

1. Introduction to BST : Binary Search Tree

- a. The left subtree of a node contains only nodes with value lesser than the node's value.
- b. The right subtree of a node contains only nodes with value greater than the node's value.
- c. Equal value must be on the left or right, any one side only. **Not two both side.**
- d. The left and right subtree each must also be a binary search tree.

2. BST Insertion: Theory, Tree will vary depending on input order. Input Order Matter here.

Time Complexity : $O(h)$, **height == n in worst case so $O(n)$**

3. BST Insertion : Implementation

4. BST Searching : $O(\text{height})$ height == n in worst case so $O(\text{height})/O(n)$

5. BST Deletion: $O(\text{height})$ height == n in worst case so $O(\text{height})/O(n)$

a. CASE 1 = No child

b. CASE 2 = One child

c. CASE 2 = Two child [By Inorder Successor]

Immediate Greater Value in its left subtree

All of these Function's Time Complexity will be :

- Normal Case : $O(\text{height})$
- Average Case : $O(\log n)$
- Worst Case : $O(n)$