

Part – C

AVL Tree data structure is a self-balancing Binary Tree. It is a type of binary tree in which the left node is less than the parent node and the right node is greater than the parent node. Which makes it easy to store data sorted in the tree. After every insertion or deletion, it checks for the balance in the tree by subtracting heights of the two children of the subtrees. And then it does the right rotate or left rotate accordingly. It might also do double rotations depending on the balance factors.

AVL trees is a type of binary tree as it stores the data sorted but only thing makes it different is the self-balancing. For a binary tree, insertion and deletion is fast because it does not check for the balancing factor every time it inserts or deletes a node. But search is slow in binary tree as it is not balanced. Whereas, as compared to an AVL tree, insertion and deletion is slower than a binary tree because of the rotations. But search is much faster because it is balanced.

Big-Oh-

Insertion =  $O(\log n)$

Deletion =  $O(\log n)$

Search =  $O(\log n)$