

Задача Magic. Цветные волшебники

Имя входного файла: `magic.in`
 Имя выходного файла: `magic.out`
 Ограничение по времени: 10 секунд
 Ограничение по памяти: 64 мегабайта

Сказочная страна представляет собой множество городов, соединенных дорогами с двухсторонним движением. Причем из любого города страны можно добраться в любой другой город либо непосредственно, либо через другие города. Известно, что в сказочной стране не существует дорог, соединяющих город сам с собой и между любыми двумя разными городами, существует не более одной дороги.

В сказочной стране живут желтый и синий волшебники. Желтый волшебник, пройдя по дороге, перекрашивает ее в желтый цвет, синий — в синий. Как известно, при наложении желтой краски на синюю, либо синей краски на желтую, краски смешиваются и превращаются в краску зеленого цвета, который является самым нелюбимым цветом обоих волшебников.

В этом году в столице страны (городе f) проводится конференция волшебников. Поэтому желтый и синий волшебники хотят узнать, какое минимальное количество дорог им придется перекрасить в зеленый цвет, чтобы добраться в столицу. Изначально все дороги не покрашены.

Начальное положение желтого и синего волшебников заранее не известно. Поэтому необходимо решить данную задачу для k возможных случаев их начальных расположений.

Формат входного файла

Первая строка входного файла содержит целые числа: n ($1 \leq n \leq 100\,000$) и m ($1 \leq m \leq 500\,000$) — количество городов и дорог в волшебной стране соответственно. Третья строка содержит одно целое число f ($1 \leq f \leq n$) — номер города, являющегося столицей сказочной страны. В следующих m строках, находится описание дорог страны. В этих m строк записано по два целых числа a_i и b_i , означающих, что существует дорога, соединяющая города a_i и b_i . Следующая строка содержит целое число k ($1 \leq k \leq 100\,000$) — количество возможных начальных расположений волшебников. Далее следуют k строк, каждая из которых содержит два целых числа — номера городов, в которых изначально находится желтый и синий волшебники соответственно.

Формат выходного файла

Для каждого из k случаев, ваша программа должна вывести в выходной минимальное количество дорог, которое придется покрасить в зеленый цвет волшебникам для того, чтобы добраться в столицу.

Пример

<code>magic.in</code>	<code>magic.out</code>
6 6	1
1	2
1 2	
2 3	
3 4	
4 2	
4 5	
3 6	
2	
5 6	
6 6	

Задача Magic-Easy. Цветные волшебники (простая)

Задача «Цветные волшебники» с ограничением 100 на количество дорог и 500 на количество городов.

Задача MoveToFront. В начало строя!

Имя входного файла: `movetofront.in`
 Имя выходного файла: `movetofront.out`
 Ограничение по времени: 4 секунды
 Ограничение по памяти: 64 мегабайта

Капрал Питуца любит командовать своим отрядом. Его любимый приказ «в начало строя». Он выстраивает свой отряд в шеренгу и оглашает последовательность приказов. Каждая приказ имеет вид «Солдаты с l_i по r_i — в начало строя!»

Пронумеруем солдат в начальном положении с 1 до n , слева направо. Приказ «Солдаты с l_i по r_i — в начало строя!» означает, что солдаты, стоящие с l_i по r_i включительно перемещаются в начало строя, сохраняя относительный порядок.

Например, если в некоторый момент солдаты стоят в порядке 2, 3, 6, 1, 5, 4, после приказа: «Солдаты с 2 по 4 — в начало строя!» порядок будет 3, 6, 1, 2, 5, 4.

По данной последовательности приказов найти конечный порядок солдат в строю.

Формат входного файла

В первой строке два целых числа n and m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$) — количество солдат и количество приказов. Следующие m строк содержат по два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

Формат выходного файла

Выведите n целых чисел — порядок солдат в конечном положении после выполнения всех приказов.

Пример

<code>movetofront.in</code>	<code>movetofront.out</code>
6 3	1 4 5 2 3 6
2 4	
3 5	
2 2	

Задача MoveToFront-Easy. Вперёд! (простая)

Задача «Вперёд!» с ограничением 100 на количество солдат и количество приказов.

Задача Tree. Декартово дерево

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

Вам даны пары чисел (a_i, b_i) , Вам необходимо построить декартово дерево, такое что i -ая вершина имеет ключи (a_i, b_i) , вершины с ключем a_i образуют бинарное дерево поиска, а вершины с ключем b_i образуют кучу.

Формат входного файла

В первой строке записано число N — количество пар. Далее следует N ($1 \leq N \leq 50\,000$) пар (a_i, b_i) . Для всех пар $|a_i|, |b_i| \leq 30\,000$. $a_i \neq a_j$ и $b_i \neq b_j$ для всех $i \neq j$.

Формат выходного файла

Если декартово дерево с таким набором ключей построить возможно, выведите в первой строке YES, в противном случае выведите NO. В случае ответа YES, выведите N строк, каждая из которых должна описывать вершину. Описание вершины состоит из 3-х чисел: номер предка, номер левого сына и номер правого сына. Если у вершины отсутствует предок или какой либо из сыновей, то выводите на его месте число 0.

Если подходящих деревьев несколько выведите любое.

Пример

tree.in	tree.out
7	YES
5 4	2 3 6
2 2	0 5 1
3 9	1 0 7
0 5	5 0 0
1 3	2 4 0
6 6	1 0 0
4 11	3 0 0