Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr/Jan Griebsch

Wintersemester 2005/06
Übungsblatt 1
6. Februar 2006

Fundamental Algorithms

Aufgabe 1

Give, in Landau notation, the relationships between the functions in the table below. Fill each field in column f and row g with one of $\{o, O, \omega, \Omega, \Theta\}$. Be as precise as possible.

	$n\log(n)$	n	$n\log\log(n)$	$n^{1.1}$	$n\log^2(n)$	$n\sqrt{n}$
$n\log(n)$						
n						
$n\log\log(n)$						
$n^{1.1}$						
$n\log^2(n)$						
$n\sqrt{n}$						

Aufgabe 2

Prove the following:

a) $g = o(f) \Rightarrow g = O(f)$
b) $g = \omega(f) \Rightarrow g = \Omega(f)$
c) $g = \Omega(f) \Leftrightarrow f = O(g)$
d) $g = \Theta(f) \Leftrightarrow f = O(g) \wedge g = O(f)$
e) $g = o(f) \Rightarrow g \neq \Omega(f)$
f) $g = \omega(f) \Rightarrow g \neq O(f)$

Make sure to prove both directions of equivalence relations. For which of the implications a),b),e),f) does the opposite direction hold as well ?

Aufgabe 3

Let SuperComputer be a very fast computer which can perform 10^9 operations per second. For some problem of size n the table below lists the number of operations neccessary. More specifically, the ith algorithm needs $t_i(n)$ operations:

$$\begin{aligned} t_1(n) &= 2 \cdot n \\ t_2(n) &= n \log_2(n) \\ t_3(n) &= 2.5 \cdot n^2 \\ t_4(n) &= \frac{1}{1000} \cdot n^3 \\ t_5(n) &= 3^n. \end{aligned}$$

Determine, for which maximal input sizes each algorithm needs at most 1 second, 1 minute, 1 hour.
How do these values change, if the computer is upgraded to be 10 times faster (i.e. can do 10^{10} operations) ?

Aufgabe 4

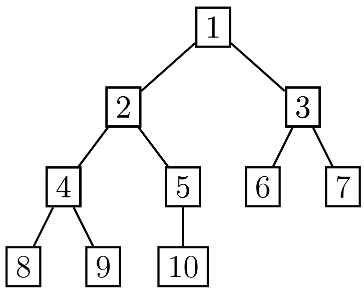
Let T be a binary tree. Prove

a) T at most $2^{\ell-1}$ nodes has within a level ℓ (with $1 \leq \ell \leq d$).

b) T has at most 2^{d-1} leafs.

Aufgabe 5

Given a heap whose nodes are numbered (in the usual graphical representation) level-wise from left to right. The first number is 1 (at the root). Example:



Let v be an inner node with number i. Show the following:

a) The left child of v has number $2i$.

b) The right child of v (if it exists) has number $2i + 1$.

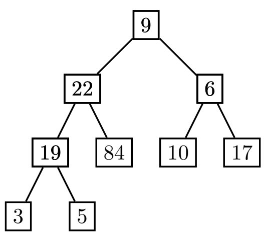
c) The father of v (if it exists) has number $\lfloor \frac{i}{2} \rfloor$.

d) The level vof v is equal to the length of the binary representation of i, i.e. $\ell(i) = \lfloor \log i \rfloor + 1$.

Aufgabe 6

Given the array $A = [9, 22, 6, 19, 84, 10, 17, 3, 5]$. Illustrate graphically the following operations

a) create_heap$(A, 9)$



b) Three times delete_min(h) on the above created heap h.

Show all intermediate steps.

Aufgabe 7

We want to define a new operation `increase_min` for heaps with pair-wise different integer keys. This operation shall increase the minimal key (the one in the root) of the heap h by d. Make sure your algorithm ensures that the heap is still valid afterwards. The heap h is given as a tree, with a pointer to the root. At each node, pointers to the children are stored.

(a) Give an efficient algorithm for `increase_min`. The other heap operations (`create_heap`, `delete_min` und `reheap`) may not be used explicitly.

(b) Give a tight upper bound (O-notation), and explain why it holds.

Aufgabe 8

Give an algorithm to print out the keys stored in a binary search tree - in increasing order. The runing time shall be linear on the number of nodes in the tree.The tree is given as a pointer to the root. At each node, pointers to the children are stored.

Aufgabe 9

Given a binary search tree with keys from 1 to 1000. We are searching an element with key 363. Which of the following sequences of keys can not possibly appear in the sequence of keys of the nodes visited during the search ?

a) $2, 252, 401, 398, 330, 344, 397, 363$

b) $924, 220, 911, 244, 898, 258, 362, 363$

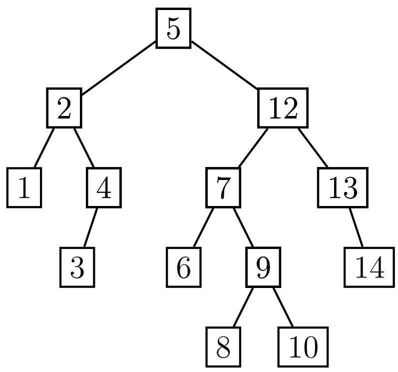
c) $925, 202, 911, 240, 912, 245, 363$

d) $2, 399, 387, 219, 266, 382, 381, 278, 363$

e) $935, 278, 347, 621, 299, 392, 358, 363$

Aufgabe 10

Given the following AVL tree:



a) Prove, that an AVL tree of height h has at most $2^{h+1} - 1$ nodes.

b) Execute `insert(11)` on the given tree. Show all intermediate steps of the operation (rebalancing, balances, etc.).

Aufgabe 11

Given an undirected graph $G = (V, E)$ with $V = \{1, 2, \dots, 15\}$. The edge set E is:

1: 2,5,8	2: 1,4,13	3: 8,11,14
4: 2,7,13,14	5: 1,6,7	6: 5,8,10
7: 4,5,10,14	8: 1,3,6	9: 15
10: 6,7,12	11: 3	12: 10,13
13: 2,4,12	14: 3,4,7	15: 9

a) Compute the adjacency matrix of the given graph.

b) Determine the DFS numbers of the nodes of a DFS traversal starting at node 5 (traverse the adjacency lists of the nodes from left to right).

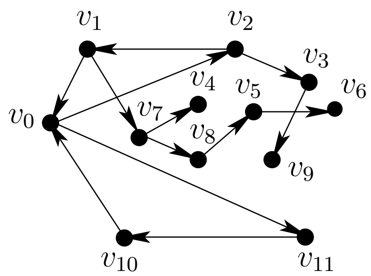
Aufgabe 12

a) Show that for each directed graph $G = (V, E)$

$$|E| = \sum_{v \in V} d^+(v) = \sum_{v \in V} d^-(v) = \frac{1}{2} \sum_{v \in V} d(v).$$

Aufgabe 13

a) Given the directed graph $G = (V, E)$ below, do a breath first search on G. At each step, give the current content of the queue. For each node, determine the BFS number.
b) Do the same for the undirected version of G: $G' = (V, \{\{v, w\} : v, w \in V \wedge (v, w) \in E\})$
.



Aufgabe 14

In the lecture, we looked at how many comparisons insertion sort needs to sort an array with n elments. Determine asymtotic time complexity including all operations (i.e. including copying elements around) in the worst case. Give an example for this worst case.

Aufgabe 15

Let T_1, T_2 be two (a, b)-trees with n_1, n_2 nodes, such that for all $x \in T_1$ and $y \in T_2$ it holds that: $\text{key}(x) < \text{key}(y)$.
Design an algorithm CONCATENATE, which merges T_1 and T_2 into a new (a, b)-tree with running time $O(\log(n_1 + n_2))$.

Aufgabe 16

Let f, g be two functions with $\mathbf{N} \to \mathbf{R}^+$. Prove

a)

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty \quad \Rightarrow \quad f(n) = O(g(n))$$

b)

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} > 0 \quad \Rightarrow \quad f(n) = \Omega(g(n))$$

Why doesn't the reverse hold ? Give a counter example for each.