3) (10 pts) ALG (AVL Trees)

This question deals with the AVL Tree shown below:

```
              30
            /      \
          15        70
         /  \      /  \
        8    20   45    100
       /         /  \     \
      6        33   62    107
                    /
                   57
```

(a) (7 pts) How many restructure operations (a single restructure operation is either a single or double rotation) would occur if each of the following items was deleted? Consider each item separately as being the only item being deleted from the tree shown above. (Note: It's possible that the answer to some parts is 0.)

Item to Delete	**Number of Restructure Operations**
6	**1 (at 30)**
20	**2 (at 15, then 30)**
33	**1 (at 45)**
57	**0**
62	**0**
100	**1 (at 70)**
107	**1 (at 70)**

Grading: 1 pt per answer, no exceptions (node of restructuring isn't necessary)

(b) (3 pts) What is the fewest number of consecutive insertion operations that would need to occur to force a rebalance at the root node of the given tree in the picture? (Hint: In order for this to occur, there has to be the requisite height imbalance at the root node 30, and no other imbalances on the path from the last inserted node to the root.)

5 **Grading: 3 pts for the correct answer, 2 pts for 4 or 6, 1 pt for 3 or 7, 0 pts otherwise**

Note: In order for this to occur, the right side would have to become a height 2 more than the left of 30. But since the node 70 is already not perfectly balanced, we must first insert 2 items (one example that works is 90 followed by 101) so that 70 is perfectly balanced. We must then get 45 perfectly balanced. One way to do this is to insert 31. Finally, either 33 or 62 must be perfectly balanced. One way to do this is inserting 67. Finally, from this point, inserting 65 will ultimately trigger a rebalance operation all the way up at 30, which, for this example, would make 45 the new root of the tree. More generally, to force a rebalance at a particular node, both trees underneath it must be valid AVL trees, and in general, for a height of n, the minimum number of nodes to create a valid AVL tree is $F_{n+3} - 1$, where F_{n+3} is the $(n+3)^{rd}$ Fibonacci number. This dictates that we need the right subtree of 30 to have $F_{4+3} - 1 = 13 - 1 = 12$ nodes. Since it currently has 7, the minimum number of insertions theoretically possible is 12 - 7 = 5. The construction given shows that it's possible to achieve this theoretical minimum.