Lab 10: Binary Search Tree (BST)

CLO: 02

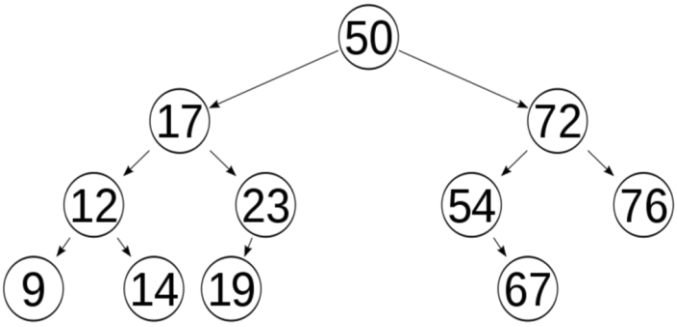
Objectives: In this lab you will learn about

- Implementing binary search trees.
- Basic functions that can be performed on BST.

Binary Search Tree (BST):
Binary Search Tree is a node-based binary tree data structure which has the following properties:

- The left subtree of a node contains only nodes with keys lesser than the node's key.
- The right subtree of a node contains only nodes with keys greater than the node's key.
- The left and right subtree each must also be a binary search tree.

A binary tree is made of nodes, where each node contains a "left" pointer, a "right" pointer, and a data element. The "root" pointer points to the topmost node in the tree. The left and right pointers recursively point to smaller "subtrees" on either side. A null pointer represents a binary tree with no elements -- the empty tree. The formal recursive definition is: a binary tree is either empty (represented by a null pointer), or is made of a single node, where the left and right pointers (recursive definition ahead) each point to a binary tree.



Basic Operations
Following are the basic operations of a tree −

1. Search − Searches an element in a tree.
2. Insert − Inserts an element in a tree.
3. Pre-order Traversal − Traverses a tree in a pre-order manner.
4. In-order Traversal − Traverses a tree in an in-order manner.
5. Post-order Traversal − Traverses a tree in a post-order manner.

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 sub_tree. Otherwise, search for the element in the right sub_tree. Follow the same algorithm for each node.

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 subtree and insert the data. Otherwise, search for the empty location in the right subtree and insert the data.

Lab Tasks

You are tasked with implementing a Patient Management System for a hospital using a Binary Search Tree (BST). The system should efficiently manage patient records based on unique Patient IDs.

Task:

- BST Insertion: Implement a function to insert patient records into the BST based on Patient IDs.
- BST Search: Create a function to search for a patient based on their Patient ID.
- BST In-order, pre-order, post-order Traversal ask user which function he or she want to use: Develop a function to display patient records in ascending order based on Patient IDs.
- Function called Max to find the patient with maximum age in Tree.
- BST Deletion: Implement a function to delete a patient record from the BST.
- Scenario Execution: Demonstrate the usage of the implemented functions in a scenario involving patient addition, search, and deletion.
- Patient IDs are unique integers.
- Patient names, medical history, and date of admission are strings.
- Age is an integer.