Lab Assignment 04

22201936

Task - 1 (a)

To store the graph, had to use 2D array to simulate a matrix. If there are r - rows and c - columns, for traversing the whole matrix, Time complexity O(rxc). Space complexity O(rxc). Accessing any individual cell is O(1).

Task - 1 (b)

Here, took help from the Linked List data structure. The connection from one node to another node and their weight stored as a tuple in the List.

Task - 2

Breadth - first search technique of graph traversal used here. First the source inserted in the queue. Then poped from the queue and add the adjacency neighbor in the queue. Repeated the process atlantill queue becomes empty.

Task 3

Depth first search technique used here to traverse the graph. maintained a stack. ~~pu~~ It Initially pushed a value in the stack and goes at its depth untill no where to go. Then poped and try for the previus node. Done it recursively untill the stack is empty

Task 4

Use A DFS technique to find ~~ani~~ circle. Starting from a node, make it visited and path True. And while backtracking ~~&~~ make the path False. If a node is visited and it's path is True that means there's a cycle exist in the graph.

Task 5

To find the shortest path from node 1 to a given destination used bfs technique and maintained two extra array level and parent ~~matrix~~ with visited array. After that backtracked the path and get the results.

Task 6

To solve this problem used BFS technique. If there's and cell with '#' or visited ignored that. And iteratively BFS through the whole matrix and mainted a array ~~A~~ for every BFS and returned the maximum value from the array.