UT-II

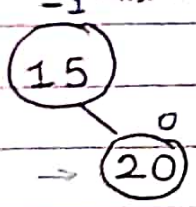
Time: 20m. DSA - Question Bank

1) Create AVL Tree (sequential sel) Show all steps and rotations
15, 20, 24, 10, 13, 7, 30, 36, 25

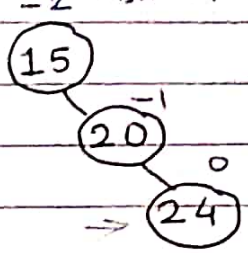
→ AVL Tree is called height-balanced tree was defined by mathematicians (Adelson, Velskii, Landis)
It is a self balancing Binary Search Tree (BST) where difference btw (height of left and Right subtree) is either 0, -1, 1.

Step1: Insert 15

Step2: Insert 20

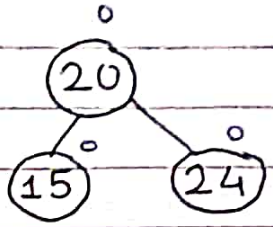


Step3: Insert 24

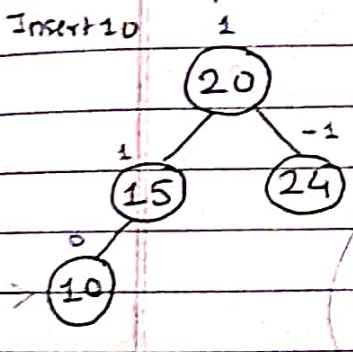


Step4:

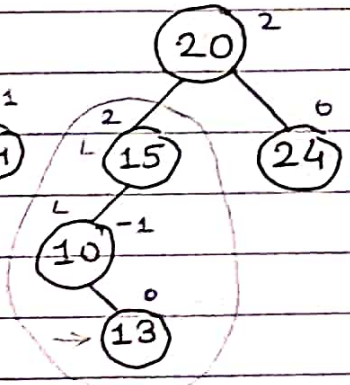
RR rotation anti ↺



Step5: Insert 10



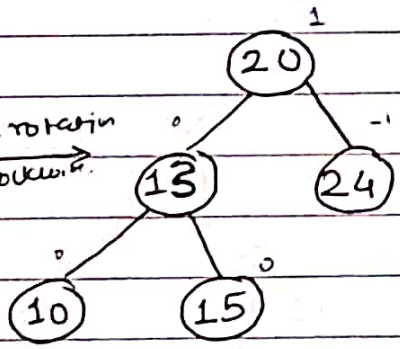
Step6: Insert 13



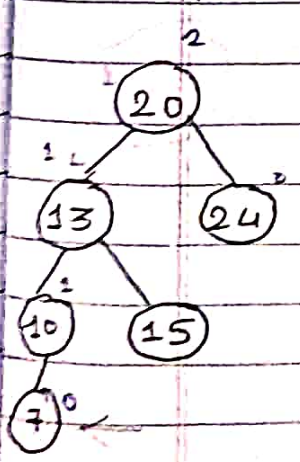
Imbalance

Step7:

LL rotation ↻

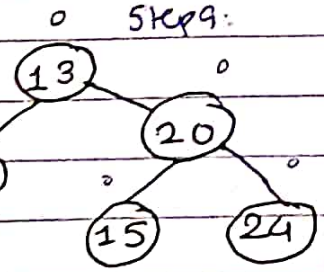


Step8: Insert 7:

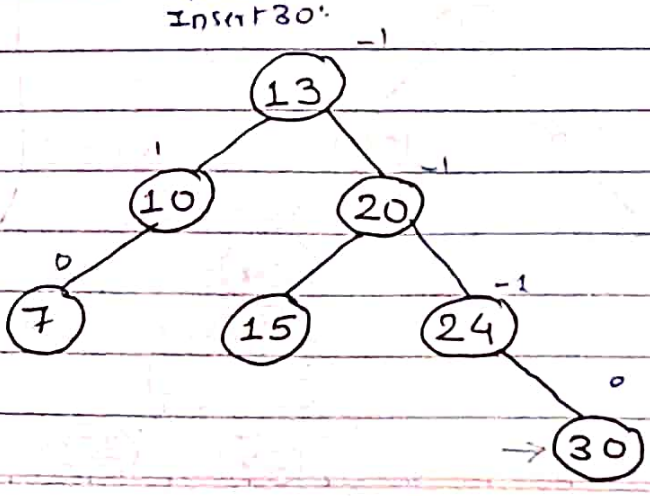


LL rotation ↻

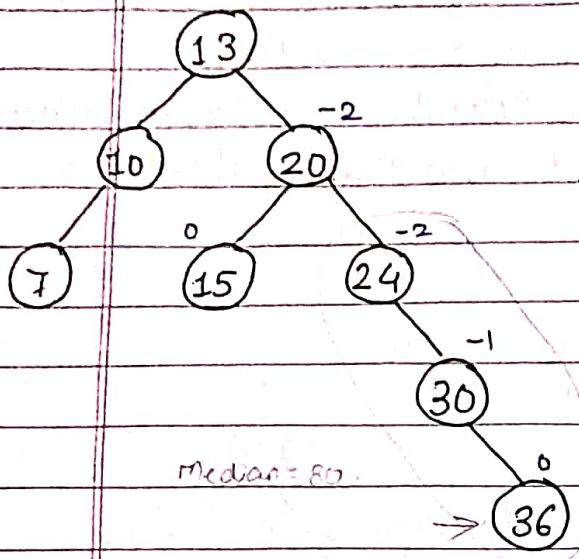
Step9:



Step10: Insert 30:

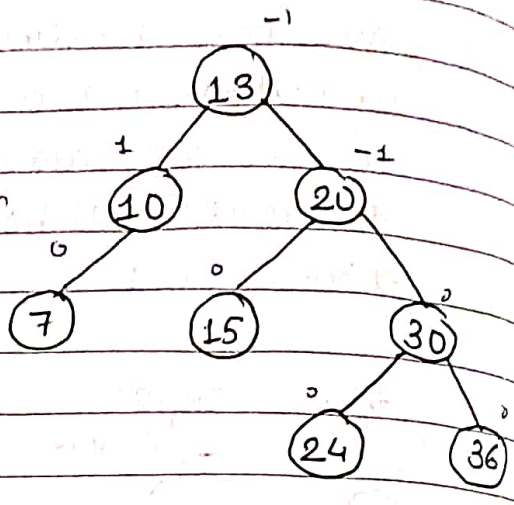


Step 11:
Insert 36.

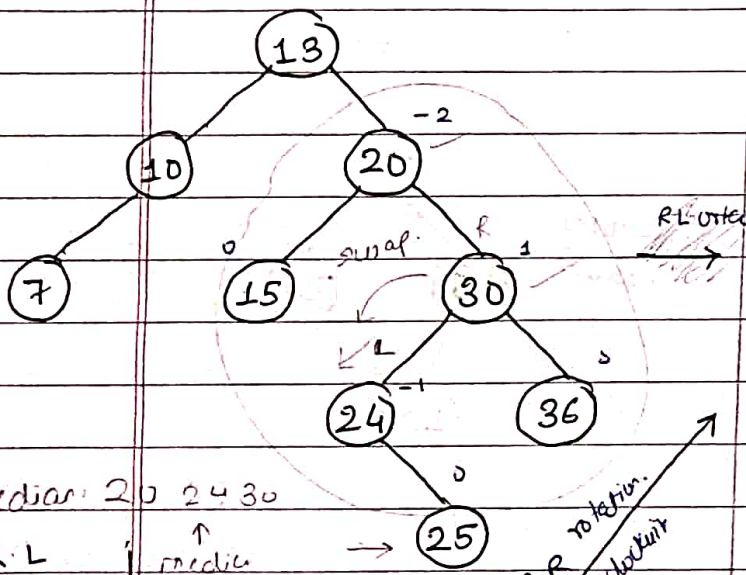


Step 12:

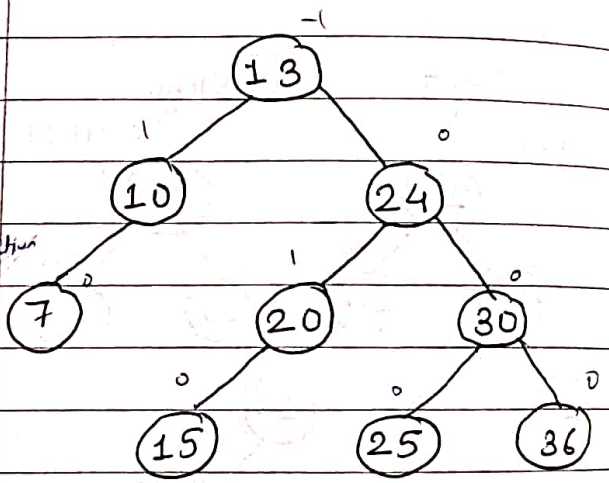
RR rotation
anticlockwise



Step 13:
Insert 25.

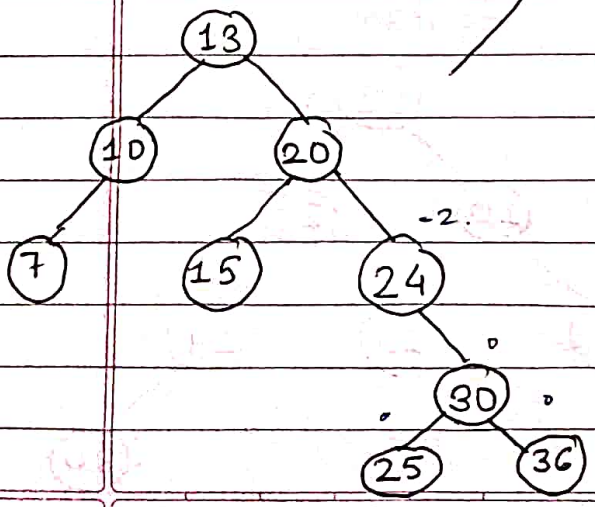


RL rotation



Step 14:

RR rotation.
anticlockwise



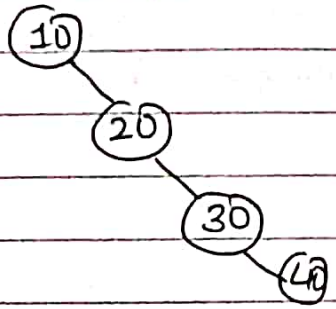
Final Ans.

2) AVL Tree differ from BST? Show result of inserting 15, 19, 22, 10, 3, 37, 25, 12, 13 one at a time into an initially empty AVL Tree.

→ Binary Search is a tree data structure that follows the condition of the binary tree. That is, Each node in a binary search tree should have the utmost two child nodes. It arranges its node elements in sorted manner. A Binary tree is referred as a binary search tree if for any node n in the tree:

1. The node elements in the left subtree of n are lesser in value than n.
2. The node elements in the right subtree of n are greater than or equal to n.

Consider following example: 10 20 30 40



Right Skewed BST.

It is observed that BST's worst case performance is closest to linear search algorithm / Linked List, that is O(n). In real time data, we cannot predict data pattern and their frequencies. So a need arises to balance BST.

Therefore we use AVL Tree.

- AVL Tree is a height balancing Tree or self balancing ~~Tree~~ binary search tree where difference between height of Left Subtree and height of Right Subtree is either 0, -1 or 1. This difference is called as Balance Factor (BF)
- In AVL Tree the height of tree is O(log n) n = no. of nodes.

Searching is efficient in AVL Tree when there are large number of nodes in tree because tree's height is balanced

To Balance the AVL Tree we perform some rotations:

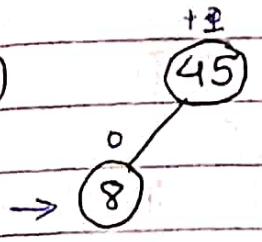
1) LL 2) RR 3) LR 4) RL.

Step 11

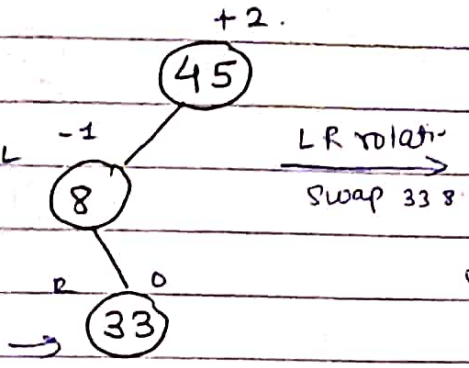
3) AVL Tree 45 8 33 85 61 10 48 76 57 99

Step 1: Insert 45

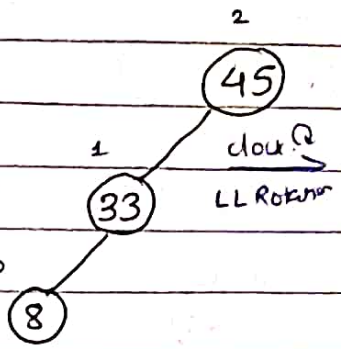
Step 2: Insert 8



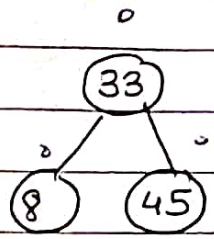
Step 3: Insert 33



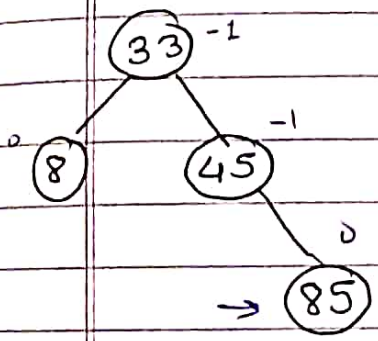
Step 4:



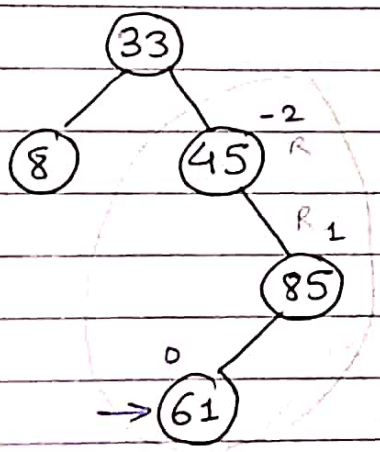
Step 5:



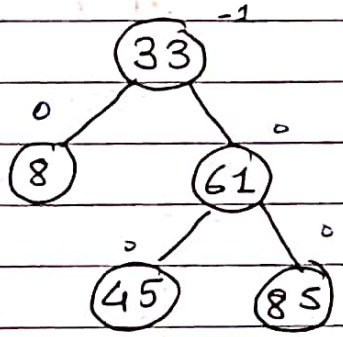
Step 6: Insert 85



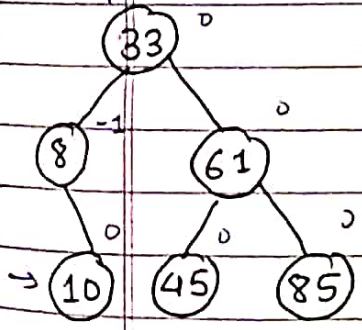
Step 7: Insert 61



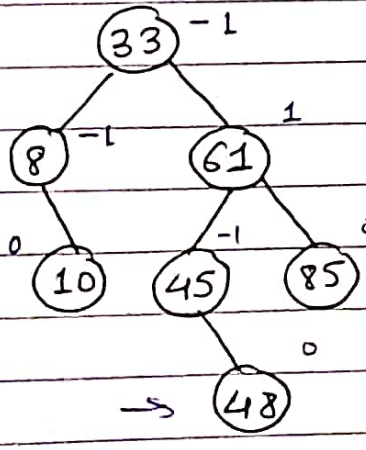
Step 8:



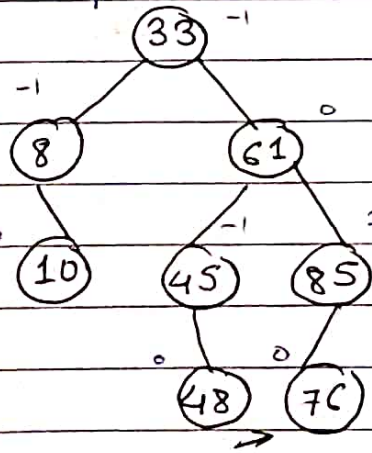
Step 9: Insert 10.



Step 10: Insert 48

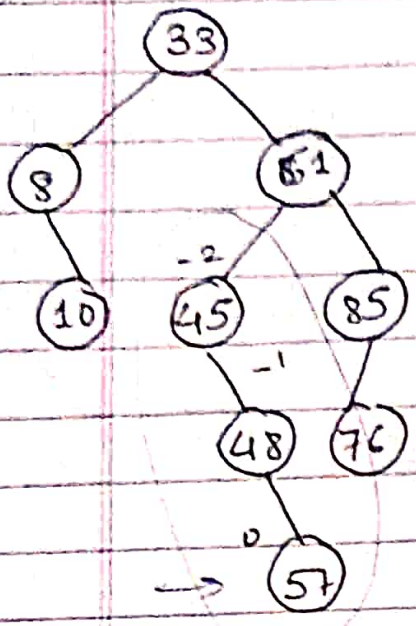


Step 11: Insert 76



Step 12:

Insert 57

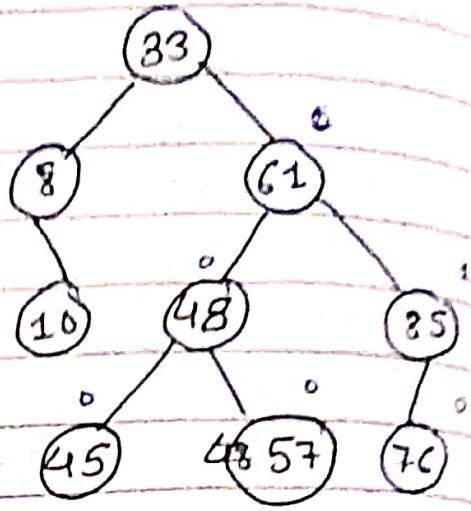


RR rotation

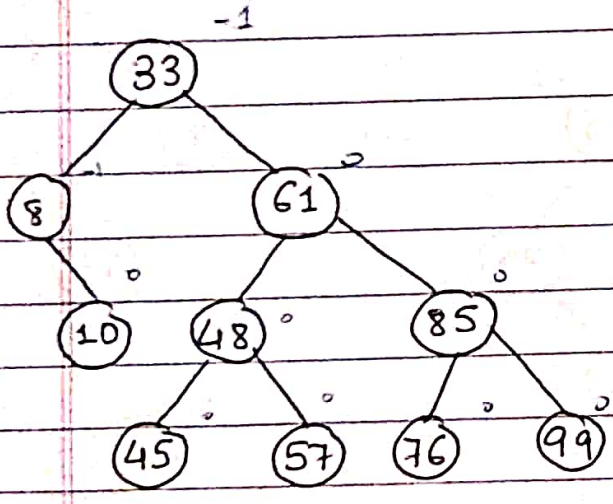
anticlockwise

Mean: 48

Step 13:

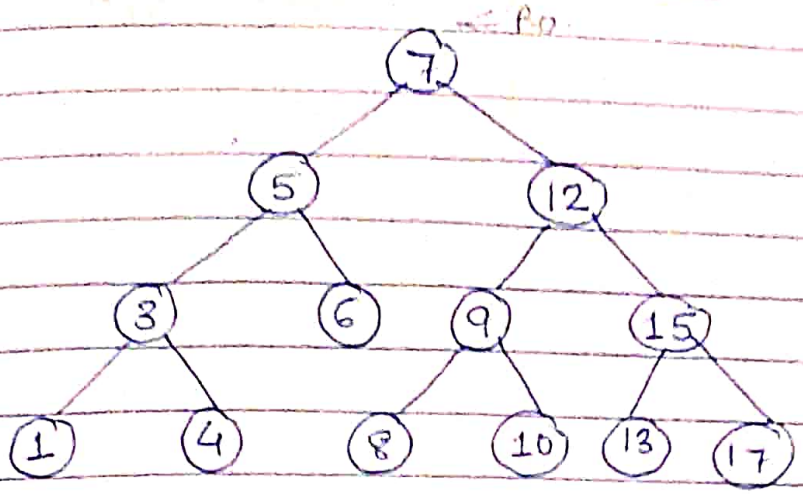


Step 14: Insert 99



4, 5, 6

7. BST find post order traversal sequence.



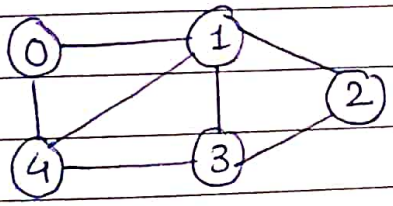
Solution: Post order:

Traverse Left subtree (L), Traverse Right (R), Traverse Rootnode (RO)

1 - 4 - 3 - 6 - 5 - 8 - 10 - 9 - 13 - 17 - 15 - 12 - 7

8. Breadth-first-Traversal of graph starting vertex 0.

Assignment.



9. Depth-First-Search

Assignment.