

Another Contest 1 Problem 2 - Graphs

Time limit: 0.6s **Memory limit:** 256M

This problem has a simple statement.

Given an undirected, unweighted graph of N vertices and M edges, and Q pairs of vertices, compute the distance between each pair of vertices.

We ran out of time trying to set strong test data for the problem though, so we did it randomly. Here's how we generated our test data.

- We picked values for N , M , and Q . The vertices are numbered from 1 to N .
- We generated, uniformly at random, a pair of distinct integers between 1 and N and added an edge between those vertices as long as they weren't already directly connected by an edge. This means the graph has no self-loops, nor parallel edges.
- The above step was repeated until the graph contains exactly M edges.
- We generated, uniformly at random, Q pairs of integers between 1 and N for the queries.

Constraints

$N = 10^5$, $M = 2 \cdot 10^5$, and $Q = 10^4$.

Note: The sample does not respect the constraints. Your solution does not need to produce the correct output on the sample to get AC.

Input Specification

The first line of input contains three positive integers, N , M , and Q .

The next M lines each contain two distinct positive integers from 1 to N , representing an edge between the two given vertices.

The next Q lines each contain two positive integers from 1 to N , representing a distance query between the two given vertices.

Output Specification

Output Q lines. For each distance query, output `-1` if the two vertices are disconnected, otherwise output the distance between the two vertices. Output answers for the queries in the order they're given, one query per line.

Sample Input

3 2 2

1 2

2 3

1 3

2 3

Sample Output

2

1