# Wireless Lab 2

## DSR

**Add a comment with your name and BN at the beginning of your submission.**

Dynamic Source Routing is a routing protocol used in wireless ad hoc networks. The DSR protocol is designed to find and maintain routes between nodes in a wireless network, without the need for a centralized routing infrastructure.

In DSR, when a source node wants to send a packet to a destination node, it first searches its route cache for a route to the destination. If a route is not found, the source node initiates a route discovery process. During route discovery, the source node broadcasts a route request packet, which is received by its neighbors. Each neighbor forwards the packet to its neighbors, until the packet reaches the destination or a node that has a route to the destination in its cache. The route discovery process creates a route from the source to the destination, which is added to the route cache of all nodes along the path. Once a route has been established, packets can be sent between the source and destination using the discovered route.

In this challenge, you will simulate the DSR route discovery process.

Given a graph of $n$ vertices and $m$ edges. Each node has no prior knowledge of the network topology.

Then two vertices are given, your task is to initiate the route discovery process from the source to the destination. Each vertex should broadcast the RREQ to the neighboring vertices. Output the path forwarded in the RREQ.

## Input Format

The first line consists of two integers *n* and *m* - the number of vertices and the number of edges.

Each of the following *m* lines contains a pair of integers *x* and *y*, that show that an edge exists between vertices *x* and *y*. For each pair of vertices there will be at most one edge between them, no edge connects a vertex to itself.

The next line contains a pair of integers *u* and *v*, asking about the route between them.

## Constraints

- $2 \le n \le 1000$
- $1 \le m \le (n.(n-1)) / 2$
- $1 \le x, y \le n, x \ne y$
- $1 \le u, v \le n, u \ne v$

## Output Format

Output *n* lines. The *i* th line represents path forwarded in RREQ for vertex *i* - $(1 \le i \le n)$.

If there are multiple possible paths, output the shortest one having the lexicographically smallest path. If a vertex won't forward a RREQ, output -1.

| Sample Input 0 | Sample Output 0 |
| --- | --- |
| 9 11<br>1 2<br>1 3<br>2 4<br>3 4<br>3 6<br>3 8<br>4 5<br>5 7<br>6 7<br>6 9<br>8 9<br>8 4 | 8 3 1<br>8 3 1 2<br>8 3<br>-1<br>8 3 6 7 5<br>8 3 6<br>8 3 6 7<br>8<br>8 9 |
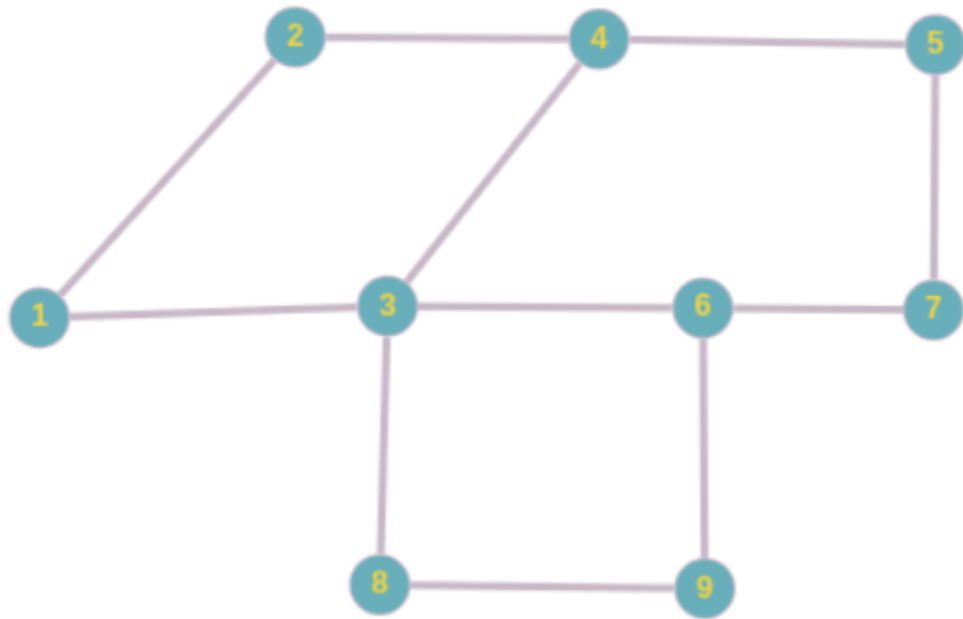
- First, Node 8 starts the route discovery and broadcasts RREQ with path [8] to 3 and 9.
- Node 3 broadcasts to 1, 4, 6 and 8 with path [8, 3].
- Node 1 broadcasts to 2 and 3 with path [8, 3, 1].
- Node 2 broadcasts to 1 and 4 with path [8, 3, 1, 2].
- Node 4 will not forward RREQ as it is the destination.
- Node 9 broadcasts to 6 and 8 with path [8, 9].
- Node 6 receives RREQ from 3 and 9, but will forward that of 3 as it is lexicographically smaller. It will forward to 7, 3 and 9 with path [8, 3, 6].
- Node 7 broadcasts to 5 and 6 with path [8, 3, 6, 7].
- Node 5 broadcasts to 4 and 7 with path [8, 3, 6, 7, 5].

| Sample Input 1 | Sample Output 1 |
|---|---|
| 6 7<br>1 2<br>1 3<br>2 4<br>3 4<br>3 5<br>4 6<br>5 6<br>1 4 | 1<br>1 2<br>1 3<br>-1<br>1 3 5<br>1 3 5 6 |

| Sample Input 2 | Sample Output 2 |
|---|---|
| 4 3<br>1 2<br>2 3<br>3 4<br>2 3 | 2 1<br>2<br>-1<br><br>-1 |

This is an **individual** assignment.

Write your name, section and BN in a comment at the top of your submission.

**Submit to**: https://www.hackerrank.com/cufe-wn-lab2