

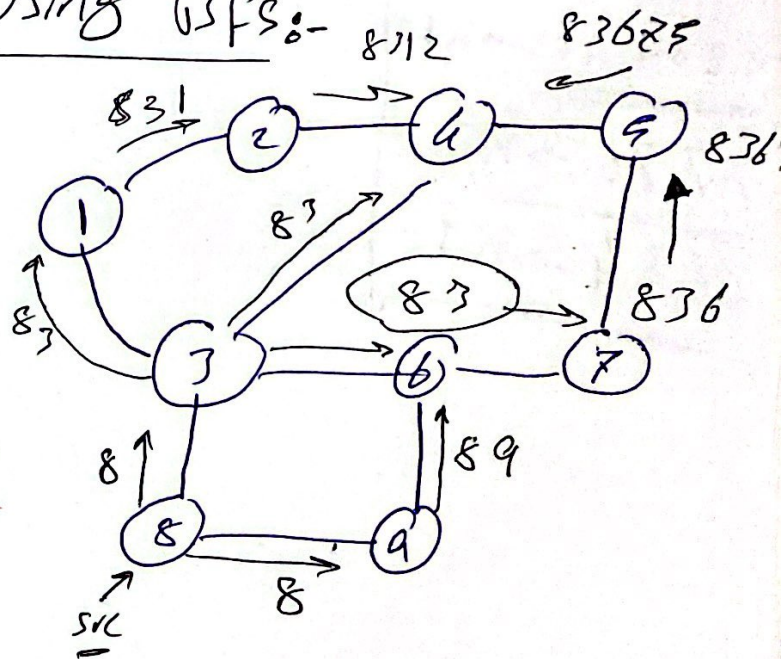
We need to solve this using BFS:-

@6, we have conflict

83 from 3 & 89 from 9

so we should select the one with shorter length & smallest lexicographical order

83



@4, we have 2 conflicts

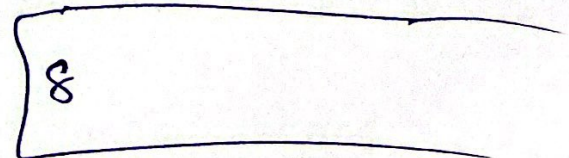
{8,3}, {8312}, {83675}

↓  
shorter length

What is the Algorithm now?

BFS  
Queue

8 iterate over all its ne  
→ insert them in the Queue.  
if they're not visited before  
→ if they were inserted, just  
check on their Path



map	Path
8	8
3	83
9	89
6	<u>836</u>   896

checked  
before

During neighbor  
iteration

checked before

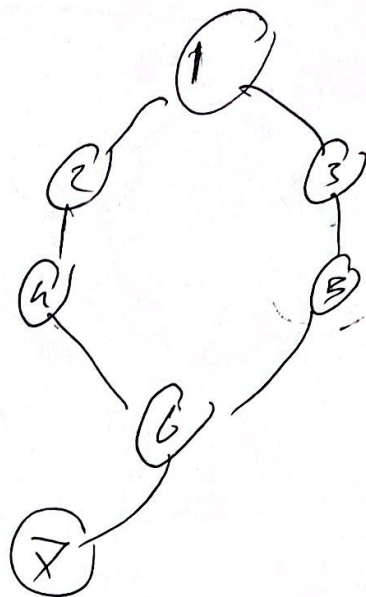
visited before → after visiting



Q  
~~1 2 3 4 5 6~~

NB

1 2 3 4 5 6 7  
 ✓ ✓ ✓ ✓ ✓ ✓ ✓



6 qps logic 11 16  
 25

Implementation : 16

#	Path
1	1
2	1 2
3	1 3
4	1 2 4
5	1 3 5
6	<u>1 2 4 6</u>
	<del>1 3 5 6</del>
7	1 2 4 6 7



DFS

Path = ""

1. Create an empty Queue & insert the src
2. Create a map for paths
3. Create a visited vector.

memo[src] = src

4. while (Q.size() > 0)

4.1 <sup>Path += "</sup> Extract the top of the Queue.

4.2. Mark it as visited.

4.3. Iterate over all its neighbours.

4.3.1. if memo[nei]

4.3.1.1. if length(current path) < ~~length~~

memo[nei] =

4.3.1.2. else if equal length

memo[nei] = min lexicographical

Path = memo[top] || top  
if not empty

~~memo[top] +~~

else

memo[nei] = Path + nei

دینا کسی کے پاس

if [vis]

insert in the Queue