University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
P. Schneider

Algorithms and Data Structures
Winter Term 2019/2020
Exercise Sheet 7

Remark: For this exercise, the material of the eleventh video lecture is relevant.

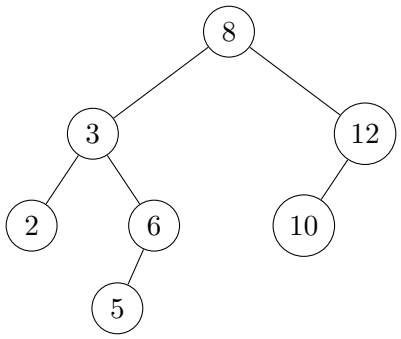
Exercise 1: Binary Search Trees - Finding the Successor

(a) Give an algorithm that takes as input a node of a binary search tree and outputs its successor in the tree, i.e., the node with the next larger key in $\mathcal{O}(d)$, where d is the depth of the tree.

(b) Explain the correctness and the running time of your algorithm.

Exercise 2: AVL Trees

Consider the following AVL tree



(a) In the above tree, perform the operations `insert`(4), `insert`(7) and `insert`(1) and the necessary rotations to re-balance the AVL-tree. Draw the state of the tree after each operation.

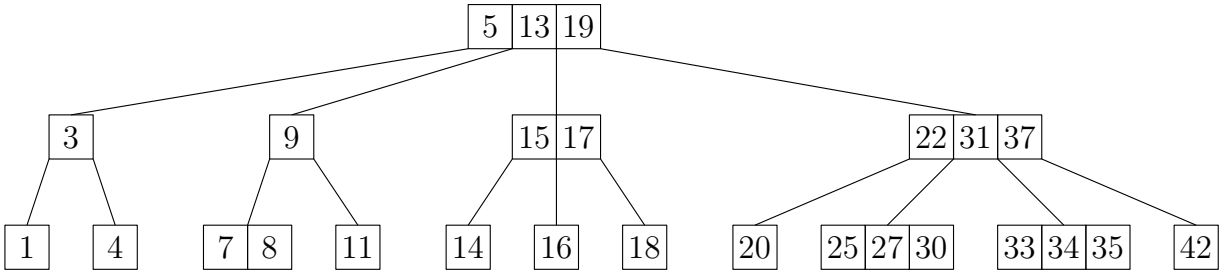
Remark: Inserting works the same as in binary search trees. Afterwards, for each ancestor of the inserted node (bottom up), repair the AVL condition (if violated) by performing an according rotation (left or right).

(b) In the resulting tree, perform the operations `delete`(5) and `delete`(7) and the necessary rotations to re-balance the AVL-tree. Draw the state of the tree after each operation.

Remark: Deleting works the same as in binary search trees. Afterwards, starting at the position of the node that was used to replace the deleted key, for each ancestor (bottom up) repair the AVL condition (if violated) by performing an according rotation (left or right or double rotations).

Exercise 3: (a, b)-Trees

Consider the following $(2, 4)$-tree



(a) In the given tree, perform the operations `insert`(2), `insert`(26) and `insert`(36). Draw the state of the $(2, 4)$-tree after each operation.

(b) In the resulting tree, perform the operations `delete`(11), `delete`(3). Draw the state of the $(2, 4)$-tree after each operation.

(c) *For exercise lesson:* In the resulting tree from part (b), perform `delete`(13) and draw the state of the $(2, 4)$-tree.

Remark: For comprehensive details on all cases of the `delete` *operation consider, e.g., "Introduction to Algorithms" by Cormen, Leierson, Rivest and Stein.*