

у2018-3-2. Кратчайшие пути. Игры

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 ($1 \leq N \leq 100$) — 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 8 5 9 100 100 0 2 8 1 3 5 4 100 100 0</pre>	
выходные данные	Скопировать
<pre>0 5 7 13 12 0 2 8 11 16 0 7 4 5 11 0</pre>	

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 ($2 \leq n \leq 30000$, $1 \leq m \leq 400000$), 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>	

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 ($1 \leq N \leq 100$) — 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
<pre>2 0 -1 -1 0</pre>	
выходные данные	Скопировать
<pre>YES 2 1</pre>	

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 ребер, от S до всех вершин.

Входные данные
В первой строке дано целых четыре числа: $1 \leq N$, $M \leq 10^4$ — количества вершин и ребер, $0 \leq K \leq 100$ — количество ребер 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 ребер из S в i , или -1 , если пути не существует.

Примеры	
д. кратчайший путь длины k	Скопировать
<pre>3 3 1 1 1 2 100 2 3 300 1 3 2</pre>	
выходные данные	Скопировать
<pre>-1 100 2</pre>	
входные данные	Скопировать
<pre>3 3 2 1 1 2 100 2 3 300 1 3 2</pre>	
выходные данные	Скопировать
<pre>-1 -1 400</pre>	

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

Следующие m строк описывают ребра графа. Каждое ребро задается тремя числами — начальной вершиной, конечной вершиной и весом ребра соответственно. Вес ребра — целое число, не превосходящее 10^{15} по абсолютной величине. 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.

Выходные данные
Выведите n строк — 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
<pre>6 7 1 1 2 10 2 3 5 1 3 100 3 5 7 5 4 10 4 3 -10 6 1 -1</pre>	
выходные данные	Скопировать
<pre>0 10 - - - *</pre>	

Statement is not available on English language

В. 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 задается тройкой чисел u_i , v_i , w_i ($1 \leq u_i$, $v_i \leq n$, $1 \leq w_i \leq 10^3$) — 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 домика a , b и c . Если пути, содержащего все три домика не существует, то выведите -1 .

Примеры	
входные данные	Скопировать
<pre>4 4 1 2 3 2 3 1 3 4 7 4 2 10 1 4 3</pre>	
выходные данные	Скопировать
<pre>11</pre>	
входные данные	Скопировать
<pre>4 2 1 2 10 2 3 5 1 2 4</pre>	
выходные данные	Скопировать
<pre>-1</pre>	

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 посещённым.

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 n in exactly t minutes.

Input
First line contains integers n and m — the number of cities and the number of roads in the country, respectively ($1 \leq n \leq 50$). 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 ($1 \leq a_i$, $b_i \leq n$; $1 \leq d_i \leq 10^8$).

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

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

Examples	
input	Скопировать
<pre>3 3 1 3 7 1 2 6 2 3 5 11</pre>	
output	Скопировать
<pre>Possible</pre>	
input	Скопировать
<pre>3 3 1 3 7 1 2 6 2 3 5 25</pre>	
output	Скопировать
<pre>Possible</pre>	
input	Скопировать
<pre>2 1 1 2 1 9</pre>	
output	Скопировать
<pre>Possible</pre>	
input	Скопировать
<pre>2 1 2 1 1 1000000000000000000</pre>	
output	Скопировать
<pre>Impossible</pre>	
input	Скопировать
<pre>4 3 1 3 10 1 2 10 2 3 10 1000</pre>	
output	Скопировать
<pre>Impossible</pre>	

H. Game

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: game.in
output: game.out

You are given directed acyclic graph. One of the graph vertices contains a token. Two player play the game. Player alternate turns. On each turn if the token is located in vertex u and graph contains edge uv , then current player can move token from vertex u to vertex v . The player which can't make move loses.

Determine, who wins, if both players play optimally.

Input
First line contains three integers n , m and s ($1 \leq n$, s , $m \leq 10^5$) — number of vertices, edges and the vertex, where token is located initially. The next m lines contain graph edges. Each edge is described by two integers v and u — starting and ending vertices ($1 \leq v$, $u \leq n$).

The given graph doesn't contain cycles.

Output
Output First player wins if the first player wins, otherwise, print Second player wins.

Examples	
input	Скопировать
<pre>3 3 1 1 2 2 3 1 3</pre>	
output	Скопировать
<pre>First player wins</pre>	
input	Скопировать
<pre>3 2 1 1 2 2 3</pre>	
output	Скопировать
<pre>Second player wins</pre>	

I. Retrograde analysis

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

You are given directed graph of n vertices and m edges. Alice and Bob play a game. Initially vertex i contains a token. On each turn a player can move the token using on of the outgoing edges. The player, which can't make move, loses.

For every vertex i determine, who wins if both players play optimally.

Input
Input consists of one or more testcases. Every testcase contains description of directed graph. Graph is described as follows.

First line contains two integers n ($1 \leq n \leq 300\,000$) and m ($1 \leq m \leq 300\,000$). Next m lines contain edges of the graph, each one is described by two integers from 1 to n — starting and ending vertices of an edge. Graph can contain loops and multiple edges.

The sum of n over all testcases in an input doesn't exceed 300 000.

The sum of m over all testcases in an input doesn't exceed 300 000.

Output
For each testcase output n lines. For each vertex i output FIRST, SECOND or DRAW, depending on who wins, if both players play optimally starting the game from vertex i .

Output an empty line after each testcase.

Example	
input	Скопировать
<pre>5 5 2 3 2 3 3 1 1 4 4 5 2 1 1 2 4 4 1 5 1 3 2 3 3 1 1 4</pre>	
output	Скопировать
<pre>DRAW DRAW DRAW FIRST SECOND FIRST SECOND FIRST FIRST SECOND SECOND</pre>	

Statement is not available on English language

J. Функция Гранди

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

Дан ориентированный ациклический граф. Посчитайте функцию Гранди для каждой стартовой вершины.

Входные данные
На первой строке вводятся два числа n и m — количество вершин и ребёр в графе ($1 \leq n$, $m \leq 100\,000$). На следующих m строках содержится по два числа x и y ($1 \leq x$, $y \leq n$).

Учите, что в графе могут быть кратные ребра.

Выходные данные
Выведите n чисел — значение функции Гранди для каждой стартовой вершины.

Примеры	
входные данные	Скопировать
<pre>3 3 1 2 1 3 1 3</pre>	
выходные данные	Скопировать
<pre>2 0 0</pre>	
входные данные	Скопировать
<pre>2 1 2 1</pre>	
выходные данные	Скопировать
<pre>0 1</pre>	

Statement is not available on English language

К. Дровосек

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

Двое играют в следующую игру: имеется дерево с отмененной вершиной (корнем). Игроки ходят по очереди. За ход игрок рубит ветку (свое ребро), причем из двух получившихся компонент связности остается только та, которая содержит корень — остальная отбрасывается и больше в игре не участвует. Проигрывает тот, кто не может сделать ход. Определите, может ли выиграть первый игрок, и если да, то укажите любой из его выигрышных ходов.

Входные данные
В первой строке входного файла находятся 2 числа, N и R — количество вершин дерева и номер корня ($1 < N \leq 100\,000$, $1 \leq R \leq N$). Далее следуют $N - 1$ строка, в каждой из которых находятся два числа — номера вершин, которые соединят очередное ребро.

Выходные данные
Выведите в выходной файл одно число 1 или 2 — номер игрока, который выигрывает при правильной игре. Если выигрывает первый игрок, то выведите также любой его выигрышный ход, т. е. порядковый номер ребра во входном файле, которое ему достаточно рубить первый ходом (число от 1 до $N - 1$).

Примеры	
входные данные	Скопировать
<pre>1 1</pre>	
выходные данные	Скопировать
<pre>2</pre>	
входные данные	Скопировать
<pre>2 2 1 2</pre>	
выходные данные	Скопировать
<pre>1 1</pre>	