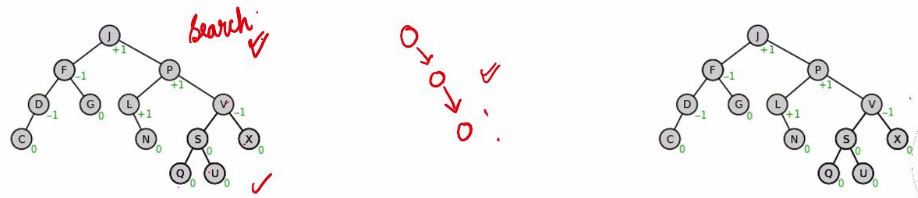
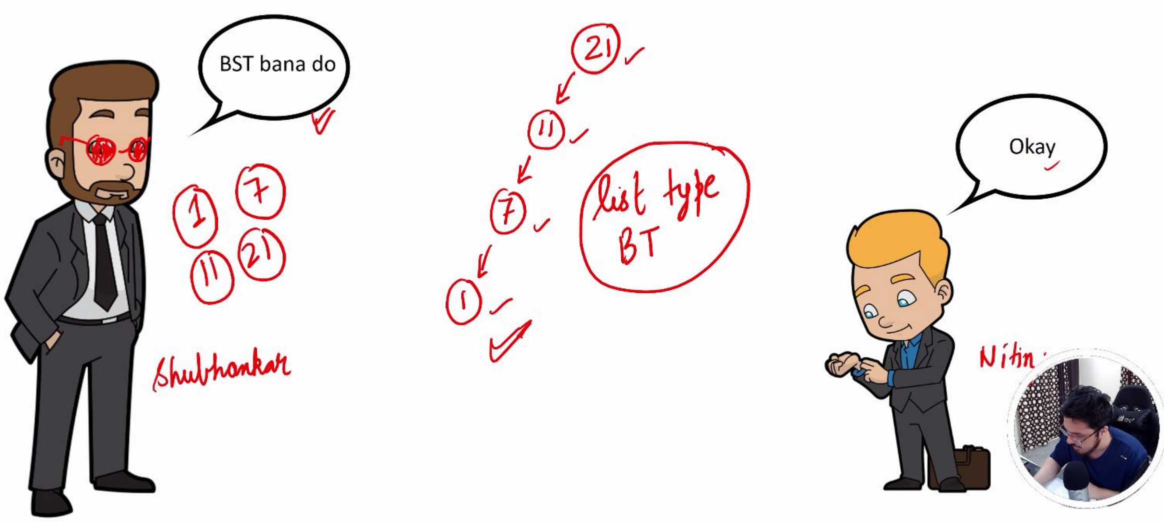
AVL Trees

By CodeWithHarry

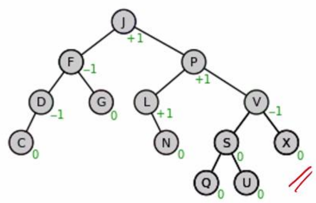


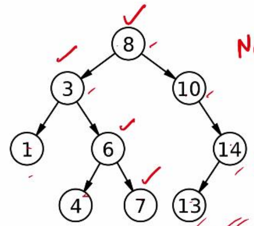
Why AVL Trees? (Cheating Employee Example)



Why do we need an AVL Tree?

- Almost all the operations in a binary search tree are of order O(h) where h is the height of the tree.
- If we don't plan our tree properly, this height can get as high as n where n is the number of nodes in a BST (skewed tree)
- To guarantee an upper bound of O(log n) for all these operations, we use balanced trees





NOC = 4

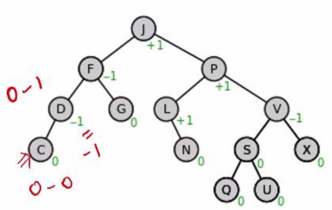
What is an AVL Tree?

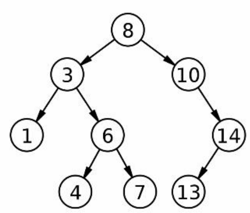
- Height balanced binary search trees
- Hieght difference between heights of left and right subtrees for every node is less than or equal to 1.
- Balanced factor = Height of right subtree – Height of left subtree
- Can be -1, 0 or 1 for a node to be balanced in a Binary search tree
- Can be -1, 0 or 1 for all nodes of an AVL tree

BF = Balanced factor

For a tree to be balanced

$|BF| \leq 1$





What is an AVL Tree?

- Height balanced binary search trees ✓
- Hieght difference between heights of left and right subtrees for every node is less than or equal to 1.
- Balanced factor = Height of right subtree – Height of left subtree ✓
- Can be -1, 0 or 1 for a node to be balanced in a Binary search tree
- Can be -1, 0 or 1 for all nodes of an AVL tree

BF = Balanced factor

For a tree to be balanced

$|BF| \leq 1$

Rotations