Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Department Informatik
Markus Püschel
Peter Widmayer
Thomas Tschager
Tobias Pröger

24th November 2016

Algorithmen & Datenstrukturen — Exercise Sheet 10 — AS 16

Hand-in: Thursday, 1st December 2016 before the start of the lecture at 10:00 in the entrance area of ML D28. Please staple all sheets together and use this sheet as the cover page. Fill out the first two fields of the form below.

Exercise class (Room & Day): ____________________

Submitted by: ____________________

Corrected by: ____________________

Bonus points: ____

Exercise 10.1 *Search Trees.*

a) Draw the resulting tree if you insert the keys $11, 2, 8, 9, 5, 17, 20, 24, 18, 30$ in this order into an initially empty natural search tree.

b) Give the preorder, postorder, and inorder traversal of the tree in a).

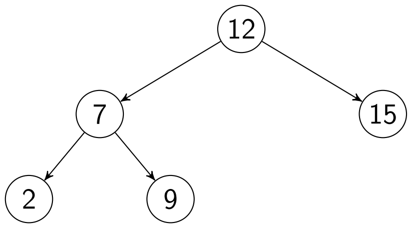
c) Remove the key 17 from the tree in a) and after that the key 11 from the resulting tree. Draw both trees.

d) Draw the result if the keys from part a) are inserted into an initially empty AVL tree.

Exercise 10.2 *Questions on Search Trees.*

a) Given a binary search tree T and the numbering of the vertices according to the preoder and the postorder traversal, how can you decide for two vertices v and w in T only considering the preorder and postorder numbers if w is in the subtree of v? Justify your answer.

b) Insert a *single* key that results in a *double rotation* into the following AVL tree. All keys that are stored in the AVL tree have to be pairwise different. Indicate *all possible* integer candidate keys.



Candidates: ____________.

Please turn over.

Exercise 10.3 *Number of different Search Trees.*

Let $\mathcal{K}_n = \{1, 2, \ldots, n\}$ be a set of keys. Derive a recursive formula for the number of different binary search trees that contain exactly the keys in $\mathcal{K}_n$. You do *not* need to eliminate the recursion.

Exercise 10.4 *Matrix Multiplication.*

You have seen Strassen's algorithm for multiplying two $(n \times n)$ matrices A and B in the lecture (cf. lecture notes, 3.11.2016). The algorithm partitions A and B into four $(n/2 \times n/2)$ submatrices and computes seven products of matrices of size $(n/2 \times n/2)$.

a) Show that, as stated in the lecture, each of the four submatrices of $A \times B$ can actually be written as the sum, respectively the difference, of some of the seven products.

b) Show that the Strassen's algorithm uses $\mathcal{O}(n^{\log_2 7})$ operations even if not only elementary multiplications, but also elementary additions and subtractions are considered. As in the lecture, define a recurrence relation $A(n)$ that counts the number of elementary operations that occur during the multiplication of two $(n \times n)$ matrices. Note that $\Theta(n^2)$ many elementary additions, respectively subtractions are necessary for additions or subtractions of two $(n \times n)$ matrices. Find a closed form of $A(n)$ by telescoping and prove the correctness using mathematical induction over n. Justify briefly why $A(n) \in \mathcal{O}(n^{\log_2 7})$ holds.