**UCS 2312 Data Structures Lab**

**Assignment 5: BSTADT and its application**

**Date of Assignment: 07.11.2022**

Create an ADT for the binary search tree data structure with the following functions. Each node which consists of integer data, address of left and right children. [CO1, K3]

1. insertBST(t,data) – insert data into BST
2. inorder(t) – display the tree using inorder traversal
3. preorder(t) – display the tree using preorder traversal
4. postorder(t) – display the tree using postorder traversal
5. levelorder(t) – display the tree hierarchically
6. findmin(t)– returns the minimum element in the tree
7. search(t,key) – returns the element found, otherwise returns NULL
8. delete(t,elt) – delete the given elt from tree
9. Demonstrate the BSTADT with the following test case Insert(t,29)

Insert(t,23) Insert(t,4) Insert(t,13 Insert(t,39 Insert(t,31 Insert(t,45 Insert(t,56 Insert(t,49

Inorder(t) 🡪 4,13,23,29,31,39,45,49,56

Levelorder(t)🡪 1st Level 🡪 29

2nd level 🡪 23, 39

3rd Level 🡪4, 31, 45

4th Level 🡪13, 56 5th Level 🡪 49

Findmin(t) 🡪 4

Find(t, 13) 🡪 Found, value is 3 Find(t,3) 🡪 Not found

1. Write an application to do the following
   1. Check whether the two BST contains the same set of elements
   2. Count the number of nodes in tree within the given range
   3. Find sum of k smallest elements in the given BST

Test case for the Application (a)

# Input: Tree 1

10

/ \

5 50

/ / \

1 40 100

# Input: Tree2

10

/ \

5 50

/ / \

1 40 100

Tree1 and Tree2 are identical with a set of elements (b)

Tree1 not complete (c)

Tree1 Range: [**5**, 45]

Output: 3

Nodes are 5, 10, 40

Tree2 Range: [**1**, 45]

Output: 4

Nodes are 1,5, 10, 40

**Best practices to be followed:**

* Design before coding
* Usage of algorithm notation
* Use of multi-file C program
* Versioning of code