

Path Reconstruction

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: pathrestore

You are given an undirected graph with n vertices and m edges, as well as two vertices s and t .

Find the lexicographically smallest shortest path from s to t . Paths are compared as vertex sequences: at the first position where two paths differ, the path with the smaller vertex number is lexicographically smaller.

To make the required path deterministic, sort each adjacency list in increasing order and run BFS from s , storing the first parent used to reach each vertex.

Input

The first line contains four integers n , m , s , and t .

The next m lines each contain two integers a and b , meaning there is an undirected edge between vertices a and b .

Vertices are numbered from 1 to n .

Output

If t is unreachable from s , print `-1`.

Otherwise, print two lines. The first line should contain d , the number of edges in the path. The second line should contain $d + 1$ integers, the vertices of the path from s to t .

Constraints

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

Example 1

Input

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

Output

```
3
1 2 4 6
```

Explanation

There are multiple shortest paths from 1 to 6, including `[1, 2, 4, 6]` and `[1, 2, 5, 6]`. The path `[1, 2, 4, 6]` is lexicographically smallest.

Example 2

Input

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

Output

```
-1
```

Explanation

Vertex 5 is not reachable from vertex 1.