Task01:

**Step 1:** Start from the root and perform **standard BST insertion** for the key.

**Step 2:** After insertion, go back up the recursion stack and **update the height** of each ancestor node.

**Step 3:** For each node, **check balance factor**:

* balance = height(left subtree) - height(right subtree)

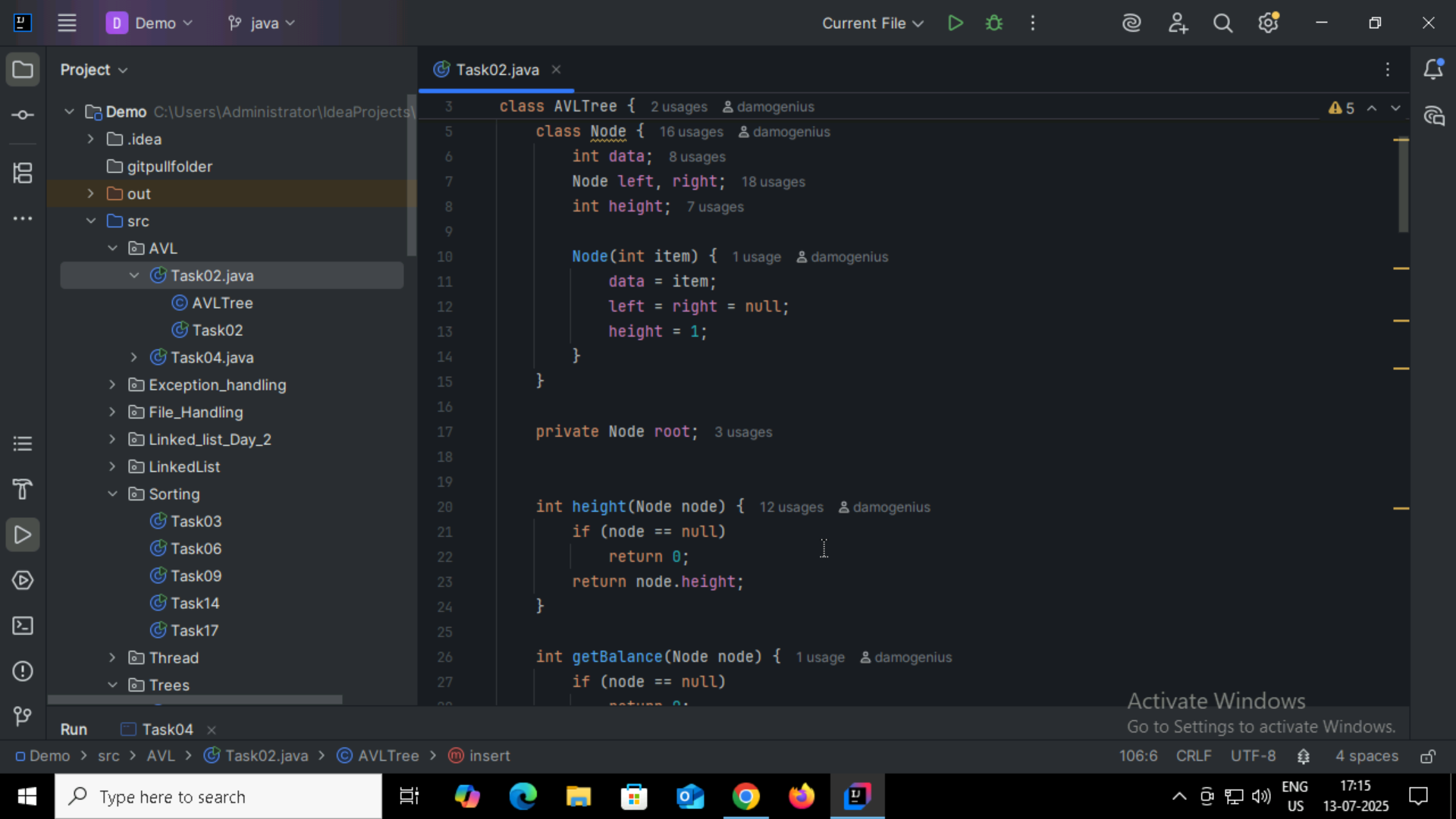
**Step 4:** If the balance factor is **outside the range [-1, 1]**, perform rotations:

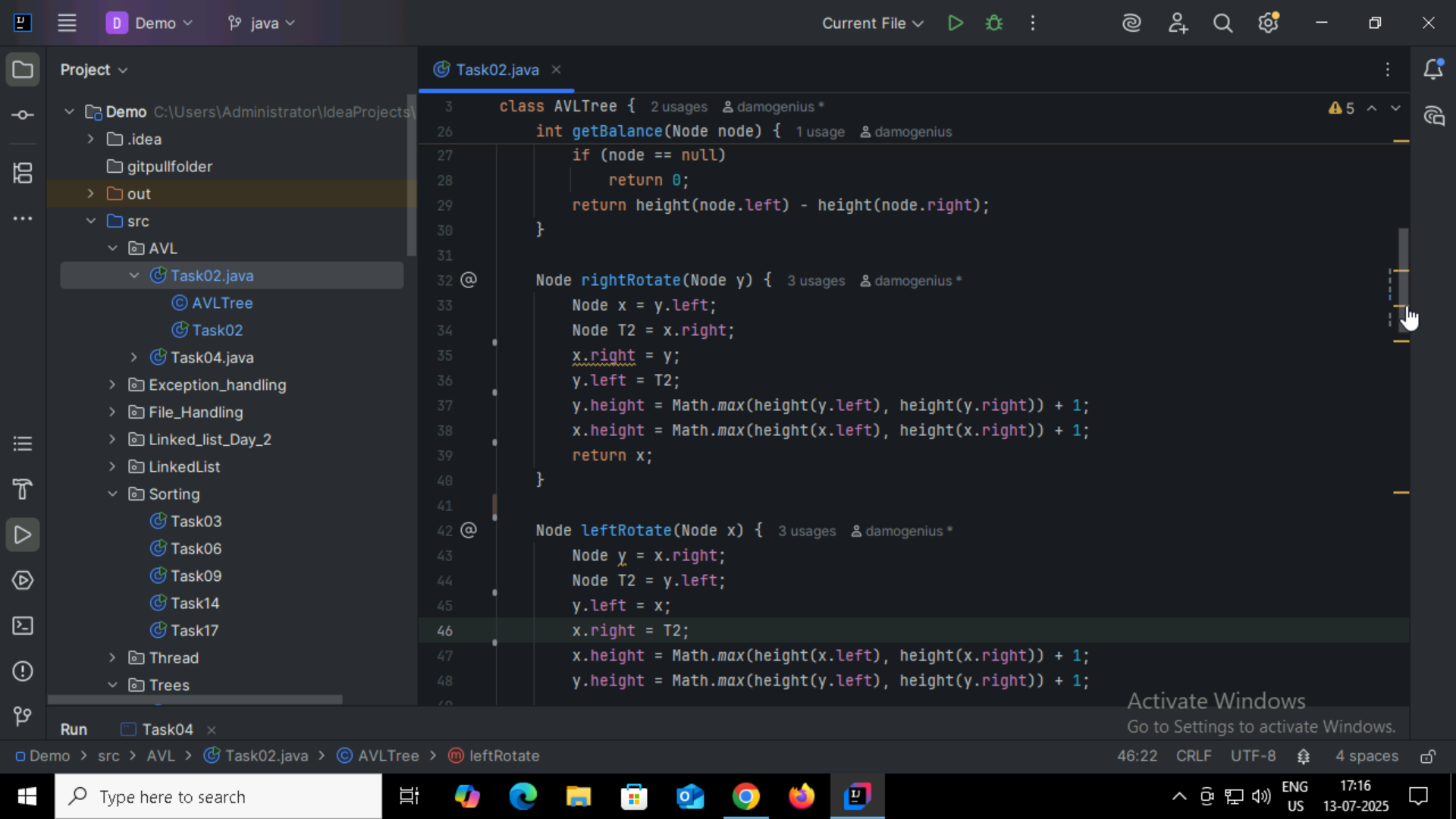
| **Case Type** | **Condition** | **Fix with** |
| --- | --- | --- |
| Left-Left (LL) | Inserted in left subtree of left child | Right Rotation |
| Right-Right (RR) | Inserted in right subtree of right child | Left Rotation |
| Left-Right (LR) | Inserted in right subtree of left child | Left → Right Rotation |
| Right-Left (RL) | Inserted in left subtree of right child | Right → Left Rotation |

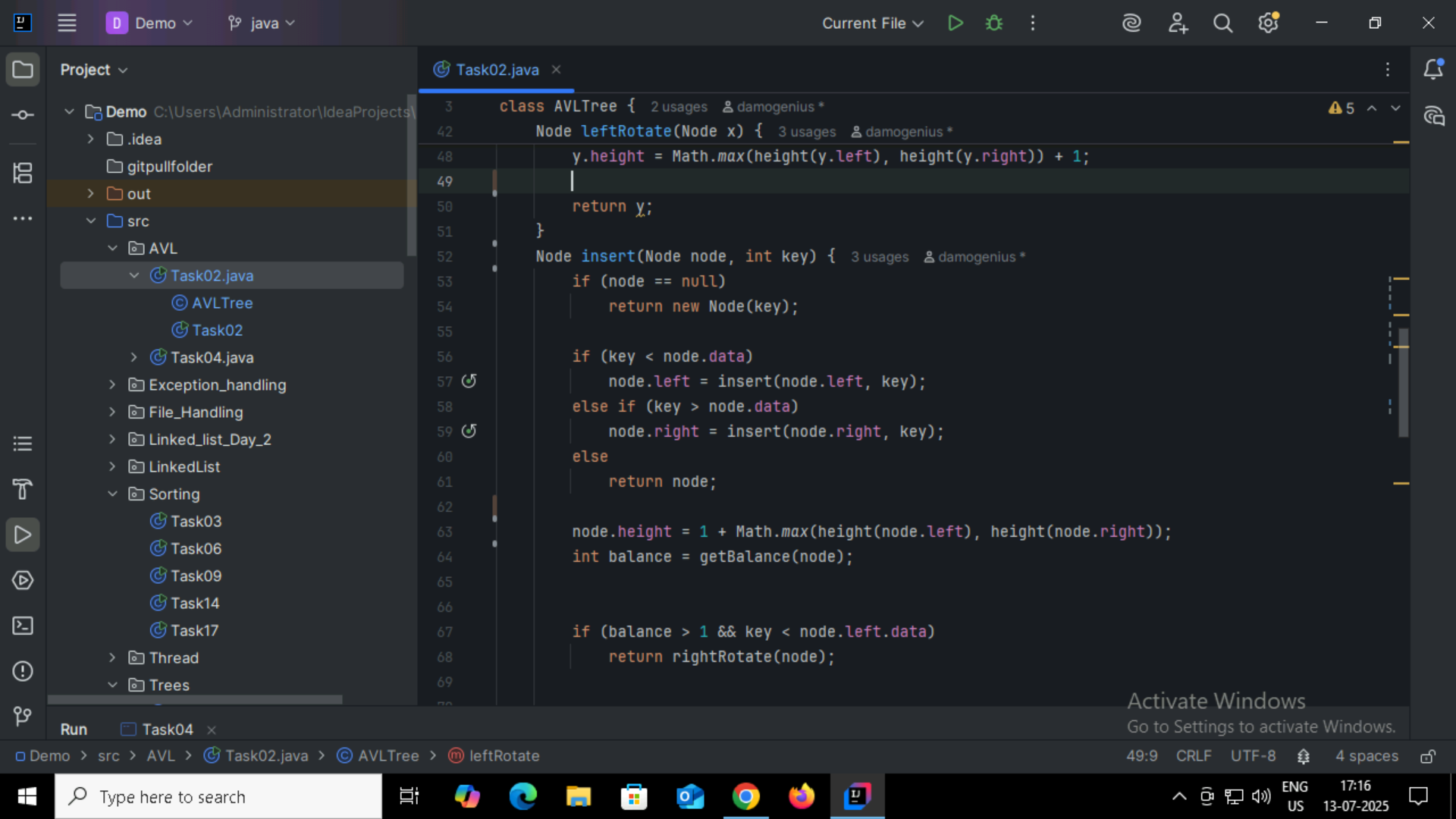
**Step 5:** Return the (possibly new) root of the subtree after rotation.

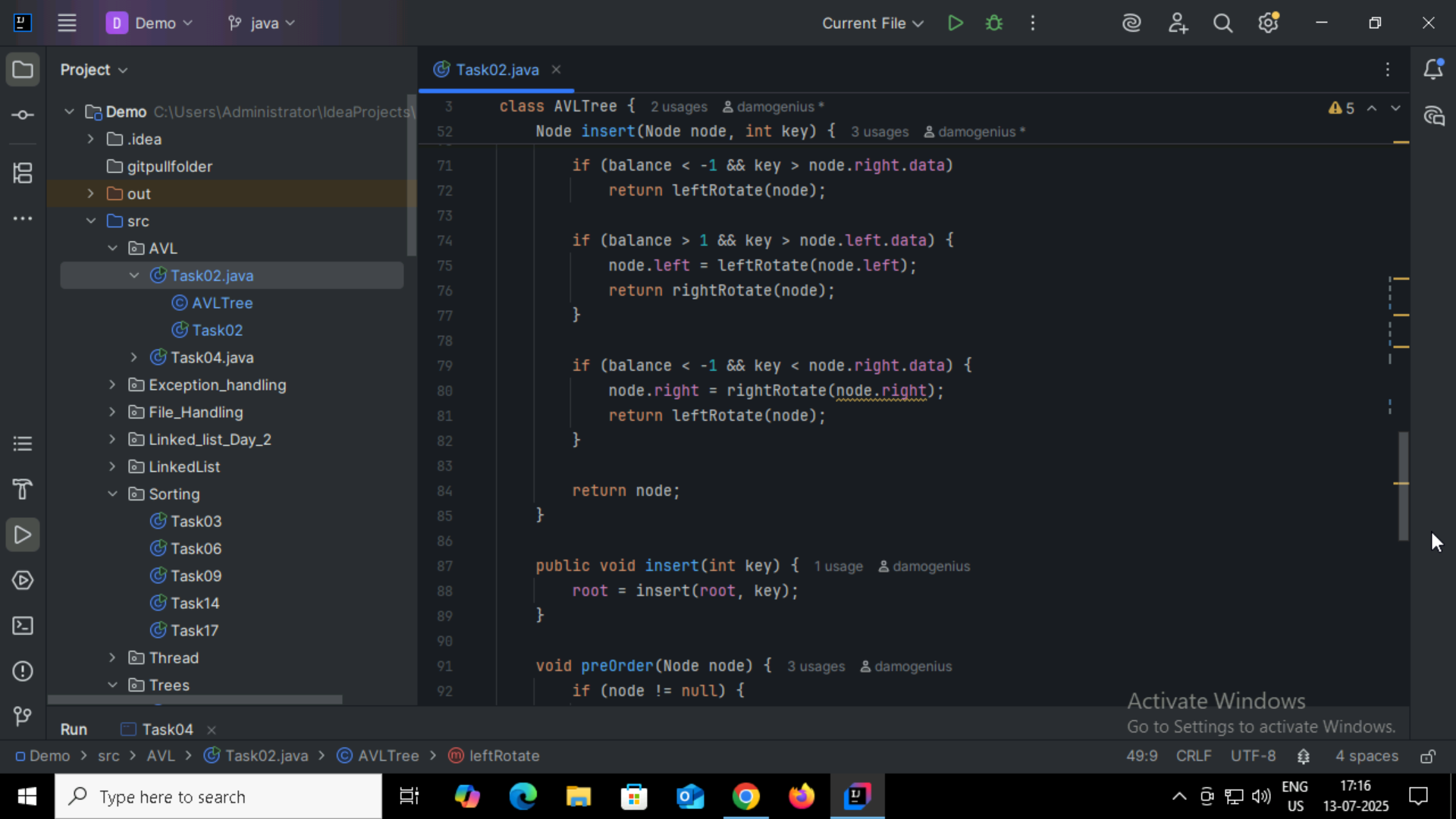
**Step 6:** Repeat until entire tree is rebalanced.

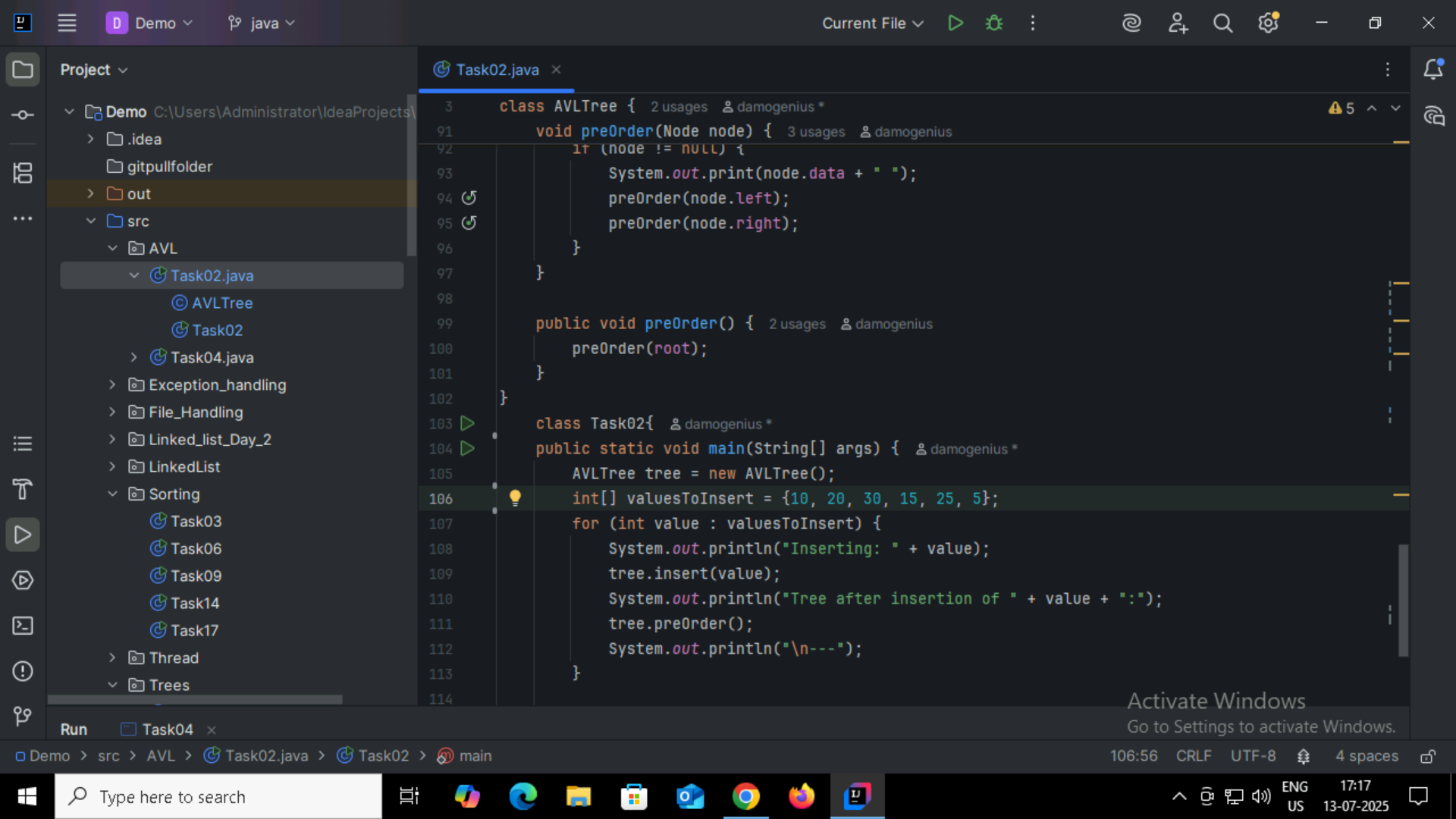
Task02:













Task03:

Insert an Element - Red Black Tree −

1. Check tree is empty. If empty, then insert new node - color Black. (Because Root Node - Black in color)

2. else if Tree - not empty then insert new node as leaf node to the end and color - Red.

3. If parent of new node is Red and its neighbours(parent’s) node is also Red,

then Flip the color of the both neighbour and Parent and Grandparents (If it is not Root Node Otherwise Flip the color of the Parent and neighbour only) i.e., Black.

4. If parent of new node is Red and its neighbours(parent’s) node is empty or NULL,

then Rotate (either Left-Left or Left-Right rotation) the new node and parent.

5. we have two types of rotation

- Left Left Rotation and

- Left Right Rotation.

6. we apply Rotation in some conditions only.

The conditions are −

- If parent of new node is Red and neighbour node is empty or NULL, then rotate left or right rotation.

- In Left-Left Rotation flip the color of the parent and grandparent.

Make the parent as Grandparent and grandparent as child

Task04:

