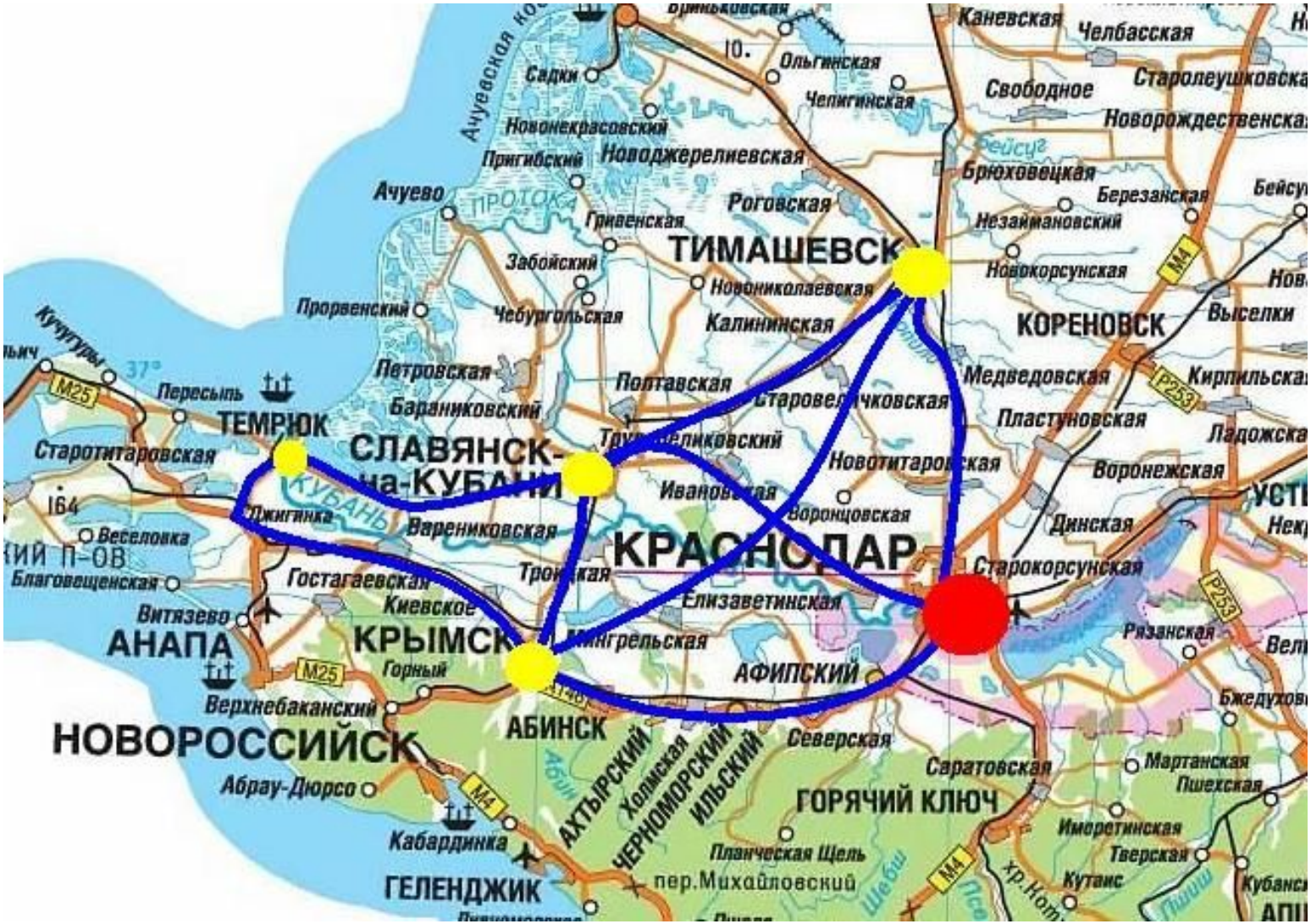
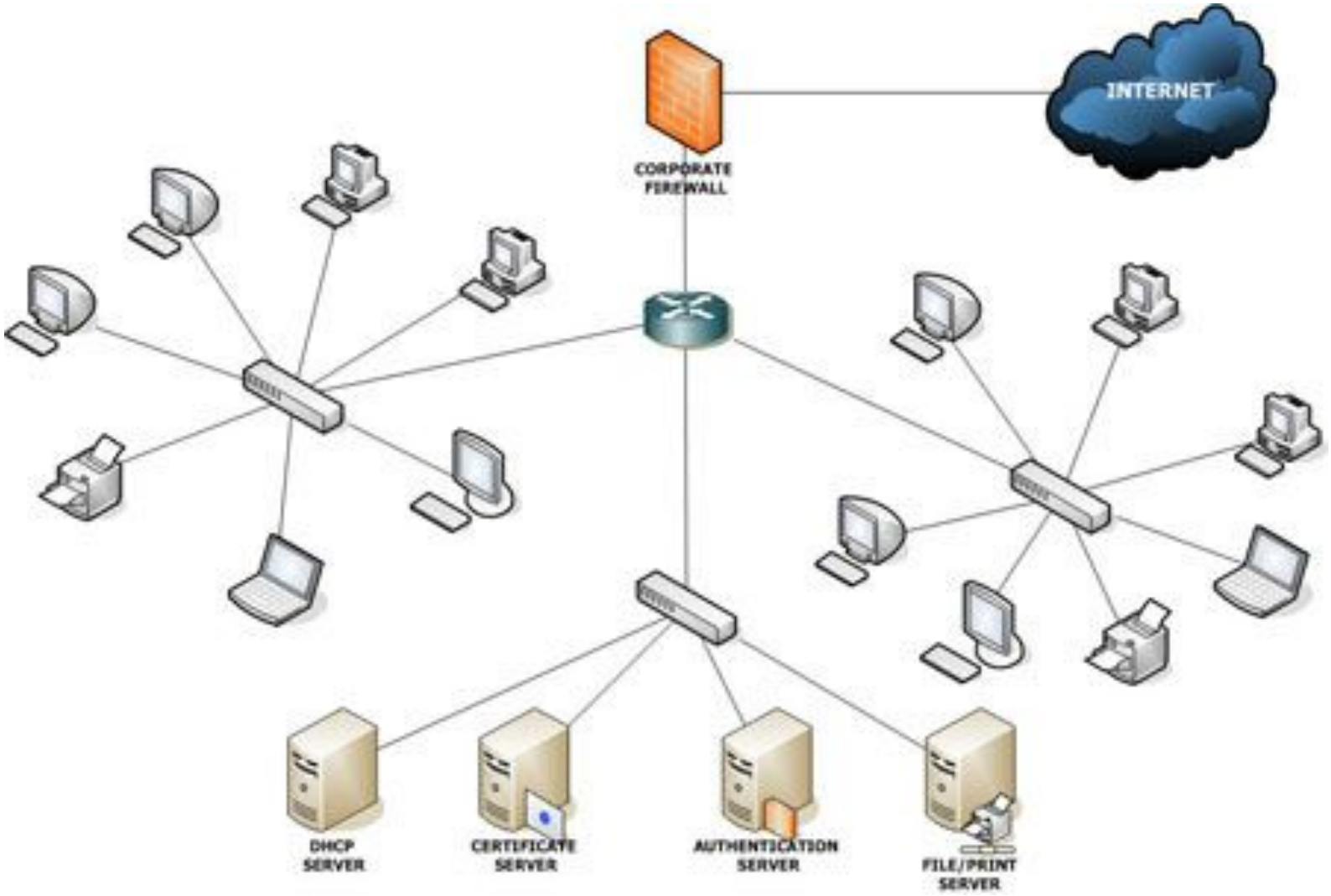
Теория графов

Примеры применения графов в реальной жизни



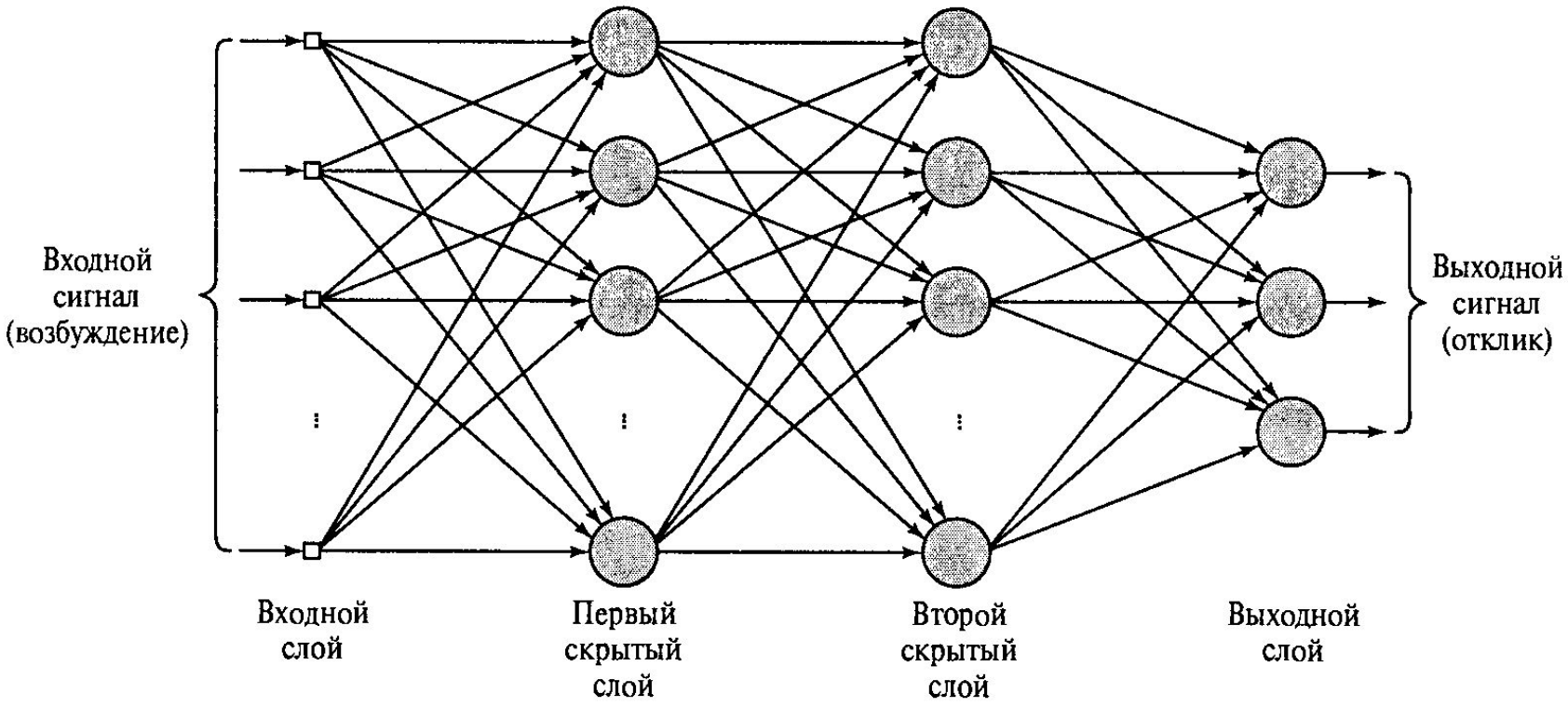
Примеры применения графов в реальной жизни

Примеры применения графов в реальной жизни

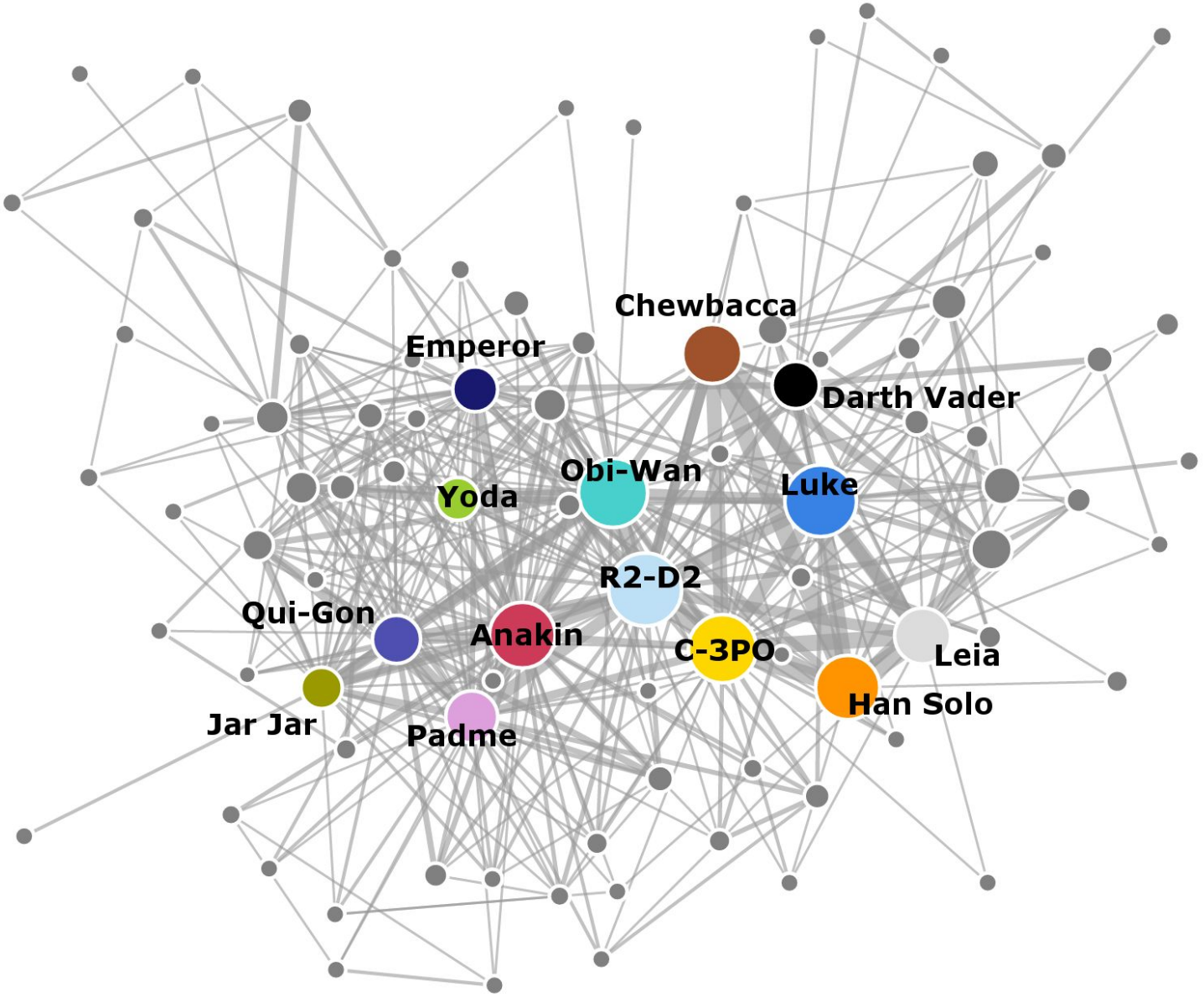


Примеры применения графов в реальной жизни

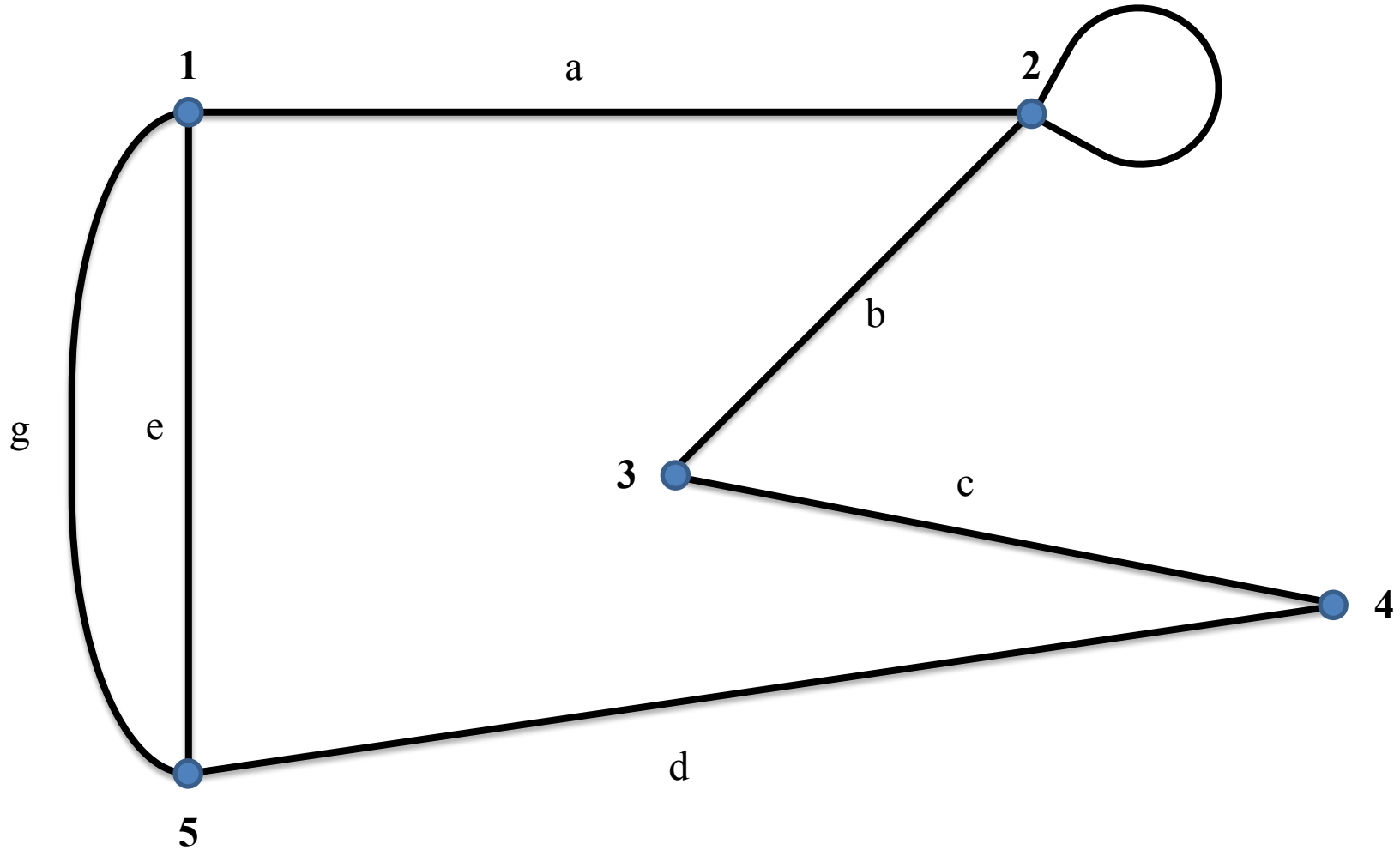
Примеры применения графов в реальной жизни



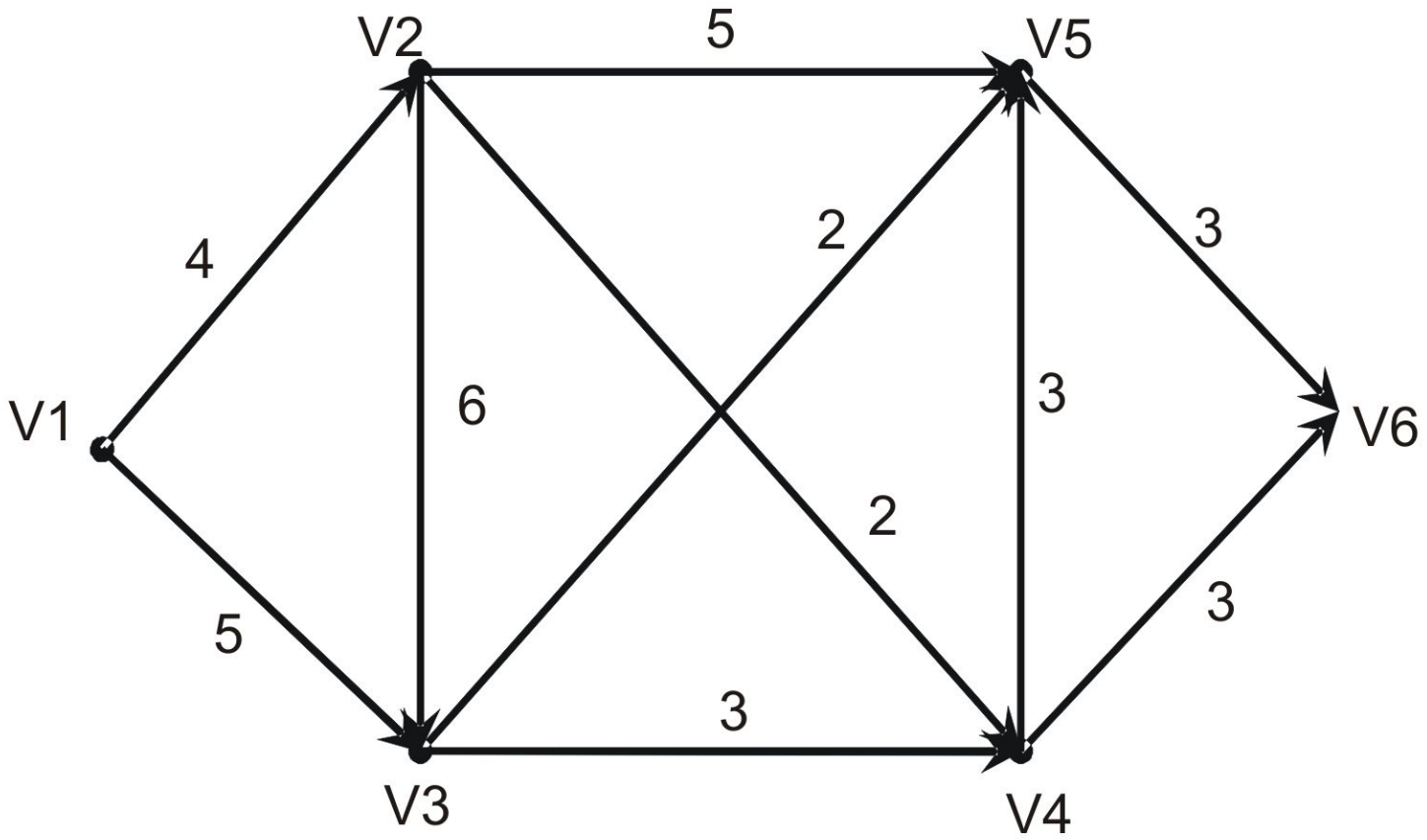
Примеры применения графов в реальной жизни



Граф – конечное множество вершин и множество ребер

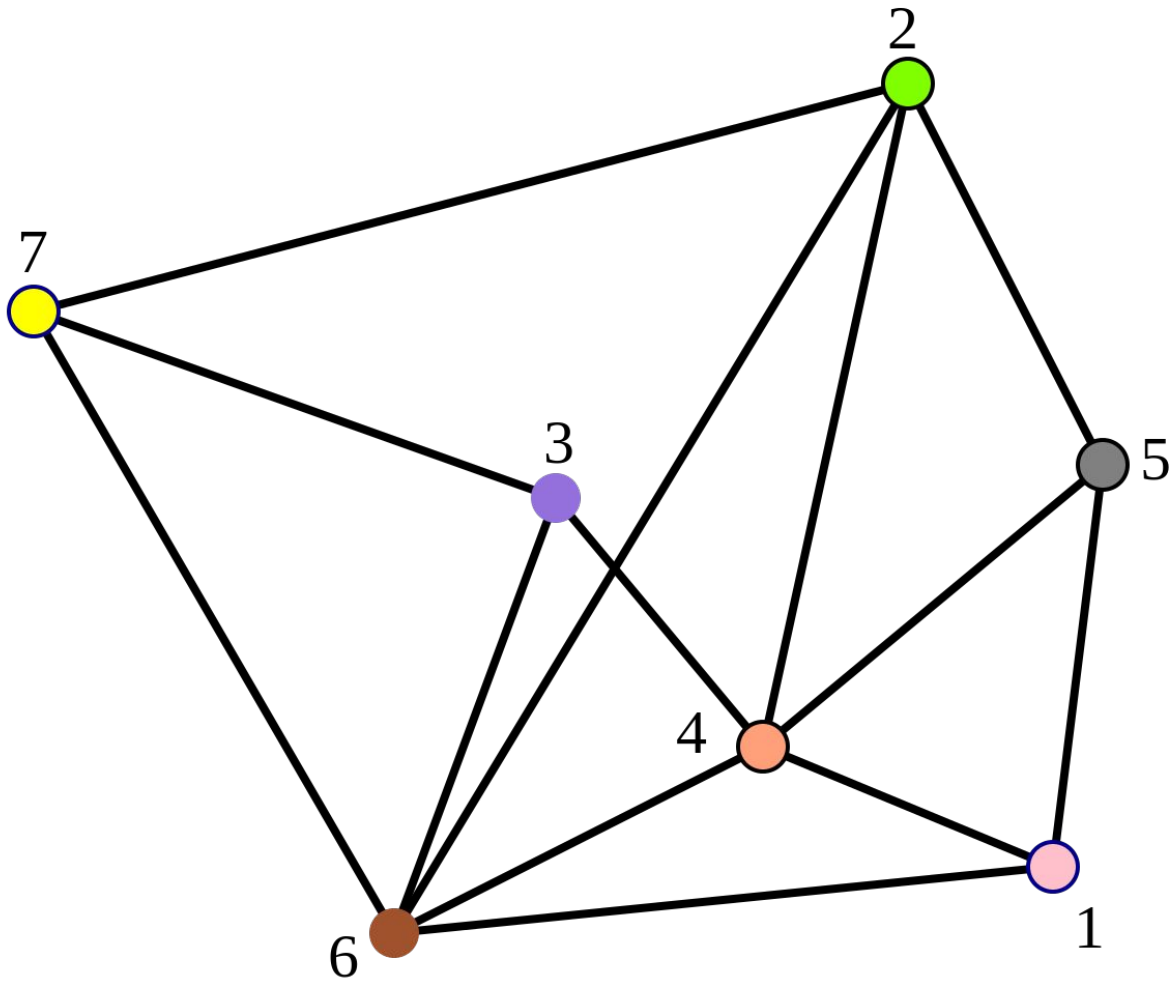


Ориентированный граф



Первая теорема теории графов

Сумма степеней всех вершин равна удвоенному числу ребер в графе



Задача

Найти максимальное число ребер в простом графе, если у него n вершин

Задача

Найти максимальное число ребер в простом графе, если у него n вершин

Решение:

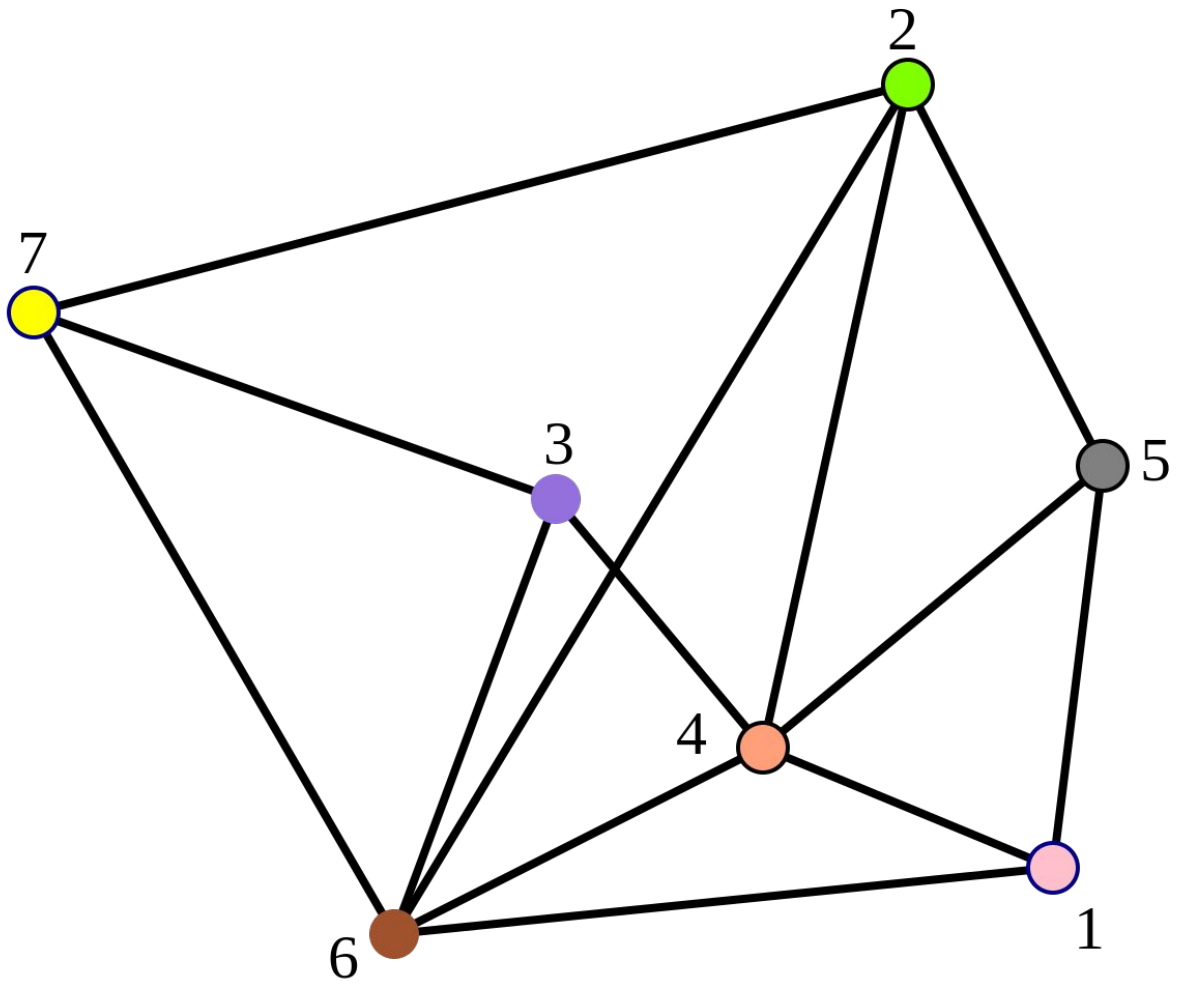
M – число ребер

$$2 * M = n * (n - 1)$$

$$M = \frac{n * (n - 1)}{2}$$

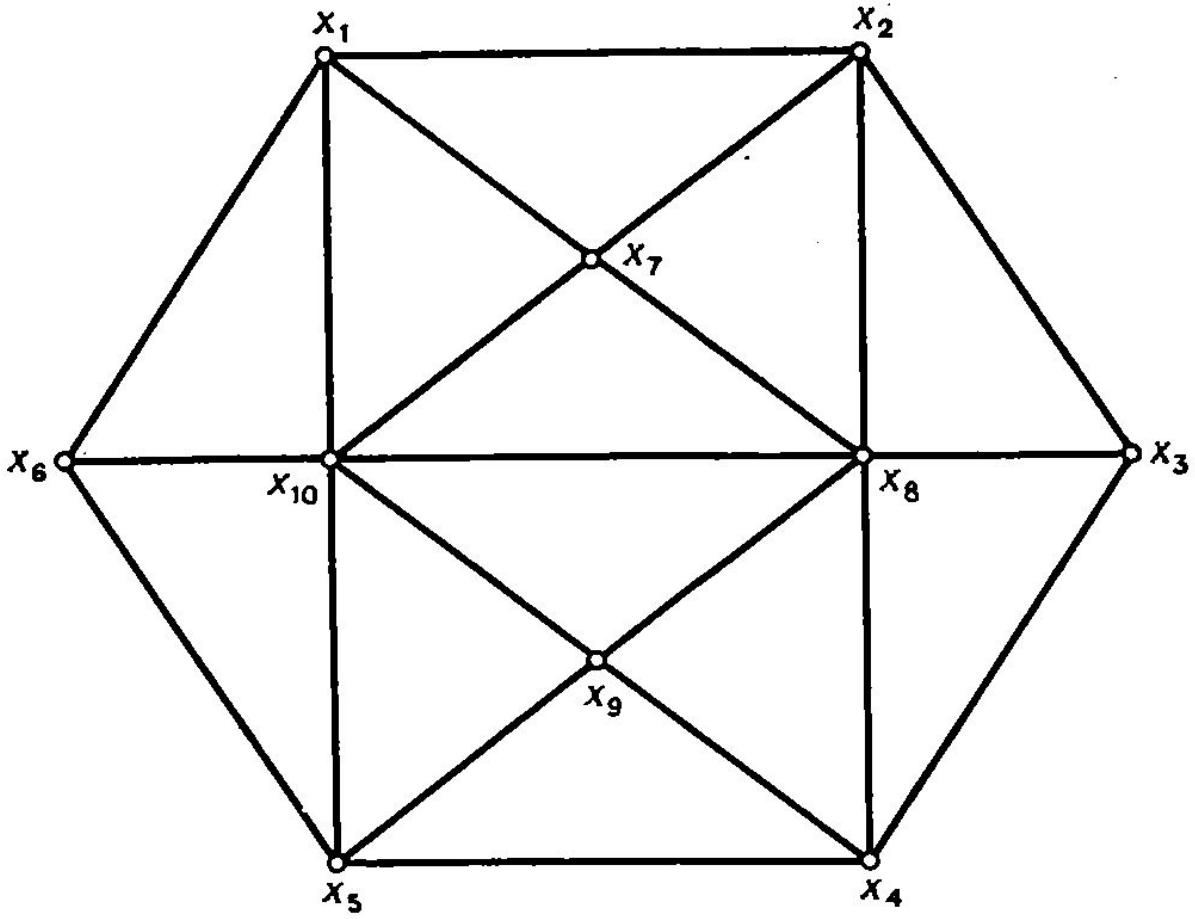
«Лемма о рукопожатиях»

В любом графе число вершин нечетной степени - четно



«Лемма о рукопожатиях»

В любом графе число вершин нечетной степени - четно



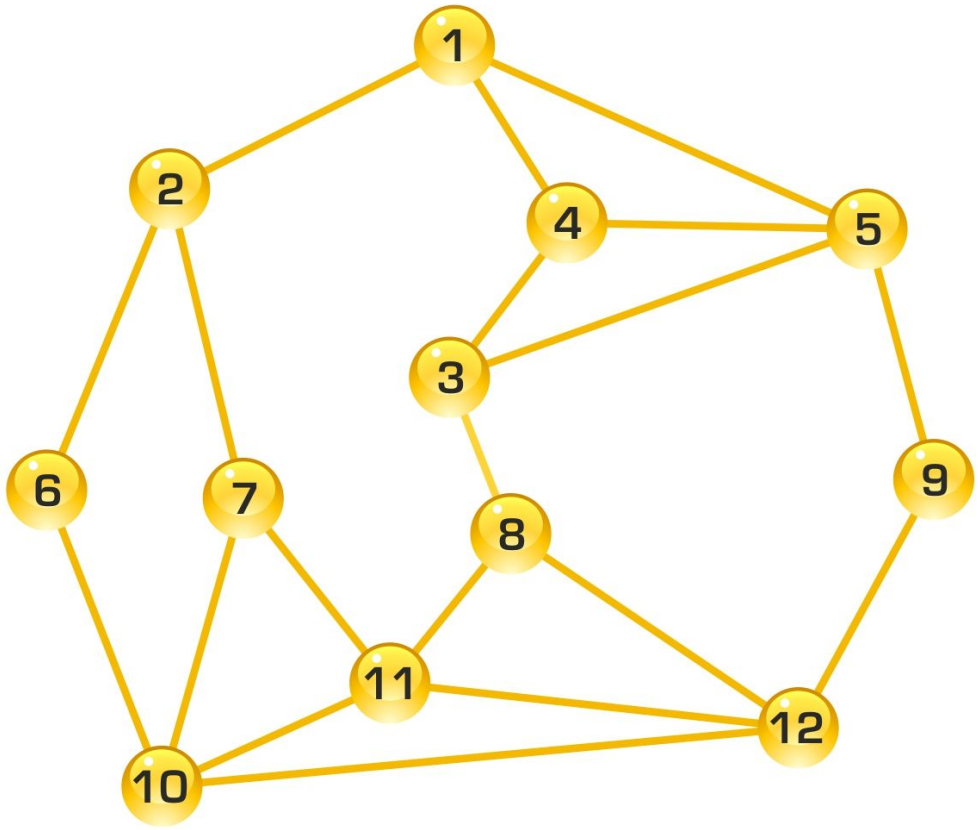
Задача

Какая из представленных числовых последовательностей может быть последовательностью степеней вершин графа

1) 2 2 2 1
2) 3
3) 5 2 2 1
4) 8 6 5 4 4 4 4 4 4 3 1 1 1 1 1 1 1 1 1 1
5) 2
6) 8 6 5 4 4 4 4 4 4 3 1 1 1 1 1 1 1 1 1
7) 5 4 3 3
8) 2 2 1 1

Путь в графе

– это последовательность ребер, в которой конец одного ребра является началом следующего ребра

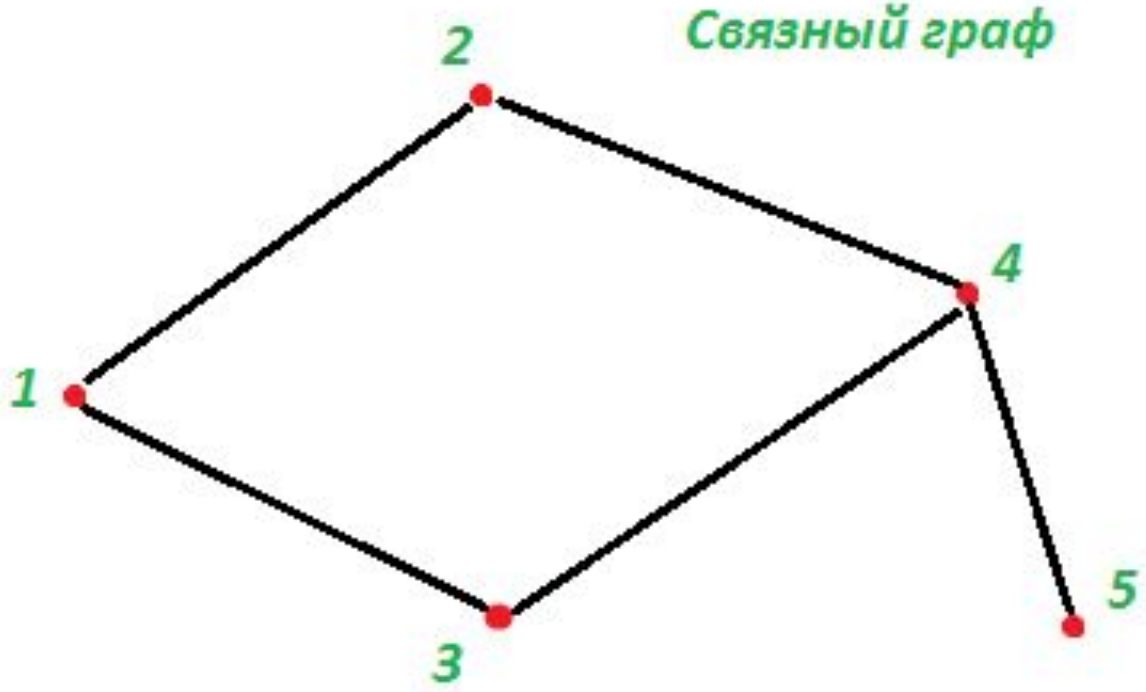


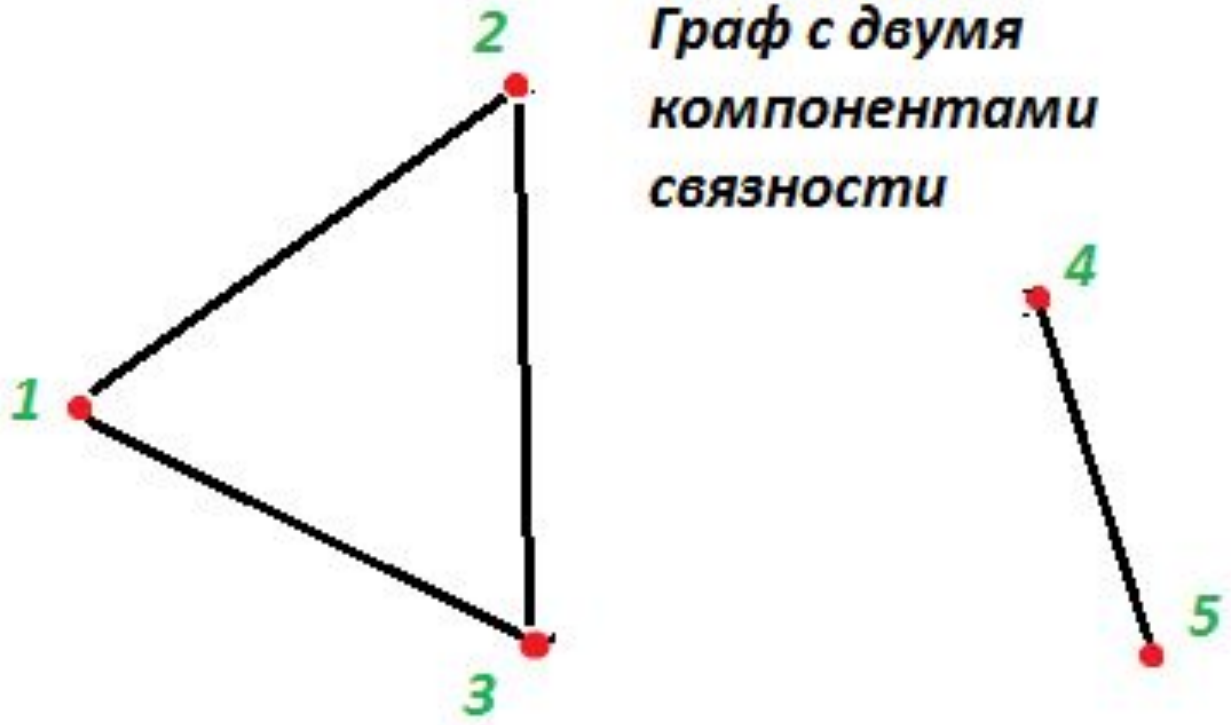
Связный граф

Граф называется **связным** если между любыми двумя его вершинами существует маршрут. В противном случае граф называется несвязным.

Любой несвязный граф состоит из нескольких связных графов, каждый из которых называется **компонентой связности графа**. В частности у связного графа ровно одна компонента связности.

Связный граф

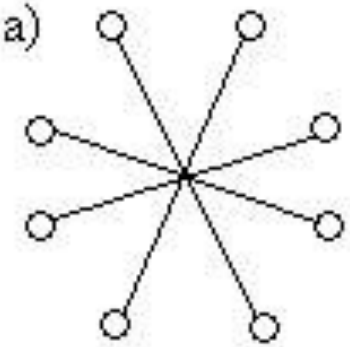




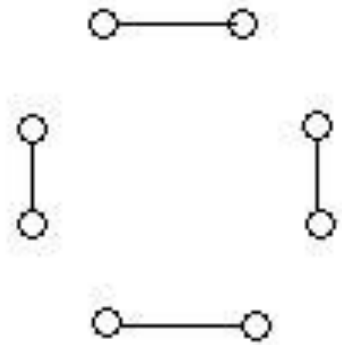
Граф с двумя компонентами связности

Определите, какие графы являются связными, а какие нет

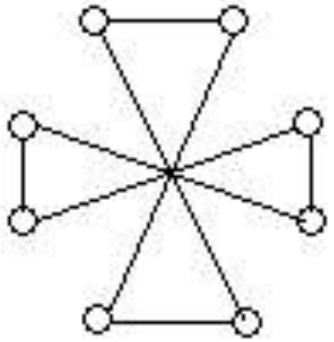
а)



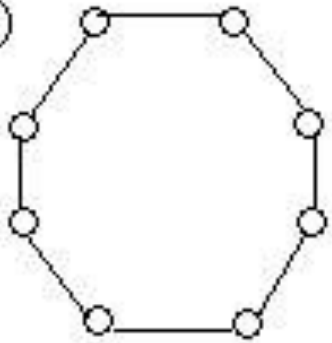
б)



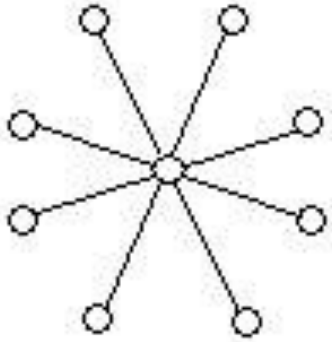
в)



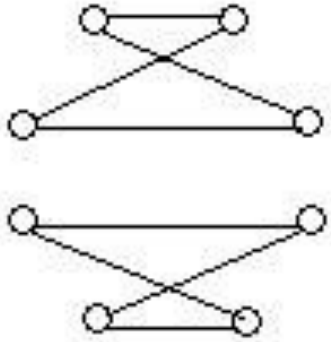
г)



д)



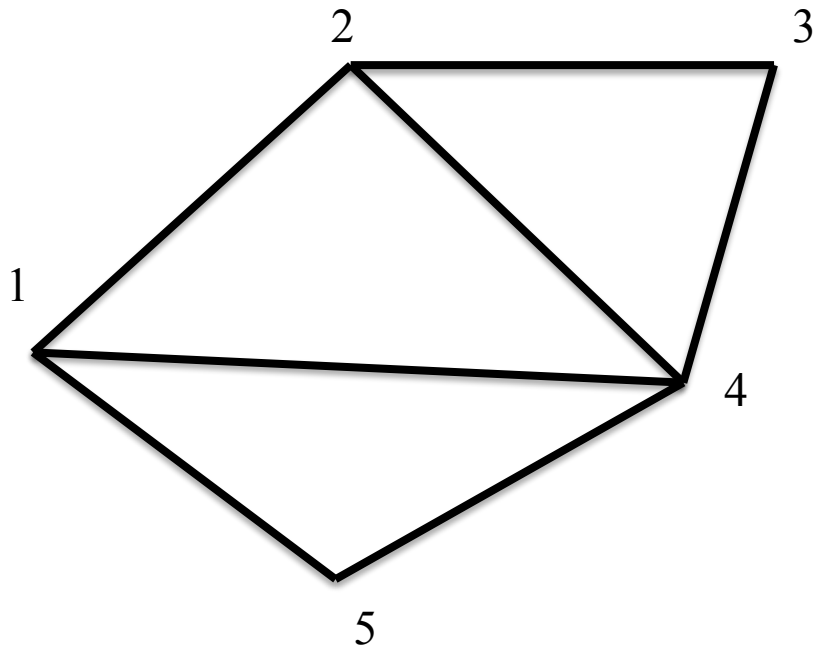
е)



Способы задания графа в памяти ПК

Матрица смежности

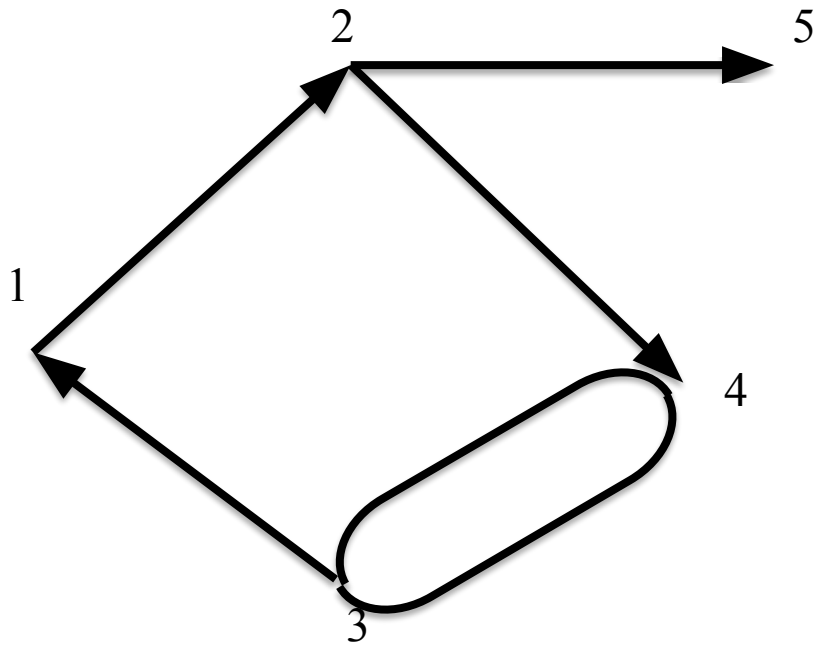
Неориентированный граф



	1	2	3	4	5
1	0	1	1	1	0
2	1	0	0	1	1
3	1	0	0	1	0
4	1	1	1	0	1
5	0	1	0	1	0

Матрица смежности

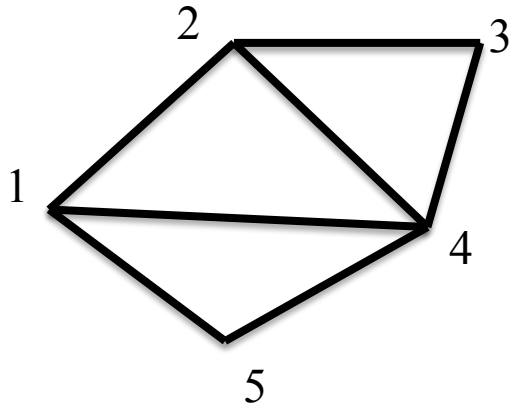
Ориентированный граф



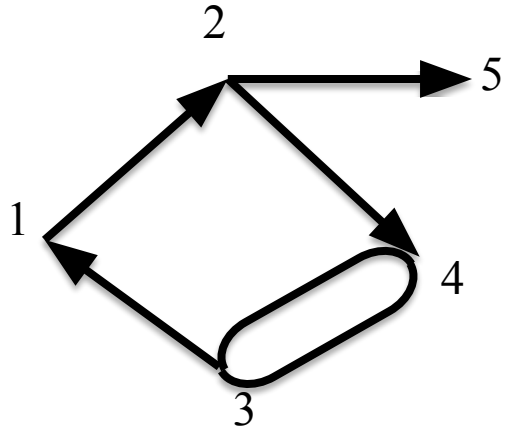
	1	2	3	4	5
1	0	1	0	0	0
2	0	0	0	1	1
3	1	0	0	1	0
4	0	0	1	0	0
5	0	0	0	0	0

Списки смежности

Для каждой вершины хранится список w[i] смежных с ней вершин



w[1]=[2, 3, 4]
w[2]=[1, 4, 5]
w[3]=[1, 4]
w[4]=[1, 2, 3, 5]
w[5]=[2, 4]



w[1]=[2]
w[2]=[4, 5]
w[3]=[1, 4]
w[4]=[1, 3]
w[5]=[ ]

Способы задания графа в Python

Задача №1 к §45

Напишите программу, которая строит списки смежности для каждой вершины графа на основе его матрицы смежности.

Входные данные

В первой строке вводится количество вершин графа N ($1 \le N \le 1000$). В следующих N строках записано по N чисел, разделённых пробелами – элементы матрицы смежности графа.

Выходные данные

Программа должна вывести списки смежности для каждой вершины графа в порядке возрастания их номеров. Номера вершин в каждом списке разделены пробелами. Нумерация начинается с единицы. Если из вершины не выходит ни одно ребро, вместо списка нужно вывести число 0.

```
n=int(input())
ms=[[0] * n for i in range(n)]
for i in range(n):
    m=map(int,input().split())
    ms[i]=list(m)
for i in range(n):
    count=0
    for j in range(n):
        if ms[i][j]==1:
            print(j+1,end=' ')
            count+=1
    if count==0:
            print(0,end=' ')
    print(sep=' ')
```

Структуры данных и алгоритмы/Алгоритмы на графах/1. Основные понятия. Способы задания графов

В галактике "Milky Way" на планете "Neptune" есть N городов, некоторые из которых соединены дорогами. Император "Maximus" галактики "Milky Way" решил провести инвентаризацию дорог на планете "Neptune". Но, как оказалось, он не силен в математике, поэтому он просит вас сосчитать количество дорог.

Входные данные

В первой строке задается число N ($0 \leq N \leq 100$). В следующих N строках содержится по N чисел, каждое из которых является единичкой или ноликом. Причем, если в позиции (i,j) квадратной матрицы стоит единичка, то i-ый и j-ый города соединены дорогами, а если нолик, то не соединены.

Выходные данные

Выведите одно число – количество дорог на планете "Neptune".

```
import numpy as np
n=int(input())
ms=[[0] * n for i in range(n)]
for i in range(n):
    m=map(int,input().split())
    ms[i]=list(m)
s=0
for i in range(n):
    for j in range(i,n):
        s+=ms[i][j]
print(s)
```

```
import numpy as np
n=int(input())
ms=[[0] * n for i in range(n)]
for i in range(n):
    m=map(int,input().split())
    ms[i]=list(m)
print(ms)
ms2=np.array(ms)
print(ms2)
print(np.sum(ms2))
```

https://pythonworld.ru/numpy/2.html

print(ms)
[[0, 1, 0, 0, 0], [1, 0, 1, 1, 0], [0, 1, 0, 0, 0], [0, 1, 0, 0, 0], [0, 0, 0, 0, 0]]
print(ms2)
[[0 1 0 0 0] [1 0 1 1 0] [0 1 0 0 0] [0 1 0 0 0] [0 0 0 0 0]]

Задача «Светофорчики»

В подземелье M тоннелей и N перекрестков, каждый тоннель соединяет какие-то два перекрестка. Мышиный король решил поставить по светофору в каждом тоннеле перед каждым перекрестком. Напишите программу, которая посчитает, сколько светофоров должно быть установлено на каждом из перекрестков. Перекрестки пронумерованы числами от 1 до N.

Входные данные

Первая строка входных данных содержит два числа N и M ($0 < N \le 100$, $0 \le M \le N*(N-1)/2$). В каждой из следующих M строк записаны по два числа i и j ($1 \le i,j \le N$), которые означают, что перекрестки i и j соединены тоннелем.

Выходные данные

Требуется вывести N чисел: k-ое число означает количество светофоров на k-ом перекрестке.

Примечание. Можно считать, что любые два перекрестка соединены не более, чем одним тоннелем. Нет тоннелей от перекрестка i до него самого.

Примеры

входные данные

7 10
5 1
3 2
7 1
5 2
7 4
6 5
6 4
7 5
2 1
5 3

выходные данные

3 3 2 2 5 2 3

```
n,m=map(int,input().split())
ms=[[0] * (n) for i in range(n)]
for i in range(m):
    a,b=map(int,input().split())
    ms[a-1][b-1]=1
    ms[b-1][a-1]=1
for i in range(n):
    print(sum(ms[i]),end=' ')
```