Exercise 5.1  Joining AVL Trees.

Let $T_1$ and $T_2$ be two AVL trees that contain the key sets $K_1$ resp. $K_2$. Let the largest key in $K_1$ be smaller than the smallest key in $K_2$, i.e. we have $K_1 \cap K_2 = \emptyset$. Design an algorithm that joins the AVL trees $T_1$ and $T_2$ in time $O(\log n)$, i.e. that computes an AVL tree containing exactly the key set $K_1 \cup K_2$.

Note: The challenge is to ensure that the resulting tree is again an AVL tree (and not an arbitrary unbalanced search tree).

Exercise 5.2  Amortized Analysis.

In this exercise, we consider arrays that grow dynamically on demand (e.g., `java.util.Vector` in the Java standard library). Specifically, we assume that we insert the values one by one. If more than $n$ elements are stored, a new array of fixed length $k > n$ is created, the old contents are copied and the new element is stored. An array of length $k$ can be created in $k$ steps, and the copying of an element is done in constant time.

a) Describe how to choose $k$ so that each insert operation has amortized constant time, and hence the insertion of $n$ elements can be done in time $\Theta(n)$. Prove using amortized analysis that your choice results in constant amortized time per insert operation.

b) Now consider the situation where we allow to delete the last element from the array. We allow any mixture of such insert and delete operations. For memory reasons, it may be useful to also shrink the array sometimes. Describe how would you shrink the array, and show that both insertion and removal require amortized constant time.

Exercise 5.3  Blum’s Median-of-Median Strategy.

We consider finding the median of a sequence using the median-of-median strategy from the lecture (see Chapter 3.1 in the book). We will consider only the highest level of recursion, so only the very first invocation of the procedure `Auswahl` that determines the $i$-th smallest element with $i = \lceil \frac{N}{2} \rceil$.

a) Given the following sequence

$$8, 13, 17, 5, 11, 29, 3, 4, 11, 10, 15, 7, 30, 57, 1, 2, 6, 9, 17, 7, 14, 13,$$

provide the two sequences on which `Auswahl` invokes itself recursively.

b) In general, how long at least and at most are each of the two sequences used in the two recursive calls of the procedure `Auswahl` for $i = \lceil \frac{N}{2} \rceil$?