

1 Computing the Minimum Number of Flight Segments

Problem Introduction

You would like to compute the minimum number of flight segments to get from one city to another one. For this, you construct the following undirected graph: vertices represent cities, there is an edge between two vertices whenever there is a flight between the corresponding two cities. Then, it suffices to find a shortest path from one of the given cities to the other one.

Problem Description

Task. Given an *undirected* graph with n vertices and m edges and two vertices u and v , compute the length of a shortest path between u and v (that is, the minimum number of edges in a path from u to v).

Input Format. A graph is given in the standard format. The next line contains two vertices u and v .

Constraints. $2 \leq n \leq 10^5$, $0 \leq m \leq 10^5$, $u \neq v$, $1 \leq u, v \leq n$.

Output Format. Output the minimum number of edges in a path from u to v , or -1 if there is no path.

Time Limits.

language	C	C++	Java	Python	C#	Haskell	JavaScript	Ruby	Scala
time (sec)	2	2	3	10	3	4	10	10	6

Memory Limit. 512MB.

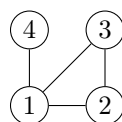
Sample 1.

Input:

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

Output:

```
2
```



There is a unique shortest path between vertices 2 and 4 in this graph: $2 - 1 - 4$.

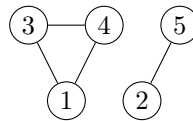
Sample 2.

Input:

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

Output:

```
-1
```



There is no path between vertices 3 and 5 in this graph.