

Name: Implementation of Breadth First Search (BFS) to solve Route Finding Problem.

Problem Description:

In this problem you will be given an undirected unweighted (or each edge equally weighted) graph G (Search Space containing all the states), a source vertex S (Start State) and a destination vertex D (Goal State). You need to find a path or route to reach D from S using the BFS algorithm. You also need to calculate the cost associated with finding the path from S to D. In this problem, the cost to reach a node from another node denotes the number of intermediate edges that have been used to reach.

In the first line you will be given two integers V and E denoting the number of vertices and edges of this graph respectively. All the vertices will have ids between 1 to V. In the following E lines, you will get the information about all the edges. In the i^{th} line you will have two integers x ($1 \leq x \leq V$) and y ($1 \leq y \leq V$) which denotes there is an undirected edge between x and y. After E lines of input, you will get two integers S and D ($1 \leq S \leq V$, $1 \leq D \leq V$) denoting the source and destination vertices of the problem.

As output, you will print an integer number d which will denote the cost to traverse from S to D. Let the cost be x. In the following x lines there will be x edges' information that will be used to traverse to reach D from S according to their usage order. If there lies no path between S and D print "INF" without quotes.

Test Cases:

Input	Output
4 5 1 2 1 3 2 4 3 4 1 4 1 4	1 1 4
6 8 1 2 1 3 1 4 1 5 2 5 3 5 4 5 5 6 1 6	2 1 5 5 6
10 13 1 7 1 5 2 6 2 10 2 7 3 8 3 4 3 6 4 6 5 10	2 2 6 6 3

6 8 7 9 8 9 2 3	
12 16 1 11 1 8 2 4 2 3 3 7 4 12 4 5 5 10 5 9 6 7 6 10 7 9 7 8 8 12 9 11 11 12 8 10	3 8 7 7 6 6 10

Part 2:

Print the **Exploration Tree**. Exploration tree denotes which nodes were explored and their exploration order. For example, look at the following exploration tree. The number beside each node denotes their order of exploration. Exploration as a term differs from discovery. Exploration of a node means, from this node, I will try to search a set of nodes that are connected to it.

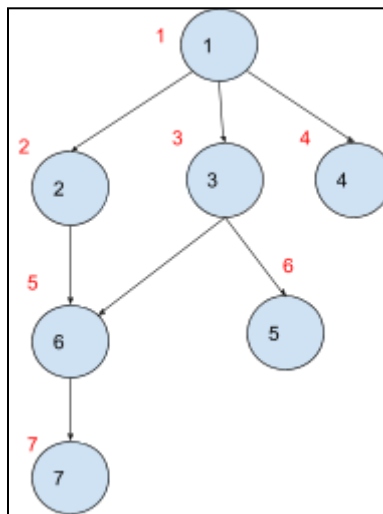


Fig. Example of a exploration tree

For each node, you will print their exploration order. If a node is not explored, print -1 for that particular node as its exploration order.