y2018-3-2. Кратчайшие пути. Игры

A. Флойд

2 секунды, 256 мегабайт

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 $(1 \le N \le 100)$ — 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.

входные данные

```
4
0 5 9 100
100 0 2 8
100 100 0 7
4 100 100 0
```

выходные данные

```
0 5 7 13
12 0 2 8
11 16 0 7
4 9 11 0
```

B. Кратчайший путь – 2

2 секунды, 256 мегабайт

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 ($2 \le n \le 30000, 1 \le m \le 400000$), 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 10^4.

Выходные данные

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

входные данные

```
4 5
1 2 1
1 3 5
2 4 8
3 4 1
2 3 3
```

выходные данные

```
0 1 4 5
```

C. Цикл отрицательного веса

2 секунды, 256 мегабайт

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 $10\,000$. 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.

входные данные

```
2
0 -1
-1 0
```

выходные данные

```
YES
2
2 1
```

D. Кратчайший путь длины K

4 секунды, 256 мегабайт

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

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

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.

входные данные

```
3 3 1 1
1 2 100
2 3 300
1 3 2
```

выходные данные

```
-1
100
2
```

входные данные

```
3 3 2 1
1 2 100
2 3 300
1 3 2
```

выходные данные

```
-1
-1
400
```

E. Кратчайшие пути

2 секунды, 256 мегабайт

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

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 «$-$».

входные данные

```
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

```
0
10
-
-
-
*
```

F. В поисках утраченного кефира

2 секунды, 256 мегабайт

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 ($3 \le n \le 100\,000$, $0 \le m \le 200\,000$) — 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 ($1 \le u_i, v_i \le n$, $1 \le w_i \le 10^9$) — 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 дороги, соединяющей домик с самим собой.

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

Выходные данные

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

входные данные

```
4 4
1 2 3
2 3 1
3 4 7
4 2 10
1 4 3
```

выходные данные

```
11
```

входные данные

```
4 2
1 2 10
2 3 5
1 2 4
```

выходные данные

```
-1
```

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.

G. Бемби

2 секунды, 256 мегабайт

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 ($1 \le n, m \le 50$).

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

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

Выходные данные

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

входные данные

```
3 3
1 3 7
1 2 6
2 3 5
11
```

выходные данные

```
Possible
```

входные данные

```
3 3
1 3 7
1 2 6
2 3 5
25
```

выходные данные

```
Possible
```

входные данные

```
2 1
1 2 1
9
```

выходные данные

```
Possible
```

входные данные

```
2 1
2 1 1
1000000000000000000
```

выходные данные

```
Impossible
```

входные данные

```
4 3
1 3 10
1 2 10
2 3 10
1000
```

выходные данные

```
Impossible
```

H. Игра

2 секунды, 256 мегабайт

Дан ориентированный невзвешенный 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 $(1 \le N, S, M \le 100\,000)$ — количество вершин рёбер и вершина, в которой находится «фишка» в начале игры соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин.

Выходные данные

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

входные данные

```
3 3 1
1 2
2 3
1 3
```

выходные данные

```
First player wins
```

входные данные

```
3 2 1
1 2
2 3
```

выходные данные

```
Second player wins
```

I. Ретроанализ для маленьких

2 секунды, 256 мегабайт

Дан ориентированный весёлый 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 $(1 \le n \le 300\,000)$ и m $(1 \le m \le 300\,000)$. 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.

входные данные

```
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
```

выходные данные

```
DRAW
DRAW
DRAW
FIRST
SECOND

FIRST
SECOND

FIRST
FIRST
SECOND
SECOND
```

J. Функция Гранди

2 секунды, 256 мегабайт

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 рёбер в графе $(1 \le n, m \le 100\,000)$. На следующих m строках содержится по два числа x и y $(1 \le x, y \le n)$.

Учтите, что в графе могут быть кратные рёбра.

Выходные данные

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

входные данные

```
3 3
1 2
2 3
1 3
```

выходные данные

```
2
1
0
```

входные данные

```
2 1
2 1
```

выходные данные

```
0
1
```

K. Дровосек

2 секунды, 256 мегабайт

Двое играют в следующую игру: имеется дерево с отмеченной вершиной (корнем). Игроки ходят по очереди. За ход игрок рузрубает ветку (стирает ребро), причем из двух получившихся компонент связности остается только та, которая содержит корень — остальная отваливается 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 соединяет 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 $N - 1$).

входные данные

```
1 1
```

выходные данные

```
2
```

входные данные

```
2 2
1 2
```

выходные данные

```
1
1
```