Name : Mayuresh Bhagwan Nehe

PRN No: - 20190802076

LAB-5

AIM:

1. To Implement BFS & DFS (Inorder Traversal) for a given Graph.

2. To check if a graph is strongly connected.

3. To print pre & post visited times.

THEORY: Traversing the graph means examining all the nodes and vertices of the graph. There are two standard methods by using which, we can traverse the graphs.

1) Breadth First Search

2)Depth First Search

Breadth First Search (BFS) Algorithm:

Breadth first search is a graph traversal algorithm that starts traversing the graph from root node and explores all the neighboring nodes. Then, it selects the nearest node and explore all the unexplored nodes. The algorithm follows the same process for each of the nearest node until it finds the goal.

Depth First Search (DFS) Algorithm:

Depth first search (DFS) algorithm starts with the initial node of the graph G, and then goes to deeper and deeper until we find the goal node or the node which has no children. The algorithm, then backtracks from the dead end towards the most recent node that is yet to be completely unexplored. The data structure which is being used in DFS is stack. The process is similar to BFS algorithm. In DFS, the edges that lead to an unvisited node are called discovery edges while the edges that lead to an already visited node are called block edges.

BFS Algorithm	DFS Algorithm
• **Step 1:** SET STATUS = 1 (ready state) for each node in G • **Step 2:** Enqueue the starting node A and set its STATUS = 2 (waiting state) • **Step 3:** Repeat Steps 4 and 5 until QUEUE is empty • **Step 4:** Dequeue a node N. Process it and set its STATUS = 3 (processed state).	• **Step 1:** SET STATUS = 1 (ready state) for each node in G • **Step 2:** Push the starting node A on the stack and set its STATUS = 2 (waiting state) • **Step 3:** Repeat Steps 4 and 5 until STACK is empty • **Step 4:** Pop the top node N. Process it and set its STATUS = 3 (processed state) • **Step 5:** Push on the stack all the neighbours of N that are in the ready state (whose STATUS = 1) and set their

• **Step 5:** Enqueue all the neighbours of N that are in the ready state (whose STATUS = 1) and set their STATUS = 2 (waiting state) [END OF LOOP] • **Step 6:** EXIT	STATUS = 2 (waiting state) [END OF LOOP] • **Step 6:** EXIT

CONCLUSION: We have successfully implemented BFS & DFS for the given graph.