WEEK-7	BINARY AND BINARY SEARCH TREE
LAB A	19-24 SEPTEMBER

1. You have been given a list of characters as {A, B, C, D, E, F, G, H, and I). Write a program to insert these elements into a complete binary tree. Also, write the code to perform all the functionalities on the constructed binary tree,

a. To find a node with specified value

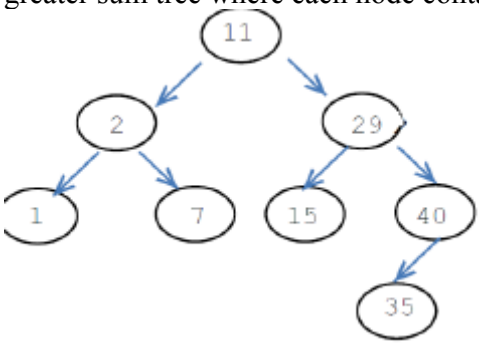
b. To compute height of a given node

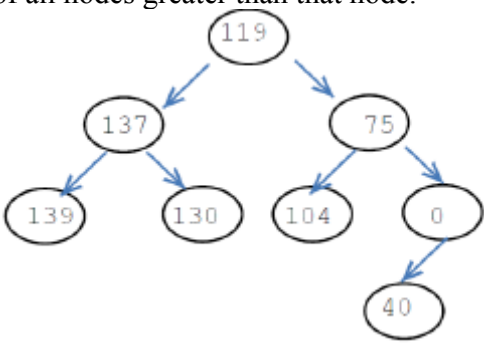
c. To compute the depth of a node

d. To check whether the given tree is complete binary tree or not

2. Write a program to construct a binary tree from its level order and in-order traversal.

3. Given a binary tree and inorder traversal, construct a new binary tree with additional pointers such that the inorder traversal is reflected in the new tree (Hint: Threaded binary tree). **Q2.** Given a BST, transform it into greater sum tree where each node contains sum of all nodes greater than that node.





4. Construct a binary tree from its level order and in-order traversal.

5. In a binary search tree, write a function to checks if the sum of all the nodes in left sub tree of the node is equal or greater or lesser than the sum of all the nodes in right sub tree of the node.

LAB B	AVL TREE
	19-24 SEPTEMBER

Q1. Write functions to

(a) Balance a given BST to form an AVL Tree.
(b) Find the cost of balancing a given AVL Tree.
(c) Traverse in Dept first and Level-Order fashion.
(d) Display the height of the AVL tree.
(e) Search for a particular element in a given AVL tree and display its height.
(f) Delete a particular value.
(g) Delete a subtree starting with a particular node value.
(h) Delete all nodes at a given height.
(i) Display values in a subtree.
(j) Write a function to merge the two given AVL Trees into a single AVL Tree.
(k) Consider, E as an element stored in a node N of an AVL tree, A. Considering an update operation which updates the element E by ±Δ.
With updated value as E±Δ at node N, A may not be a valid AVL tree. Call the updated tree as A'.
Write a program to make the tree A' as a valid AVL tree.