Overview

Author	Timothy Stranex	Bruce Merry	Carl Hultquist
Problem	**train**	**taxi**	**shrubs**
Source	train.c train.cpp	taxi.c taxi.cpp	shrubs.c shrubs.cpp
Input file	stdin	stdin	stdin
Output file	stdout	stdout	stdout
Time limit	1 second	1 second	1 second
Number of tests	10	10	20
Points per test	10	10	5
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Train jumping

Author

Timothy Stranex

Introduction

Dave, a Parkour enthusiast, is planning a spectacular demonstration of jumping from the roof of a moving train onto another. To prepare for the stunt, he will practise the jump between two stationary trains. He knows exactly how the trains lie and to make the stunt as safe as possible, he would like the distance he jumps to be as short as possible. You must help him find this shortest distance.

Task

Viewed from above, the trains are described by two 2D line segments. You must calculate the shortest distance between them. The distance between two points (x_1, y_1) and (x_2, y_2) is given by

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Example

Suppose the first train is described by the line segment from $(0, 0)$ to $(0, 20)$ and the second is described by the line segment from $(10, 10)$ to $(20, 20)$. The shortest jump is from $(0, 10)$ to $(10, 10)$. The shortest distance is 10 units.
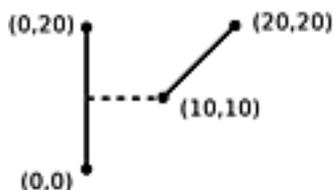
Input (stdin)

The input consists of four lines. Each line contains two space-separated integers describing a point. The first two points are the endpoints of the line segment for the first train. The final two are the endpoints of the line segment for the second train.

Sample input

```
0 0
0 20
10 10
20 20
```

Output (stdout)

The output consists of a single floating point value, the shortest jumping distance.

Sample output

```
10
```

Constraints

All the x and y coordinates will be greater than or equal to 0 and less than 10^6.

Time limit

1 second.

Scoring

If the absolute difference between the answer and the correct answer is less than 1, you will score 100% for the test case. Otherwise, or if the answer is formatted incorrectly, you will score 0% for the test case.

Taxi Strike

Author

Bruce Merry

Introduction

As a commuter in South Africa, you know that when taxi drivers go on strike, they often shut down roads. In some cases, they will shut down an important road and it will no longer be possible to make certain trips!

Task

Write a program that will accept a street map of a city, and answer queries of the form "if taxi drivers shut down road X, will it still be possible to get from A to B?" All roads are bidirectional and connect two distinct intersections. It is guaranteed that there will be route between every pair of intersections if all the roads are available.

Example
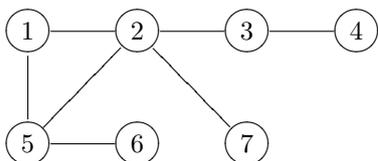
Consider the city layout shown above. If the edge from 2 to 3 is shut down then it is no longer possible to get from 6 to 4 or from 5 to 3, although it is still possible to get from 1 to 7. If the road from 1 to 2 is shut down it is still possible to get from 1 to 2, by going via 5.

Input (stdin)

The first line of input contains three space-separated integers, N (the number of intersections), R (the number of roads), and Q, the number of queries. The next R lines each contain two space-separated integers A and B, indicating a road between intersections A and B. The remaining Q lines each contain three space-separated integers A, B and C, representing the query "if taxi drivers shut down road C, is it still possible to get from A to B?" Intersections are numbered from 1 to N and roads from 1 to R.

Sample input

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

Output (stdout)

The output consists of Q lines. Each line should contain either "YES" or "NO" (excluding quotes) in answer to the corresponding query from the input.

Sample output

```
NO
NO
YES
YES
YES
```

Constraints

- $1 \le N \le 10000$
- $1 \le R \le 10000$
- $1 \le Q \le 10000$

50% constraints

In 50% of the test cases, $R + 1 = N$.

Time limit

1 second.

Scoring

A correct answer scores 100% while an incorrect one scores 0%.

Maintaining the Shrubberies

Author

Carl Hultquist

Introduction

The Knights who say Ni are extremely proud of all the shrubberies that they have managed to collect over the years. However, they've received so many that tending to them all has become quite a task — this is largely due to the fact that they have spread their shrubberies all around the forest. Keeping paths through the forest tidy enough for them to push around their cart of garden shears takes a lot of effort, and so they have resolved to tidy only as much path as is necessary for them to be able to reach all the shrubberies (possibly by taking paths that pass through other shrubberies).

Each day, one of the Knights goes forth and finds a new path between two of the shrubberies (possibly between two shrubberies between which a path already exists, and possibly with different length to any existing paths). At the end of the day, the Knights regroup and decide which paths they will keep tidy the next day, such that the total length of paths kept tidy is a minimum and that it is possible to reach any shrubbery by travelling only along tidy paths.

Task

Given the number of shrubberies that the Knights have and the path discovered by the Knights on a number of days, determine the minimum total length of path that the Knights must keep tidy each day.

All paths are bidirectional, and whilst two paths may cross the Knights aren't very smart and will remain on the path they are on until they reach a shrubbery.

Example

Input (stdin)

The first line of input will contain two space-separated integers, N and M, which are the number of shrubberies and the number of days over which the Knights discover new paths, respectively. The next M lines will each contain three space-separated integers A_i, B_i and D_i, which

are the numbers of the two shrubberies connected and the distance of the path respectively. The shrubberies are numbered from 1 to N, and the Knights start and end each day at the Nth shrubbery.

Sample input

```
4 6
1 2 10
1 3 8
3 2 3
1 4 3
1 3 6
2 1 2
```

Output (stdout)

Your output should be exactly M lines, with a single integer on each line that is the total length of path that must be kept tidy by the Knights. If it is not possible to reach all of the shrubberies from the Knights' starting point (at shrubbery N) using the paths found so far, then your program should output -1 for that day.

Sample output

```
-1
-1
-1
14
12
8
```

Constraints

- $1 \le N \le 200$
- $1 \le M \le 6000$
- $1 \le D_i \le 10000$

50% constraints

- $1 \le M \le 2000$

Time limit

1 second.

Scoring

100% for correctly calculating the length of path to be tidied on each day, 0% if the output for any day is incorrect.