Homework Sheet 5

Author	**Matriculation Number**	**Tutor**
Abdullah Oğuz Topçuoğlu	7063561	Maryna Dernovaia
Ahmed Waleed Ahmed Badawy Shora	7069708	Jan-Hendrik Gindorf
Yousef Mostafa Farouk Farag	7073030	Thorben Johr

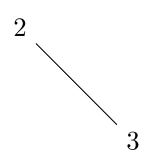
Exercise 1

We begin with an empty AVL tree and perform the following operations step by step:

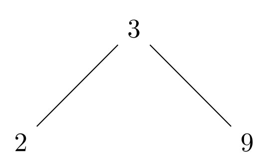
1. Insert(2)
2. Insert(3)
3. Insert(9)
4. Insert(5)
5. Insert(8)
6. Insert(6)
7. Remove(8)
8. Remove(5)
9. Insert(4)
10. Insert(5)
11. Remove(6)

Insert(2): 2

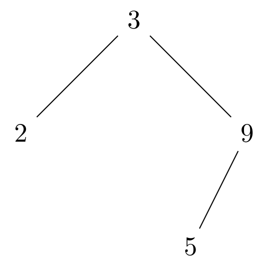
Insert(3):



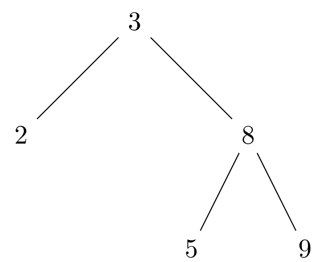
Insert(9):



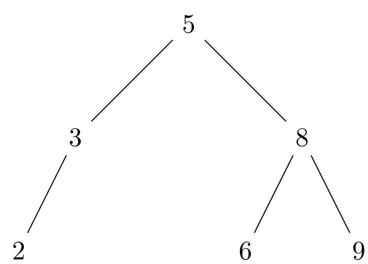
Insert(5):



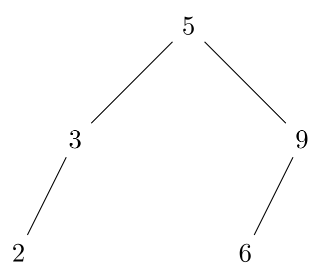
Insert(8):



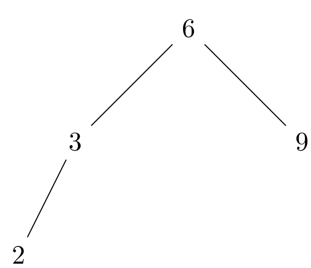
Insert(6):



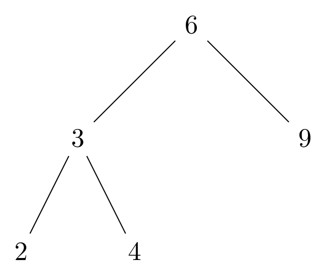
Remove(8):



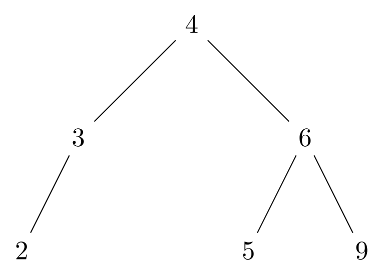
Remove(5):



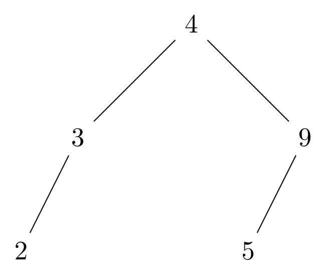
Insert(4):



Insert(5):



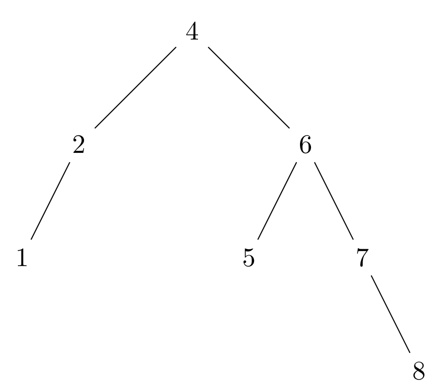
Remove(6):



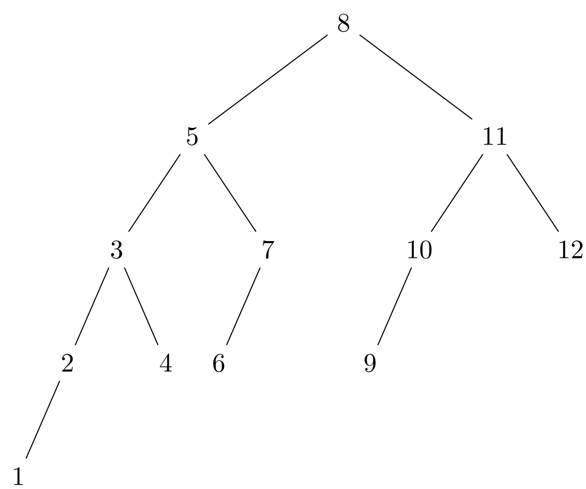
Exercise 2

(a)

An AVL tree of maximum height with 7 nodes:



An AVL tree of maximum height with 12 nodes:



(b)

Proof by Induction:

- **Base Case:** For height $h = 0$, the minimum number of nodes in an AVL tree is $F_{0+2} - 1 = F_2 - 1 = 1 - 1 = 0$, which is correct since an AVL tree of height 0 has no nodes.

- **Inductive Step:** Assume that for some height $h \geq 0$, the minimum number of nodes in an AVL tree of height h is $F_{h+2} - 1$. We need to show that this holds for height $h + 1$.

- Consider an AVL tree of height $h + 1$. To minimize the number of nodes, we can have one subtree of height h and another subtree of height $h - 1$ (since the height difference must be at most 1 for AVL trees).

- By the inductive hypothesis the minimum number of nodes in the subtree of height h is $F_{h+2} - 1$ and in the subtree of height $h - 1$ is $F_{(h-1)+2} - 1 = F_{h+1} - 1$.

- Therefore the total minimum number of nodes in the AVL tree of height $h + 1$ is:

  $$N(h+1) = (F_{h+2} - 1) + (F_{h+1} - 1) + 1 = F_{h+2} + F_{h+1} - 1 = F_{(h+1)+2} - 1$$

  (using the Fibonacci property $F_n = F_{n-1} + F_{n-2}$).

- Thus by induction the statement holds for all heights $h \geq 0$.

Exercise 3

(a)

To implement the Min(h) operation, we can follow the left child pointers starting from the node represented by handle h until we reach a node that does not have a left child. This node will be the node with the smallest key in the subtree rooted at h since AVL trees are just BSTs with a constraint on the height of the nodes. The algorithm is as follows:

```
Handle Min(Handle h)
    current := h
    while current.left != NULL do
        current := current.left
    return current
```

The running time of this algorithm is O(log n) because in an AVL tree, the height is O(log n), and we may need to traverse from the root of the subtree down to the leaf.