

Задача А. Кратчайшие пути

Имя входного файла: path.in
Имя выходного файла: path.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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 .

Формат входных данных

Первая строка входного файла содержит три целых числа n , m , s — количество вершин и рёбер в графе и номер начальной вершины соответственно ($2 \leq n \leq 2000$, $1 \leq m \leq 5000$).

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.

Формат выходных данных

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 «-».

Примеры

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 В. 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.

Примеры

negcycle.in	negcycle.out
2	YES
0 -1	2
-1 0	2 1

Задача С. Флойд

Имя входного файла: floyd.in
Имя выходного файла: floyd.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Полный ориентированный взвешенный граф задан матрицей смежности. Постройте матрицу кратчайших путей между его вершинами.

Гарантируется, что в графе нет циклов отрицательного веса.

Формат входных данных

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.

Формат выходных данных

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 .

Примеры

floyd.in	floyd.out
4	0 5 7 13
0 5 9 100	12 0 2 8
100 0 2 8	11 16 0 7
100 100 0 7	4 9 11 0
4 100 100 0	

Задача D. Рейсы во времени

Имя входного файла: `time.in`
Имя выходного файла: `time.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Между N населёнными пунктами совершаются пассажирские рейсы на машинах времени.

В момент времени 0 вы находитесь в пункте A . Вам дано расписание рейсов. Требуется оказаться в пункте B как можно раньше (то есть в наименьший возможный момент времени).

При этом разрешается делать пересадки с одного рейса на другой. Если вы прибываете в некоторый пункт в момент времени T , то вы можете уехать из него любым рейсом, который отправляется из этого пункта в момент времени T или позднее (но не раньше).

Формат входных данных

Первая строка входного файла содержит число N — количество населённых пунктов ($1 \leq N \leq 1000$). Вторая строка содержит два числа A и B — номера начального и конечного пунктов. Третья строка содержит число K — количество рейсов ($0 \leq K \leq 1000$). Следующие K строк содержат описания рейсов, по одному на строке. Каждое описание представляет собой четвёрку целых чисел. Первое число каждой четвёрки задаёт номер пункта отправления, второе — время отправления, третье — пункт назначения, четвёртое — время прибытия. Номера пунктов — натуральные числа из диапазона от 1 до N . Пункт назначения и пункт отправления могут совпадать. Время измеряется в некоторых абсолютных единицах и задаётся целым числом, по модулю не превышающим 10^9 . Поскольку рейсы совершаются на машинах времени, то время прибытия может быть как больше времени отправления, так и меньше, или равным ему.

Гарантируется, что входные данные таковы, что добраться из пункта A в пункт B всегда можно.

Формат выходных данных

Выведите в выходной файл минимальное время, когда вы сможете оказаться в пункте B .

Примеры

<code>time.in</code>	<code>time.out</code>
2 1 1 2 1 1 2 10 1 10 1 9	0
1 1 1 3 1 3 1 -5 1 -5 1 -3 1 -4 1 -10	-10

Задача E. Pink Floyd

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

Группа Pink Floyd собирается отправиться в новый концертный тур по всему миру. По предыдущему опыту группа знает, что солист Роджер Уотерс постоянно нервничает при перелетах. На некоторых маршрутах он теряет вес от волнения, а на других — много ест и набирает вес.

Известно, что чем больше весит Роджер, тем лучше выступает группа, поэтому требуется спланировать перелеты так, чтобы вес Роджера на каждом концерте был максимально возможным.

Группа должна посещать города в том же порядке, в котором она дает концерты. При этом между концертами группа может посещать промежуточные города.

Формат входных данных

Первая строка входного файла содержит три натуральных числа n , m и k — количество городов в мире, количество рейсов и количество концертов, которые должна дать группа соответственно ($n \leq 100$, $m \leq 10\,000$, $2 \leq k \leq 10\,000$). Города пронумерованы числами от 1 до n .

Следующие m строк содержат описание рейсов, по одному на строке. Рейс номер i описывается тремя числами b_i , e_i и w_i — номер начального и конечного города рейса и предполагаемое изменение веса Роджера в миллиграммах ($1 \leq b_i, e_i \leq n$, $-100\,000 \leq w_i \leq 100\,000$).

Последняя строка содержит числа a_1, a_2, \dots, a_k — номера городов, в которых проводятся концерты ($a_i \neq a_{i+1}$). В начале концертного тура группа находится в городе a_1 .

Гарантируется, что группа может дать все концерты.

Формат выходных данных

Первая строка выходного файла должна содержать число l — количество рейсов, кото-

рые должна сделать группа. Вторая строка должна содержать l чисел — номера используемых рейсов.

Если существует такая последовательность маршрутов между концертами, что Роджер будет набирать вес неограниченно, то первая строка выходного файла должна содержать строку “infinitely kind”.

Примеры

floyd.in	floyd.out
4 8 5	6
1 2 -2	5 6 5 7 2 3
2 3 3	
3 4 -5	
4 1 3	
1 3 2	
3 1 -2	
3 2 -3	
2 4 -10	
1 3 1 2 4	