University of Central Punjab

Faculty of Information Technology

Data Structures and Algorithms
Spring 2023

Graded Lab 04	
Topic	• Binary Search Trees
Objective	The basic purpose of this lab is to test students on recursive Binary Search Trees (BST)

Graded Task

Task: Implement a program that allows users to create two binary search trees (BSTs) by providing input values and then checks if they are identical using recursion.

Specifications:

1. Implement a binary search tree structure/class with the following operations:
- `void **insert**(int value)`: Inserts a new node with the given value into the BST.
- `bool **search**(int value)`: Searches for a node with the given value in the BST and returns true if found, false otherwise.
- `bool **isIdentical**(BST* other)`: Takes a pointer to another BST object as input and returns true if the two BSTs are identical, and false otherwise.

2. Implement a function to take user input and create the two BSTs:
- The function should prompt the user to enter the values for the first BST, one value at a time.
- After entering all the values for the first BST, the function should prompt the user to enter the values for the second BST.
- The user can input any number of values for each BST.

3. The `**isIdentical**()` function should perform a recursive comparison of the trees, checking if the values at each node match and if the left and right subtrees are also identical.

4. Use the `**isIdentical**()` function to check if the two BSTs are identical.

5. Print the results, indicating whether the BSTs are identical or not.

Example output:
Create the first binary search tree
Enter space-separated values for BST: 8 4 12 2 6 10 14

Create the second binary search tree
Enter space-separated values for BST: 8 4 12 2 6 10 14

The two binary search trees are identical.