

Хогвартс

Общежитие Хогвартса представляет из себя систему комнат, соединенных между собой дверьми. Учивые преподаватели Хогвартса разбросали по некоторым комнатам ловушки, чтобы жильцы общежития лишний раз не выходили из своих комнат.

Вам, как любопытному студенту-волшебнику, непременно нужно выделить компоненты связности комнат по которым могут перемещаться студенты. А так же Вам интересно определить, какие компоненты связности содержат комнаты с ловушками, чтобы самому не попасться.

Входные данные

Первая строка содержит два натуральных числа n и m - количество комнат и дверей соответственно ($1 \leq n \leq 100000, 0 \leq m \leq 200000$).

Следующие m строк содержат описание дверей: каждая строка содержит два числа a и b - номера комнат, которые соединены дверью ($1 \leq a, b \leq n$).

Затем следует строка из n символов, где i -ый символ равен '0', если i -ая комната безопасна для студентов, и '1', если в ней находится ловушка.

Выходные данные

В первой строке выведите целое число k - количество компонент связности доступных для студентов комнат.

Во второй строке выведите n натуральных чисел a_1, a_2, \dots, a_n , где a_i ($1 \leq a_i \leq k + 1$) - номер компоненты связности, которой принадлежит i -ая комната (компоненты связности нумеруют так, что a_i -ая компонента связности относится к комнате с наименьшим номером, не принадлежащей к $a_1, a_2 \dots a_{i-1}$ компонентам).

В третьей строке выведите k символов, где i -ый символ равен '0', если i -ая компонента является безопасной для студентов, и '1', если в ней находится ловушка.

STDIN

```
4 2
1 3
2 4
0 0 1 0
```

STDOUT

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

Тиндер

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 ($1 \leq n \leq 100\,000$; $0 \leq m, q \leq 100\,000$; $(n + m) * q \leq 10^7$).

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

```
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
```

STDOUT

```
YES
YES
YES
NO
YES
```

Самокаты

В Университете $MIT(O)$ существует N корпусов. Корпуса соединены между собой M односторонними велодорожками. У студентов этого университета существует возможность арендовать самокаты. Единственное условие: самокат, взятый у определенного корпуса должен вернуться на свое исходное место.

Студенту интересно узнать, если он возьмет самокат у корпуса A то до каких корпусов он может доехать, не нарушив правило аренды.

Вам поступает Q запросов вида $A\ B$ на каждый из которых необходимо вывести "YES", если из A можно доехать в B , не нарушив при этом условие проката (после поездки студенту необходимо вернуть самокат на исходную точку). В противном случае необходимо вывести "NO".

Входные данные

В первой строке вводятся три целых числа $1 \leq N, M, Q \leq 10^4$.

Следующие M строк содержат по два целых числа $1 \leq start, end \leq N$, описывающие односторонние велодорожки.

Далее следует Q строк, содержащие по два целых числа $1 \leq A, B \leq N$ - запросы.

Выходные данные

На каждый запрос в новой строке выведите "YES" либо "NO".

STDIN

```
7 8 5
1 2
2 3
3 4
4 5
5 1
5 6
6 7
7 6
1 4
5 6
6 7
2 5
1 7
```

STDOUT

```
YES
NO
YES
YES
NO
```

Потерянное условие

Входные данные

В первой строке входного файла даны два натуральных числа n и m

($1 \leq n, m \leq 10^5$) - количество вершин и ребер графа соответственно.

Далее в m строках перечислены рёбра графа. Каждое ребро задается парой чисел - номерами начальной и конечной вершин соответственно.

Выходные данные

Выведите новые названия вершин, чтобы путь из любой вершины был всегда в вершину с большим номером

STDIN

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

STDOUT

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