

Problem B

Tower Defense

Input file: *testdata.in*

Time limit: 1 second

Problem Description

Orcs are attacking our kingdom! To protect the king we must build a tower to stop orcs from reaching the capital where the king stays. There are N cities in the kingdom and they are numbered from 0 to $N - 1$. The city numbered 0 is the capital. Cities are connected by M bidirectional roads. Every city can reach all other cities by a series of roads, so orcs appearing at any city can get to the capital.

Orcs can appear at any city and start moving to the capital. To defend this orc invasion you must choose a city to build a tower, so that no matter how the orcs choose their path to the capital, they will be blocked by this tower. We call a tower “effective” if it can do so. Note that there could be more than one city where you can build an effective tower.

We now consider K different possible starting cities where the orcs may appear. In each case the orcs will appear in a single starting city and start moving to the capital, and you must find a city to build a tower to stop them. All K starting city cases are independent and the tower for one starting city case has nothing to do with the tower in another starting city case.

Technical Specification

- $1 \leq N \leq 5000$
- $0 \leq M \leq 1000000$
- $1 \leq K \leq 100$

- Orcs cannot appear in the capital.
- You cannot build towers on the capital, nor the city where orcs will appear.

Input Format

There are multiple test cases. Each test case starts with a line containing N, M , and K . Each of the following M lines contains two integers, i and j , indicating that there is a road connecting city i and city j . Each of the following K lines has the starting city the orcs would appear. A line containing three zeros for N, M , and K indicates the end of input.

Output Format

For each starting city output a line containing the cities where an effective tower could be built, and the cities should be separated by a space character. If there are more than one city where an effective tower could be built, output them in increasing order of their shortest distance to the starting city. If there are not feasible solutions, output 'NOFEASIBLE' in one line.

Sample Input

```
5 4 4
0 1
1 2
2 3
3 4
4
3
2
1
5 5 1
0 1
0 2
1 3
2 3
3 4
```

4
0 0 0

Sample Output

3 2 1
2 1
1
NOFEASIBLE
3