Data Structures Lab Manual

Lab 7: Binary Tree , Binary Search Tree

Prepared for: Students of Data Structures

Department of Computer Science  
Fast School of Computing

# Objectives

1. To understand the structure, properties, and applications of Binary Trees and Binary Search Trees.
2. To implement traversal techniques (Inorder, Preorder, Postorder).
3. To perform insertion, deletion, and searching operations in BSTs.
4. To analyze recursion and iterative methods in tree operations.
5. To strengthen problem-solving and algorithmic thinking in hierarchical data structures.

# Lab Outcomes

After successfully completing this lab, students will be able to:  
1. Construct and traverse binary trees and BSTs using recursive and iterative approaches.

2. Perform insertion, searching, and deletion operations efficiently.

3. Differentiate between Binary Tree and Binary Search Tree behavior.

4. Apply traversal methods to solve real-world problems such as expression evaluation and hierarchical organization.

# Lab Task

# Part 1: Construct and Display a Binary Tree

Task 1: Write a program to construct a Binary Tree using user input (level order or pre-defined nodes). Display the tree using Inorder, Preorder, and Postorder traversals.  
  
Expected Output Example:  
Preorder: 1 2 4 5 3  
Inorder: 4 2 5 1 3  
Postorder: 4 5 2 3 1

**Note: Read the file and load that into binary search tree and then display it by using three traversals. File is uploaded with Lab.**

# Part 2: Implement Binary Search Tree

Task 2: Create a Binary Search Tree (BST) and perform insertion operations for a given sequence of integers. Display the tree using Inorder traversal (which should print sorted order).  
  
Example Input: 50, 30, 70, 20, 40, 60, 80  
Expected Output: 20 30 40 50 60 70 80

Task 3: Write a recursive function to search for a given element in a BST. Print whether the element exists or not.  
  
Example Input: Search 60  
Expected Output: Element 60 found in the BST.

Note: Count the number of comparisons made during the search.

Task 4: Write both iterative and recursive code to insert nodes in binary search tree.

# Part 3: Find Height and Leaf Count of a Binary Tree

Task 5: Write functions to compute:  
1. Height of the tree  
2. Number of leaf nodes  
  
Expected Output Example:  
Height of tree: 3  
Total leaf nodes: 4

# Part 4: Mirror a Binary Tree

Task 6: Write a recursive function that converts a Binary Tree into its mirror. Display Inorder traversal before and after mirroring.

Expected Output:  
Inorder before mirror: 4 2 5 1 3  
Inorder after mirror: 3 1 5 2 4

Hint: Swap left and right child pointers recursively.

# Submission Guidelines

- Submit your .cpp file with proper comments.  
- Make sure your program compiles and runs successfully.