Task 1:

Write Algo for AVL tree

INSERTION STEPS:

1. CREATE NODE

- Make a new box (node)

- Put your number in it

- Mark it as empty on both sides

2. CHECK IF TREE IS EMPTY

- If tree is empty

→ Put new box as the first box (root)

- If not empty

→ Move to step 3

3. INSERT LIKE REGULAR BINARY TREE

- If new number is smaller

→ Go left

- If new number is bigger

→ Go right

- Keep going until you find empty spot

4. CHECK BALANCE

- Count levels on left side

- Count levels on right side

- Subtract: left - right

- If difference bigger than 1

→ Tree needs fixing

5. FIX THE TREE (IF NEEDED)

- If left side too heavy

→ Rotate right

- If right side too heavy

→ Rotate left

SEARCH STEPS:

1. START AT TOP

- Begin at first box (root)

2. COMPARE NUMBERS

- Is it the number you want?

→ You found it!

3. CHOOSE DIRECTION

- If your number is smaller

→ Go left

- If your number is bigger

→ Go right

4. KEEP GOING

- Repeat steps 2-3 until you:

→ Find the number, or

→ Reach empty spot (number not found)

5. GIVE RESULT

- Tell if you found the number

- Or say it's not in the tree

Task 2:

Write code for AVL tree

Hint: try to insert nodes

While inserting get the balance of the tree

Create 2 methods for left rotate and right rotate

Try to insert

Finally display