

Breadth-First Search

time limit per test: 1 second
memory limit per test: 976.6 megabytes
input: standard input
output: standard output
points: 100 · problem code: bfs · author: kahootist

Problem Statement

Implement breadth-first search in an undirected graph.

During the traversal, always select the smallest unvisited vertex as the next one.

Input

Your first line will contain v and e as space-separated integers. v represents the number of vertices in the graph, labelled from 0 to $v - 1$, and e represents the number of edges in the graph.

Your next line will contain s , the source vertex to start the DFS from.

Your next e lines will contain 2 space-separated numbers, specifying that there is an edge between the two specified vertices.

Output

The BFS traversal, separated by line breaks.

Constraints

For all test cases:

- $2 \leq v \leq 10^3$
- $0 \leq s \leq v - 1$
- $1 \leq e \leq 10^6$

Example 1

Input

```
6 5
0
0 1
0 2
1 3
1 4
2 5
```

Output

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

Python Template

```
v, e = map(int, input().split())
s = input()
edges = [tuple(map(int, input().split(" "))) for _ in range(e)]

# print(...)
```