[redacted]

Binary Search Tree :

BST is a binary tree in which every node contains smaller values only in its left subtree and only larger in its right subtree.

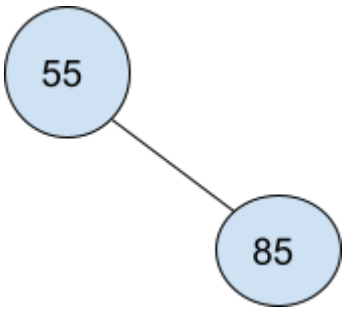
Given Array 55, **85**, 27, 67, 7, 39, 83, 73, 86, 31, 32 [redacted]

Part A) Construction :

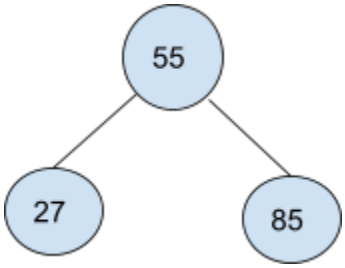
Given an array, we will consider the first index as the root node.

Insert 55 :

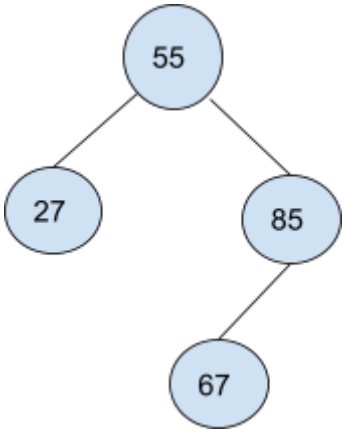
Insert 85 :
85 is greater than root 55 so we have to add it to right side.



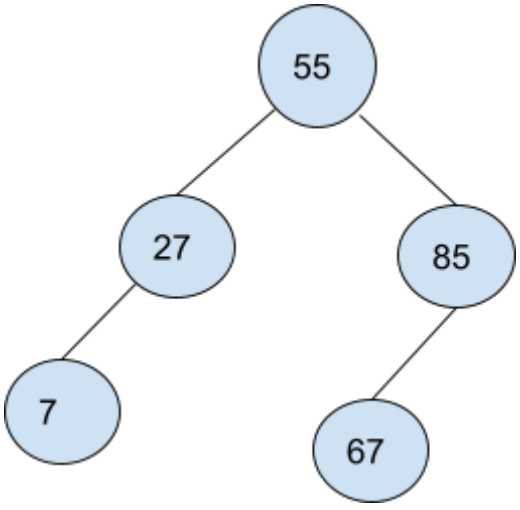
Insert 27 :
27 is less than 55 so we will add it root's left side.



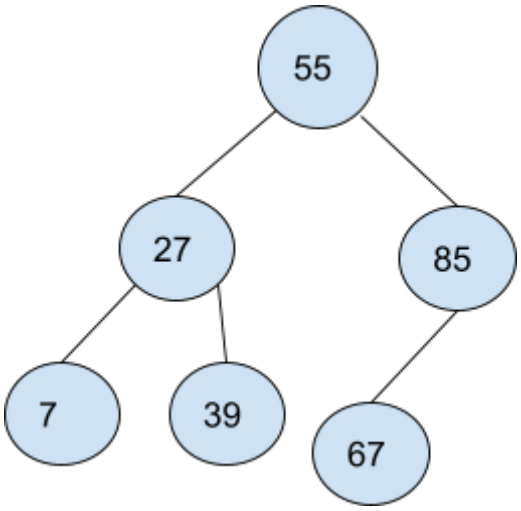
Insert 67 :
67 > 55 so it goes to right side.
Again 67 will be checked with 85 it's 67<85 so it will go to left child of 85.



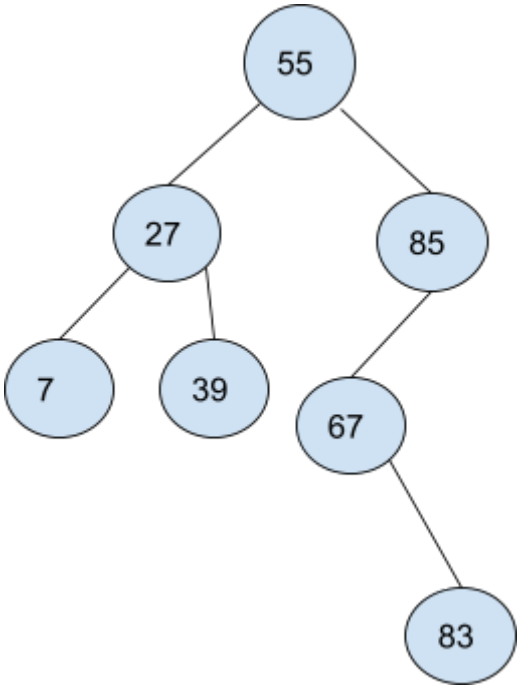
Insert 7 :
7<55 so it goes left child of 55
7<27 so it goes again left child of 27



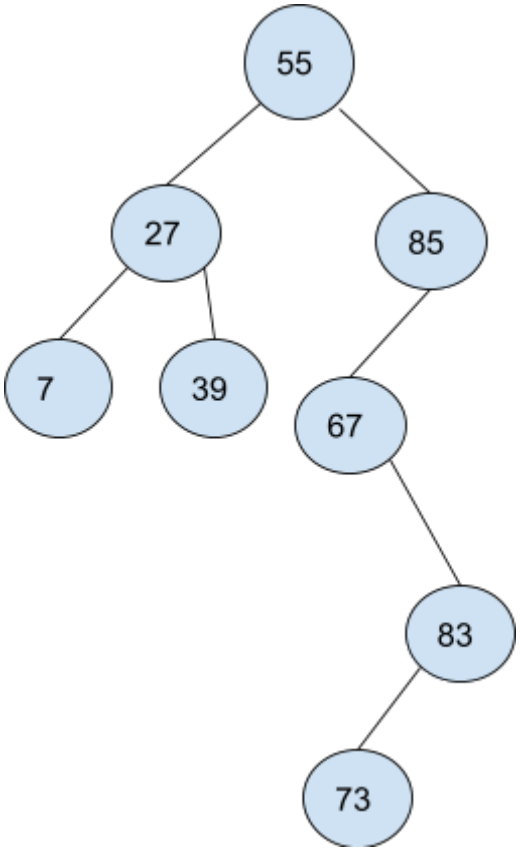
Insert 39 :
39<55 so it goes to left child of 55
39>27 so it goes right child of 27



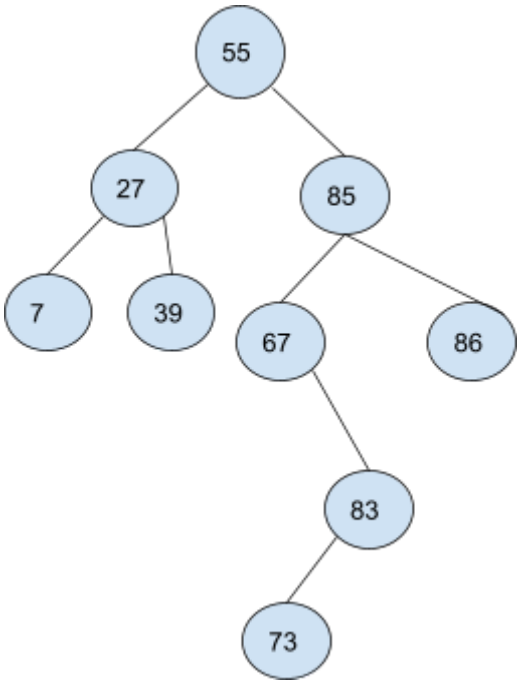
Insert 83 :
83>55 so it goes to right child of 55
83<85 so it goes to left child of 85
83>67 ao it goes to right child of 67



Insert 73 :
73>55 so it goes to right child of 55
73<85 so it goes to left child of 85
73>67 so it goes to right child of 67
73<83 so it goes to left child of 83

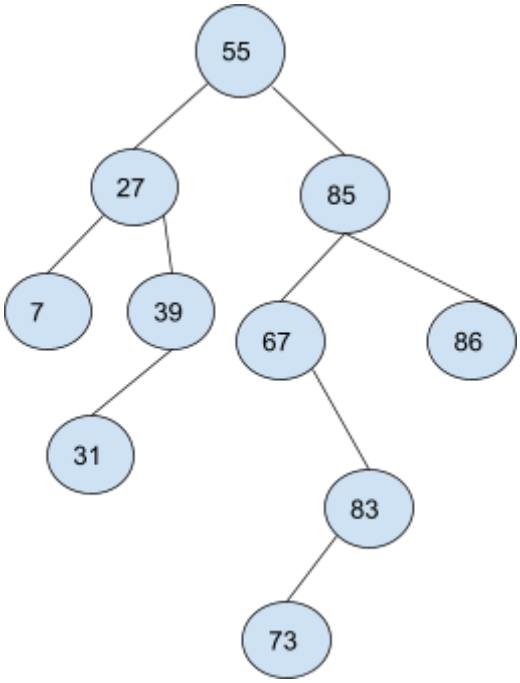


Insert 85 :
85>55 so it goes to right child of 55
86>85 so it goes to right child of 85
After that no more nodes are there so we have to insert.

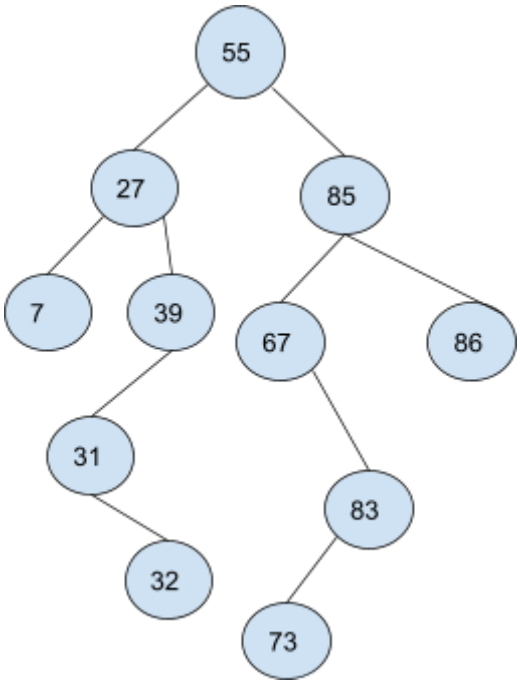


Insert 31 :
31<55 so it goes to left child of 55
31>27 so it goes to right child of 27
31<39 so it goes to the left child of 39.

After that no more nodes are there so we have to insert.



Insert 32 :
32<55 so it goes to left child of 55
32>27 so it goes to right child of 27
32<39 so it goes to the left child of 39.
32>31 so it goes to the right child of 31



Finally, BST has been constructed.

Preorder traversal :
In this traversal method, the root node is visited first, then the left subtree and finally the right subtree.
We start from 55, and following pre-order traversal, we first visit 55 itself and then move to its left subtree 27. 27 is also traversed pre-order. The process goes on until all the nodes are visited. The output of pre-order traversal of this tree will be
56,27,7,39,31,32,85,67,83,73,86

InOrder traversal :
This will give us sorted array as we are checking the last child in each sub tree.
the left subtree is visited first, then the root and later the right sub-tree.

7,27,31,32,39,55,64,73,83,85,86

Postorder traversal :
In this traversal method, the root node is visited last, hence the name. First we traverse the left subtree, then the right subtree and finally the root node.

We start from 55, and following Post-order traversal, we first visit the left subtree 27. 27 is also traversed post-order. The process goes on until all the nodes are visited. The output of post-order traversal of this tree will be

7,32,31,39,27,73,83,67,86,85,55

BFS :
BFS uses queue to maintain search operation
BFS maintain two categorsies :
Visited and Not visited

DFS :
DFS uses stack to maintain search operation

B)

Binary search tree performance :
If we want to find 32 and 73 from Binary search tree, Worst case for binary tree would be o(n) as number of comparisons needed to find the node marked is 5