

у2020-3-2. Кратчайшие пути. Игры

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

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 связанный 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	
<div>входные данные</div> <div>4 5 1 2 1 1 3 5 2 4 1 3 4 1 2 3 3</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>0 1 4 5</div>	<div>Скопировать</div>

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	
<div>входные данные</div> <div>2 0 -1 -1 0</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>YES 2 2 1</div>	<div>Скопировать</div>

D. 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 целых 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_i — начало и конец ребра, а также его вес ($1 \leq a_i, b_i \leq N, -10^3 \leq w_i \leq 10^3$).

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	
<div>входные данные</div> <div>3 3 1 1 1 2 100 2 3 100 1 3 2</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>-1 100 2</div>	<div>Скопировать</div>
<div>входные данные</div> <div>3 3 2 1 1 2 100 2 3 100 1 3 2</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>-1 -1 400</div>	<div>Скопировать</div>

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 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 ребра и петли.

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	
<div>входные данные</div> <div>6 7 1 1 2 10 2 3 5 1 3 100 3 5 7 5 4 10 4 3 -10 6 1 -1</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>0 10 - - - *</div>	<div>Скопировать</div>

Ф. 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	
<div>входные данные</div> <div>4 4 1 2 3 2 3 1 3 4 7 4 2 10 1 4 3</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>11</div>	<div>Скопировать</div>
<div>входные данные</div> <div>4 2 1 2 10 2 3 5 1 2 4</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>-1</div>	<div>Скопировать</div>

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.

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 её 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 ($1 \leq t \leq 10^8$).

Выходные данные

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

Примеры	
<div>входные данные</div> <div>3 3 1 3 7 1 2 6 2 3 5 11</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>Possible</div>	<div>Скопировать</div>
<div>входные данные</div> <div>3 3 1 3 7 1 2 6 2 3 5 25</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>Possible</div>	<div>Скопировать</div>
<div>входные данные</div> <div>2 1 1 2 1 9</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>Possible</div>	<div>Скопировать</div>
<div>входные данные</div> <div>2 1 2 1 1 1000000000000000000</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>Impossible</div>	<div>Скопировать</div>
<div>входные данные</div> <div>4 3 1 3 10 1 2 10 2 3 10 1000</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>Impossible</div>	<div>Скопировать</div>

Н. Игра

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

Дан ориентированный невязвешенный ациклический граф. На одной из вершин графа стоит «фишка». Двое играют в игру. Пусть «фишка» находится в вершине u , и в графе есть ребро (u, v) . Тогда за ход разрешается перевести «фишку» из вершины u в вершину v . Проигрывает тот, кто не может сделать ход.

Входные данные

В первой строке входного файла находятся три натуральных числа N, M и S ($1 \leq N, S, M \leq 100\,000$) — количество вершин графа и вершина, в которой находится «фишка» в начале игры соответственно. Далее в M строках перечислены ребра графа. Каждое ребро задается парой чисел — номерами начальной и конечной вершин.

Выходные данные

Если выигрывает игрок, который ходит первым, выведите «First player wins», иначе — «Second player wins».

Примеры	
<div>входные данные</div> <div>3 3 1 1 2 2 3 1 3</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>First player wins</div>	<div>Скопировать</div>
<div>входные данные</div> <div>3 2 1 1 2 2 3</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>Second player wins</div>	<div>Скопировать</div>

I. Ретроанализ для маленьких

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

Дан ориентированный невязвешенный ациклический граф из n вершин и m ребер. Оля и Коля в игру. Изначально фишка стоит в вершине i . За ход можно передвинуть фишку по любому из исходящих ребер. Тот, кто не может сделать ход, проигрывает. Ваша задача — для каждой вершины i определить, кто выиграет при оптимальной игре обоих.

Входные данные

Входные данные состоят из одного или нескольких тестов. Каждый тест содержит описание ориентированного графа. Граф описывается так: на первой строке записаны два целых числа n ($1 \leq n \leq 300\,000$) и m ($1 \leq m \leq 300\,000$). Следующие m строк содержат ребра графа, каждое описывается парой целых чисел от 1 до n . Пара $a\ b$ обозначает, что ребро ведет из вершины a в вершину b . В графе могут быть петли и кратные ребра.

Сумма n по всем тестам не превосходит 300 000, сумма m по всем тестам также не превосходит 300 000.

Выходные данные

Для каждого теста выведите для каждой вершины FIRST, SECOND или DRAW в зависимости от того, кто выиграет при оптимальной игре из этой вершины. Ответы к тестам разделяйте пустой строкой.

Пример	
<div>входные данные</div> <div>5 5 1 2 2 3 3 1 1 4 4 5 2 1 1 2 4 4 1 2 2 3 3 1 1 4</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>DRAW DRAW FIRST SECOND FIRST SECOND FIRST SECOND SECOND</div>	<div>Скопировать</div>

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 будут даны числа n и m — количество вершин и ребер в графе ($1 \leq n, m \leq 100\,000$). На следующих m строках содержится по два числа x и y ($1 \leq x, y \leq n$).

Учтите, что в графе могут быть кратные ребра.

Выходные данные

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

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

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 отмеченной вершиной (корнем). Игроки ходят по очереди. За ход игрок рубит ветку (старает ребро), причем из двух получившихся компонент связности остается только та, которая содержит корень — оставшаяся становится 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 соединит 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 разрубить первым ходом (число от 1 до $N - 1$).

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