



**East West University**

**Department of Computer Science & Engineering**

**A/2, Jahurul Islam Avenue, Jahurul Islam City, Aftabnagar, Dhaka-1212**

---

**Lab Manual** : 06  
**Course Code** : CSE207  
**Course Title** : Data Structure  
**Instructor** : Dr. Maheen Islam, Associate Professor, CSE

### **Objective:**

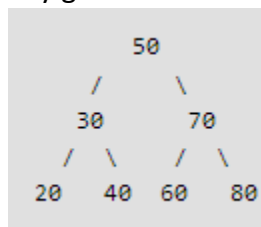
The objective of this lab is to provide basic concept of Binary Search Tree. At the end of the lab, students are able:

- To learn how to create BST
- To learn how to perform different traversal of BST
- To learn how to perform Insertion, Deletion, Searching of BST

### **Binary Search Tree:**

A Binary Search Tree (BST) is a tree in which all the nodes follow the below-mentioned properties –

- The left sub-tree of a node has a key less than or equal to its parent node's key.
- The right sub-tree of a node has a key greater than to its parent node's key.



### **Exercise 1:**

#### **Insert Operation**

Whenever an element is to be inserted, first locate its proper location. Start searching from the root node, then if the data is less than the key value, search for the empty location in the left sub tree and insert the data. Otherwise, search for the empty location in the right sub tree and insert the data.

**Exercise 2:****Search Operation**

Whenever an element is to be searched, start searching from the root node. Then if the data is less than the key value, search for the element in the left subtree. Otherwise, search for the element in the right subtree. Follow the same algorithm for each node.

**Exercise 3:****Traversal Operation**

Perform **Pre, Post and In-order** traversal of BST

**Exercise 4:****Deletion Operation**

Perform deletion operation of BST. You have to perform the following three operations:

- 1) Node to be deleted is leaf: Simply remove from the tree.
- 2) Node to be deleted has only one child: Copy the child to the node and delete the child
- 3) Node to be deleted has two children: Find inorder successor of the node. Copy contents of the in-order successor to the node and delete the inorder successor. Note that inorder predecessor can also be used.

**Exercise 5:****Smallest and Largest Element Searching Operation**

Find maximum and minimum element from BST