

### Лабораторная работа 2-2. Графы, кратчайшие пути

Statement is not available on English language

# А. Флойд

ограничение по времени на тест: 2 секунды ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод вывод: стандартный вывод

Полный ориентированный взвешенный граф задан матрицей смежности. Постройте матрицу кратчайших путей между его вершинами. Гарантируется, что в графе нет циклов отрицательного веса.

### Входные данные

В первой строке вводится единственное число N ( $1 \le N \le 100$ ) — количество вершин графа. В следующих N строках по N чисел задается матрица смежности графа (j-ое число в i-ой строке — вес ребра из вершины i в вершину j). Все числа по модулю не превышают 100. На главной диагонали матрицы — всегда нули.

#### Выходные данные

Выведите N строк по N чисел — матрицу расстояний между парами вершин, где j-ое число в i-ой строке равно весу кратчайшего пути из вершины i в j.

### Пример

Пример	
входные данные	Сору
4	
0 5 9 100	
100 0 2 8	
100 100 0 7	
4 100 100 0	
выходные данные	Сору
0 5 7 13	
12028	
11 16 0 7	
49110	

Statement is not available on English language

# В. Кратчайший путь-2

ограничение по времени на тест: 2 секунды ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод вывод: стандартный вывод

Дан неориентированный связный взвешенный граф. Найдите кратчайшее расстояние от первой вершины до всех вершин.

#### Входные данные

В первой строке входного файла два числа: n и m ( $2 \le n \le 30000$ ,  $1 \le m \le 400000$ ), где n — количество вершин графа, а m — количество ребер.

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — неотрицательное целое число, не превосходящее  $10^4$ .

#### Выходные данные

Выведите *п* чисел — для каждой вершины кратчашее расстояние до нее.

# Пример

входные данные	Сору
45	
121	
135	
248	
341	
45 121 135 248 341 233	

выходные данные Copy

0145

### Statement is not available on English language

# С. Цикл отрицательного веса

ограничение по времени на тест: 2 секунды ограничение по памяти на тест: 256 мегабайт

> ввод: стандартный ввод вывод: стандартный вывод

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

### Входные данные

Во входном файле в первой строке число N ( $1 \le N \le 100$ ) — 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.

## Выходные данные

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

входные данные	Сору
2 0-1 -10	
выходные данные	Сору
YES 2 2 1 1	

Statement is not available on English language

# Кратчайший путь длины K

ограничение по времени на тест: 4 секунды ограничение по памяти на тест: 256 мегабайт

> ввод: стандартный ввод вывод: стандартный вывод

Дан ориентированный граф. Найдите кратчайшие пути, состоящие из K рёбер, от S до всех вершин.

#### Входные данные

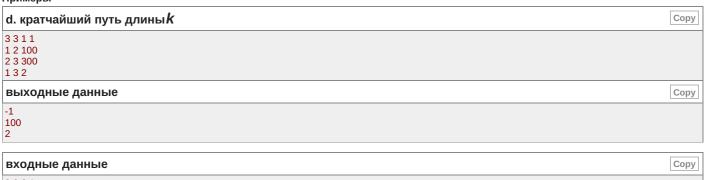
В первой строке дано целых четыре целых числа:  $1 \le N$ ,  $M \le 10^4$  — количества вершин и рёбер,  $0 \le K \le 100$  — количество рёбер в кратчайших путях,  $1 \le S \le N$  — начальная вершина.

В последующих M строках даны тройки целых чисел  $a_i, b_i, w$  — начало и конец ребра, а также его вес ( $1 \le a_i, b_i \le N$ ,  $-10^5 \le w \le 10^5$ ).

# Выходные данные

Выведите ровно N чисел по одному в строке. i-е число — длина минимального пути из ровно K рёбер из S в i, или  $\,$ -  $\,$ 1, если пути не существует.

#### Примеры



3321

1 2 100 2 3 300



Statement is not available on English language

# Е. Кратчайшие пути

ограничение по времени на тест: 2 секунды ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод вывод: стандартный вывод

Вам дан взвешенный ориентированный граф и вершина *S* в нём. Для каждой вершины графа *U* выведите длину кратчайшего пути от вершины *S* до вершины *U*.

### Входные данные

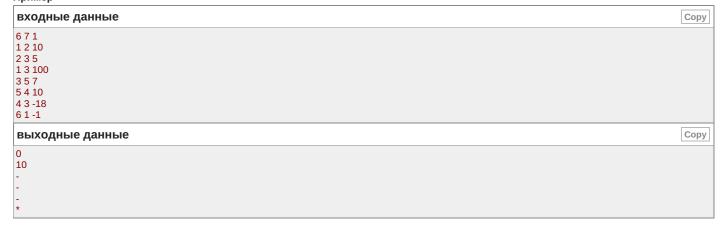
Первая строка входного файла содержит три целых числа n, m, s — количество вершин и ребёр в графе и номер начальной вершины соответственно ( $2 \le n \le 2\,000$ ,  $1 \le m \le 5\,000$ ).

Следующие m строчек описывают рёбра графа. Каждое ребро задаётся тремя числами— начальной вершиной, конечной вершиной и весом ребра соответственно. Вес ребра— целое число, не превосходящее  $10^{15}$  по абсолютной величине. В графе могут быть кратные рёбра и петли.

#### Выходные данные

Выведите n строчек — для каждой вершины u выведите длину кратчайшего пути из s в u. Если не существует пути между s и u, выведите s. Если не существует кратчайшего пути между s и u, выведите s.

#### Пример



Statement is not available on English language

## F. В поисках утраченного кефира

ограничение по времени на тест: 2 секунды ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод вывод: стандартный вывод

Школьник Вася хочет найти запасы спрятанного кефира. По легенде, кефир находится в домиках a, b или c. Вася хочет проверить каждый из этих трёх домиков, потратив на это минимальное количество времени.

Местность, в которой находится Вася представляет собой n домиков, пронумерованных числами от 1 до n. Некоторые из домиков соединены дорогами, по которым можно ходить в обе стороны. Время прохождения i-й дороги составляет  $W_i$  секунд. Путём в графе называется непустая последовательность вершин, такая что все соседние вершины соединены дорогой. Требуется помочь Васе найти путь, содержащий вершины a, b, c, такой что суммарное время прохождения всех дорог на пути минимально. При этом, если мы прошли по какойто дороге дважды (или более), то и время её прохождения следует учитывать соответствующее количество раз. Начинать свой путь Вася может из любой вершины.

Гарантируется, что *a*, *b*, *c* — попарно различные домики.

### Входные данные

В первой строке ввода записаны два числа n и m ( $3 \le n \le 100~000$ ,  $0 \le m \le 200~000$ ) — количество домиков в ЛКШ и дорог между ними соответственно.

Следующие m строк содержат описания дорог, по одному в строке. Каждая из дорог задаётся тройкой чисел  $u_i$ ,  $v_i$ ,  $w_i$  ( $1 \le u_i$ ,  $v_i \le n$ ,  $1 \le w_i \le 10^9$ ) — номерами соединённых домиков и временем, затрачиваемым на прохождение данной дороги. По каждой дороге разрешено ходить в обе стороны. Гарантируется, что любая пара домиков соединена не более чем одной дорогой. Также гарантируется, что нет дороги, соединяющей домик с самим собой.

В последней строке записаны три попарно различных числа  $a, b, c \ (1 \le a, b, c \le n)$ .

### Выходные данные

Выведите одно целое число — минимальное возможное время, которое нужно затратить на прохождение пути, содержащего домики *a*, *b* и *c*. Если пути, содержащего все три домика не существует, то выведите -1.

#### Примеры



#### Примечание

В первом примере путь 1–2–3–4 является минимальным (11 секунд). Например, путь 1–2–4–3 не подходит, так как занимает больше времени (20 секунд), а путь 3–4–2 не подходит, так как домик *а* оказывается не посещенным.

Во втором примере не существует способа добраться от домика b до домика c, поэтому искомого пути не существует.

# G. Long long trip

time limit per test: 2 seconds memory limit per test: 256 megabytes input: standard input output: standard output

There is a country with n cities. The cities are numbered 1 through n. There are some bidirectional roads in the country. Each road connects a pair of cities. More precisely, for each i, road i connects the cities  $a_i$  and  $b_i$ .

Limit is a deer that likes to travel along the roads. Traveling along road i (in either direction) takes him exactly  $d_i$  minutes. Limit does not like cities, so he never waits in a city.

Limit is currently in the city 1, starting his travels. In exactly t minutes, he wants to be in the city n. You are to find, whether Limit can reach city t minutes.

### Input

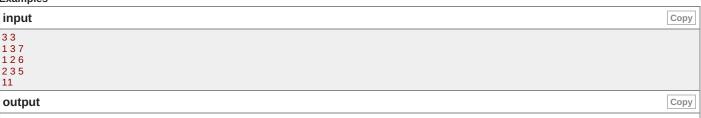
First line contains integers n and m — the number of cities and the number of roads in the country, respectively ( $1 \le n \le 50$ ). Next m lines describe the roads. Each line consists of integers  $a_i$ ,  $b_i$  and  $d_i$  — the endpoints of the road and its length ( $1 \le a_i$ ,  $b_i \le n$ ;  $1 \le d_i \le 10^4$ ).

The last line contains integer t — the number of minutes Limit wants to travel for  $(1 \le t \le 10^{18})$ .

#### Output

Output "Possible" if Limit can reach city n in exactly t minutes, output "Impossible" otherwise.

#### **Examples**



Possible	
input	Сору
3 3	
33 137	
126	
235	
25	
output	Сору
Possible	
input	Сору
21	
121	
9	
output	Сору
Possible	
input	Сору
21	
211	
10000000000000000	
output	Сору
Impossible	
input	Сору
43	
1 3 10	
1 2 10	
2 3 10	
1000	
output	Сору
Impossible	

# H. Dwarf Tower

time limit per test: 3 seconds memory limit per test: 256 megabytes

input: dwarf.in output: dwarf.out

Little Vasya is playing a new game named "Dwarf Tower". In this game there are n different items, which you can put on your dwarf character. Items are numbered from 1 to n. Vasya wants to get the item with number 1.

There are two ways to obtain an item:

- You can buy an item. The i-th item costs  $C_i$  money.
- You can craft an item. This game supports only *m* types of crafting. To craft an item, you give two particular different items and get another one as a result.

Help Vasya to spend the least amount of money to get the item number 1.

#### input

The first line of input contains two integers n and m ( $1 \le n \le 200\,000$ ;  $0 \le m \le 500\,000$ ) — the number of different items and the number of crafting types.

The second line contains n integers  $c_i$  — values of the items  $(0 \le c_i \le 10^9)$ .

The following m lines describe crafting types, each line contains three distinct integers  $a_i$ ,  $x_i$ ,  $y_i - a_i$  is the item that can be crafted from items  $x_i$  and  $y_i$  ( $1 \le a_i$ ,  $x_i$ ,  $y_i \le n$ ;  $a_i \ne x_i$ ;  $x_i \ne y_i$ ;  $y_i \ne a_i$ ).

# Output

The output should contain a single integer — the least amount of money to spend.

#### Example

LAMIIIPIC	
input	Сору
5 3 5 0 1 2 5 5 2 3 4 2 3 1 4 5	
1 4 5	

output	Сору
2	

<u>Codeforces</u> (c) Copyright 2010-2018 Mike Mirzayanov The only programming contests Web 2.0 platform

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js