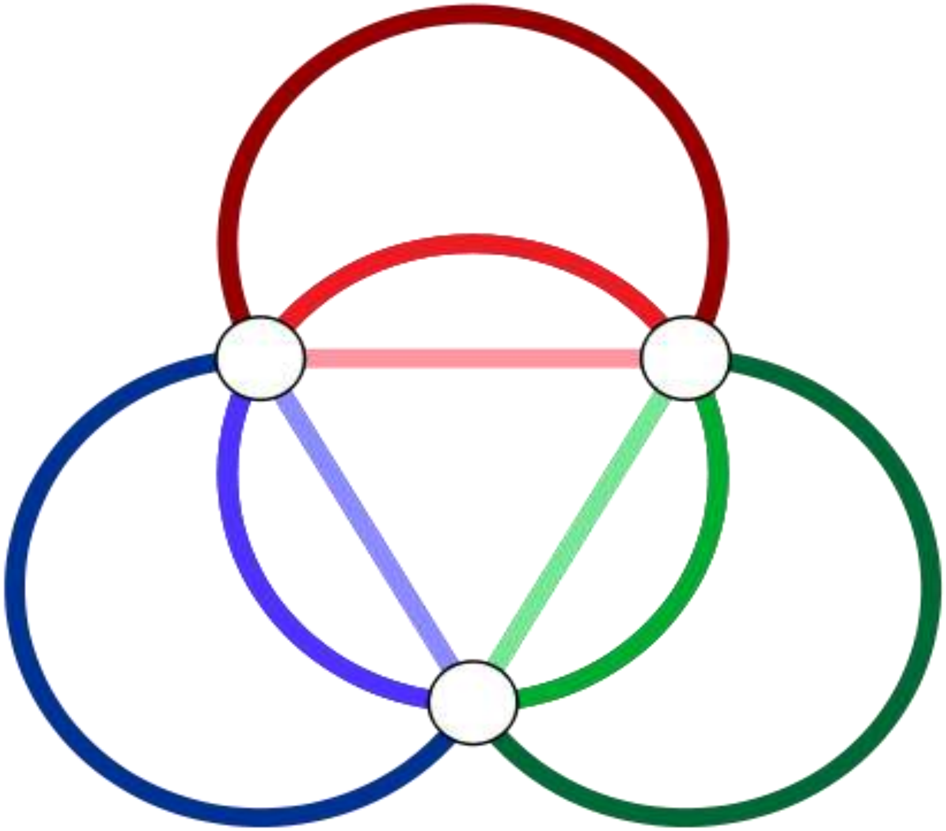


G_4 **- Мультиграф**

Псевдограф — граф с кратными ребрами и петлями.

Мультиграф — граф имеющий кратные ребра (несколько параллельных ребер) без петель.

Мультиграф или псевдограф?



Есть ли петли?

Степень вершины графа

Число рёбер, выходящих из вершины, называется **степенью вершины**

Степень вершины графа

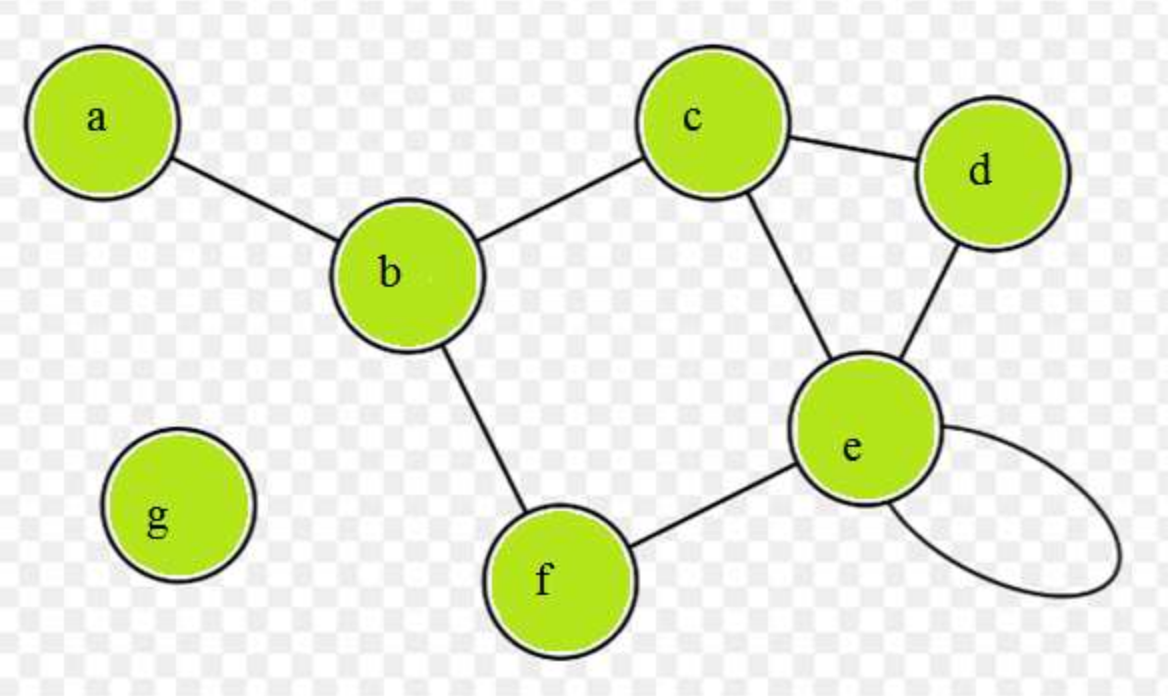
Степень вершины еще называют валентностью и обозначают **d(v), deg(v).**

Вершина графа, для которой d(v)=0, является изолированной, если d(v)=1, то висячей.

Вершина называется нечетной, если d(v) – нечетное число, четной если d(v) – четное число.

Степень каждой вершины полного графа на единицу меньше числа его вершин.

Определить степени вершин графа



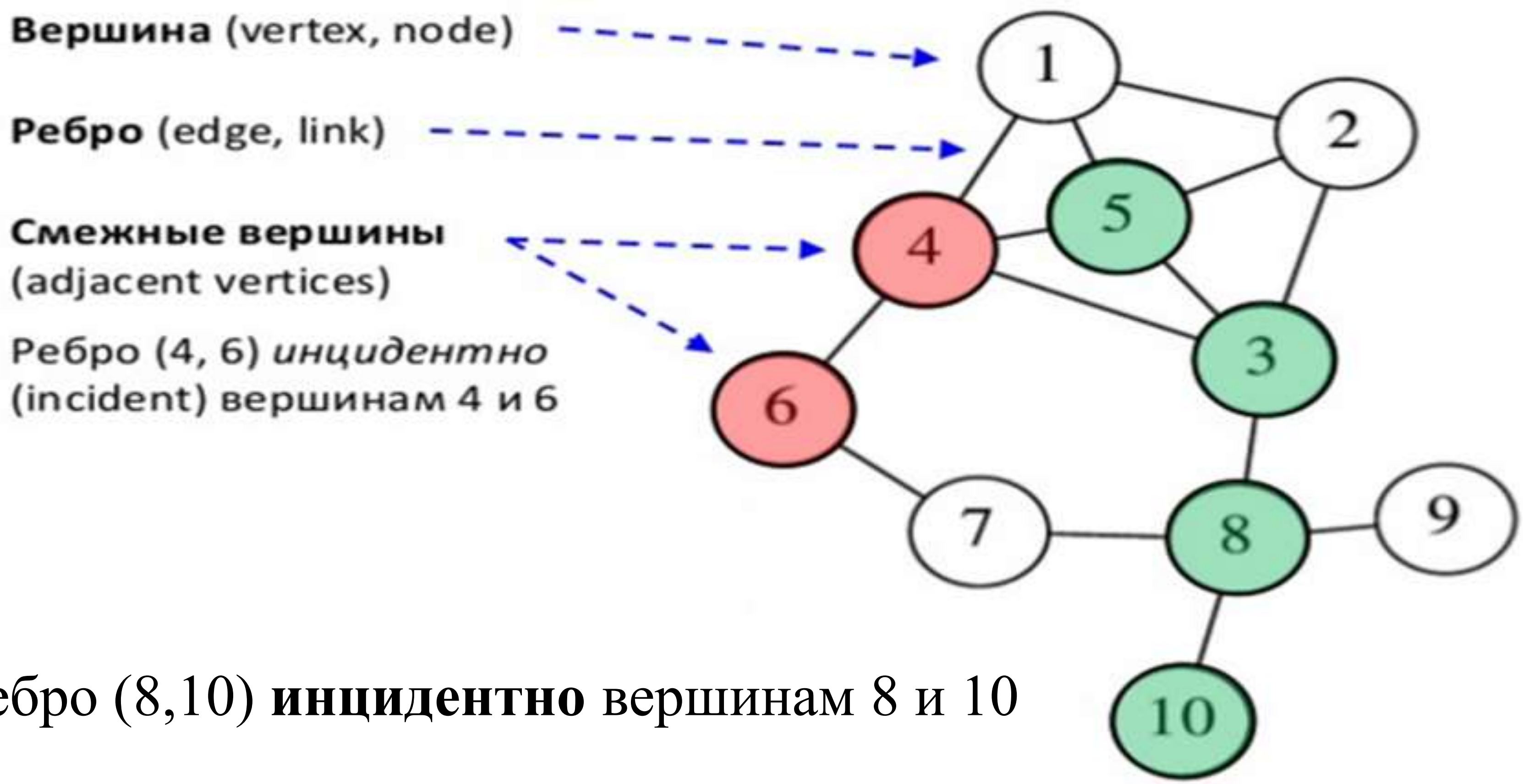
- d(g)=0
- d(a)=1
- d(b)=3
- d(f)=2
- d(e)=5

- g – изолированная вершина
- a – висящая вершина

Свойства степени вершины

- В графе G(V,R) сумма степеней всех его вершин – число четное, равное удвоенному числу ребер графа.
- Число нечетных вершин любого графа четно.
- Во всяком графе с n вершинами, где $n \geq 2$ всегда найдутся, по меньшей мере, две вершины с одинаковыми степенями.
- Если в графе с n вершинами ($n \geq 2$) в точности две вершины имеют одинаковую степень, то в этом графе всегда найдется либо в точности одна вершина степени 0, либо в точности одна вершина степени n-1.

- Вершины, соединенные ребром, называются ***смежными***. Ребра, имеющие общую вершину, также называются ***смежными***.
- Ребро и любая из его двух вершин называются ***инцидентными***.

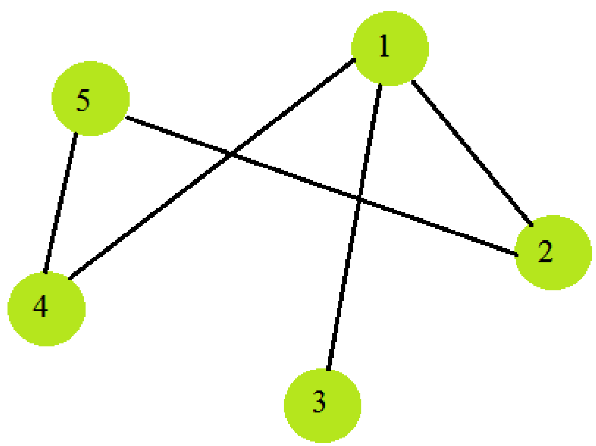


Ребро (8,10) **инцидентно** вершинам 8 и 10

4. Матричное представление графов

Матрица смежности K – квадратная матрица порядка *n*, строки и столбцы которой соответствуют вершинам графа, а элементы определяют число выходящих ребер.

$$\mathbf{K}_{ij} = \begin{cases} \textbf{1, если существует ребро из вершины } \boldsymbol{i} \textbf{ в вершину } \boldsymbol{j}, \\ \textbf{0, иначе.} \end{cases}$$



$$K = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

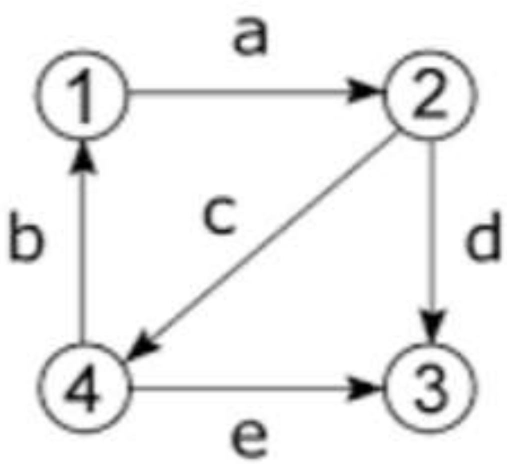
Данный способ подходит для ориентированных и неориентированных графов. Для неориентированных графов матрица A является симметричной. Для мультиграфа $K_{ij} \geq 1$.

Матрица инцидентности

I - строки которой соответствуют вершинам графа, столбцы – ребрам графа, а элементы определяют выходит ребро из вершины или входит в нее.

$$I_{ij} = \begin{cases} \mathbf{1}, & \text{ребро выходит из вершины,} \\ \mathbf{-1}, & \text{ребро входит в вершину,} \\ \mathbf{0}, & \text{ребро отсутствует или является петлёй} \end{cases}$$

	a	**b**	**c**	**d**	**e**
1	1	-1	0	0	0
2	-1	0	1	1	0
3	0	0	0	-1	-1
4	0	1	-1	0	1



Весовая матрица графа

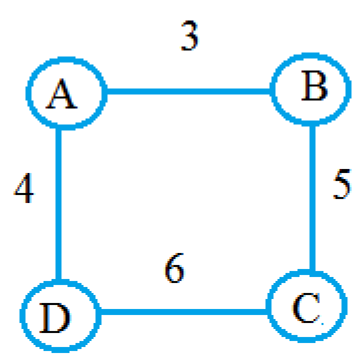
Для описания графа часто используют квадратную таблицу, которая описывает все возможные связи между узлами. Если, например, на пересечении строки A и столбца B записано какое-то число, это означает, что есть ребро, соединяющее вершины A и B; пустая ячейка означает, что такого пути между вершинами нет. Матрицу, в клетках которой записывают вес ребра или расстояние между вершинами, называют весовой. На рисунке показана схема, на которой указаны длины дорог, соответствующий ей граф и его весовая матрица:

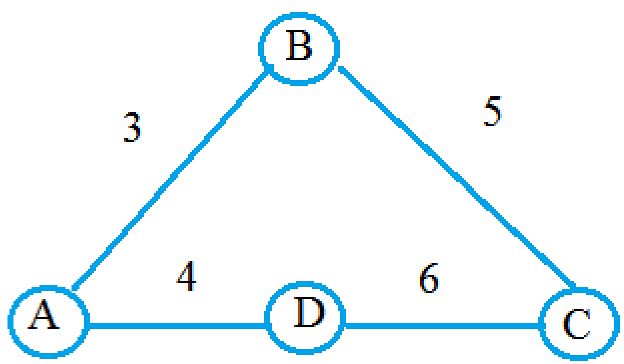
	A	B	C	D
A		12	8	
B	12		5	6
C	8	5	2	4
D		6	4	

Соответствие матрицы и графа

Наконец, полезно научиться рисовать граф по заданной весовой матрице. Нужно только помнить, что это можно сделать разными способами. Для последней приведенной матрицы рисунок может быть, например, таким: После того как освоены основные понятия, можно переходить непосредственно к задачам, которые включены в ЕГЭ и ГИА.

	A	B	C	D
A		3		4
B	3		5	
C		5		6
D	4		6	





5.Задача о Кёнигсбергских мостах

Кенигсбергцы предлагали приезжим следующую задачу: пройти по всем мостам и вернуться в начальный пункт, причём на каждом мосту следовало побывать только один раз.

Леонард Эйлер
(1707 - 1783)

История возникновения теории графов

Основы теории графов как математической науки заложил в 1736 г. **Леонард Эйлер**, рассматривая задачу о кенигсбергских мостах. Сегодня эта задача стала классической.

Бывший *Кенигсберг (ныне Калининград)* расположен на реке Прегель. В пределах города река омывает два острова. С берегов на острова были перекинуты мосты. Старые мосты не сохранились, но осталась карта города, где они изображены.

Выводы Эйлера

В 1736 году задача о семи мостах заинтересовала выдающегося математика Леонарда Эйлера, о чём он написал в письме итальянскому математику Мариони в 1736 г.

Эйлер пишет, что он смог найти правило, пользуясь которым легко определить, можно ли пройти по всем мостам, не проходя дважды ни по одному из них (в случае семи мостов Кёнигсберга это невозможно).

Эйлер пришёл к следующим выводам:

- Число нечётных вершин графа должно быть чётно. Не может существовать граф, который имел бы нечётное число нечётных вершин.
- Если все вершины графа чётные, то можно, не отрывая карандаша от бумаги, начертить граф, при этом можно начинать с любой вершины графа и завершить его в той же вершине.
- Граф с более чем двумя нечётными вершинами невозможно начертить одним росчерком.

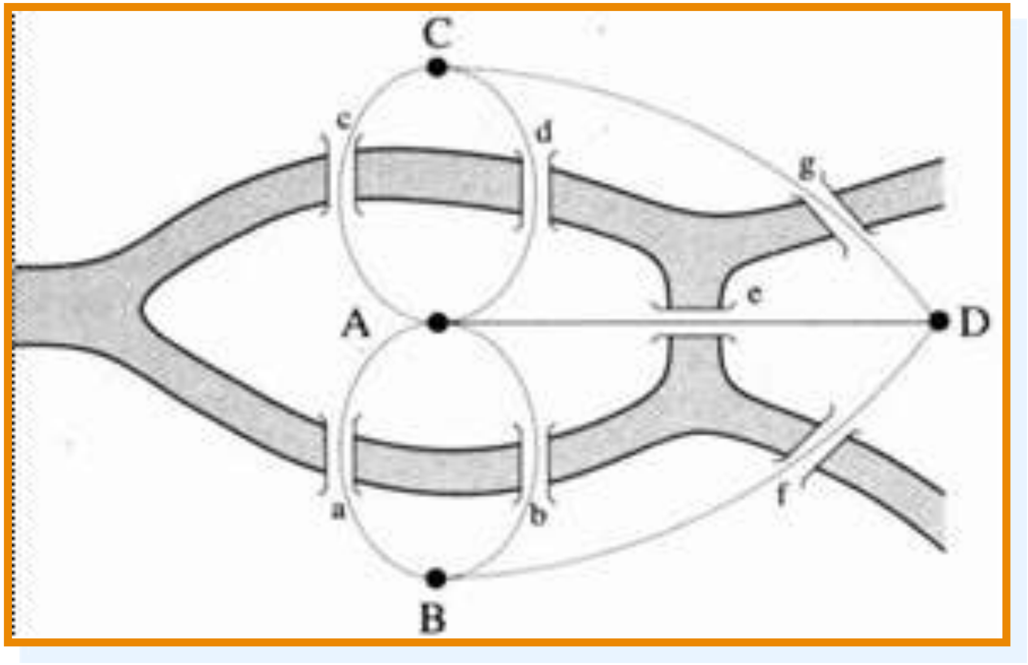
Позже эти выводы стали теоремами в теории графов

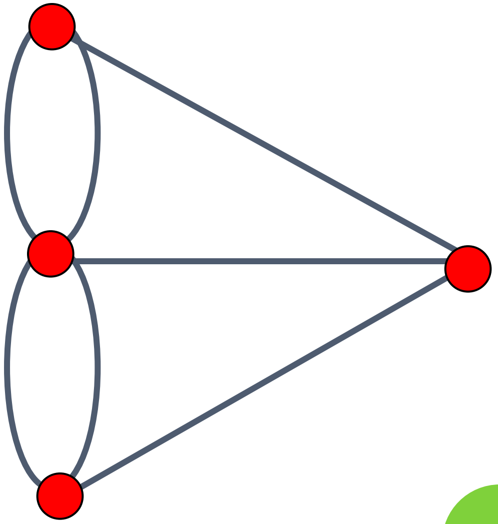
Решение задачи о Кенигсбергских мостах

Пройти по Кенигсбергским мостам, соблюдая заданные условия, нельзя.

Граф имеет четыре нечетные вершины.

Граф с более чем двумя нечётными вершинами невозможно начертить одним росчерком.

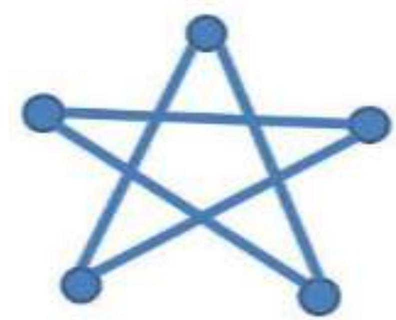




6.Эйлеров граф и эйлеров цикл

Граф называется **эйлеровым**, если существует замкнутая цепь содержащая каждое ребро заданного графа проходят по одному разу (**эйлеров цикл**).
Его можно нарисовать одним росчерком

Теорема 1: Связный граф обладает эйлеровым циклом, только тогда, когда степени всех его вершины - четные.



Одним росчерком

Теоремы Эйлера

Теорема 2: В графе сумма степеней всех его вершин – число чётное, равное удвоенному числу ребер графа:

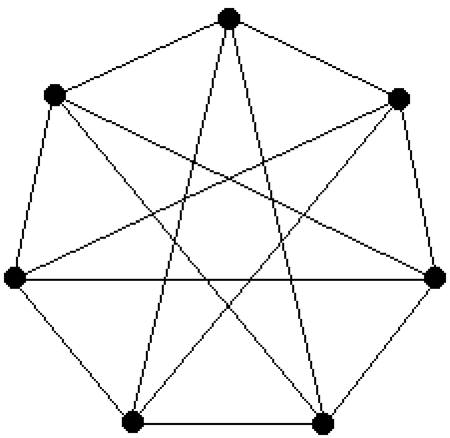
Степень А +степень В + степень С +…=2*число рёбер

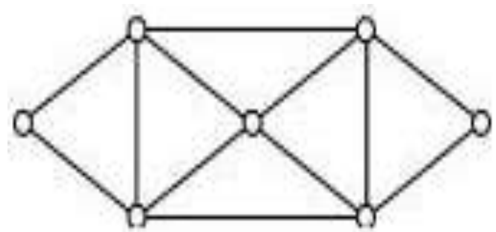
Теорема 3: Число нечетных вершин любого графа – чётно.

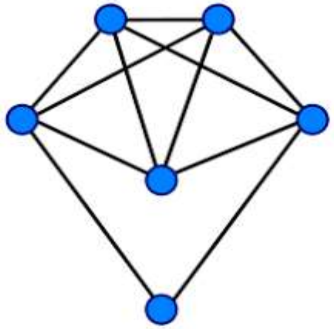
Следствия:

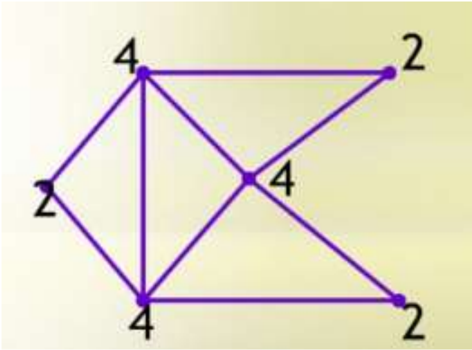
- Число вершин многогранника, в которых сходится нечётное число рёбер, чётно.
- Нечётное число знакомых в любой компании всегда чётно.

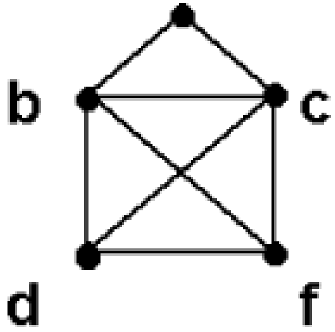
Эйлеровы графы



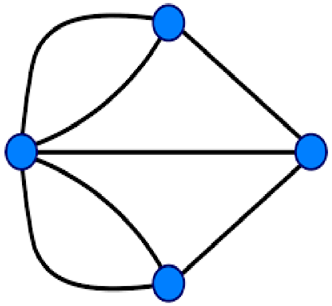


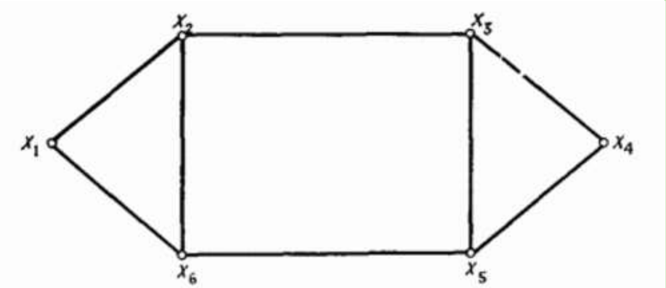






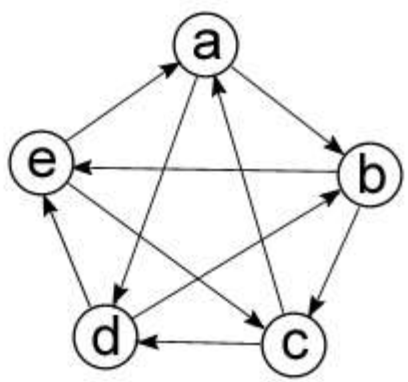
Не эйлеровы графы

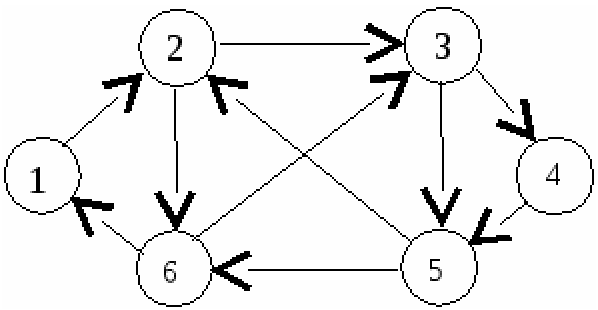




Признак Эйлерова графа для ориентированного графа

Теорема 4. Связный ориентированный граф обладает эйлеровым циклом, только когда у каждой вершины входящая и исходящая степени равны.

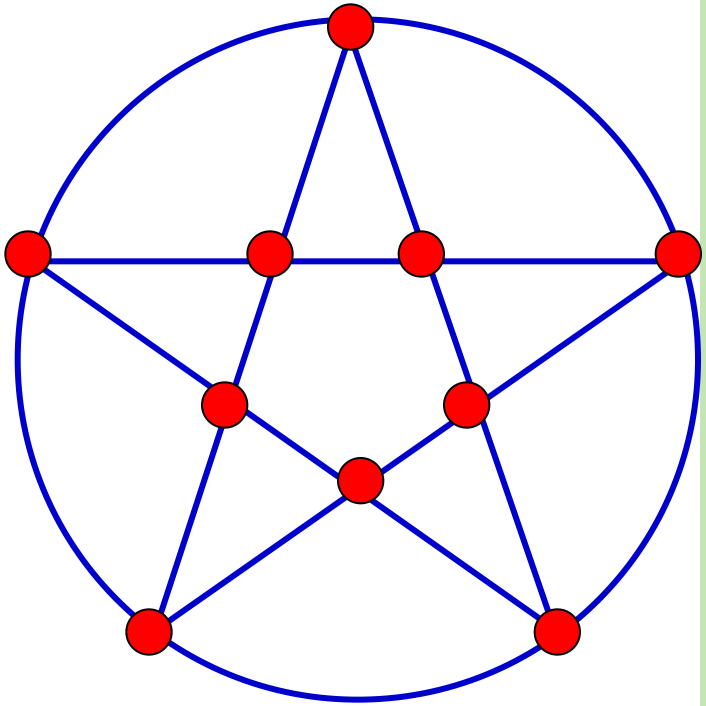




Эйлеров граф

Одним росчерком

Если все вершины графа четные, то можно не отрывая карандаш от бумаги («одним росчерком»), проводя по каждому ребру только один раз, начертить этот граф. Движение можно начать с любой вершины и закончить его в той же вершине.

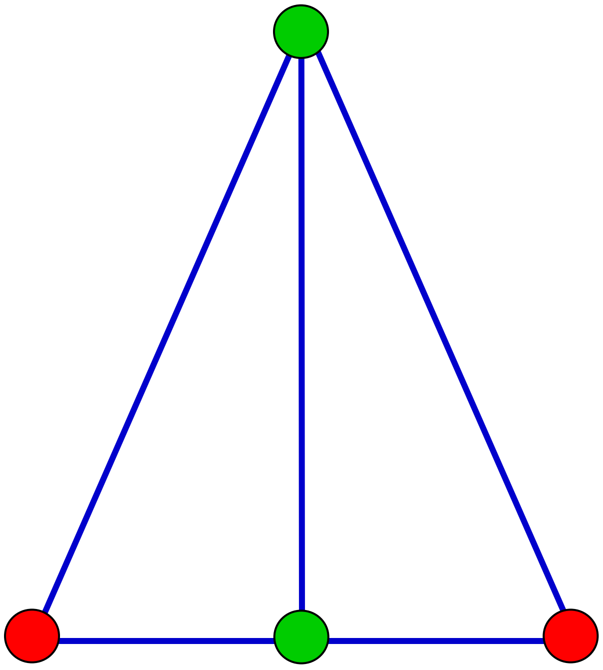


дальше

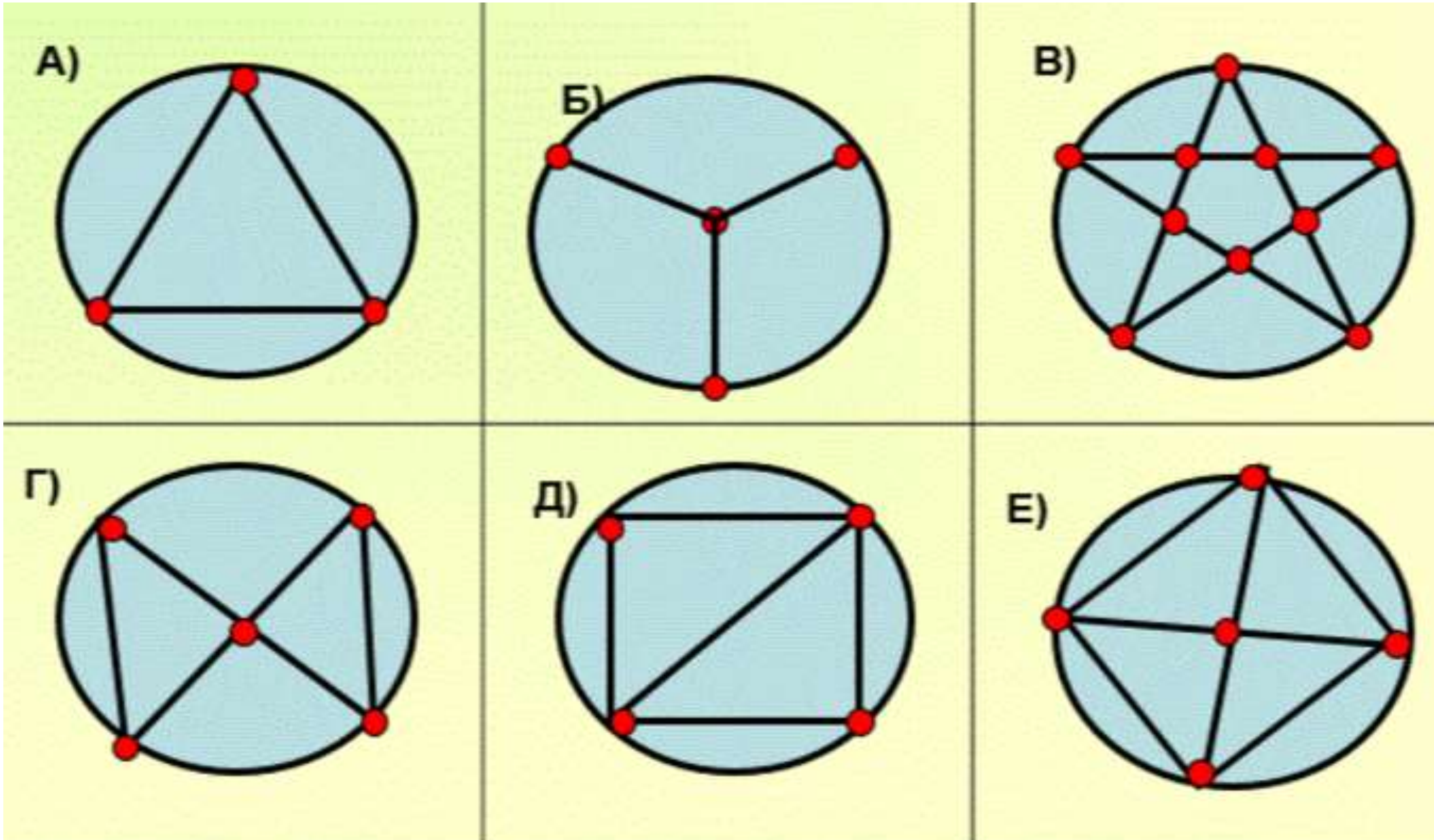
Одним росчерком

Граф, имеющий всего две нечетные вершины, можно начертить, не отрывая карандаш от бумаги, при этом движение нужно начать с одной из этих нечетных вершин и закончить во второй из них.

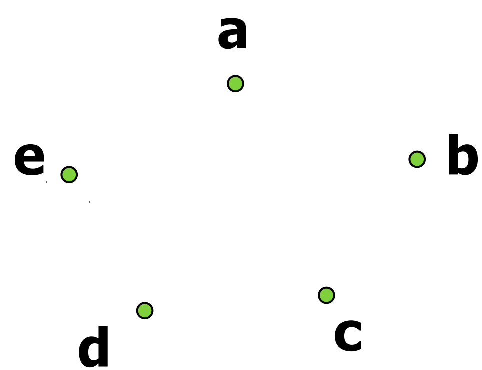
Какие вершины являются нечетными?



Нарисуйте приведенные фигуры одним росчерком не проводя более 1 раза по одной и той же линии



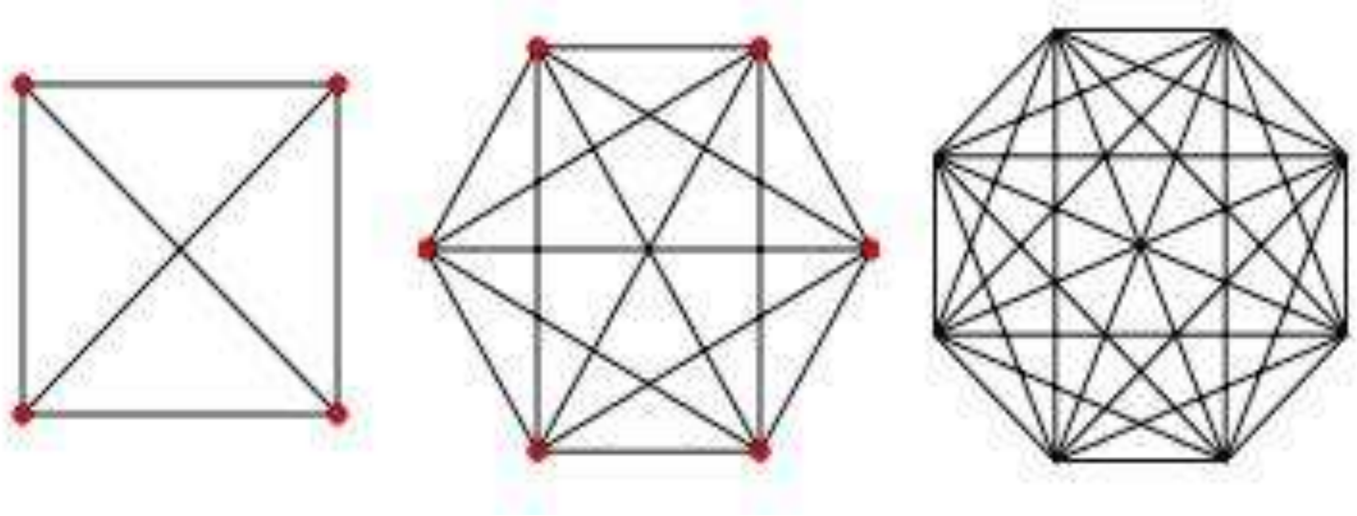
Пустой граф



G_5 – пустой граф – не содержит ни одного ребра, $R = \varnothing$.

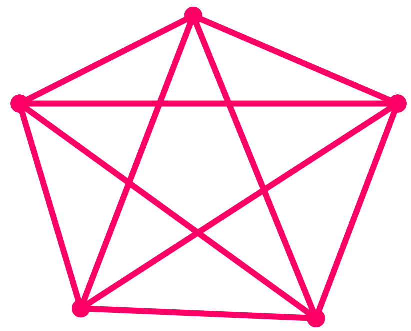
Полный граф

- **Полным графом** называется граф, в котором каждая пара различных вершин соединена ребром: парой дуг по одной в каждую сторону. Полный граф с n вершинами имеет $n(n-1)/2$ рёбер. Является регулярным графом степени $n-1$.
- **Каждые две различные вершины графа соединены одним и только одним ребром**



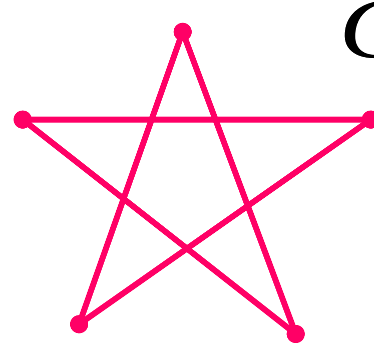
Полный граф и дополнение

Граф называется **полным**, если любые две его различные вершины соединены одним и только одним ребром.

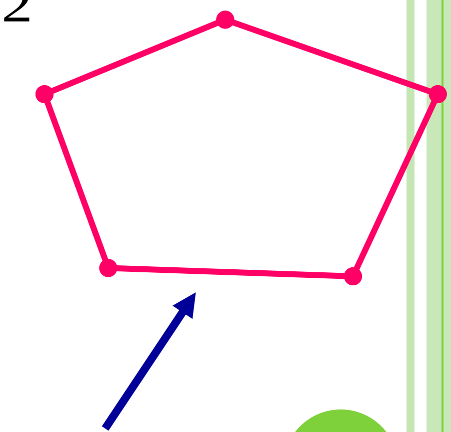


G_2

Дополнением графа называется граф с теми же вершинами и имеющий те и только те ребра, которые необходимо добавить к исходному графу, чтобы он стал полным.



G_5



Дополнение графа G_5

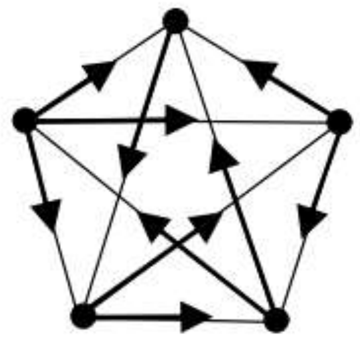
до графа G_2

Полный ориентированный граф и Турниры

Полным ориентированным графом называется граф, каждая пара вершин которого соединена в точности одним ориентированным ребром.

В *ориентированном полном* графе имеются пары ребер, по одному в каждом направлении, соединяющие любые две различные вершины.

Турниром называется орграф, любую пару вершин которого соединяет только одна дуга. Турнир отличается от полного ориентированного графа тем, что у него нет противоположно направленных дуг.



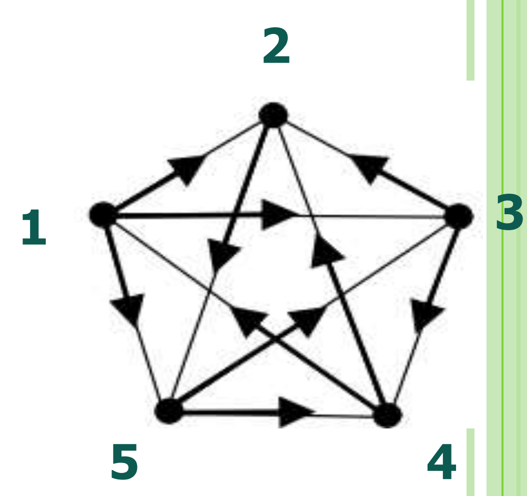
Турниры

Этот термин произошел от соревнований по круговой системе, графическое представление которых соответствует полному ориентированному графу. В турнирах по круговой системе играют несколько команд каждая со всеми остальными по разу. Игра обязательно должна закончится чьей-то победой.

Вершины соответствуют командам и дуга присутствует в графе, если команда 1 победила команду 2.

В таком ориентированном графе нет параллельных дуг и петель и между каждыми двумя вершинами обязательно есть дуга.

Определите команду победительницу!



Турнирная таблица
Плей-офф КХЛ сезона 2013-2014гг

1/4 финала

Динамо (Мск)	3
Локомотив	4
СКА	4
ЦСКА	0
Лев	4
Медвешчак	0
Донбасс	4
Динамо (Р)	3
Металлург (М)	4
Адмирал	1
Барыс	4
Автомобилист	0
Ак Барс	2
Сибирь	4
Салават	4
Торпедо	3

1/2 финала

СКА	2
Локомотив	4
Лев	4
Донбасс	2
Металлург (М)	4
Сибирь	0
Барыс	2
Салават	4

финал конференций

ЗАПАД

Лев	4
Локомотив	1

ВОСТОК

Металлург (М)	4
Салават	1

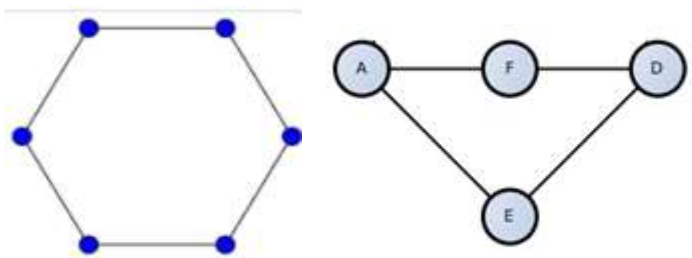
финал чемпионата Кубок Гагарина

Лев	3
Металлург (М)	4

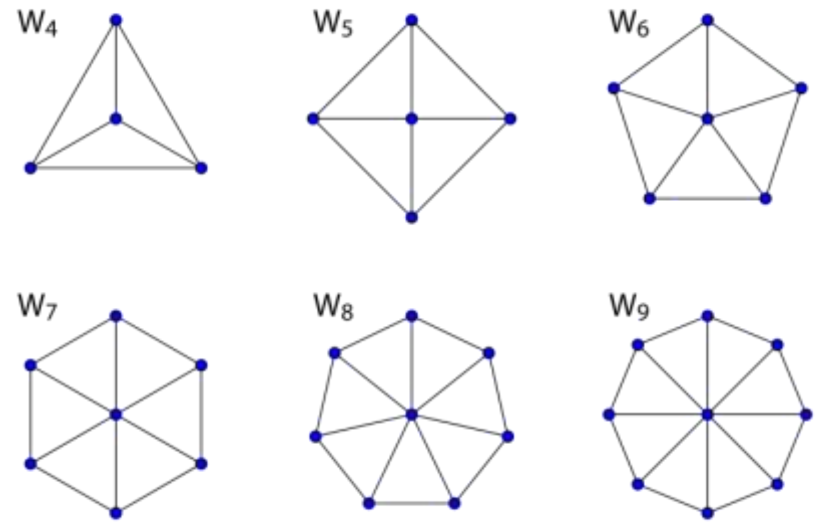
Плей-офф КХЛ сезона 2013-2014. Pribalt.info

Другие типы графов

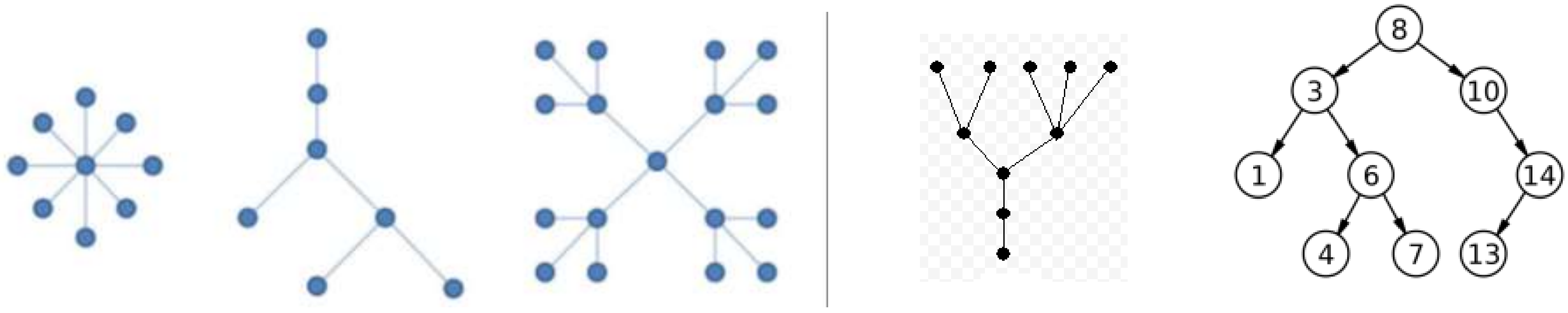
Циклы



Колёса



Деревья



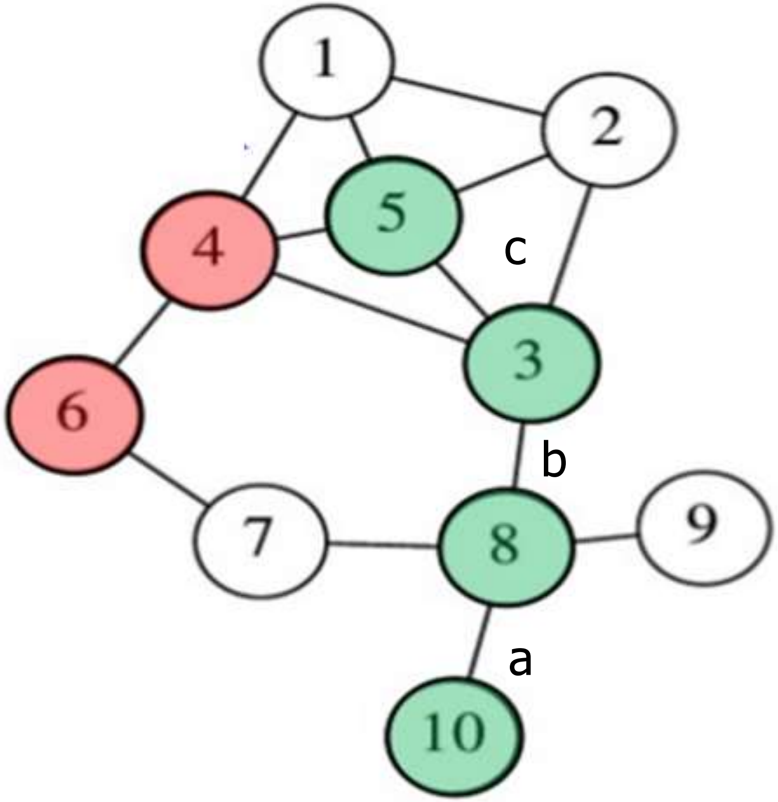
Путь или маршрут в графе

- **Путь** из вершины А в вершину В - это последовательность смежных ребер (или вершин), по которым можно дойти из А в В.
- **Длиной пути** во взвешенном графе называют сумму длин звеньев этого пути. Количество k ребер в пути называется длиной пути.

Путь от вершины 10 до вершины 5:

Через ребра: (a,b,c), длина -3.

Через вершины: (10,8,3,5)

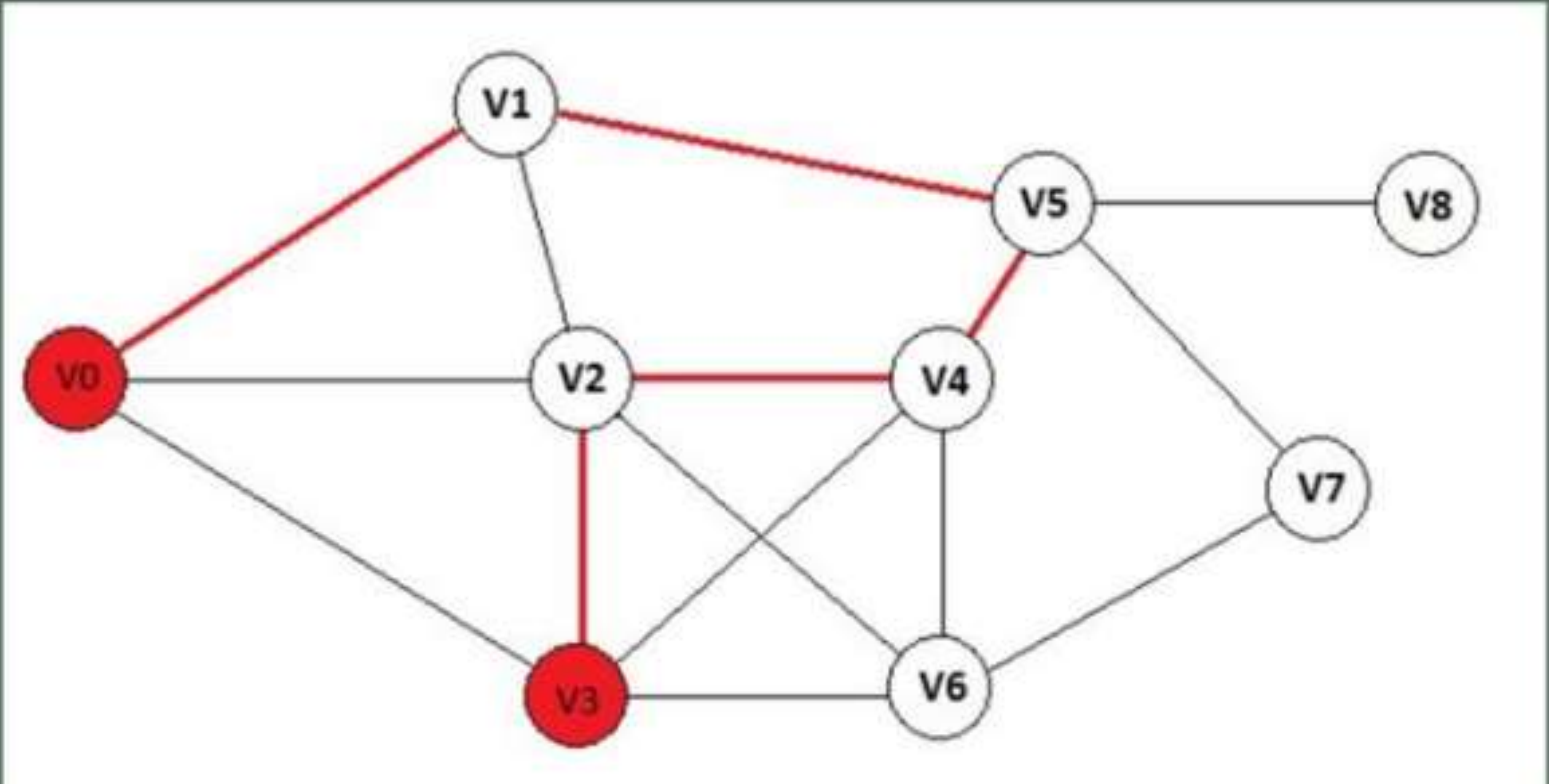


Цепь, цикл, дерево

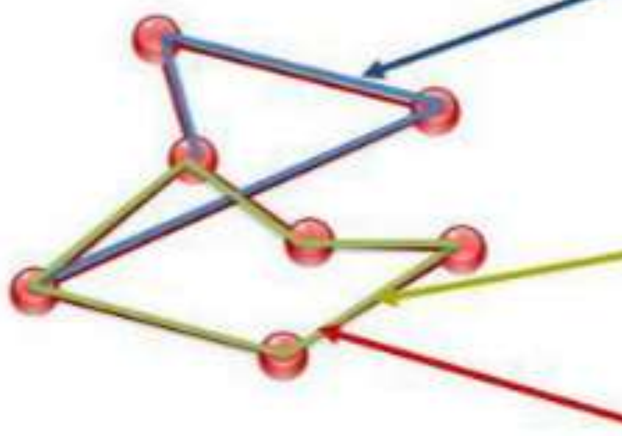
- **Маршрут графа** – последовательность вершин в графе. Может быть замкнутым когда начало совпадает с концом или незамкнутым. В маршруте могут повторяться и вершины и ребра.
- **Простая цепь**–маршрут , в котором все вершины и ребра различны.
- **Цепь** – путь по вершинам и ребрам, включающий любое ребро графа не более одного раза.
- **Цикл** – цепь, начальная и конечная вершины которой совпадают.
- **Дерево** – это граф, в котором между любыми двумя вершинами существует единственный путь. В дереве не должно быть циклов

Если маршрут в простом графе задан последовательностью вершин $v_0, v_1, \ldots, v_k$, то вершины v_o, v_k называют **концами маршрута**.

Например: концами маршрута V0-V1-V5-V4-V2-V3 являются вершины V0 ,V3



Сеть и дерево

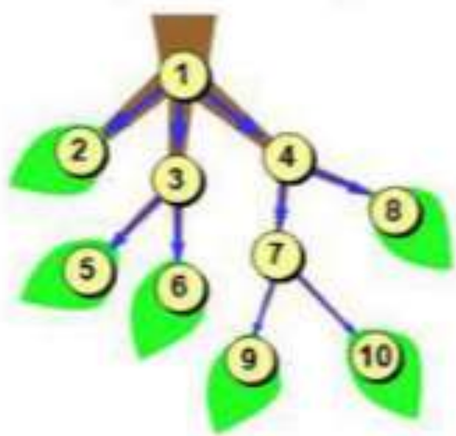


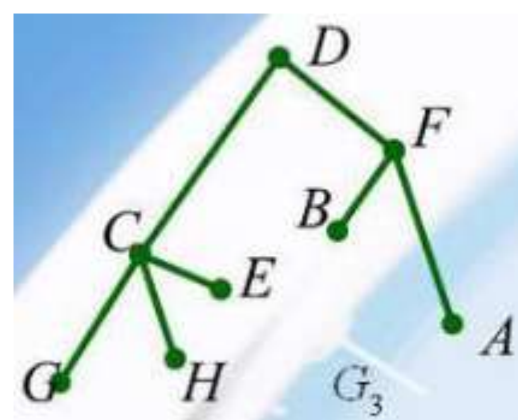
Цепь – путь по вершинам и рёбрам граф, в который любое ребро графа входит не более одного раза.

Цикл - цепь, начальная и конечная вершины которой совпадают.

Сеть - граф с циклом.

Дерево – это граф, в котором нет циклов





1805-1865

7. Уильям Гамильтон

Уильям Роуэн Гамильтон (*William Rowan Hamilton*) ирландский математик, механик, физик-теоретик XIX века». Известен фундаментальными открытиями

в математике:

- **основы векторного анализа,**
- **вариационное исчисление,**

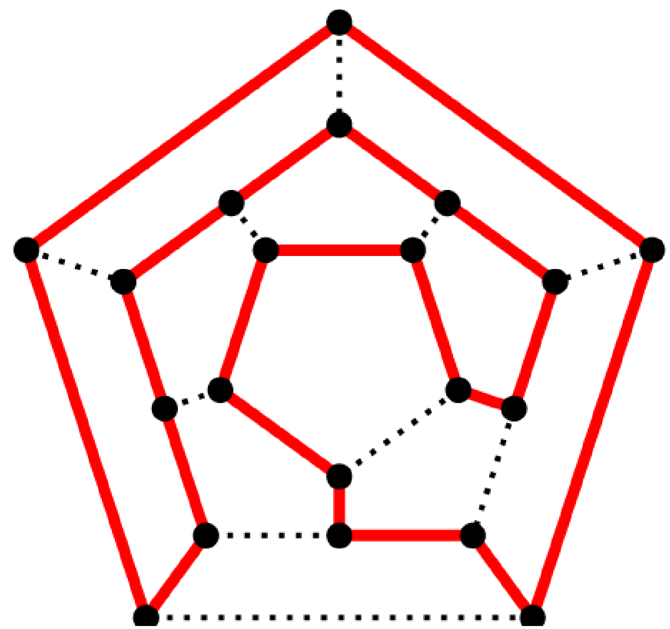
обоснование комплексных чисел,

аналитической механике и оптике.

В 1857 г. ирландский математик **У. Гамильтон** предложил игру, названную «Путешествие вокруг света». Вершины додекаэдра символизировали города: Брюссель, Амстердам, Эдинбург, Пекин, Прага, Дели, а рёбра соединяющие их дороги.

Игра сводилась к обходу по ребрам всех вершин правильного додекаэдра, при условии, что ни в одну из вершин нельзя заходить более одного раза.

Сначала игра была объемная – сфера, веревки, гвозди, потом для удобства ее сделали плоской в форме додекаэдра.

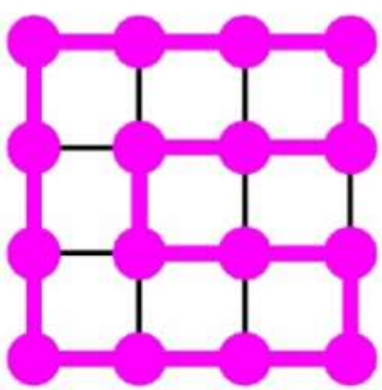


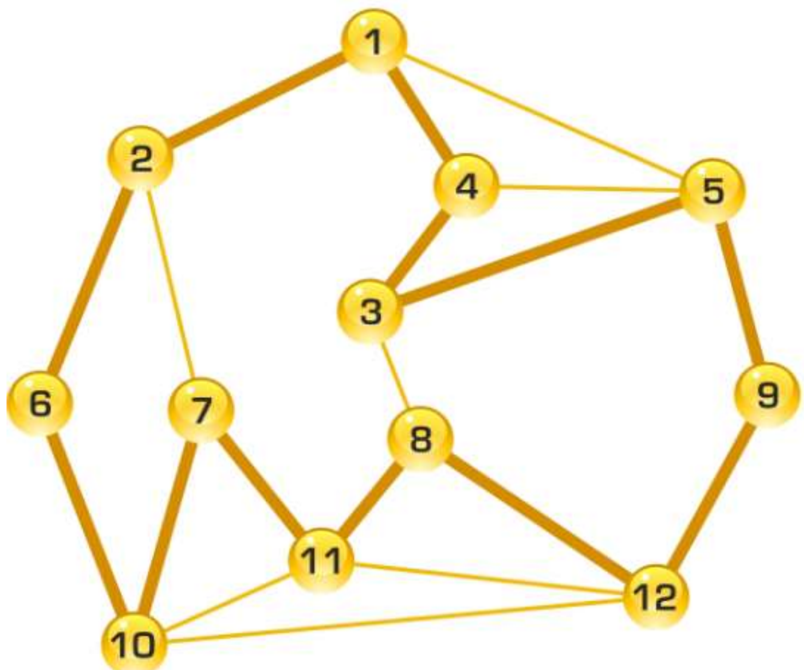
«Путешествие вокруг света»

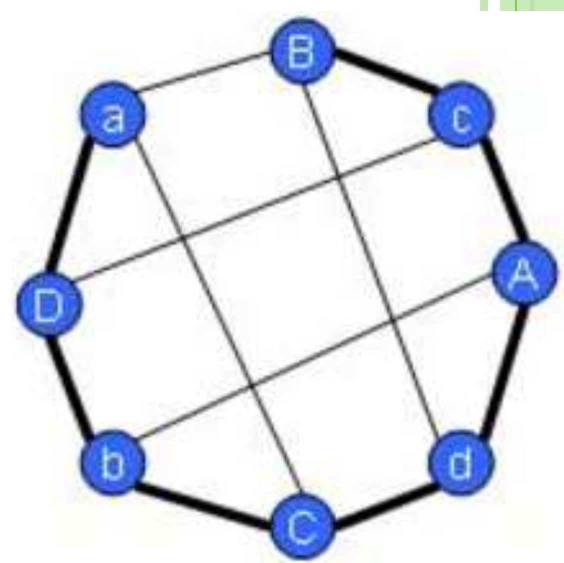
7. Гамильтоновы графы

Гамильтонов цикл - цикл в графе, содержащий все вершины графа по одному разу.

Гамильтонов граф - граф, в котором есть гамильтонов цикл.





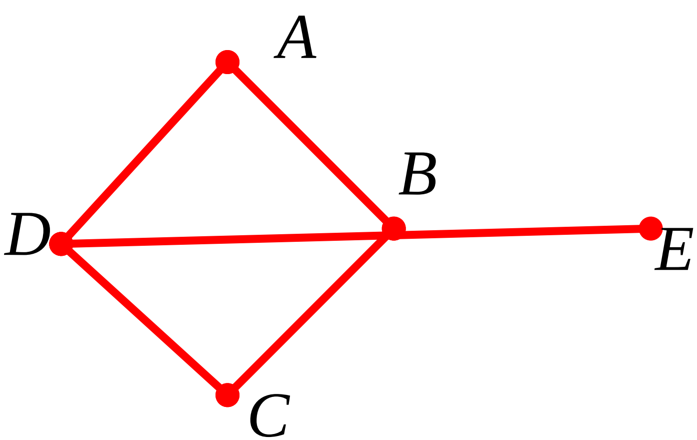


Гамильтоновы графы

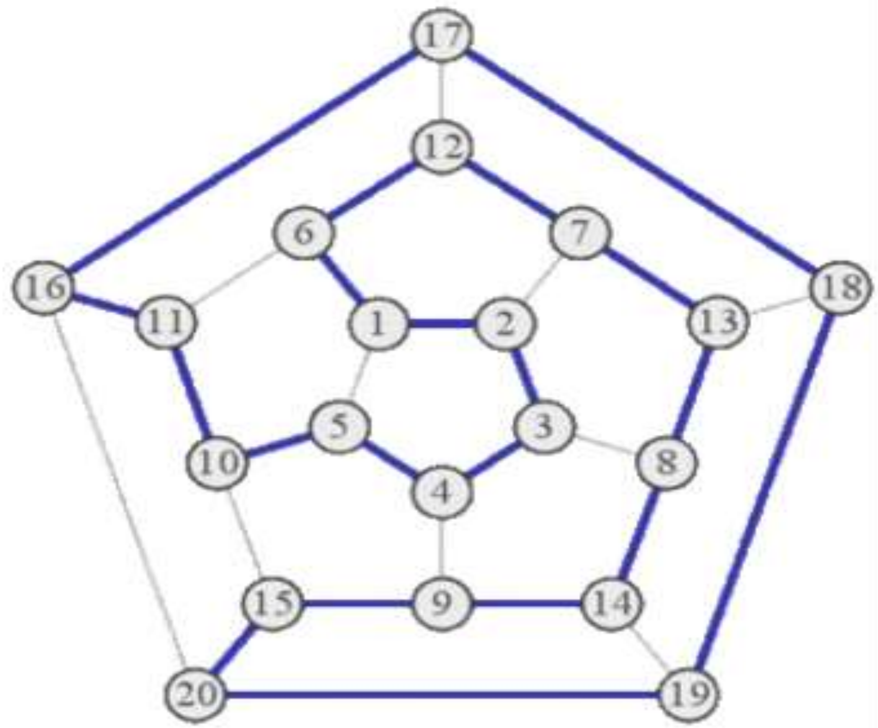
Гамильтонов путь (цикл)

Гамильтоновым путем графа называется путь(цикл), проходящий через каждую его вершину только один раз.

Примеры гамильтоновых графов



(C, D, A, B, E) – гамильтонов путь



Граф додекаэдра с выделенным циклом Гамильтона

Критерий же существования гамильтонова цикла в произвольном графе еще не найден.

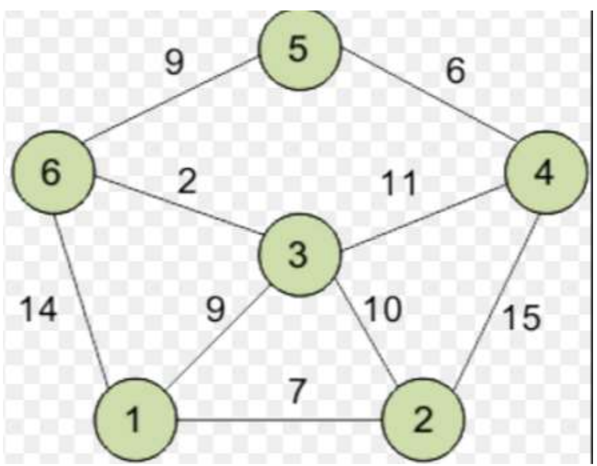
Решение этой проблемы имеет практическую ценность, так как к ней близка известная задача о коммивояжере, который должен объехать несколько пунктов и вернуться обратно. Он обязан побывать в каждом пункте в точности по одному разу и заинтересован в том, чтобы затратить на поездку как можно меньше времени.

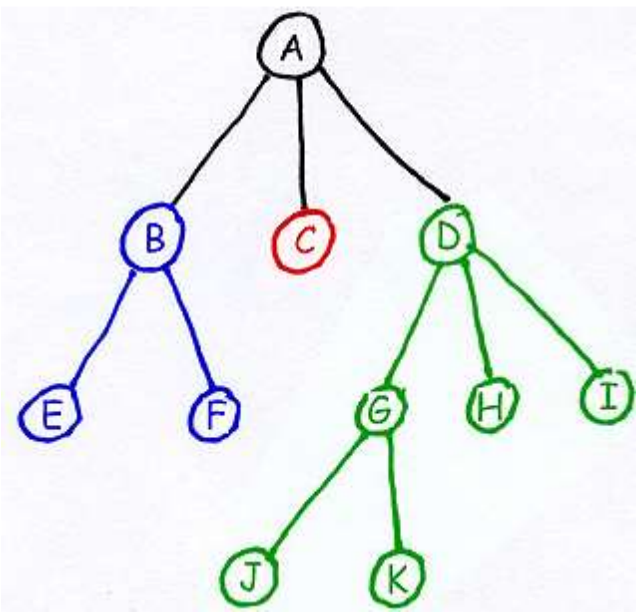
А для этого требуется определить все варианты посещения городов и подсчитать в каждом случае затрату времени. Большинство известных теорем имеет вид "если граф имеет достаточное число ребер, то граф является гамильтоновым графом".

8. Поиск кратчайшего пути в графе

Методы решения

1) Построение графа
2) Построение дерева кратчайших путей
3) Перебор возможных путей без построения дерева
4) Использование алгоритма Дейкстры
5) Другие алгоритмы





Алгоритмы поиска кратчайшего пути в графе

Алгоритм	Применение
Алгоритм Дейкстры	Находит кратчайший путь от одной из вершин графа до всех остальных. Алгоритм работает только для графов без ребер отрицательного веса ($w_{ij} \geq 0$)
Алгоритм Беллмана-Форда	Находит кратчайшие пути от одной вершины графа до всех остальных во взвешенном графе. Вес ребер может быть отрицательным
Алгоритм поиска A* (A star)	Находит путь с наименьшей стоимостью от одной вершины к другой, используя алгоритм поиска по первому наилучшему совпадению на графе
Алгоритм Флойда-Уоршелла	Находит кратчайшие пути между всеми вершинами взвешенного ориентированного графа
Алгоритм Джонсона	Находит кратчайшие пути между всеми парами вершин взвешенного ориентированного графа (должны отсутствовать циклы с отрицательным весом)
Алгоритм Ли (волновой алгоритм)	Находит путь между вершинами *s* и *t* графа, содержащий минимальное количество промежуточных вершин (трассировки электрических соединений на кристаллах микросхем и на печатных платах)
Алгоритмы Viterbi, Cherkassky, …	

Алгоритм Дейкстры

- **Алгоритм Дейкстры** (Dijkstra's algorithm, 1959) – алгоритм поиска кратчайшего пути в графе из заданной вершины во все остальные (single-source shortest path problem)
- Находит кратчайшее *расстояние* от одной из вершин графа до всех остальных
- Применим только для графов без ребер отрицательного веса и петель ($w_{ij} \geq 0$)

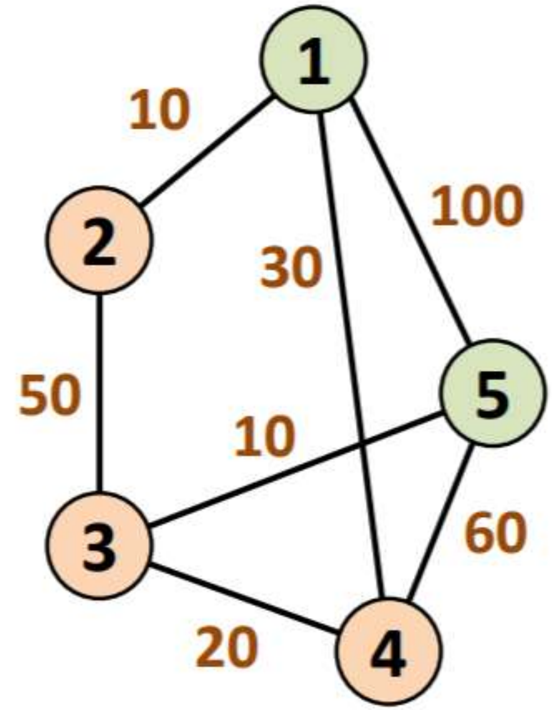
- **Эдсгер Дейкстра** (Edsger Wybe Dijkstra) – нидерландский ученый (структурное программирование, язык Алгол, семафоры, распределенные вычисления)
- Лауреат премии Тьюринга (ACM A.M. Turing Award)

1. Дейкстра Э. Дисциплина программирования = A discipline of programming. — М.: Мир, 1978. — С. 275.
2. Дал У., Дейкстра Э., Хоор К. Структурное программирование = Structured Programming. — М.: Мир, 1975. — С. 247.

Алгоритм Дейкстры

- **Пример:** найти кратчайший путь из вершины 1 в вершину 5

- Введем обозначения:
  - *H* – множество посещенных вершин
  - *D*[*i*] – текущее известное кратчайшее расстояние от вершины *s* до вершины *i*
  - prev[*i*] – номер вершины, предшествующей *i* в пути

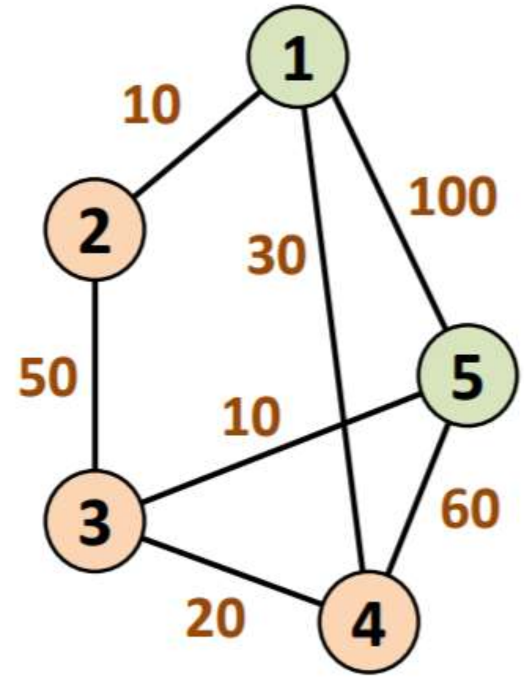


Алгоритм Дейкстры

1. Устанавливаем расстояние $D[i]$ от начальной вершины s до всех остальных в ∞

2. Полагаем $D[s] = 0$

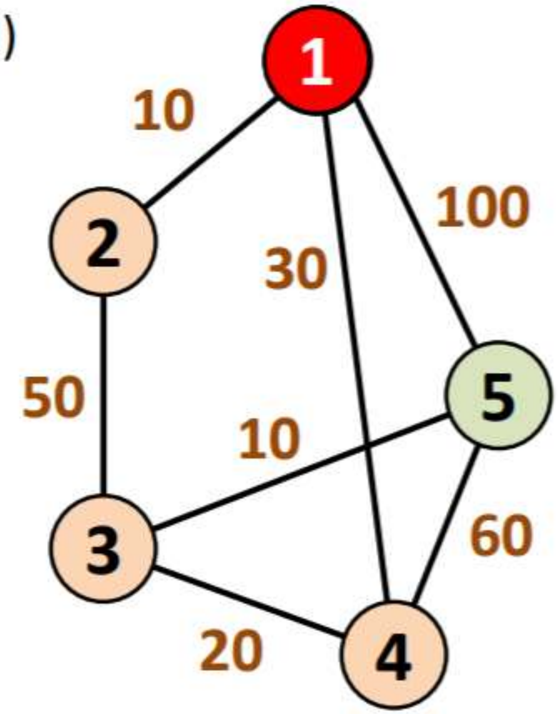
3. Помещаем все вершины в очередь с приоритетом Q (min-heap): приоритет вершины i это значение $D[i]$



	1	2	3	4	5
$D[i]$	0	∞	∞	∞	∞

Алгоритм Дейкстры

4. Запускаем цикл из *n* итераций (по числу вершин)
   1. Извлекаем из очереди *Q* вершину *v* с минимальным приоритетом – ближайшую к *s* вершину
   2. Отмечаем вершину *v* как посещенную (помещаем *v* во множество *H*)
   3. Возможно пути из *s* через вершину *v* стали короче, выполняем проверку: для каждой вершины *u* смежной с вершиной *v* и не включенной в *H* проверяем и корректируем расстояние *D*[*u*]



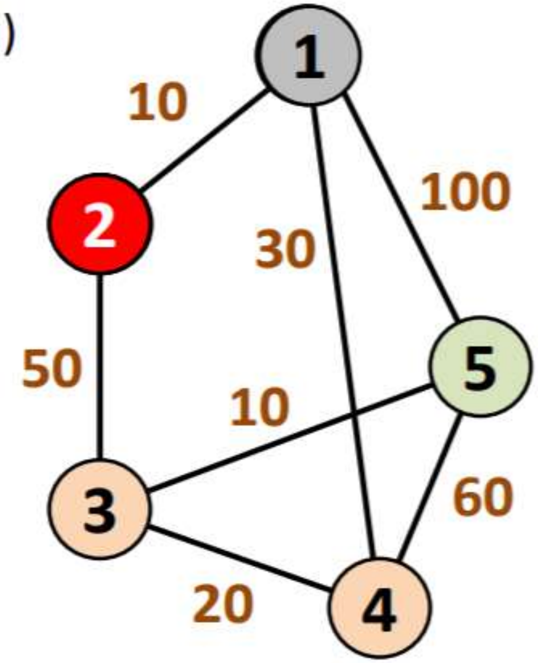
короче

D[2] = 10
D[4] = 30
D[5] = 100

D[*i*]	0	10	∞	30	100

Алгоритм Дейкстры

4. Запускаем цикл из *n* итераций (по числу вершин)
   1. Извлекаем из очереди *Q* вершину *v* с минимальным приоритетом – ближайшую к *s* вершину
   2. Отмечаем вершину *v* как посещенную (помещаем *v* во множество *H*)
   3. Возможно пути из *s* через вершину *v* стали короче, выполняем проверку: для каждой вершины *u* смежной с вершиной *v* и не включенной в *H* проверяем и корректируем расстояние *D*[*u*]



D[3] = 60

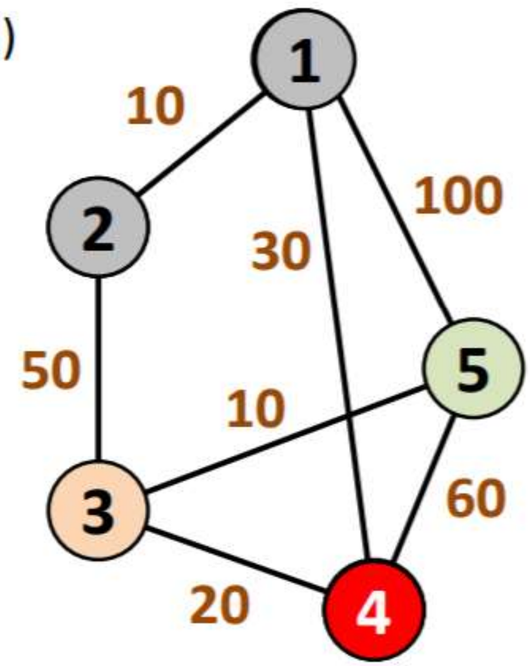
D[*i*]	0	10	∞	30	100

D[*i*]	0	10	60	30	100

Определили путь до 3 вершины

Алгоритм Дейкстры

4. Запускаем цикл из *n* итераций (по числу вершин)
   1. Извлекаем из очереди *Q* вершину *v* с минимальным приоритетом – ближайшую к *s* вершину
   2. Отмечаем вершину *v* как посещенную (помещаем *v* во множество *H*)
   3. Возможно пути из *s* через вершину *v* стали короче, выполняем проверку: для каждой вершины *u* смежной с вершиной *v* и не включенной в *H* проверяем и корректируем расстояние *D*[*u*]



D[3] = 50
D[5] = 90

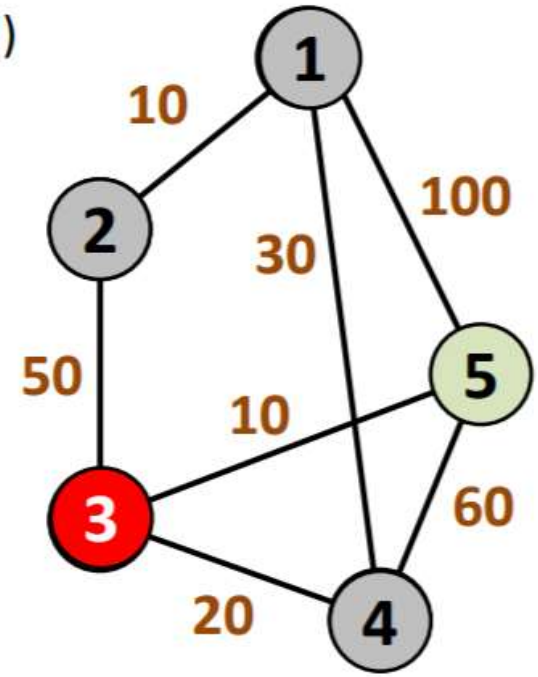
короче

D[*i*]	0	10	60	30	100
D[*i*]	0	10	50	30	90

Нашли более короткий путь до 3 вершины 50 через 4 ую
И общий путь 1,4,5 -90

Алгоритм Дейкстры

4. Запускаем цикл из *n* итераций (по числу вершин)
    1. Извлекаем из очереди *Q* вершину *v* с минимальным приоритетом – ближайшую к *s* вершину
    2. Отмечаем вершину *v* как посещенную (помещаем *v* во множество *H*)
    3. Возможно пути из *s* через вершину *v* стали короче, выполняем проверку: для каждой вершины *u* смежной с вершиной *v* и не включенной в *H* проверяем и корректируем расстояние *D*[*u*]



D[5] = 60

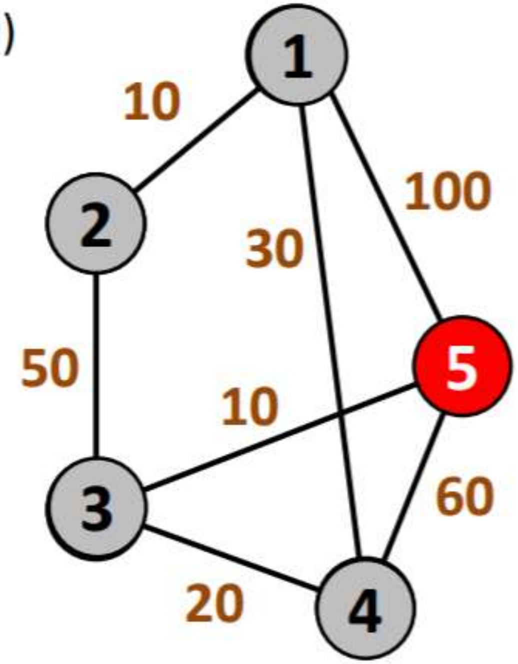
D[*i*]	0	10	50	30	90

D[*i*]	0	10	50	30	60

Нашли путь до 5 вершины через 1,4,3,5

Алгоритм Дейкстры

4. Запускаем цикл из *n* итераций (по числу вершин)
   1. Извлекаем из очереди *Q* вершину *v* с минимальным приоритетом – ближайшую к *s* вершину
   2. Отмечаем вершину *v* как посещенную (помещаем *v* во множество *H*)
   3. Возможно пути из *s* через вершину *v* стали короче, выполняем проверку: для каждой вершины *u* смежной с вершиной *v* и не включенной в *H* проверяем и корректируем расстояние *D*[*u*]

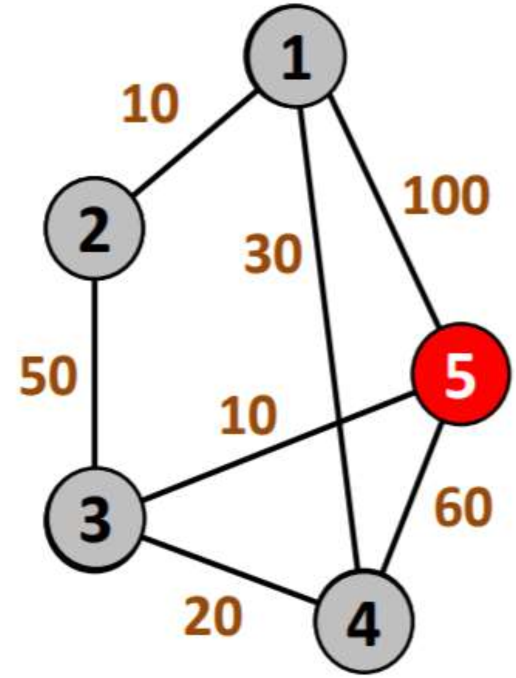


Осталась последняя вершина 5 проверяем все пути до нее, нет ли короче

***D*[*i*]**	**0**	**10**	**50**	**30**	**60**

Алгоритм Дейкстры

- В массиве D[1:*n*] содержаться длины кратчайших путей из начальной вершины *s* = 1
  - *D*[1] – длина пути из 1 в 1
  - *D*[2] – длина пути из 1 в 2
  - *D*[3] – длина пути из 1 в 3
  - *D*[4] – длина пути из 1 в 4
  - *D*[5] – длина пути из 1 в 5



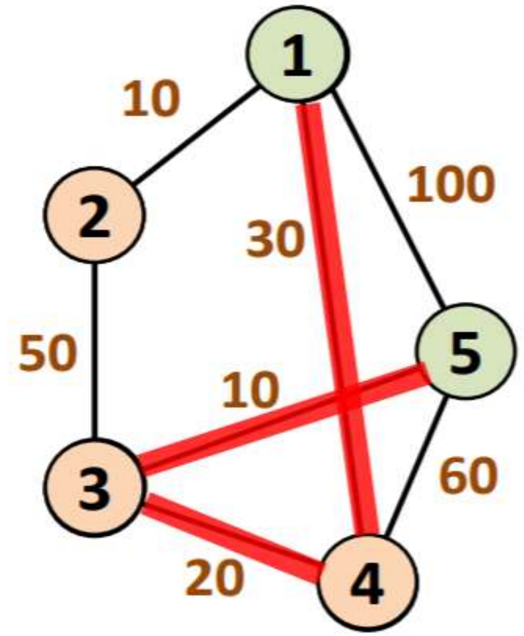
***D*[*i*]**					
	0	**10**	**50**	**30**	**60**

- **Как определить какие *вершины* входят в кратчайший путь из *s* = 1 в *d* = 5?**
- **Как восстановить путь?**

Алгоритм Дейкстры

- **Восстановление кратчайшего пути**
- Массив prev[*i*] содержит номер вершины, предшествующей *i* в пути

	1	2	3	4	5
prev[*i*]	**-1**	**1**	**4**	**1**	**3**



***D*[*i*]**	**0**	**10**	**50**	**30**	**60**

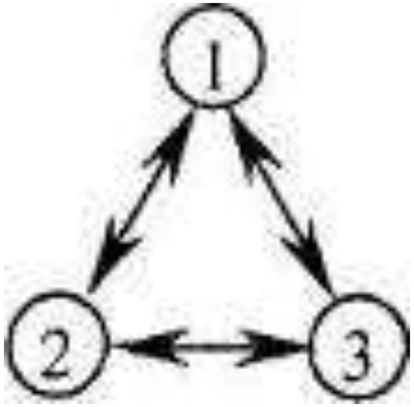
- **Восстанавливаем путь с конца**
  - Вершина 5
  - Вершина prev[5] = 3
  - Вершина prev[3] = 4
  - Вершина prev[4] = 1

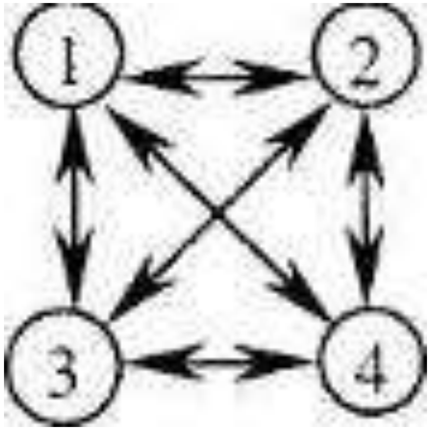
Кратчайший путь (1, 4, 3, 5)

Задачи с применением графов

Задача 1. По окончании деловой встречи специалисты обменялись визитными карточками (каждый вручил свою карточку каждому). Сколько всего визитных карточек было роздано, если во встрече участвовали: 1) 3 человека; 2) 4 человека; 3) 5 человек?

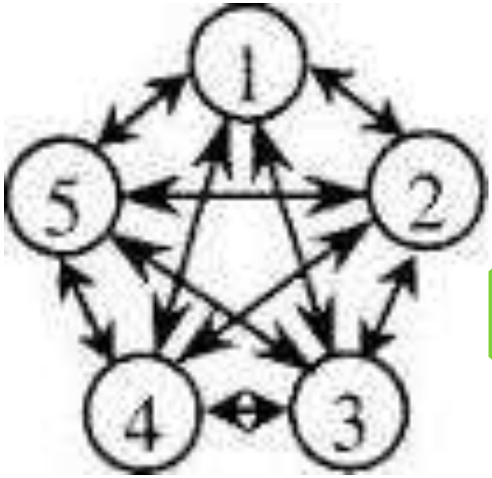
1) Во встрече участвовали 3 человека:





2) Во встрече участвовали 4 человека:

3) Во встрече участвовали 5 человек.

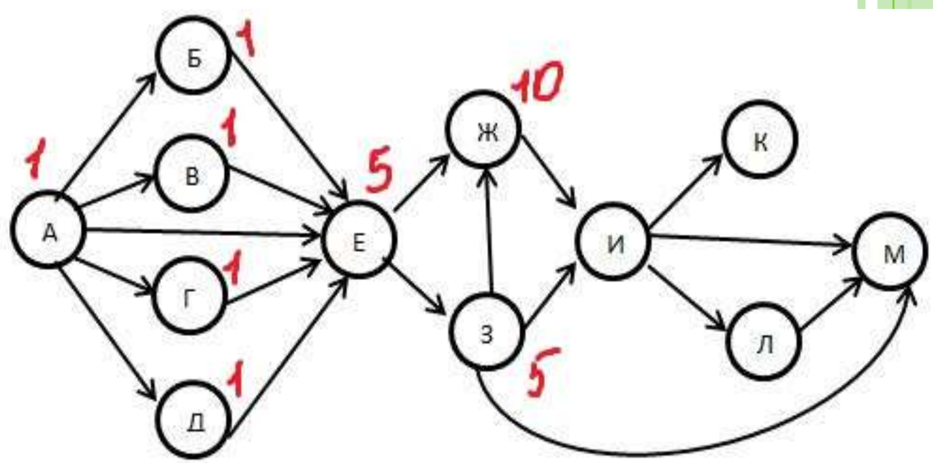


Задание № 3

- Выбор рисунка графа, соответствующего весовой матрице
- Выбор весовой матрицы по некоторому критерию
- Оптимальный маршрут путешественника
- Оптимальный маршрут по весовой матрице
- Сопоставление вершин графа и весовой матрицы

Задание № 15

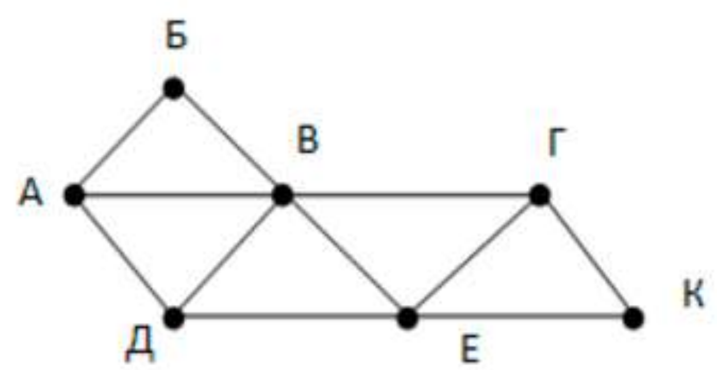
- Поиск количества путей в графе



3.1.Поиск длины маршрута

(№ 88) На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

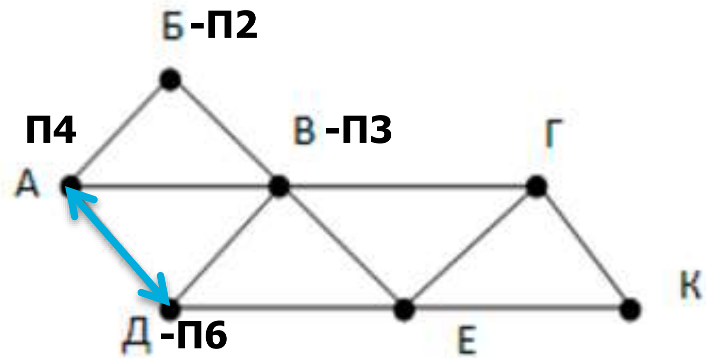
	П1	П2	П3	П4	П5	П6	П7
П1			30		25		18
П2			17	12			
П3	30	17		23		34	15
П4		12	23			46	
П5	25						37
П6			34	46			18
П7	18		15		37	18	



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта А в пункт Д.

Поиск длины маршрута от А до Д

	П1	П2	П3	П4	П5	П6	П7
П1			30		25		18
П2			17	12			
П3	30	17		23		34	15
П4		12	23			46	
П5	25						37
П6			34	46			18
П7	18		15		37	18	



Решение

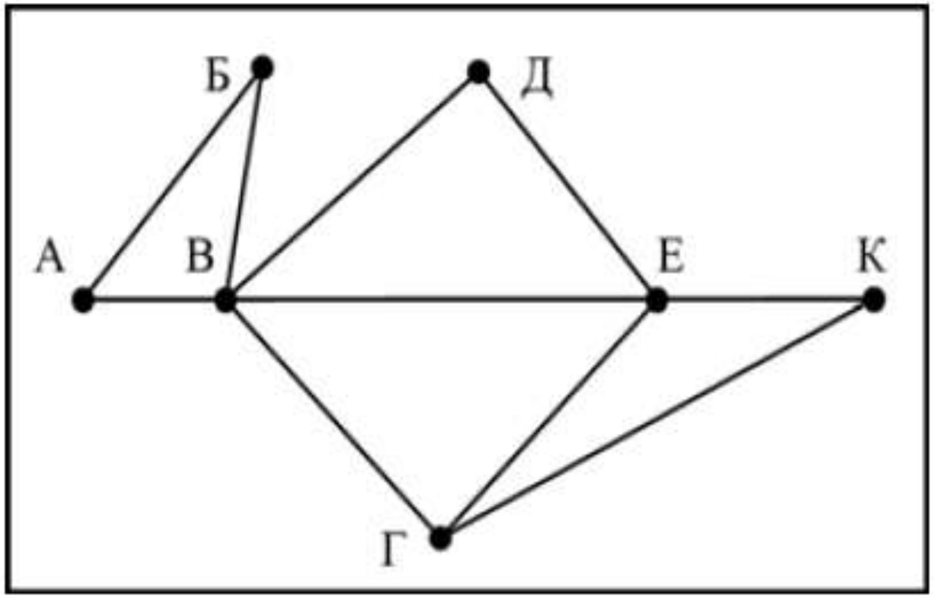
1) Определим какая вершина графа соответствует каждому пункту таблицы. Начнем с вершины В – туда входит максимальное число ребер – 5. Найдем в таблице строку, в которой пять чисел. Это пункт – П3.
2) П3 связана с единственной вершиной, в которую входят 2 ребра. По таблице определим что это за вершина – это Б или П4. Вершина Б связана с двумя вершинами В и А, то есть А это П4, а вершина Д – это П6.
3) Между этими вершинами есть прямой путь **П4-П6** равный **46.**

Ответ: 46

3.2.Поиск длины маршрута от В до Е

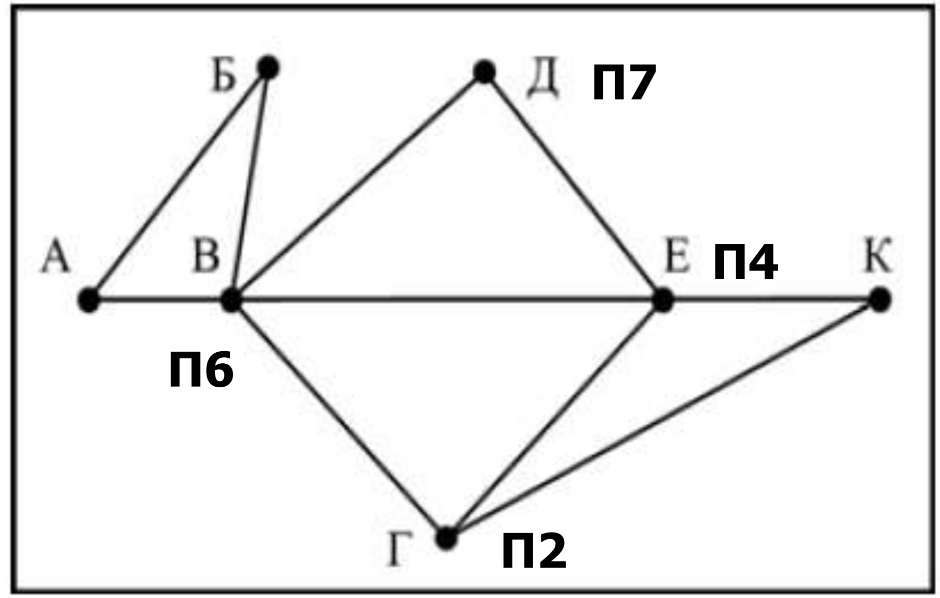
На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах). Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта В в пункт Е. В ответе запишите целое число

	П1	П2	П3	П4	П5	П6	П7
П1		45		10			
П2	45			40		55	
П3					15	60	
П4	10	40				20	35
П5			15			55	
П6		55	60	20	55		45
П7				35		45	



3.2.Поиск длины маршрута от В до Е

	П1	П2	П3	П4	П5	П6	П7
П1		45		10			
П2	45			40		55	
П3					15	60	
П4	10	40				20	35
П5			15			55	
П6		55	60	20	55		45
П7				35		45	



1)Надо посчитать сколько путей ведет из пункта В и найти в таблице соответствующую строку. Это строка П6 в таблице.

2)В вершину Е входит 4 дуги, значит ей соответствует строка П4.

Это строки в таблице № 6 и № 4.

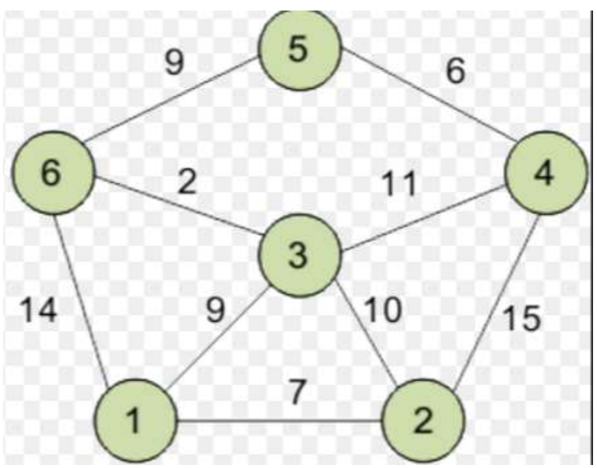
Надо определить длину дороги из пункта В в пункт Е.

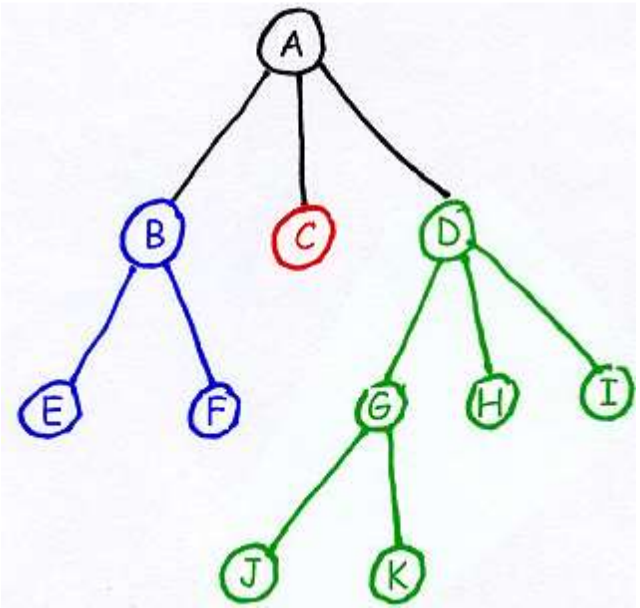
Ответ: 20

Поиск кратчайшего пути

Методы решения

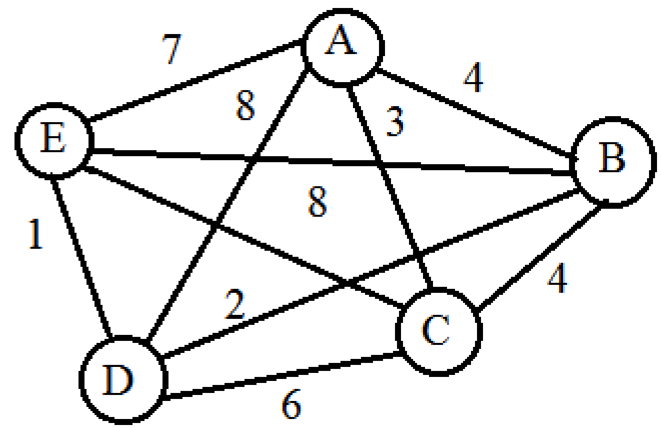
1) Построение графа
2) Построение дерева кратчайших путей
3) Перебор возможных путей без построения дерева
4) Использование алгоритма Дейкстры





1 способ. Построение графа

	A	B	C	D	E
A		4	3	8	7
B	4		4	2	2
C	3	4		6	3
D	2	2	6		1
E	7	8	3	1	



Определим самый короткий путь от вершины А до вершины Е

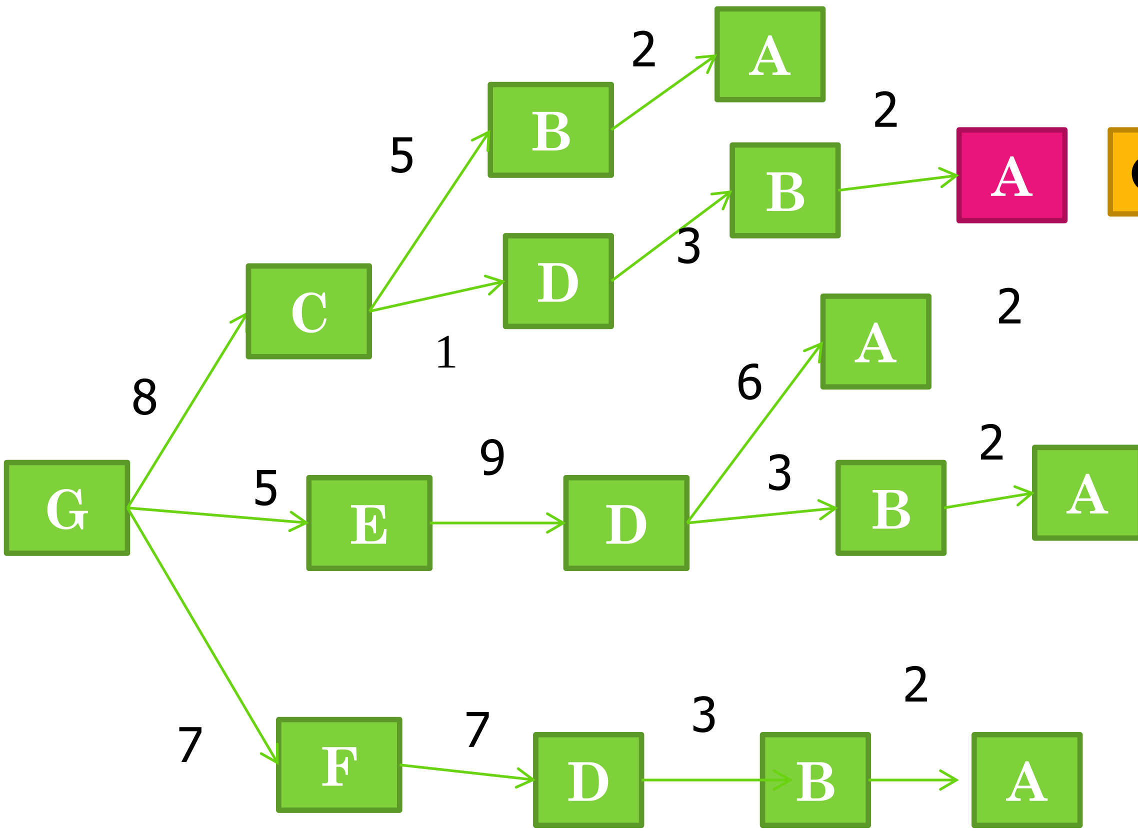
2 способ. Построение дерева
Поиск кратчайшего пути

- Между населёнными пунктами A, B, C, D, E, F, G построены дороги, протяжённость которых приведена в таблице. Отсутствие числа в таблице означает, что прямой дороги между пункта ми нет.
- Определите длину **кратчайшего пути** между пунктами A и G. Передвигаться можно только по указанным дорогам.

	A	B	C	D	E	F	G
A		2		6			
B	2		5	3			
C		5		1			8
D	6	3	1		9	7	
E				9			5
F				7			7
G			8		5	7	

Кратчайший путь между A и G

	A	B	C	D	E	F	G
A		2		6			
B	2		5	3			
C		5		1			8
D	6	3	1		9	7	
E				9			5
F				7			7
G			8		5	7	



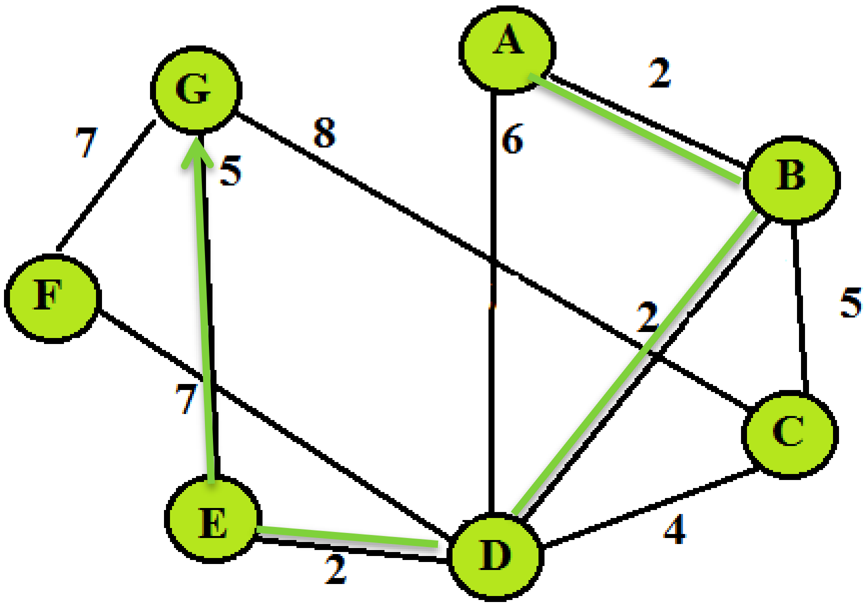
Ответ: 14

Задание № 3

- Между населёнными пунктами A, B, C, D, E, F, G построены дороги, протяжённость которых приведена в таблице. Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.
- Определите длину кратчайшего пути между пунктами A и G. Передвигаться можно только по указанным дорогам.

	A	B	C	D	E	F	G
A		2		6			
B	2		5	2			
C		5		4			8
D	6	2	4		2	7	
E				2			5
F				7			7
G			8		5	7	

Кратчайший путь между пунктами A и G



	A	B	C	D	E	F	G
A		2		6			
B	2		5	2			
C		5		4			8
D	6	2	4		2	7	
E				2			5
F				7			7
G			8		5	7	

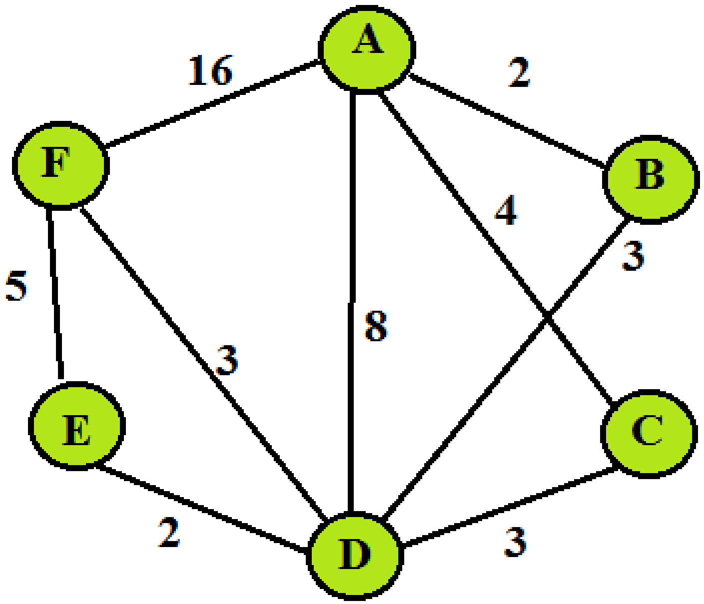
- ABCDFG=2+5+4+7+7=25
- ABDFG=2+2+7+7=18
- ADFG=6+7+7=20
- ABCDEG=2+5+4+2+5=18
- **ABDEG=2+2+2+5=11**
- ADEG=6+2+5=13
- ABCG=2+5+8=15
- ADCG=6+4+8=18

Задание № 3

- Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. Отсутствие числа в таблице значит, что прямой дороги между пунктами нет.
- Определите **длину кратчайшего пути между A и F,** проходящего через пункт E и не проходящего через пункт B. Передвигаться можно только по указанным дорогам.

	A	B	C	D	E	F
A		2	4	8		16
B	2			3		
C	4			3		
D	8	3	3		2	3
E				2		5
F	16			3	5	

Min путь из A в F, проходящий через E и не проходящий через пункт B



	A	B	C	D	E	F
A		2	4	8		16
B	2			3		
C	4			3		
D	8	3	3		2	3
E				2		5
F	16			3	5	

ACDEF=4+3+2+5=14

ADEF=8+2+5=15

Ответ: ACDEF=14