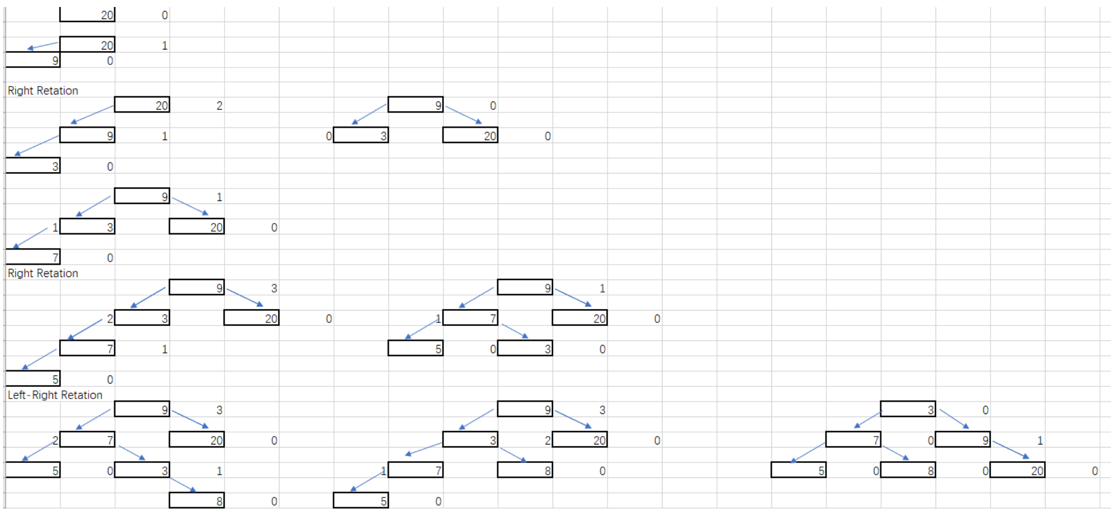
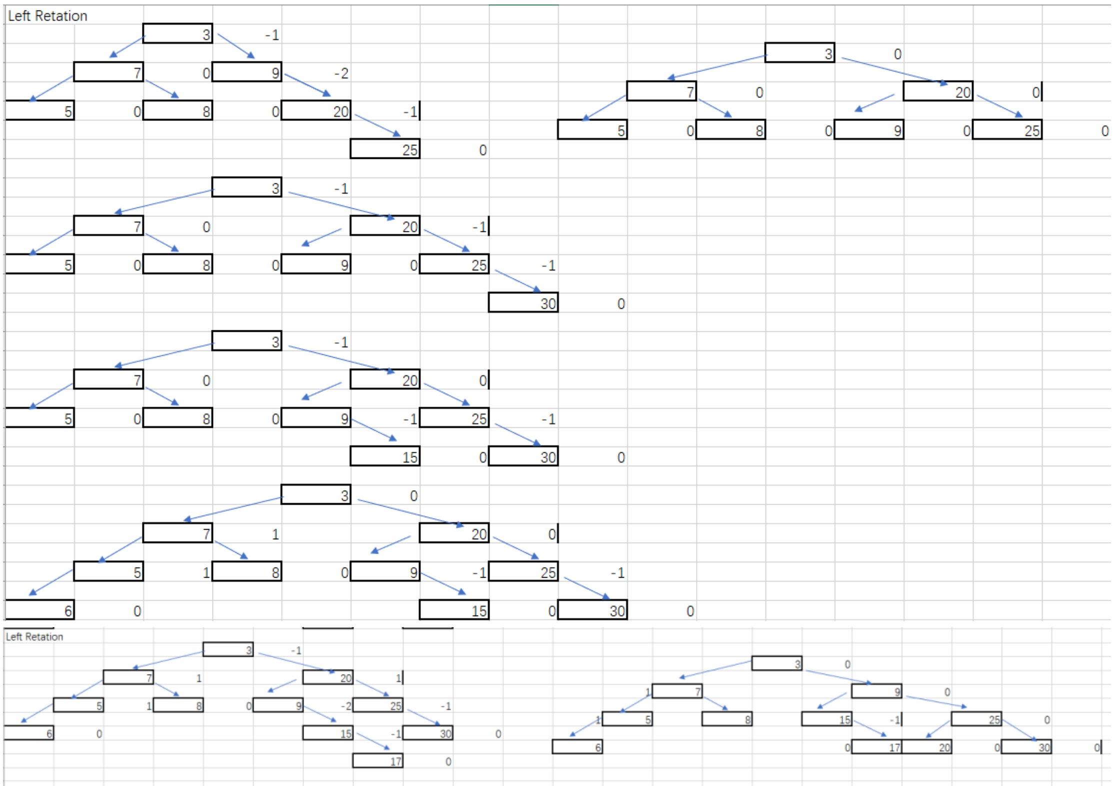
1. Draw a series of figures demonstrating the insertion of the values

20, 9, 3, 7, 5, 8, 25, 30, 15, 6, 17

into an initially *empty* AVL tree. Insert the values in the order they appear in the given sequence. You must:

- Show the resulting AVL tree immediately before and after each insertion step that causes the tree's rebalancing.
- Calculate the ***balance degree*** (as the difference between the heights of the left and the right subtrees rooted at a node) and label each node of the AVL tree before and after the necessary rebalancing.
- Clearly indicate the node(s) at which rotation is performed. Here, let v be the first node you
- encounter in going up from the newly added node, say v, toward the root of the AVL tree T such that v is unbalanced. And let x denote the child of v with greater height (and note that x must be an ancestor of z). Determine whether the subtree of T rooted at v is *right* or *left heavy*. Similarly, specify whether the subtree of T rooted at x is *right* or *left heavy*.
- Performing a *single* or a *double rotation* as the rebalancing (repairing) operation on the AVL tree, specify the type of the rotation that you apply: ***Single Left Rotation***, ***Single Right Rotation***, ***Left-Right Rotation***, or ***Right-Left Rotation***.
- Each time a new value has been inserted, ensure that both the *Binary Search Tree Property* and the *Height-Balance* (*AVL Tree*) *Property* are maintained.



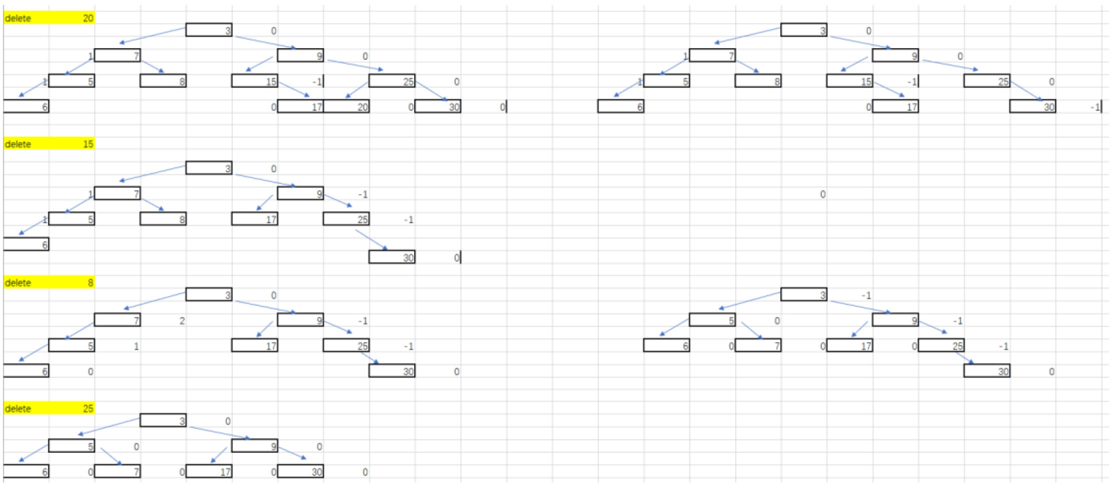


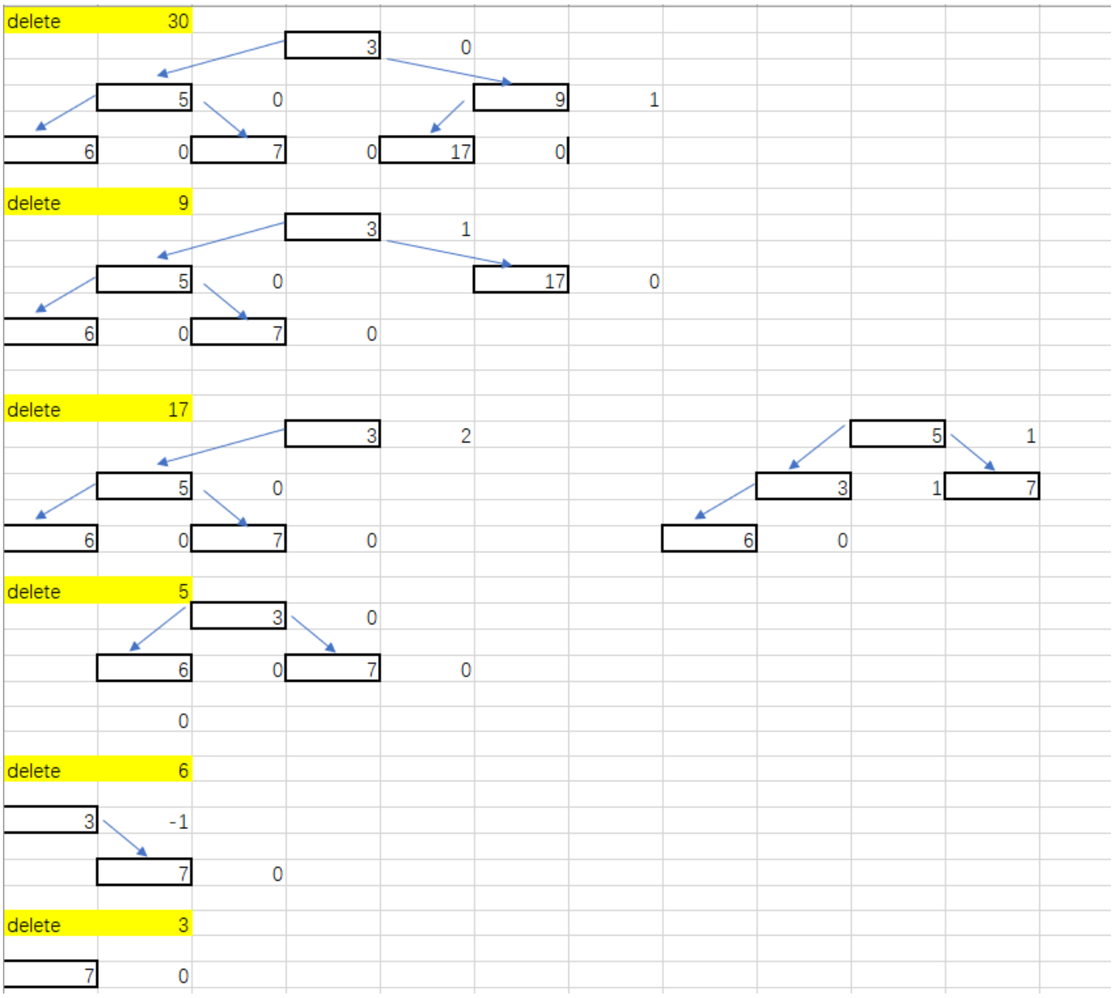
2. Now, draw a series of figures showing the deletion of the values:

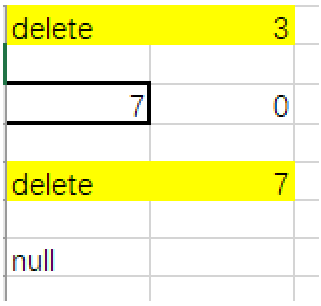
20, 15, 8, 25, 30, 9, 17, 5, 6, 3, 7

from the AVL tree built in the previous part of the task. Delete the values in the order they appear in the given sequence. Draw the AVL tree after each deletion and rotation (if any), and complement each of the figures with all the aforementioned details.

When you delete a node with two children, you must always replace it with the ***largest element smaller than the key of the deleted node*** (as opposed to another possible rule: the smallest element larger than the key of the deleted node). **Following this rule is important** to let us check your solution quickly.







3. What order should we insert the elements {1, 2, ... ,7} into an empty AVL tree so that we do not have to perform any rotations on it?

This order can be {1, 2, 5, 3, 4, 6, 7}, this will not have to perform any rotations