

Лабораторная работа 2-2. Графы, кратчайшие пути

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 ($1 \leq N \leq 100$) — 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

входные данные	Скопировать
<pre>4 0 5 9 100 100 0 2 8 100 100 0 7 4 100 100 0</pre>	
выходные данные	Скопировать
<pre>0 5 7 13 12 0 2 8 11 16 0 7 4 9 11 0</pre>	

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 ($2 \leq n \leq 30000$, $1 \leq m \leq 400000$), 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 n чисел — для каждой вершины кратчайшее расстояние до нее.

Пример

входные данные	Скопировать
<pre>4 5 1 2 1 1 3 5 2 4 8 3 4 1 2 3 3</pre>	
выходные данные	Скопировать
<pre>0 1 4 5</pre>	

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 ($1 \leq N \leq 100$) — 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.

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	Скопировать
2 0 -1 -1 0	
выходные данные	Скопировать
YES 2 2 1	

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: $1 \leq N$, $M \leq 10^4$ — количества вершин и рёбер, $0 \leq K \leq 100$ — количество рёбер в кратчайших путях, $1 \leq S \leq N$ — начальная вершина.

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

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

d. кратчайший путь длинны K	Скопировать
3 3 1 1 1 2 100 2 3 300 1 3 2	
выходные данные	Скопировать
-1 100 2	

входные данные	Скопировать
3 3 2 1 1 2 100 2 3 300 1 3 2	
выходные данные	Скопировать
-1 -1 400	

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 ($2 \leq n \leq 2\,000, 1 \leq m \leq 5\,000$).

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 «*». Если не существует кратчайшего пути между S и U , выведите «-».

Пример

входные данные	Скопировать
6 7 1 1 2 10 2 3 5 1 3 100 3 5 7 5 4 10 4 3 -18 6 1 -1	
выходные данные	Скопировать
0 10 - - - *	

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 какой-то 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 ($3 \leq n \leq 100\,000, 0 \leq m \leq 200\,000$) — 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 ($1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^9$) — 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 ($1 \leq a, b, c \leq n$).

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

входные данные	Скопировать
4 4 1 2 3 2 3 1 3 4 7 4 2 10 1 4 3	

выходные данные	Скопировать
11	

входные данные	Скопировать
4 2 1 2 10 2 3 5 1 2 4	

выходные данные	Скопировать
-1	

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 *a* оказывается не посещенным.

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

Г. Бемби

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Существует страна, в которой *n* городов. Города пронумерованы от 1 до *n*. Также в этой стране существуют двунаправленные дороги. Каждая дорога соединяет пару городов. Для каждого *i*, автомобильная дорога *i* соединяет города *a_i* и *b_i*.

Бемби — это олень, который любит путешествовать по дорогам. Движение по дороге *i* (в любом направлении) занимает у оленя *d_i* минут. Бемби ненавидит города и из-за этого никогда в них не задерживается.

Бемби начинает путешествие из города номер 1. Через *t* минут он желает оказаться в городе *n*. Вы должны узнать, может ли Бемби достигнуть город *n* ровно через *t* минут.

Входные данные
Первая строка содержит два целых числа *n* и *m* — количество городов и дорог в стране ($1 \leq n, m \leq 50$).

Следующие *m* строк описывают дороги. Каждая строка состоит из чисел *a_i*, *b_i* и *d_i* — концы дороги и ее длина ($1 \leq a_i, b_i \leq n; 1 \leq d_i \leq 10^4$).

Последняя строка содержит целое число *t* — количество минут, за которое Бемби желает добраться до города *n* ($1 \leq t \leq 10^{18}$).

Выходные данные
Выведите "Possible" если Бемби сможет достичь цели ровно за *t* минут, иначе выведите "Impossible".

Примеры

входные данные	Скопировать
3 3 1 3 7 1 2 6 2 3 5 11	

выходные данные	Скопировать
Possible	

входные данные	Скопировать
3 3 1 3 7 1 2 6 2 3 5 25	

выходные данные	Скопировать
Possible	

входные данные	Скопировать
2 1 1 2 1 9	

выходные данные	Скопировать
Possible	

входные данные	Скопировать

2 1 2 1 1 1000000000000000000	
выходные данные	Скопировать
Impossible	

входные данные	Скопировать
4 3 1 3 10 1 2 10 2 3 10 1000	
выходные данные	Скопировать
Impossible	

Условие недоступно на русском языке

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 ($1 \leq n \leq 200\,000; 0 \leq m \leq 500\,000$) — the number of different items and the number of crafting types.

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

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 ($1 \leq a_i, x_i, y_i \leq n$; $a_i \neq x_i$; $x_i \neq y_i$; $y_i \neq a_i$).

Output

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

Example

input	Скопировать
5 3 5 0 1 2 5 5 2 3 4 2 3 1 4 5	
output	Скопировать
2	