PROBLEM 3 *Counting Shortest Paths* [13 points]

Given a graph $G = (V, E)$, and a starting node s, let $\ell(s, t)$ be the length of the shortest path in terms of number of edges between s and t. Design an algorithm that computes the number of distinct paths from s to t that have length exactly $\ell(s, t)$.

Answer

Graph G = (V, E)
Starting node = s
Ending node = t
Length of shortest path from s to t = l(s, t)

For this problem, we just need to use the Breadth First Search Algorithm and modify it a bit. While running BFS, we can keep track of all the nodes [Distance from the start and number of paths from the start].

To do this, we can make use of some extra arrays\maps:

- Distance *to store the current distance from the start.*
- Paths *to store the number of paths from the start.*

First, we can initialize the arrays/maps, Distance with infinity (start with 0) and Paths with 0 (start with 1 as there is one path from start to itself).

Then we can update the arrays/maps in the following manner:

- If the distance of some neighbor of node n is bigger than the distance of current node n + 1, then we set the distance of the neighbor = to the current node and the Paths[neighbor] = Path[current].
- If the distance of some neighbor of node n is equal to the distance of the current node n +1, then we add the Paths[current] to Paths[neighbor].

In the Paths[x], where x is some node, is stored the number of unique paths from the start to node x. Looking at how this algorithm operates, it goes through each node only once, thus the time complexity will be the same as BFS **O(n+m).**

Some possible algorithm…

- Start performing the BFS algorithm.
- For every neighbor Y of each vertex X, after pushing or not to the queue do:
    - If distance[Y] > distance[X] + 1, then decrease the distance[Y] to distance[X] + 1 and assign the number of paths of vertex X to the number of paths of vertex Y.
    - Else if distance[Y] = distance[X] + 1, then add the number of paths of vertex X to the number of paths of vertex Y.
    - When the BFS queue is empty, then stop.

In the Paths array/map, we have stored all the number of distinct paths from start to x – node.