TASK 5

Name: Sami Imran

Roll No: 033

1. DFS with Stack & Node

Depth-First Search (DFS) is a traversal algorithm used for exploring trees and graphs. Instead of using recursion, we can implement DFS using an explicit stack, which manually keeps track of nodes to be visited.

In this approach, we start by pushing the root node onto a stack. Then, we repeatedly pop a node from the stack, process it, and push its children onto the stack in a way that ensures we always explore the last-added node first (LIFO order). This means the left child is added last so it is processed first, maintaining the correct DFS behavior. The process continues until the stack is empty. This method is useful for avoiding recursion depth limits and improving control over the traversal process.

2. Research about "Inorder, Preorder, Postorder" and Implement in DFS

There are three common types of DFS traversal in trees:

1. Inorder Traversal (Left → Root → Right)

In **inorder traversal**, we first visit the **left subtree**, then process the **root node**, and finally visit the **right subtree**. This order is especially useful for **Binary Search Trees (BSTs)** because it retrieves elements in sorted order.

2. Preorder Traversal (Root → Left → Right)

In **preorder traversal**, we process the **root node** first, then visit the **left subtree**, followed by the **right subtree**. This traversal is commonly used when we need to **copy or serialize** a tree structure because it processes nodes before their children.

3. Postorder Traversal (Left → Right → Root)

In **postorder traversal**, we first visit the **left subtree**, then the **right subtree**, and finally process the **root node**. This method is useful for **deleting a tree** because it ensures child nodes are removed before the parent node, preventing memory leaks or dangling references.

Each of these traversal methods follows the **DFS approach**, where we go deep into a branch before backtracking. They are widely used in searching, expression trees, and hierarchical data structures.