

## Задача А. Равноправие и дороги

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В древнем Риме имеется  $n$  перекрёстков и  $m$  соединяющих их дорог. Никакая дорога не соединяет перекрёсток с самим собой. По каждой дороге разрешено ходить либо только патрициям, либо только плебеям.

После свержения монархии и установления Римской республики сенат постановил отменить данные ограничения на некоторых дорогах. В ходе жарких дебатов было решено сделать это таким образом, чтобы, с одной стороны, количество дорог без ограничений было минимальным, а с другой стороны, от любого перекрёстка можно было дойти до любого другого, пользуясь только дорогами без ограничений. Несложно заметить, что для достижения данных требований необходимо отменить ограничения на ровно  $n - 1$  дороге.

Разумеется, и патриции, и плебеи хотят сохранить за собой как можно больше привилегий, поэтому настаивают, чтобы дороги только для них встречались в проекте отмены ограничений как можно реже. Всего было подано  $q$  проектов,  $i$ -й из них предлагает отменить ограничения на **ровно**  $a_i$  дорогах для патрициев и, соответственно, на  $n - 1 - a_i$  дорогах для плебеев. Проекты состоят только из этих чисел, конкретные дороги в них не указаны.

Сенат поручил вам определить, какие из представленных проектов можно выполнить, то есть выбрать нужное количество дорог патрициев и дорог плебеев для отмены ограничений так, чтобы были выполнены все условия.

### Формат входных данных

Первая строка входных данных содержит три целых числа  $n$ ,  $m$  и  $q$  — количество перекрёстков в Риме, количество дорог и количество внесённых в сенат проектов соответственно ( $2 \leq n \leq 100\,000$ ,  $1 \leq m \leq 200\,000$ ,  $1 \leq q \leq 100\,000$ ).

Следующие  $m$  строк описывают дороги. Каждое описание состоит из трёх чисел  $v_i$ ,  $u_i$  и  $t_i$ , где первые два числа задают номера перекрёстков, соединённых данной дорогой, а третье число равно 0, если по данной дороге разрешено ходить только патрициям, и 1, если только плебеям ( $1 \leq v_i, u_i \leq n$ ,  $v_i \neq u_i$ ,  $0 \leq t_i \leq 1$ ).

После этого идут  $q$  строк, описывающих проекты. Каждая из них содержит единственное целое число  $a_i$  ( $0 \leq a_i \leq n - 1$ ).

### Формат выходных данных

Для каждого из  $q$  проектов выведите в отдельной строке «YES» (без кавычек), если существует набор дорог, удовлетворяющих требованиям сената о достижимости, который при этом содержит ровно  $a_i$  дорог только для патрициев и ровно  $n - 1 - a_i$  дорог только для плебеев. Если набора дорог с данными свойствами не существует, то выведите «NO» (без кавычек).

### Примеры

стандартный ввод	стандартный вывод
3 3 3	NO
1 2 0	YES
2 3 0	YES
1 3 1	
0	
1	
2	

## Задача В. АлгоЛэнд

Имя входного файла: `algoland.in`  
Имя выходного файла: `algoland.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Недавно королева страны AlgoLand придумала новый способ отмывания денег для своего королевского двора. Она решила, что всякий житель, желающий совершить путешествие из одного города страны в другой, должен расплатиться за это желание своими деньгами.

В стране AlgoLand есть  $N$  городов, пронумерованных от 1 до  $N$ . Некоторые города соединены дорогами, движение по которым разрешено в двух направлениях. Начиная движение по какой-нибудь дороге, путешественник обязательно должен доехать до ее конца.

Предположим теперь, что житель страны хочет совершить путешествие из города  $A$  в город  $B$ . Новый указ королевы гласит, что при проезде по любой дороге страны во время этого путешествия, полицейские могут взять с этого жителя таможенную пошлину в пользу королевского двора (а могут и не взять). Если при этом у жителя недостаточно денег для уплаты пошлины, то он автоматически попадает в тюрьму. Указ также устанавливает величину пошлины для каждой дороги страны. Так как королева заботится о жителях своей страны, то она запретила полицейским брать с жителя пошлину более чем два раза во время одного путешествия.

Отметим, что если существует несколько способов попасть из города  $A$  в город  $B$ , то житель может выбрать для путешествия любой из них по собственному желанию.

Напишите программу, которая определяет, какую минимальную сумму денег должен взять с собой житель, чтобы гарантированно не попасть в тюрьму во время путешествия.

### Формат входных данных

Первая строка входного файла содержит числа  $N$  и  $M$  ( $2 \leq N \leq 10^4$ ,  $1 \leq M \leq 10^5$ ), разделенные пробелом — количества городов и дорог. Следующие  $M$  строк описывают дороги. Каждая из этих строк описывает одну дорогу и содержит три числа  $X, Y, Z$  ( $1 \leq X, Y \leq N$ ,  $X \neq Y$ ,  $1 \leq Z \leq 10^9$ ) разделенных пробелами, означающие, что дорога соединяет города  $X$  и  $Y$  и пошлина за ее проезд равна  $Z$  денежных единиц. Последняя строка содержит числа  $A$  и  $B$  ( $1 \leq A, B \leq N$ ,  $A \neq B$ ) - номера начального и конечного городов путешествия. Гарантируется, что существует хотя бы один способ проезда из  $A$  в  $B$ .

### Формат выходных данных

Единственная строка выходного файла должна содержать одно число, равное минимальной сумме денег, которую должен взять с собой житель, чтобы иметь возможность совершить путешествие из города  $A$  в город  $B$  и при этом гарантированно не попасть в тюрьму независимо от действий полицейских.

### Примеры

<code>algoland.in</code>	<code>algoland.out</code>
5 6 1 5 1 5 4 1 5 2 2 4 2 1 3 2 1000000000 3 1 1000000000 1 3	1000000000

## Задача С. Разрезание графа

Имя входного файла: `cutting.in`  
Имя выходного файла: `cutting.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф. Над ним в заданном порядке производят операции следующих двух типов:

- `cut` — разрезать граф, то есть удалить из него ребро;
- `ask` — проверить, лежат ли две вершины графа в одной компоненте связности.

Известно, что после выполнения всех операций типа `cut` рёбер в графе не осталось. Найдите результат выполнения каждой из операций типа `ask`.

### Формат входных данных

Первая строка входного файла содержит три целых числа, разделённые пробелами — количество вершин графа  $n$ , количество рёбер  $m$  и количество операций  $k$  ( $1 \leq n \leq 50\,000$ ,  $0 \leq m \leq 100\,000$ ,  $m \leq k \leq 150\,000$ ).

Следующие  $m$  строк задают рёбра графа;  $i$ -я из этих строк содержит два числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ ), разделённые пробелами — номера концов  $i$ -го ребра. Вершины нумеруются с единицы; граф не содержит петель и кратных рёбер.

Далее следуют  $k$  строк, описывающих операции. Операция типа `cut` задаётся строкой «`cut u v`» ( $1 \leq u, v \leq n$ ), которая означает, что из графа удаляют ребро между вершинами  $u$  и  $v$ . Операция типа `ask` задаётся строкой «`ask u v`» ( $1 \leq u, v \leq n$ ), которая означает, что необходимо узнать, лежат ли в данный момент вершины  $u$  и  $v$  в одной компоненте связности. Гарантируется, что каждое ребро графа встретится в операциях типа `cut` ровно один раз.

### Формат выходных данных

Для каждой операции `ask` во входном файле выведите на отдельной строке слово «YES», если две указанные вершины лежат в одной компоненте связности, и «NO» в противном случае. Порядок ответов должен соответствовать порядку операций `ask` во входном файле.

### Пример

<code>cutting.in</code>	<code>cutting.out</code>
3 3 7	YES
1 2	YES
2 3	NO
3 1	NO
ask 3 3	
cut 1 2	
ask 1 2	
cut 1 3	
ask 2 1	
cut 2 3	
ask 3 1	

## Задача D. Электричество в каждый дом!

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

Известный чешский математик Отакар Борувка крайне увлечен проектированием электросети Моравии. В Моравии  $N$  городов, и различные строительные фирмы уже предложили Отакару  $M$  проектов построения ЛЭП между какими-то двумя городами. Известно, что в погоне за индивидуальностью и неповторимым строительным почерком, каждая фирма предлагает все свои проекты по одинаковой стоимости, отличной от стоимостей проектов других фирм. Также известно, что каждая фирма предлагает не более *трех* проектов. Можете считать, что фирмы достаточно сообразительны, чтобы не предлагать проектов соединения какого то города с самим собой, но вполне может возникнуть ситуация, что одна или несколько фирм предлагают больше одного проекта соединения одной и той же пары городов.

Борувка собрал всех своих друзей и поручил им задачу спроектировать электросеть минимальной стоимости. Как вы, наверное, уже догадались, электрическая сеть является остовным деревом, а Борувку интересуют только сети, стоимость постройки которых минимальна.

Борувка всегда был уверен, что минимальное остовное дерево у графа одно, и представьте себе его удивление, когда каждый из друзей принес ему свой проект, утверждая что его-то дерево и есть минимальное. Подозревая неладное, он думает, что причиной разных ответов стали ребра одинакового веса. Помогите ему — посчитайте количество возможных электросетей минимальной стоимости, состоящих из ЛЭП, предложенных Борувке.

### Формат входных данных

В первой строке входного файла заданы два целых положительных числа  $N$  и  $M$ , не превосходящие 100 000.

Следующие  $M$  строк содержат по 3 целых числа каждая:  $1 \leq a_i, b_i \leq N$  и  $1 \leq c_i \leq 10^9$  — города, соединенные соответствующей ЛЭП, и ее стоимость. Гарантируется, что для любого числа  $c_i$  найдется не более трех ЛЭП, имеющих такую стоимость.

### Формат выходных данных

Выведите одно число — ответ на задачу. Так как это число может оказаться довольно большим, выведите остаток от деления на  $10^9 + 7$ .

## Примеры

countspans.in	countspans.out
2 2 1 2 1 2 1 1	2
3 5 1 2 2 2 3 3 3 1 3 3 1 1 2 3 1	1
5 10 2 5 8 4 5 8 2 4 3 4 2 3 4 2 3 5 4 6 5 1 6 1 3 5 3 1 5 4 5 7	6