Semester VI (B.Tech)

Jaypee University of Engineering and Technology, Guna

Advanced Programming Lab-3 18B17CI673

Lab Exercise 3: Red Black Trees

Title: Implementation and Analysis of Red Black Trees

Mode: Self Learning

Outcomes:

1. Understand the concept of self-balancing Binary Search Trees
2. Learn about Red Black Trees and their properties
3. Implement and analyze the performance of Red Black Trees

Methodology:

Theory: Introduction to self-balancing Binary Search Trees, Red Black Trees and their properties, rotations and color changes, and their advantages and disadvantages.

Implementation: Implementing Red Black 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 Red Black Trees with other Binary Search Trees.

Steps:

1. Study the concept of self-balancing Binary Search Trees and Red Black Trees.
2. Implement the Red Black 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 Red Black Trees with other Binary Search Trees like AVL Trees, Splay Trees, etc.
6. Write a report on the implementation and analysis of Red Black Trees.

Experiments:

1. Write a program to construct a Red Black Tree using sequence of insertions. Elements should be taken from the user. You should also print the resultant tree after every insertion.
2. Write a program to delete the element from a given red black tree. You also also print the leftover Red Black Tree.