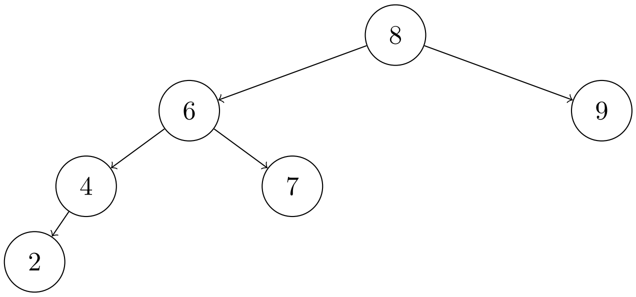
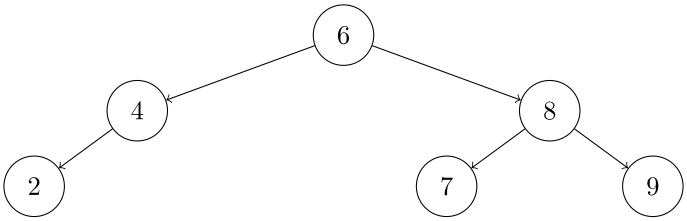
1 AVL Trees

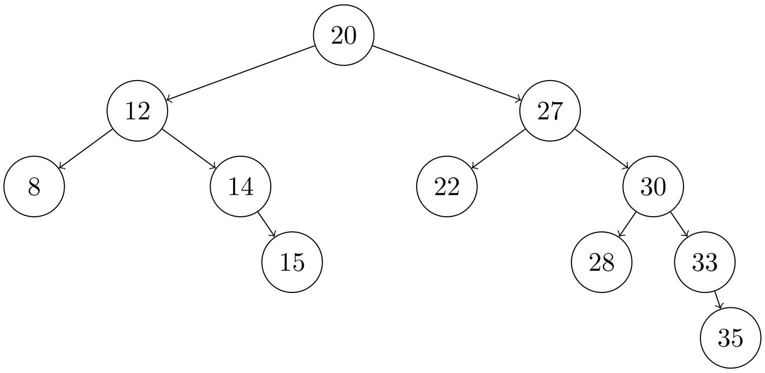
Problem 1. Perform a right rotation on the root of the following tree. Be sure to specify the subtrees used in the rotation.



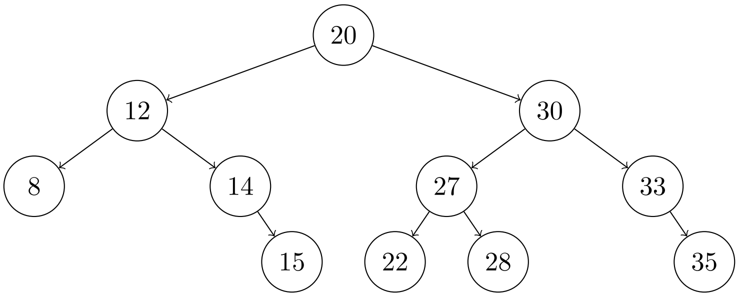
After a right rotation of the subtree rooted at 8, which happens to the the whole tree, we obtain:



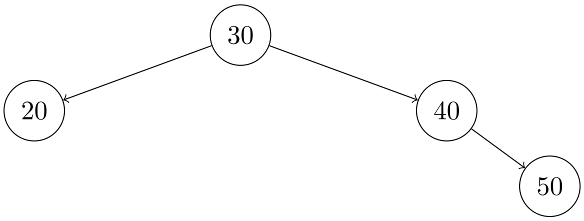
Problem 2. Show the left rotation of the subtree rooted at 27. Be sure to specify the subtrees used in the rotation.



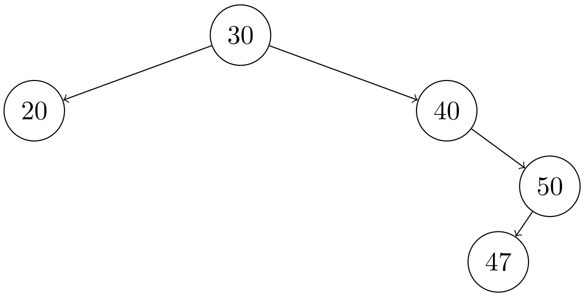
After a left rotation of the subtree rooted at 27, we obtain:



Problem 3. Using the appropriate AVL tree algorithm, insert the value 47 into the following tree. Show the tree before and after rebalancing.

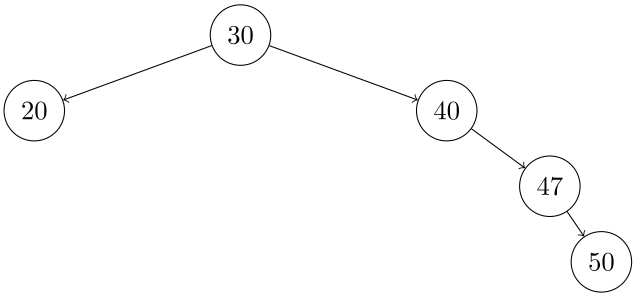


After inserting 47 (but before rebalancing), the tree will look like this:

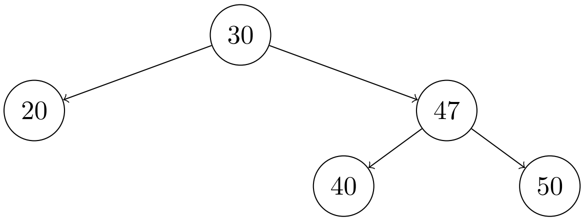


At this point, the tree is out of balance at node with keys 40 and 30. The balancing acts from bottom to top, so we right rotate the subtree rooted at 50:

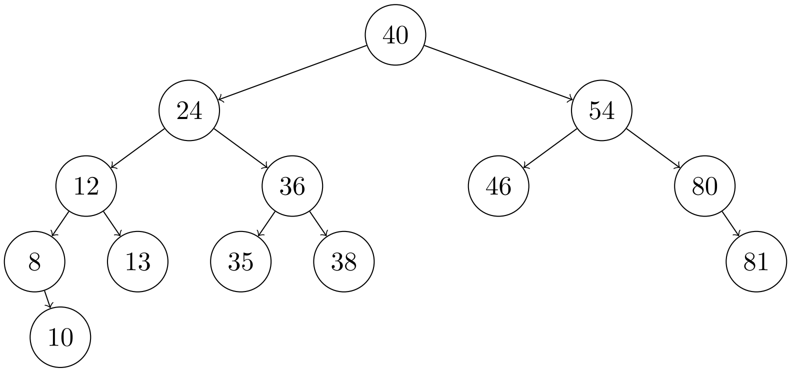
The resulting tree looks like this:



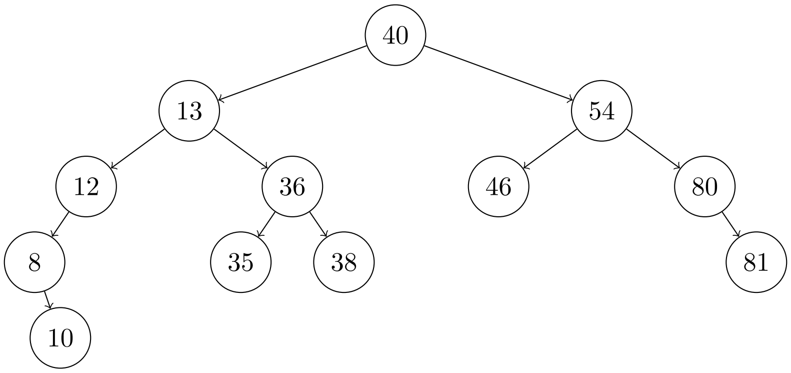
Now, we want to left rotate the subtree rooted at 40. The resulting tree looks like this:



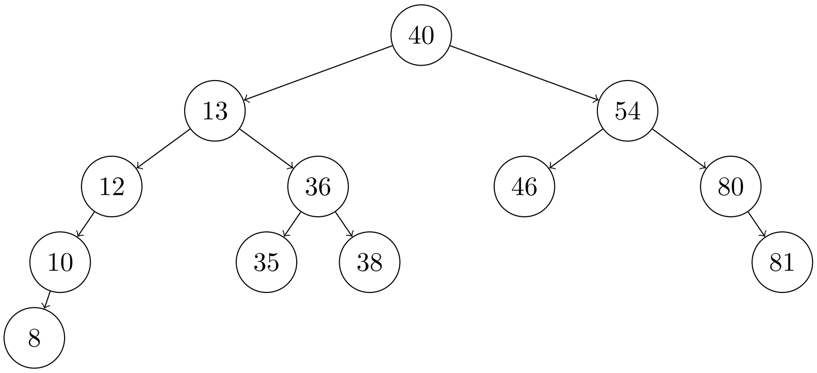
Problem 4. Using the appropriate AVL tree algorithm, remove the value 24 from the following tree. Show the tree before and after *each* rebalancing.



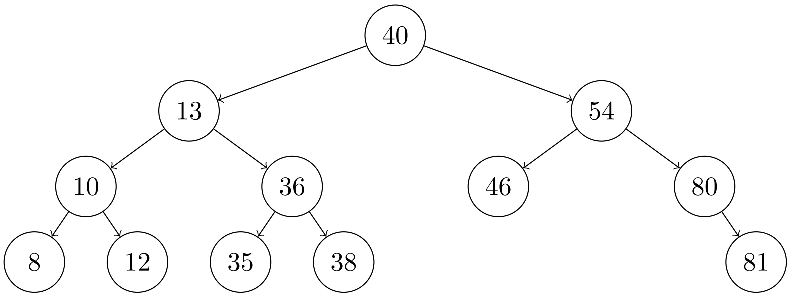
After removing the node 24, the tree looks like this:



Now the tree is unbalanced at nodes 12 and 40. The balancing acts from bottom to top, so we want to first left rotate the subtree rooted at 8. The resulting tree looks like this:



And now we want to right rotate the subtree rooted at node 12. The resulting tree looks like this:



Now the tree is balanced.