TU Bergakademie Freiberg
Fakultät Mathematik und Informatik
Institut für Numerische Mathematik und Optimierung
Oliver Ernst
Elisabeth Ullmann

28. Oktober 2005

Algorithmen und Datenstrukturen

Wintersemester 2005/06

3. Übungsblatt

Aufgabe 10:

Liegt die Funktion $f(n) = 2^{n+1}$ in $O(2^n)$? Liegt die Funktion $g(n) = 2^{2n}$ in $O(2^n)$?

Aufgabe 11:

Gegeben sei folgende Aufzählung von 30 Funktionen.

$\log_2(\log_2^* n)$	$2^{\log_2^* n}$	$\sqrt{2}^{\log_2 n}$	n^2	$n!$	$(\log_2 n)!$
$\left(\frac{3}{2}\right)^n$	n^3	$\log_2^2 n$	$\log_2(n!)$	2^{2^n}	$n^{1/\log_2 n}$
$\log\log n$	$\log_2^* n$	$n2^n$	$n^{\log_2 \log_2 n}$	$\log n$	1
$2^{\log_2 n}$	$(\log_2 n)^{\log_2 n}$	e^n	$4^{\log_2 n}$	$(n+1)!$	$\sqrt{\log_2 n}$
$\log_2^*(\log_2 n)$	$2^{\sqrt{2\log_2 n}}$	n	2^n	$n\log_2 n$	$2^{2^{n+1}}$

Ordnen Sie diese absteigend nach Wachstumsordnung, d.h. finden Sie eine Ordnung $f_1, \ldots, f_{30}$ mit $f_k = \Omega(f_{k+1})$ $(k = 1, \ldots, 29)$. Zerlegen Sie Ihre Liste in Äquivalenzklassen so, dass zwei Funktionen f und g genau dann äquivalent sind, wenn $f = \Theta(g)$.

Aufgabe 12:

Leiten Sie mit Hilfe eines Rekursionsbaumes asymptotisch enge Schranken für die Rekursion $T(n) = T(\alpha n) + T((1-\alpha)n) + cn$ her, wobei $\alpha \in (0, 1)$ und $c > 0$ Konstanten sind.

Aufgabe 13:

Geben Sie asymptotische (untere und obere) Schranken an für die Rekursion $T(n) = 16T(n/4) + n^2$.

Aufgabe 14: (Programmieraufgabe)

Bestimmen Sie durch Ausprobieren, ab welchem Wert von n Ihr Merge-Sort Programm ein Feld mit n Elementen schneller sortiert als Ihr Insertion-Sort Programm. Nehmen

Sie als nächstes als Eingabe aufsteigend sortierte Folgen der Länge n für verschiedene Werte von n. (Es reicht aus, einfach die Zahlen von 1 bis n zu nehmen.) Prüfen Sie, ob für diese Eingaben Insertion-Sort, mit einer asymptotischen Laufzeit von $\Theta(n)$ im günstigsten Fall, in Ihren Versuchen auch tatsächlich stets schnelle ist als Merge-Sort.