Semester VI (B.Tech)

Jaypee University of Engineering and Technology, Guna

Advanced Programming Lab-3 18B17CI673

Lab Exercise 2: Binary Search Tree and AVL Trees

Title: Implementation and Analysis of Binary Search Trees and AVL Trees

Mode: Self Learning

Outcomes:

1. Understand the concept of Binary Search Trees and AVL Trees
2. Learn how to insert, delete, and search for elements in Binary Search Trees and AVL Trees
3. Analyze the performance of Binary Search Trees and AVL Trees

Methodology:

Theory: Introduction to Binary Search Trees, operations like insertion, deletion, and searching in Binary Search Trees, Balanced Binary Search Trees and their importance, AVL Trees, and their properties, rotations and balancing factors, etc.

Implementation: Implementing Binary Search Trees and AVL Trees in C++/Java/Python or any other programming language of your choice, with operations like insertion, deletion, and searching.

Analysis: Analyzing the time and space complexity of the implemented operations and comparing the performance of Binary Search Trees and AVL Trees.

Steps:

1. Study the concept of Binary Search Trees and AVL Trees.
2. Implement the Binary Search Trees and AVL Trees using the chosen programming language.
3. Test the implemented operations (insertion, deletion, and searching) for different inputs.
4. Analyze the performance of the implemented operations in terms of time and space complexity.
5. Compare the performance of Binary Search Trees and AVL Trees.
6. Write a report on the implementation and analysis of Binary Search Trees and AVL Trees.

Experiments:

1. Write a program to construct a Binary Search Tree and check by inserting the elements. 25,15, 5, 20, 40, 3, 6, 16, 21.
2. Write a program for deleting the elements from the above Binary Search Tree.

3. Check it by deleting the elements 25, 40.
4. Write a program to construct the AVL tree using proper rotations by inserting the elements. Check by inserting the elements 5, 10, 15, 20, 15, 40, 45, 80, 90