y2019-3-2. Кратчайшие пути. Игры

A. Флоид

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 входного файла целые числа N ($1 \le N \le 100$) — количество вершин графа. В следующих N строках по N чисел задается матрица смежности графа (-1 на месте в i-ой строке i-ого 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
4
0 5 9 100
100 0 3 9
100 100 0 7
4 100 100 0
выходные данные
0 5 7 12
12 0 2 7
11 6 0 7
4 9 11 0
```

B. 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 кратчайшие 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 10^7.

Выходные данные

Выведите n чисел — для каждой вершины кратчайшие расстояние до нее.

Пример

```
входные данные
4 4
1 2 1
2 3 1
3 4 1
1 4 5
выходные данные
0 1 2 3
```

C. 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 10000. Если ребра нет, то соответствующее число равно 100000.

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
2
0 -1
-1 0
выходные данные
YES
2
1 2
```

D. Кратчайший путь длины K

ограничение по времени на тест: 2 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 3-ди всех вершин.

Входные данные

В первой строке даны целые четыре целых числа: $2 \le N, M \le 10^5$ — количество вершин и ребер, $0 \le K \le 100$ — количество ребер в кратчайших путях, $1 \le S \le N$ — начальная вершина.

В последующие M строках даны тройки целых чисел a, b, w — начало и конец ребра, а также его вес ($1 \le a, b \le N$, $-10^7 \le w \le 10^7$).

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

```
входные данные
3 3 1 1
1 2 100
2 3 100
1 3 1
выходные данные
101
-1
-1
```

```
входные данные
3 3 2 1
1 2 100
2 3 100
1 3 1
выходные данные
-1
-1
400
```

E. 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 t в нем. Для каждой вершины графа в выведите длину кратчайшего пути от вершины 1 до вершины i.

Входные данные

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

Следующие m строчек описывают ребра графа. Каждое ребро задается тремя числами — начальной вершиной, конечной вершиной и весом ребра соответственно. Вес ребра — целое число, не превосходящее 10^{12} по абсолютной величине. В графе могут быть кратные ребра и петли.

Выходные данные

Выведите n строчек — для каждой вершины в выведите длину кратчайшего пути из t в i. Если не существует пути между t и i, выведите «?». Если не существует кратчайшего пути между t и i, выведите «-».

Пример

```
входные данные
5 5 1
1 2 3
2 3 5
3 4 -100
4 5 10
5 4 10
1 3 -1
выходные данные
0
3
-1
-
-
```

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 утраченного кефира. По ленинграду, кефир находится в домике 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 k домиков, пронумерованных числами от 1 до k. Некоторые из домиков соединены дорогами, каждая дорога соединяет два различных домика и имеет длину, выражающуюся целым числом. Путь из домика x в домик y называется кратчайшим, если среди всех соединяющих вершины x и y путей, он имеет минимальную суммарную длину. Требуется помочь Васе найти путь, содержащий вершины a, b c, такой что суммарное время прохождения всех дорог из пути минимально. При этом, если мы прошли по какой-то дорогу дважды или более, то к время на 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 Пёгле и дорог между ними соответственно.

Следующие m строк содержат описание дорог, по порядку в строке. Каждая из дорог задаётся тройкой чисел a_i, b_i, w_i ($1 \le a_i, b_i \le n$, $1 \le w_i \le 10^9$) — 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 дорог, соединяющей домик с самим собой.

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

Выходные данные

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

Примеры

```
входные данные
4 4
1 2 1
2 3 1
3 4 1
1 4 1
1 2 4
выходные данные
11
```

```
входные данные
4 2
1 2 10
3 4 10
2 3 1
1 2 4
выходные данные
-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 3 до домика c, поэтому искомого пути не существует.

G. 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 4 города. Города пронумерованы от 1 до 4. Также в этой стране существуют двухнаправленные 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 ровно d_i минут. Бемби начинает с города 1 и на это никогда в нем не задерживается.

Бемби начинает путешествие из города номер 1. Через 7 минут он хотелось оказаться в городе 4. Вы должны узнать, может ли Бемби достичь города 4 ровно через 7 минут.

Входные данные

Первая строка содержит два целых числа n и m — количество городов и дорог в стране ($2 \le n, m \le 50$).

Следующие m строк описывают дороги. Каждая строка состоит из четырех a_i, b_i и d_i — номер дороги и ее длина ($1 \le a_i, b_i \le n$, $1 \le d_i \le 10^9$).

Последняя строка содержит целое число t — количество минут, за которое Бемби желает добраться до города 4 ($0 \le t \le 10^{18}$).

Выходные данные

Выведите "Possible" если Бемби сможет достичь цели ровно за 7 минут, иначе выведите "Impossible".

Примеры

```
входные данные
3 3
1 2 3
2 3 4
1 3 7
7
выходные данные
Possible
```

```
входные данные
3 3
1 2 3
2 3 2
1 3 5
7
выходные данные
Possible
```

```
входные данные
2 1
1 2 1
7
выходные данные
Possible
```

```
входные данные
2 1
1 2 1
1000000000000000000
выходные данные
Impossible
```

```
входные данные
3 2
1 2 10
2 3 10
1000
выходные данные
Impossible
```

H. Игра

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

Дан ориентированный неизвешенный ациклический граф. На первой 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 переместить «фишку» из вершины u в вершину v. Проигрывает тот, кто не может сделать ход.

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 ($2 \le n \le 300\,000$) и m ($0 \le m \le 300\,000$). Следующие m строк содержат ребра графа, каждое описывается парой целых чисел от 1 до n. Пары (v, u) обозначают, что ребро ведет из вершины v в вершину u. 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

```
входные данные
5 5
1 2
2 3
3 4
4 5
4 3
4 3
1 2
2 3
3 1
3 2
2 1
выходные данные
DRAW
DRAW
FIRST
FIRST
SECOND

FIRST
FIRST
SECOND
SECOND
```

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 ($1 \le n$, $m \le 100\,000$). На следующих m строках содержится по два числа a и b ($1 \le a, b \le n$).

Уточнив, что в графе могут быть кратные ребра.

Выходные данные

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

Примеры

```
входные данные
3 3
1 2
1 3
2 3
выходные данные
1
0
0
```

```
входные данные
2 1
1 2
выходные данные
1
0
```

K. 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 удаляет ветку (открыт ребро), причем из двух получившихся компонент связности остается только та, которая содержит корень — остальные отключаются и больше в игре он участвует. Проигрывает тот, кто не может сделать ход. Определите, может ли выиграть первый игрок, и если да, то указать любой из его выигрышных ходов.

Входные данные

В первой строке входного файла находится T числа, N и R — количество вершин дерева и номер корня ($1 < N \le 100\,000$, $1 \le R \le N$). Далее следует $N - 1$ строка, в каждой из которых находится два числа — номера вершин, которые соединяет очередное ребро.

Выходные данные

Выведите в выходной файл одно число 1 или 2 — номер игрока, который выигрывает при оптимальной игре. Если выигрывает первый игрок, то выведите после любой его выигрышный ход, т.е. порядковый номер ребра из входного файла, которое мы достаточно разрубить, первым ходом (числа от 1 до $N - 1$).

Примеры

```
входные данные
1 3 1
1 2
1 3
выходные данные
1
```

```
входные данные
1 3 1
1 2
2 3
выходные данные
2
```

Codeforces (c) Copyright 2010-2021 Mikhaev Mikhaevich
Соревнования по программированию 2.0