What are the worst case time complexities of searching in binary tree, BST and AVL tree respectively?  
(A) O(n) for all  
(B) O(Logn) for all  
(C) O(n) for binary tree, and O(Logn) for others  
(D) O(n) for binary tree and BST, and O(Logn) for AVL

**Solution:** As discussed, search operation in binary tree and BST have worst case time complexity of O(n). However, AVL tree has worst case time complexity of O(logn). So, the correct option is (D).

The binary search tree provides us with some interesting time complexities.

For searching, For inserting and finally for deletion, we have to traverse all elements, worst-case complexity of O(n)

Which of the following traversals is sufficient to construct BST from given traversals

1) Inorder 2) Preorder 3) Postorder

Ans: 2 or 3 is sufficient

Consider the following Binary Search Tree

10

/

5 20

/ /

4 15 30

/

11

If we randomly search one of the keys present in above BST, what would be the expected number of comparisons?

Ans: Expected number of comparisons = (1\*1 + 2\*2 + 3\*3 + 4\*1)/7 = 18/7 = 2.57