

## Задача А. Обход в ширину

Имя входного файла: `bfs.in`  
Имя выходного файла: `bfs.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дан ориентированный граф. В нём необходимо найти расстояние от одной заданной вершины до другой.

### Формат входных данных

В первой строке входного файла содержится три натуральных числа  $N$ ,  $S$  и  $F$  ( $1 \leq S, F \leq N \leq 100$ ) — количество вершин в графе и номера начальной и конечной вершин соответственно. Далее в  $N$  строках задана матрица смежности графа. Если значение в  $j$ -м элементе  $i$ -й строки равно 1, то в графе есть направленное ребро из вершины  $i$  в вершину  $j$ .

### Формат выходных данных

В единственной строке должно находиться минимальное расстояние от начальной вершины до конечной. Если пути не существует, выведите 0.

### Пример

bfs.in	bfs.out
4 4 3 0 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0	2

## Задача В. Island. Островные государства

Имя входного файла: `island.in`  
Имя выходного файла: `island.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Суровые феодальные времена переживала некогда великая островная страна Байтландия. За главенство над всем островом борются два самых сильных барона. Таким образом, каждый город страны контролируется одним из правителей. Как водится издревле, некоторые из городов соединены двусторонними дорогами. Бароны очень не любят друг друга и стараются делать как можно больше пакостей. В частности, теперь для того чтобы пройти по дороге, соединяющей города различных правителей, надо заплатить пошлину — один байтландский рубль.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город  $N$  на Всебайтландское собрание программистов. Разумеется, он хочет затратить при этом как можно меньше денег и помочь ему здесь, как обычно, предлагается Вам.

### Формат входных данных

В первой строке входного файла записано два числа  $N$  и  $M$  ( $1 \leq N, M \leq 100\,000$ ) — количество городов и количество дорог соответственно.

В следующей строке содержится информация о городах —  $N$  чисел 1 или 2 — какому из баронов принадлежит соответствующий город.

В последних  $M$  строках записаны пары  $1 \leq a, b \leq N, a \neq b$ . Каждая пара означает наличие дороги из города  $a$  в город  $b$ . По дорогам Байтландии можно двигаться в любом направлении.

### Формат выходных данных

Если искомого пути не существует, выведите единственное слово **impossible**. В противном случае в первой строке напишите минимальную стоимость и количество посещенных городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

### Пример

island.in	island.out
7 8 1 1 1 1 2 2 1 1 2 2 5 2 3 5 4 4 3 4 7 1 6 6 7	0 5 1 2 3 4 7
5 5 1 2 1 1 2 1 2 2 3 3 5 1 4 4 5	1 3 1 4 5

## Задача С. Путь в случайном графе

Имя входного файла: `fb.in`  
Имя выходного файла: `fb.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дан случайный взвешенный ориентированный граф из  $n$  вершин,  $m$  рёбер. Гарантируется, что в графе нет отрицательных циклов. Ваша задача – найти расстояние от вершины  $s$  до вершины  $t$ . Что такое случайный граф? Рёбра равномерно случайного веса между равномерно случайными вершинами.

### Формат входных данных

На первой строке  $n\ m\ s\ t$  ( $1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 200\,000$ ,  $1 \leq s, t \leq n$ ). На следующих  $m$  строках тройки чисел  $a_i\ b_i\ w_i$  ( $1 \leq a, b \leq n$ ,  $|w| < 10^9$ ), рёбра из  $a_i$  в  $b_i$  веса  $w_i$ .

### Формат выходных данных

Одно число – длину кратчайшего пути.

Если пути из  $s$  в  $t$  не существует, выведите число  $10^{18}$ .

### Примеры

fb.in	fb.out
2 0 1 2	1000000000000000000
3 4 1 3 1 2 20 2 3 -15 1 3 10 3 1 -4	5

## Задача D. Path. Кратчайший путь

Имя входного файла: `path.in`  
Имя выходного файла: `path.out`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 megabytes

Дан взвешенный ориентированный граф и вершина  $s$  в нем. Требуется для каждой вершины  $u$  найти длину кратчайшего пути из  $s$  в  $u$ .

### Формат входных данных

Первая строка входного файла содержит  $n$ ,  $m$  и  $s$  — количество вершин, ребер и номер выделенной вершины соответственно ( $2 \leq n \leq 2\,000$ ,  $1 \leq m \leq 5\,000$ ).

Следующие  $m$  строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — целое число, не превосходящее  $10^{15}$  по модулю. В графе могут быть кратные ребра и петли.

### Формат выходных данных

Выведите  $n$  строк — для каждой вершины  $u$  выведите длину кратчайшего пути из  $s$  в  $u$ ,  $*$  если не существует путь из  $s$  в  $u$  и  $-$  если не существует кратчайший путь из  $s$  в  $u$ .

### Пример

path.in	path.out
6 7 1	0
1 2 10	10
2 3 5	-
1 3 100	-
3 5 7	-
5 4 10	*
4 3 -18	
6 1 -1	

## Задача Е. Расстояние между вершинами

Имя входного файла: `distance.in`  
Имя выходного файла: `distance.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дан неориентированный взвешенный граф. Найти вес минимального пути между двумя вершинами.

### Формат входных данных

Первая строка входного файла содержит натуральные числа  $N$ ,  $M$ , вторая строка содержит натуральные числа  $S$  и  $F$  ( $N \leq 5\,000$ ,  $M \leq 100\,000$ ,  $1 \leq S, F \leq N$ ,  $S \neq F$ ) — количество вершин и ребер графа а также номера вершин, длину пути между которыми требуется найти.

Следующие  $M$  строк по три натуральных числа  $b_i$ ,  $e_i$  и  $w_i$  — номера концов  $i$ -ого ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100\,000$ ).

### Формат выходных данных

Первая строка должна содержать одно натуральное число — вес минимального пути между вершинами  $S$  и  $F$ . Во второй строке через пробел выведите вершины на кратчайшем пути из  $S$  в  $F$  в порядке обхода.

Если путь из  $S$  в  $F$  не существует, выведите  $-1$ .

### Пример

<code>distance.in</code>	<code>distance.out</code>
4 4 1 3 1 2 1 3 4 5 3 2 2 4 1 4	3 1 2 3

## Задача F. Расстояние между вершинами — 2

Имя входного файла: `distance.in`  
Имя выходного файла: `distance.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Коль Дейкстру́ писать без кучи,  
То тайм-лимит ты получишь...  
А в совсем крутой задаче  
Юзай кучу Фибоначчи!

---

Спектакль преподавателей ЛКШ.июль-2007

Дан взвешенный граф. Требуется найти вес минимального пути между двумя вершинами.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно. Вторая строка входного файла содержит натуральные числа  $s$  и  $t$  — номера вершин, длину пути между которыми требуется найти ( $1 \leq s, t \leq n$ ,  $s \neq t$ ).

Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номера концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100$ ).

$n \leq 100\,000$ ,  $m \leq 200\,000$ .

### Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами  $s$  и  $t$ .

Если путь из  $s$  в  $t$  не существует, выведите -1.

### Пример

distance.in	distance.out
4 4 1 3 1 2 1 3 4 5 3 2 2 4 1 4	3

## Задача G. Кратчайший путь коня

Имя входного файла: `knight1.in`  
Имя выходного файла: `knight1.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

На шахматной доске размером  $8 \times 8$  заданы две клетки. Соедините эти клетки кратчайшим путем коня.

### Формат входных данных

Программа получает на вход координаты двух клеток, каждая в отдельной строке. Координаты клеток задаются в виде буквы (от “a” до “h”) и цифры (от 1 до 8) без пробелов.

### Формат выходных данных

Программа должна вывести путь коня, начинающийся и заканчивающийся в данных клетках и содержащий наименьшее число клеток.

### Пример

knight1.in	knight1.out
a1 h8	a1 b3 a5 b7 d8 f7 h8

## Задача Н. Кратчайший путь

Имя входного файла: `shortest.in`  
Имя выходного файла: `shortest.out`  
Ограничение по времени: 2 секунда  
Ограничение по памяти: 256 мегабайт

Надеюсь, все вы умеете искать в неориентированном графе кратчайший путь. В этой задаче вам предлагается свое умение продемонстрировать.

Вам дан неориентированный взвешенный граф. Веса ребер — целые числа от 1000 до 2000. Нужно несколько раз (не более 1000) ответить на следующий запрос: длина кратчайшего пути из некоторой вершины  $s$  в некоторую вершину  $t$ .

### Формат входных данных

На первой строке числа  $N$  и  $M$  ( $1 \leq N \leq 10^5$ ,  $0 \leq M \leq 2 \cdot 10^5$ ) — количество вершин и ребер нашего графа, соответственно. Вершины нумеруются целыми числами от 1 до  $N$ . Далее  $M$  строк содержат информацию о ребрах графа. Каждое ребро задается тремя числами — номер начала, номер конца и вес. Все веса — целые числа от 1000 до 2000. В графе могут быть и петли, и кратные ребра. Следующая строка содержит число  $K$  ( $1 \leq K \leq 10^3$ ) — количество запросов. В следующих  $K$  строках задаются запросы. Каждый запрос описывается двумя числами — из какой вершины, и в какую должен вести путь.

### Формат выходных данных

Для каждого запроса выведите на отдельной строке целое число — длину кратчайшего пути. Если кратчайшего пути не существует следует вывести  $-1$ .

### Пример

<code>shortest.in</code>	<code>shortest.out</code>
5 5	-1
1 2 2000	3000
1 3 1000	0
1 4 1200	2000
2 3 1500	
3 4 1500	
4	
1 5	
4 2	
3 3	
1 2	

### Замечание

Путем в графе называется такая последовательность ребер, что конец  $i$ -го совпадает с началом  $i + 1$ -го. Длиной пути называется суммарный вес ребер. Путь является кратчайшим, если его длина минимальна.



## Задача I. Цикл отрицательного веса

Имя входного файла: `negcycle.in`  
Имя выходного файла: `negcycle.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дан ориентированный граф. Определите, есть ли в нем цикл отрицательного веса, и если да, то выведите его.

### Формат входных данных

Во входном файле в первой строке число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа. В следующих  $N$  строках находится по  $N$  чисел — матрица смежности графа. Все веса ребер не превышают по модулю 10 000. Если ребра нет, то соответствующее число равно 100 000.

### Формат выходных данных

В первой строке выходного файла выведите «YES», если цикл существует или «NO» в противном случае. При его наличии выведите во второй строке количество вершин в искомом цикле и в третьей строке — вершины входящие в этот цикл в порядке обхода.

### Пример

<code>negcycle.in</code>	<code>negcycle.out</code>
2	YES
0 -1	2
-1 0	1 2