1. Give a small example of adding a number to an AVL-tree causing an unbalanced tree of type right-right. Rebalance your tree.

2. Give a small example of adding a number to an AVL-tree causing an unbalanced tree of type right-left. Rebalance your tree.

3. Give a small example of removing a number from an AVL-tree causing an unbalanced tree of type right-right. Rebalance your tree.

4. Remove from the following AVL-tree the node labelled 8, then add a node labelled 9.

5. Remove the node labelled 9 from the following AVL-tree:

6. Give an example showing that the difference in depth between two leaves in an AVL-tree can be larger than 2.
7. Design an algorithm to determine the smallest number in an AVL tree. What is the worst-case time complexity of your algorithm?

8. True or false: The smallest and the largest number in an AVL tree can always be found on either the last or the one-before-last level?

9. Suppose that we add to the set of stack operations not only the operation \texttt{multipop}(S, k) but also the operation \texttt{multipush}(k, e) which pushes \(k\) times the element \(e\) on the stack. Is the amortized cost of stack operations still in \(\mathcal{O}(1)\)?

10. We consider the usual stack with operations \texttt{push} and \texttt{pop}. We assume that the size of the stack never exceeds \(k\) (for some fixed natural number \(k\)). We add an operation that after \(k\) operations makes a copy of the stack. Show that the cost of \(n\) stack operations is in \(\mathcal{O}(n)\); use the accounting method.

   (In addition, we can consider the stack with operations \texttt{push}, \texttt{pop}, and \texttt{multipop} where the operation for \texttt{copy} is added, assuming again that the size of the stack never exceeds \(k\).)

11. Use the accounting method to analyze the amortized complexity of \texttt{increment} in the binary counter. (Cf the lecture.)

12. Reconsider the exercise from exercise class 5 concerning implementing a queue using two stacks. Give the amortized complexity of the operations \texttt{enqueue} and \texttt{dequeue} from the implementation of the queue using two stacks. Use aggregate analysis and the accounting method.