Problem Definition:-

1. We have a protocal called (DSR) Dynamic Source Routing

2. Our goal is to find the route and save them for each node, from source to destination

3. Each node broadcasts a route request to all its neighbours each neighbour broadcast the packet to all his neighbours except the one which has sent the packet & the previously visited. → VIS Arr

4. the process is repeated till reaching the Destination.

5. the Discovery process creates a route from the src to the destination, which is added to the route each of each node. → [illegible]

6. we want to simulate this process.

Input:

n → # nodes   src → dest

m → # edges

Out put:

the path forwarded in the RREQ for each vertex

Constrains $2 \le n < 1000$ $1 \le m \le (n(n-1))/2$

Scanned with CamScanner

Test Case 1:

8 → 3, 9 Path [8]

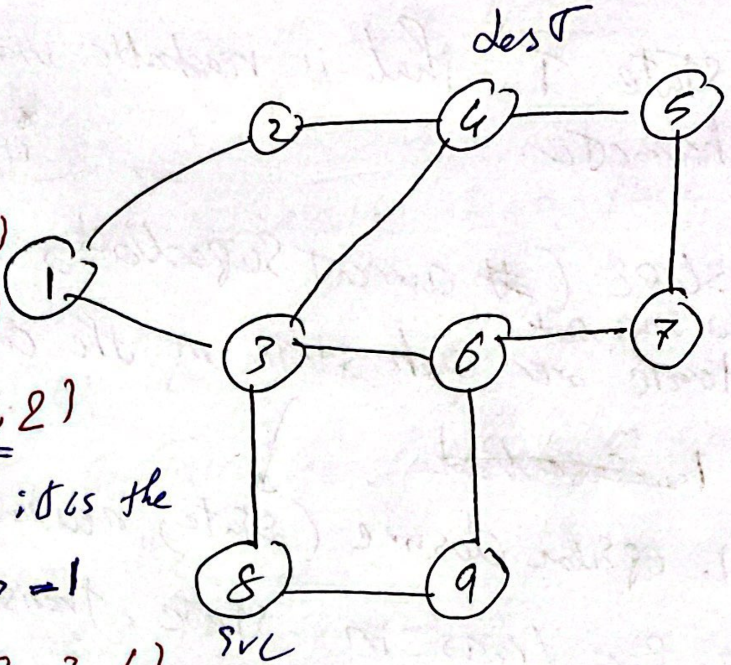
3 → 1, 4, 6, 8 Path[8,3]

1 → 2, 3 Path [8, 3, 1]

2 → ~~1~~, 4 Path [8, 3, 1, 2]

4 → will not apply as it's the
Dest → -1

6 → 3, 9, 7 Path [8, 3, 6]



Path

8

~~8→~~

9 → 6, 8 Path [8, 9]

7 → 5, 6 Path [8, 9, 6, 7)
Path [8, 3, 6, 7]
lexographically smaller

5 → 4 7
Path
83675

خالص كدا حتة ال DP

الموضوع، لو زورت ال Node
قبل كدا، لو مخزن Path
قارن ال Path الحالي باللي
متخزن و خزن ال lexographically
اصغر

871 ✓
8712 ✓
83 ✓
-1 ✓
83675 ✓
876 ✓
8367 ✓
8 ✓
89 ✓

مظبوط

Algorithm:

1. Read the Graph as List → read # as strings
2. store the src & the destination
3. create a visited array to mark if a node is previously visited.
4. create empty string to contain the path.
5. create a map to contain the path from each node
6. apply DFS from the src (G, vis, src, "src") memo)
~~7. Iterate~~ if not marked in memo → =!

initialize memo[src] = src
vis[src] = true

```
DFS (&Graph, &vis, src, path, memo, dest)
    if (memo[src])
        memo[src] = min( path, memo[src])
vis[src] = true →  return
    for nei in Graph[src]        → strings?
        if vis[nei]                    if dest continue
            memo[nei] = min(path + nei, memo[nei])
            continue
        else
            path += nei
            memo[nei] = path      } contradict the first condition
            DFS(G, vis, nei, path, memo)
```

متكتبش دي

غالباً مينفعش

min → 1. shorter in length
      2. lexicographical order