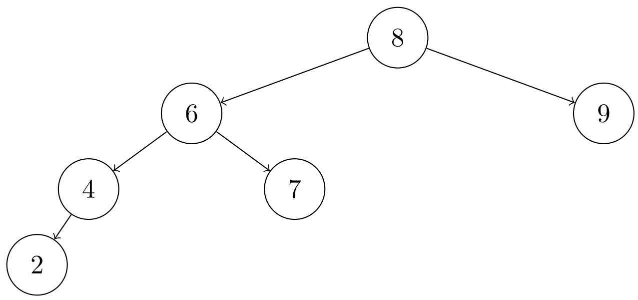
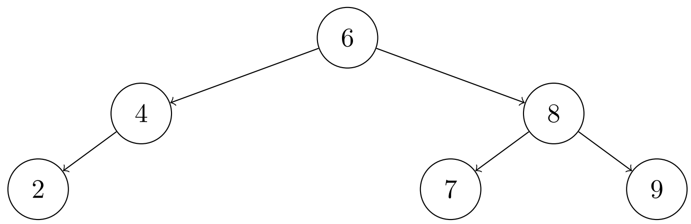
1 AVL Trees

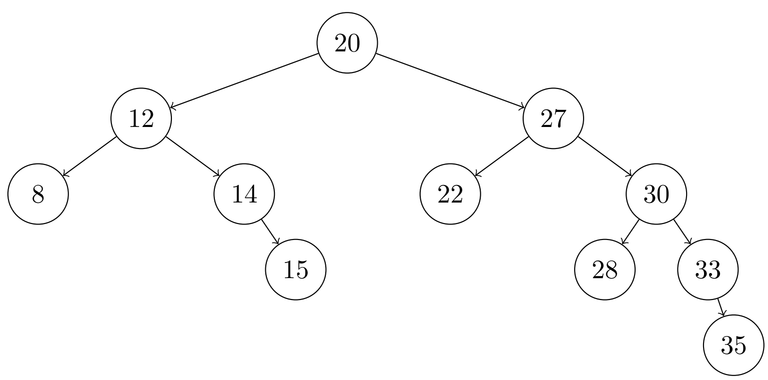
Problem 1. Perform a right rotation on the root of the following tree. Be sure to specify the subtrees used in the rotation.

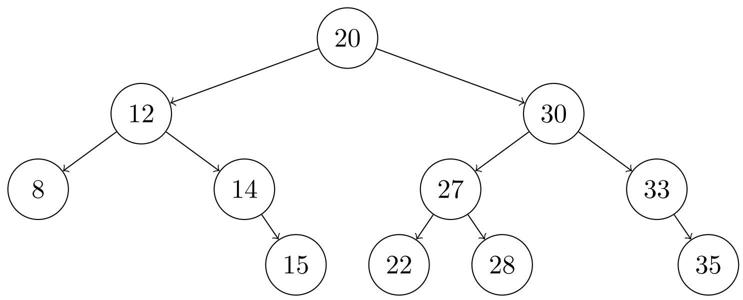




the subtree used in this rotation was rooted at 6 (with children 4, 2, 7). Thereby, it also affected the original root, 8 and its subtrees.

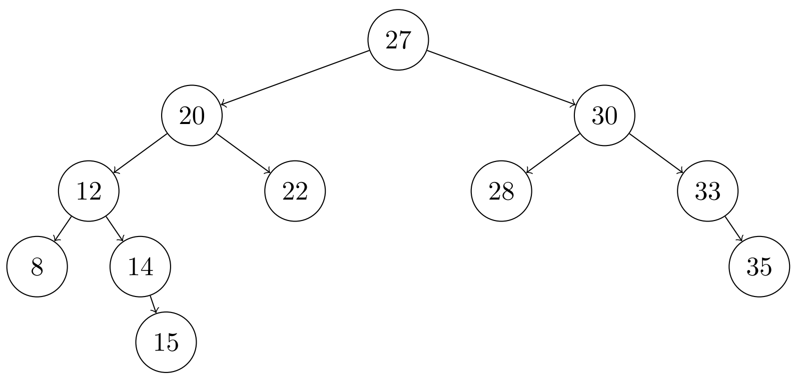
Problem 2. Show the left rotation of the subtree rooted at 27. Be sure to specify the subtrees used in the rotation.



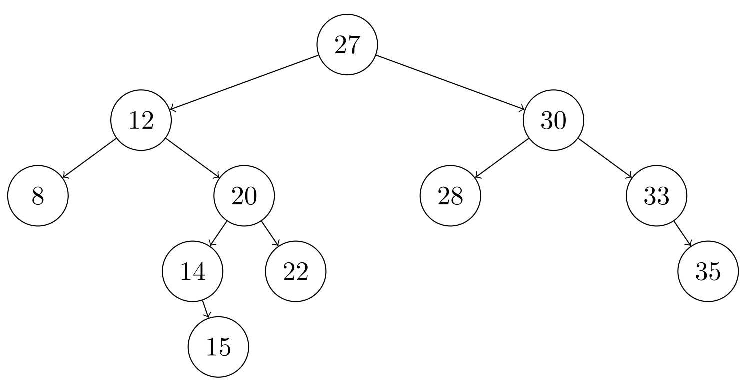


The subtree used in this rotation was rooted at 27 (with immediate children 22,30). Thereby, the children of 22 and 30 were also affected. The root 27 was left rotated to become 30 and everything was shifted accordingly.

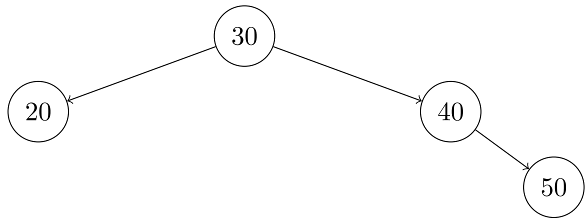
other potential solution.



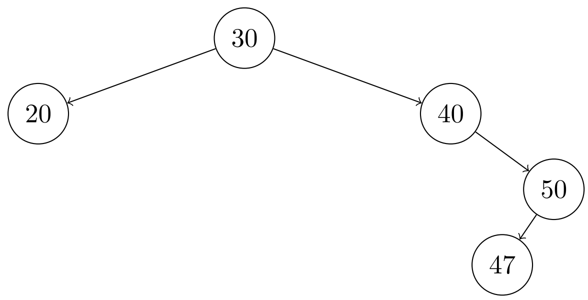
there is an imbalance at node 20. We need to left rotate 14 such that the tree looks like this



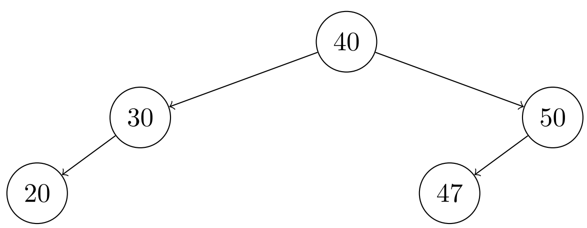
Problem 3. Using the appropriate AVL tree algorithm, insert the value 47 into the following tree. Show the tree before and after rebalancing.



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



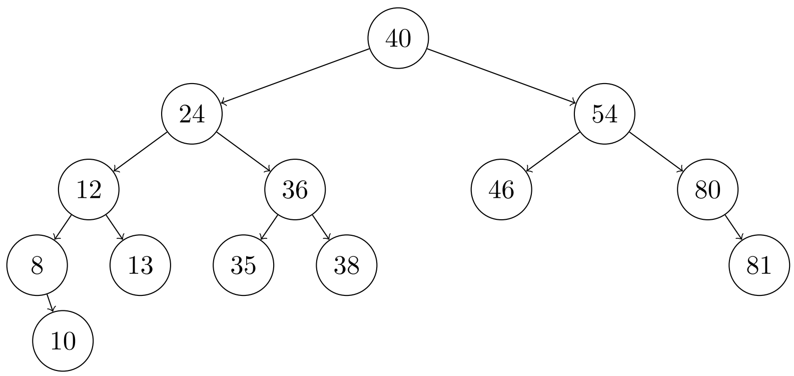
At this point, the tree is out of balance at node 40 (and thereby 30) so we perform a left rotation at 40. The resulting tree looks like this:



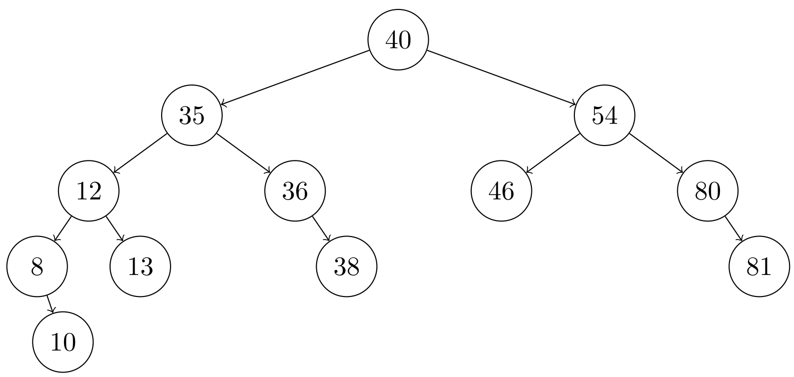
Is the tree now balanced? If not, perform the next rotation here. Repeat until the tree is balanced.

Yes the tree is balanced. There are no roots with children with a difference greater than 2.

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



We remove 24 as indicated in the instructions. I used the min of the right tree as the new root. The resulting tree looks like this.



At this point, the tree is in balance.