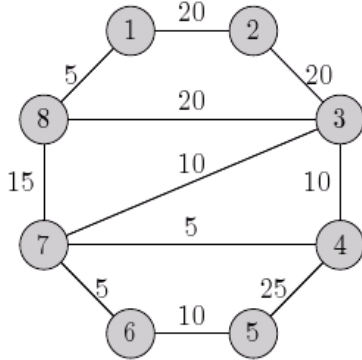


Графы: каркас минимальной стоимости

На рисунке изображён неориентированный граф, на примере которого мы будем разбирать алгоритмы этого раздела. Во всех задачах этого раздела мы будем считать, что вершины графа — «города», рёбра — соединяющие их «дороги», не имеющие направления, но зато имеющие *вес* (длину или, например, стоимость проезда)



40	Список рёбер	2 балла
-----------	---------------------	----------------

Для описанных далее алгоритмов нам будет удобно хранить граф не в виде матрицы смежности, а в виде *списка рёбер*. В самом простом варианте список рёбер графа можно представить так:

```
int from[100];
int to[100];
int weight[100];
```

В каждом из массивов будет заполнено M первых элементов (по количеству рёбер в графе). Для графа на рисунке содержимое массивов будет таким:

from:	1	2	3	4	5	6	7	1	3	3	4
to:	2	3	4	5	6	7	8	8	8	7	7
weight:	20	20	10	25	10	5	15	5	20	10	5

Граф задан в файле `input.txt` в следующем виде: N (количество вершин), M (количество рёбер), затем M строк по 3 числа: номера двух вершин и длина соединяющего их ребра. Прочитайте этот граф и заполните три массива `from`, `to` и `weight`.

41	Построение каркаса	4 балла
-----------	---------------------------	----------------

Каркасом связного неориентированного графа называется его связный подграф, который содержит все вершины исходного графа и не содержит циклов. Если в связном неориентированном графе N вершин, то в его каркасе будет ровно $N - 1$ ребро.

Строить каркас графа можно, добавляя рёбра к изначально пустому графу, при этом следя за тем, чтобы после добавления очередного ребра не образовался цикл. Для этого будем, как и раньше, отмечать «использованные» вершины в массиве `int used[100]`. Пусть изначально `used[1] = 1` и `used[i] = 0` для всех $i \neq 1$. Повторим следующие операции ровно $N - 1$ раз:

1. Найдём ребро (a, b) такое, что вершина a использована, а вершина b — нет. Обратите внимание, что рёбра у нас не имеют направлений, поэтому для ребра номер i возможны четыре варианта: обе вершины (и `from[i]`, и `to[i]` использованы, обе неиспользованы или использована либо одна, либо другая; нас интересуют последние два случая.
2. Пометим вершину b как использованную.
3. Напечатаем (a, b) .

Действуя согласно этому алгоритму, мы построим подграф исходного графа из $N - 1$ рёбер, не содержащий циклов (благодаря проверкам на `used`) — т. е. некоторый каркас нашего графа.

42	Алгоритм Прима	5 баллов
-----------	-----------------------	-----------------

Модифицируем предыдущую задачу для того, чтобы построить каркас минимального веса. При выборе очередного ребра будем брать не первое попавшееся ребро, а ребро с самым маленьким весом (при этом проверяя, что оно нас устраивает: одна из вершин должна быть использованной, другая — нет).

При помощи такого алгоритма можно построить каркас с минимально возможным суммарным весом рёбер. Этот метод называется алгоритмом Прима.