VIRTUAL LABS ASSIGNMENT

Computer Science & Engineering

Experiment: Binary Search Tree (BST)

Course: __________________________

Institute: ________________________

Student Name: _____________________

Roll No: __________________________

Date: August 17, 2025

Platform: Virtual Labs (IIIT Hyderabad – Data Structures I)

Experiment link inside report & README

Objective

Demonstrate core Binary Search Tree (BST) operations using Virtual Labs: Insertio
Search, and Deletion. Observe how the BST property maintains an ordered struct
supports efficient lookup.

Background (Theory - brief)

A BST is a binary tree in which each node’s key is greater than all keys in its left subtree and less than all keys in its right subtree. The average-case time for search/insert/delete is O(log n) given balanced structure. This assignment demon
these operations visually.

References:
- Virtual Labs – Binary Search Tree (IIIT Hyderabad)
  https://ds1-iiith.vlabs.ac.in/exp/binary-search-trees/index.html

Procedure (What you will do on vlab.co.in)

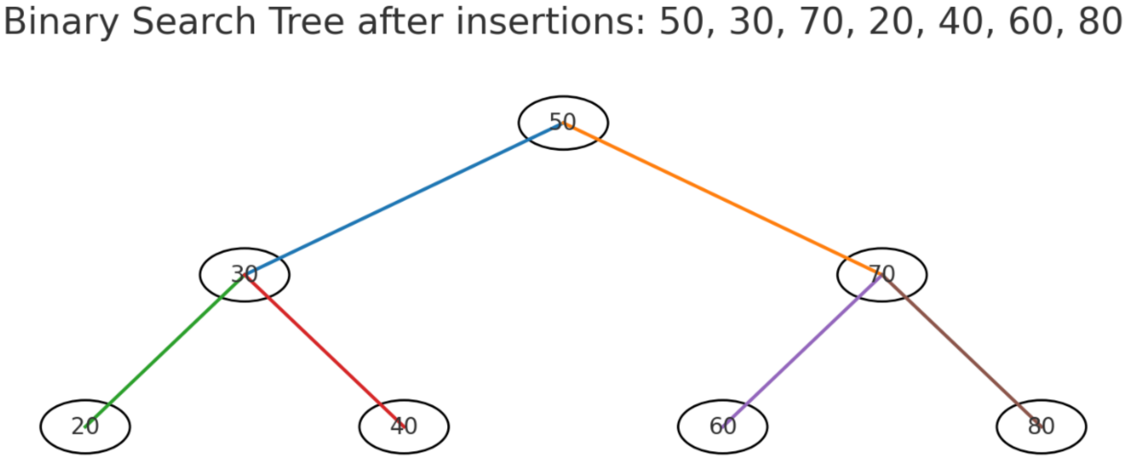
1) Open the experiment entry page and read Aim/Concept.
2) Go to BST-Insert → Demo and insert the sequence: 50, 30, 70, 20,
3) Observe the visual tree after each insertion.
4) Take a screenshot of the final tree (Win: Win+Shift+S, macOS: Shi
5) Optionally, test Search (e.g., look up 60 and 35) and Delete (e.g
6) Save the screenshot as images/bst_result.png and include it in th

Experiment links:
• Entry: https://ds1-iiith.vlabs.ac.in/exp/binary-search-trees/index.html
• Insert: https://ds1-iiith.vlabs.ac.in/exp/binary-search-trees/bst-insert/bstInsert.html
• Search: https://ds1-iiith.vlabs.ac.in/exp/binary-search-trees/bst-search/bstSearch.html
• Delete: https://ds1-iiith.vlabs.ac.in/exp/binary-search-trees/bst-delete/bstDelete.html

Observations:

- Newly inserted keys appear as leaves, preserving the BST ordering.
- Searching descends left/right depending on comparison with the current node.
- Deletion cases vary (leaf, one-child, two-children replacement by inorder successor).

Figure: Sample BST after the insertion sequence 50, 30, 70, 20, 40, 60, 80 (replace with your actual Visual Labs screenshot).


Binary Search Tree after insertions: 50, 30, 70, 20, 40, 60, 80
50
30
70
20
40
60
80

Viva-Voce (Sample)

Q1: What is the BST property?
A1: For any node, keys in the left subtree are smaller; keys in the

Q2: What are the average and worst-case time complexities of search
A2: Average O(log n) (balanced); worst-case O(n) (skewed).

Q3: How do you delete a node with two children?
A3: Replace it with its inorder successor (or predecessor), then del

Conclusion: The Virtual Labs BST modules illustrate how insert/search/delete mai
ordering, affecting search performance. Visualization helps connect the algorithm
structure.