

Adjacency List Traversal

time limit per test: 1 second (3 seconds for Python)

memory limit per test: 500 megabytes

input: standard input

output: standard output

points: 100 · problem code: adjlisttraversal

You are given an undirected graph with vertices numbered from 1 to n . Starting at vertex s , run a breadth-first search. When a vertex has multiple unvisited neighbors available, smaller-numbered vertices must be visited first. To do this, sort each adjacency list before running the search.

Input

The first line contains three integers n , m , and s , the number of vertices, the number of edges, and the starting vertex. The next m lines each contain two integers a and b , meaning there is an undirected edge between vertices a and b .

Output

Output one line containing the vertices in the order they are first visited by the BFS, separated by spaces.

Constraints

- $1 \leq n \leq 200000$
- $0 \leq m \leq 300000$
- $1 \leq s \leq n$
- $1 \leq a, b \leq n$

Example 1

Input

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

Output

```
1 2 3 4 5
```

Explanation

The BFS starts at 1. Vertices 2 and 3 are reached first, and vertex 2 is visited before vertex 3 because it has the smaller number.

Example 2

Input

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

Output

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

Explanation

From vertex 4, the sorted neighbors are 2 and 6. Vertex 2 is processed first, but vertex 6 was already discovered before vertices 1 and 3.