



Aror University of Art, Architecture, Design & Heritage Sukkur

Department of AI-Multimedia and Gaming

BS-AI (Sec# A and B), Fall 2024 Batch

Lab 09: Binary Search Tree (BST)

Subject: Data Structure (CSC221)

Date: 5 November, 2024 Instructor: Abdul Ghafoor

Objective

By the end of this lab, students will:

1. Understand the concept of Binary Search Trees (BSTs).
2. Implement a BST data structure using either linked lists or arrays.
3. Insert elements into the BST while maintaining its properties.
4. Calculate the total number of nodes in the BST.
5. Search for elements in the BST.
6. Implement and perform tree traversals (In-order, Pre-order, Post-order, and Level-order).

Note: Students may use built-in Java packages like `java.util.Queue` or `Stack` for traversal. BST properties must be maintained during insertion.

Part 01: Create a Binary Search Tree Class

Step 1: Implement a Node Class

- Define a class `Node` with:
 - `int value` (value of the node).
 - `Node left` (pointer to the left child).

- `Node right` (pointer to the right child).

Step 2: Implement a BST Class

- Define a class `BinarySearchTree` with:
 - A `Node root` (the root of the tree).
 - A method `void insert(int value)` to add elements to the BST:
 - Insert elements while maintaining BST properties (values smaller than the current node go to the left, values larger go to the right).
 - A method `int countNodes()` to calculate and return the total number of nodes in the BST.
 - A method `boolean search(int value)` to search for a value in the BST.
-

Part 2: Implement Traversals

Implement the following traversal methods:

1. **In-order Traversal:** Left \rightarrow Root \rightarrow Right (used to retrieve sorted order).
 2. **Pre-order Traversal:** Root \rightarrow Left \rightarrow Right.
 3. **Post-order Traversal:** Left \rightarrow Right \rightarrow Root.
 4. **Level-order Traversal:** Traverse the tree level by level using a queue.
-

Part 3: Testing the Binary Search Tree

Insertion Test

1. Insert at least 7-10 elements into the BST. For example:
`50, 30, 20, 40, 70, 60, 80`.
2. After inserting each element, print the total number of nodes in the BST using `countNodes()`.

Search Test

1. Search for specific values (e.g., `20, 60, 90`) in the BST using the `search()` method.

2. Print whether the value exists in the tree.

Traversal Tests

1. After inserting all elements, perform all four traversals.
 2. Display the results of each traversal:
 - **In-order Traversal** (should print the elements in sorted order).
 - **Pre-order Traversal**.
 - **Post-order Traversal**.
 - **Level-order Traversal**.
-

Conclusion

This lab enables students to practice the foundational concepts of binary search trees. By implementing insertions, searching, counting nodes, and performing traversals, students will develop a deeper understanding of BST operations and their applications in problem-solving.