**National University of Computer and Emerging Sciences**



Laboratory Manual

for

Data Structures Lab

|  |  |
| --- | --- |
| Lab Instructor(s) | Anosha Khan |
| Section | BCS 3G |
| Semester | Fall 2024 |

**Department of Computer Science**

FAST-NU, Lahore, Pakistan

**Binary Search Tree**

**Objectives:** In this lab, students will practice Binary Search Tree.

Implement the following Tree Node:

struct Node

{

int data;

Node\*left;

Node \*right;

};

**Problem Statement**

Implement a binary search tree class “BST” which contains the root of type **Node** as a data member.

class BST

{

Node\* root;

};

**Implement the following member functions for BST:**

1. A default Constructor that sets the root to **nullptr**.
2. Implement a function ‘insert’. It should insert the data while considering the insertion rules. If the data already exists in the BST, simply return false and true otherwise.   
   bool insert(int v)
3. A copy constructor which uses recursion to deep copy another Binary Search Tree object.
4. Do inorder tranversal recursively.
5. Now do inorder tranversal iteratively.

void inorderPrint () const

1. Use level order traversal for the printing of trees, level by level.  
   void levelorderPrint () const
2. A function “search”. The function then uses recursion to return a pointer to the corresponding node. If the key does not exist, the function returns **nullptr**.  
   Node\* search(intkey)
3. Use inorder **LVR** to implement a recursive function “countNodes” to return the count of total nodes in BST.

int countNodes() const

1. Use Preorder traversal VLR to implement a recursivefunction “leafCount” to return the count of leaf nodes in BST.

int leafCount() const

1. Use Postorder LRV to implement the Destructor for BST.