

Тайны временных порталов

Вы - известный исследователь временных порталов, чей отец, также ученый, исчез во времени много лет назад. Он был страстным исследователем и хотел открыть новые миры, чтобы узнать, что скрыто в прошлом.

Но когда вы были ребенком, он попал в один из временных порталов и не смог вернуться. Вы выросли и унаследовали его научное наследие, но также узнали об опасностях временных порталов. Вы хотите найти своего отца и узнать, что случилось с ним, и вам нужно найти самый быстрый путь между двумя временными пространствами, чтобы найти ответы на свои вопросы.

Однако, вы знаете, что изменение прошлого может привести к параллельным временным линиям, и вы хотите сохранить ход истории в соответствии с ее естественным порядком.

Из дневника отца Вам стало известно, что для сохранения хода истории разрешенно перемещаться между двумя временными пространствами через третье, только если их номера отличаются с третьим на не более, чем K .

Вам необходимо уметь отвечать на запрос типа $a\ b$ - наикратчайшее расстояние, между двумя временными пространствами с учетом выполнения условия, либо -1 , если условие невозможно выполнить.

Если существует путь из a в b , то обратный путь, может не существовать.

Входные данные

Первая строка содержит три целых числа, разделенными пробелами: N - количество временных пространств, M - количество односторонних связей между ними и K - максимальная дальность перемещения сквозь третье пространство ($1 \leq N, M \leq 10^3; K \leq 20$)

Следующие M строк задают связи между пространствами; i — я из этих строк содержит три числа a_i, b_i и w_i - два временных пространства и время затраченное на переход между ними ($1 \leq a_i, b_i \leq n; 1 \leq w_i \leq 10^6$)

После задается целое число Q - количество запросов ($1 \leq Q \leq 10^3$) В следующие Q строк содержат два целых числа $a\ b$ ($1 \leq a, b \leq n$).

Выходные данные

Для каждого из запросов выведите на отдельной строке ответ на запрос исходя из условия.

STDIN

5 4 1

1 2 7

2 5 3

5 4 4

4 3 1

4

1 5

5 3

4 5

1 3

STDOUT

-1

5

-1

-1

MIT(O)-Хайп-Парк

Два преподавателя Факультета Информационных технологий и программирования из Университета *MIT(O)* не переносят друг друга.

После переезда университета в только что построенный *MIT(O)*-Хайп-Парк они потребовали себе кабинеты в зданиях, максимально удалённых друг от друга.

Но есть нюанс. Высшее руководство факультета иногда проводит собрания, на котором обязаны присутствовать все преподаватели, в связи с этим, путь между выбранными зданиями должен существовать.

Обратите внимание, что *MIT(O)*-Хайп-Парк еще не достроен, а значит, что возможна такая ситуация, что из одного корпуса не существует пути до другого. Вам поручено найти расстояние между двумя такими максимально удаленными зданиями, между которыми существует путь.

Входные данные

В первой строке находятся два целых числа N и M – количество зданий и количество дорог, соединяющих здания ($1 \leq N \leq 100, 0 \leq M \leq N \cdot (N - 1) / 2$).

Далее в M строках расположены описания дорог: 3 целых числа s_i, e_i, l_i – здания, в которых начинается и заканчивается дорога и длина дороги соответственно ($1 \leq s_i, e_i \leq N, 0 \leq l_i \leq 100$), дороги двунаправленные.

Если не существует способа выделить 2 здания, согласно условию, необходимо вывести число 0.

Выходные данные

выведите одно число – искомое расстояние.

STDIN

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

STDOUT

```
3
```

Фейсконтроль

В далекой империи графов намечается большая вечеринка в честь дня рождения императора.

Изначально было запланировано, что прийти на нее смогут только образцовые жители страны графов (пропаганда отрицательных циклов запрещена законом). Но в последний момент его величеству захотелось прослыть графом прогрессивных взглядов, и он разрешил вход на вечеринку гражданам с отрицательным циклом длины меньше k .

Также оказалось, что желающих посетить мероприятие слишком много, и императорская стража не сможет проверить всех посетителей без вашей помощи. От вас требуется написать программу, способную определить, достоин ли граф пройти на вечеринку, или нет.

Примечание: отрицательных циклов в графах не более одного, каждый приглашенный граф является связным.

Входные данные

В первой строке записаны числа N и k , где N - число графов в очереди и k - максимальная длина цикла.

Дальше идёт описание N графов:

В первой строке описания каждого графа дается число вершин $n \leq 10^5$ и число ребер $m \leq 10^5$. Далее в m строках описываются ребра в формате: начало ребра $1 \leq start$, конец ребра $1 \leq end$ и вес ребра $1 \leq w \leq 10^6$.

Выходные данные

В N строках выведите YES, если очередной гость может пройти на вечеринку, в противном случае выведите NO.

STDIN

STDOUT

```
1 3
6 5
6 5 2859
6 3 5706
5 4 16828
4 2 2996
3 1 5437
```

```
YES
```

В поисках Вязьмы

Как-то вашему другу, студенту ИТМО потребовалось добраться от университета до общежития на Вяземском переулке.

К сожалению, у него разрядился телефон, поэтому он взял в руки бумажную карту и попросил Вас найти ему кратчайшие пути от университета до тех мест, куда он покажет пальцем.

На карте от точки A до точки B было несколько маршрутов, кроме того в некоторых точках можно было сделать петлю и вернуться в саму себя.

Вы решили написать программу, которая ответит на Q запросов вашего друга.

Входные данные

В первой строке задаются $N, M, Q, Start$,

$N \leq 10^6$ – количество точек на карте

M – количество путей между точками на карте

$Q \leq 10^6$ – количество вопросов друга

$1 \leq Start \leq 10^6$ – точка, в которой вы сейчас находитесь.

Далее идут M строк, содержащие по 3 числа: $Start, End, Weight$

$Start$ – точка с которой начинается путь

End – точка на которой путь завершается

$Weight$ – длина.

Далее идут еще Q строк, содержащих по одному числу: Номер точки, в которую хочет добраться студент.

Выходные данные

Q чисел.

Если путь существует, то требуется вывести его длину.

Если пути нет, то выведите -1

STDIN

```
10 6 2 5
3 4 293
3 2 717
9 6 448
7 2 539
10 3 668
10 6 895
4
2
```

STDOUT

```
-1
-1
```

Спасение Подземного Царства

Далеко-далеко, в подземном царстве находится древний город, который обитает многочисленным населением. Город издревле защищался от злых сил, проникающих из недр земли, поэтому была создана сложная система подземных туннелей и катакомб, предназначенная для перемещения солдат и магических существ, обеспечивающих защиту города.

Времена меняются, и волшебные защитные силы города иссякают. Чтобы спасти город, вам предстоит найти кратчайший путь между двумя ключевыми точками подземного мира и доставить мощный магический артефакт.

Подземный мир представлен ориентированным ациклическим графом (DAG), где вершины - это комнаты и перекрестки туннелей, а ребра - это туннели, соединяющие эти комнаты и перекрестки. Вес ребра обозначает время, затрачиваемое на прохождение туннеля между двумя вершинами.

Вам предстоит разработать алгоритм, который найдет кратчайший путь между стартовой точкой (S) и целевой точкой (T), минимизируя затрачиваемое время на перемещение. Ваша задача - доставить магический артефакт в кратчайший срок, чтобы успеть спасти подземное царство.

Входные данные

N ($2 \leq N \leq 1000$) - количество вершин в графе. M ($1 \leq M \leq 10^5$) - количество ребер в графе. S и T - индексы стартовой и целевой точек ($1 \leq S, T \leq N$, $S \neq T$).

Далее следует M строк, каждая из которых содержит информацию о ребрах графа, представленную тремя целыми числами: a , b , и w ($1 \leq a, b \leq N$, $1 \leq w \leq 100$), где a и b - индексы связанных вершин, а w - вес ребра (время прохождения туннеля).

Выходные данные

Выведите одно число - минимальное время, необходимое для перемещения между стартовой и целевой точками. Если путь недостижим, выведите -1.

STDIN

```
6 8
1 6
1 2 3
1 3 5
2 3 1
2 4 3
3 4 1
3 5 2
4 6 3
5 6 2
```



STDOUT

```
8
```



Тиндер 2

Известно, что знакомства транзитивны и симметричны и рефлексивны. Два человека сильно знакомы, если они знакомы напрямую. Два человека a и b знакомы, если существует цепочка людей начинающийся с a и заканчивающийся в b , в которой каждый человек сильно знаком с соседом по цепочке. Например, из определения следует, что сильное знакомство является частным случаем знакомства и тп.

К сожалению знакомства забываются, а в нашей задаче это ещё и влечёт цепочку разрушений знакомств. Например, вы знаете человека X только через Y , и сильное знакомство между X и Y прервётся, то ваше знакомство с X тоже прекратится.

Вы очень умный студент ИС'а, вы решили лабу по алгосам за час и теперь вы задались проверкой гипотез о знакомствах.

Для проверки гипотез вы решили написать симулятор знакомств, SIMS5 на c++, но вот не задача, генерировать новые знакомства очень сложная задача, поэтому вы решили пока остановить реализацию данной функции.

Реализуйте выполнение списка операций. Операции двух типов:

- ? $a\ b$ – проверить знакомы ли a и b .
- $a\ b$ – сильные знакомые a и b больше не знакомые.

Входные данные

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

Следующие m строк задают рёбра графа; i -я из этих строк содержит два числа a_i и b_i ($1 \leq a_i, b_i \leq n$), разделённые пробелами --- номера концов i -го ребра.

Далее следуют q строк, описывающих операции. Операции задаются строкой:

-/? $a\ b$ ($1 \leq a, b \leq n$).

Выходные данные

Для каждой операции ? во входном файле выведите на отдельной строке слово YES или NO исходя из условия.

STDIN

STDOUT

3 3 8

1 2

2 3

1 3

? 1 1

? 1 3

- 1 2

? 1 3

- 2 3

? 2 3

- 1 3

? 1 1

YES

YES

YES

NO

YES