Institut für Telematik
Karlsruher Institut für Technologie (KIT)
Prof. Dr. M. Zitterbart
Dipl.-Inform. C. Haas
Dipl.-Inform. J. Wilke

Algorithmen I
Sommersemester 2011

1. Übungsblatt
Komplexitätstheorie und Sortieren

Aufgabe 1

Asymptotische Wachstumsraten

(a) Zeigen Sie, dass $f(n) = n!$ in $O(n^n)$ liegt. (4P.)

(b) Zeigen Sie, dass $f(n)$ sogar in $o(n^n)$ liegt. (8P.)

(c) Ordnen Sie die folgenden Funktionen nach ihrem Wachstumsgrad absteigend. Geben Sie dazu eine Reihenfolge $f_1, f_2, \ldots, f_{30}$ der unten angegebenen Funktionen f_i an, sodass $f_i = \Omega(f_{i+1})$ gilt. Teilen Sie die Liste dabei in Äquivalenzklassen auf, sodass sich zwei Funktionen $f(n)$ und $g(n)$ genau dann in der gleichen Klasse befinden, wenn $f(n) = \Theta(g(n))$ gilt. (15P.)

$2 \cdot n^2$	$42n^2 + 3$	$(\sqrt{2})^{\lg n}$	n^2	0	$2^{2^{n+1}}$
$(3/2)^n$	n^3	2^n	$n \lg n$	2^{2^n}	$n^{1/\lg n}$
$\ln \ln n$	$\frac{n^2}{\pi}$	$n \cdot 2^n$	$n^{\lg \lg n}$	$\ln n$	1
$2^{\lg n}$	$(\lg n)^{\lg n}$	e^n	$4^{\lg n}$	n	$\sqrt{\lg n}$

Aufgabe 2

Sortieren Sie die Kartennummer ihres KIT Ausweises.

(a) Benutzen Sie Insertionsort. Geben Sie den Zustand des Feldes nach jedem der Insert-Schritte an. (5P.)

(b) Benutzen Sie Mergesort. (5P.)

(c) Benutzen Sie Quicksort. Wählen Sie immer das erste Element als Pivot-Element. (5P.)

Aufgabe 3

Betrachten Sie das Suchproblem:

- **Eingabe:** Eine Sequenz von n Zahlen $A = \langle a_1, a_2, \ldots, a_n \rangle$
- **Ausgabe:** Ein Index i mit $v = A[i]$, falls v in A vorkommt, und dem speziellen Wert nil, wenn der Wert v nicht in A vorkommt.

Hinweis: Sie können annehmen, dass das gesuchte Element nur 1 mal in der Liste vorkommt.

(a) Schreiben Sie ein Programm *LineareSuche(A,v)* in Pseudocode für die **lineare Suche**, die eine Sequenz nach v durchsucht. Untersuchen Sie den Algorithmus im besten, schlechtesten und durchschnittlichen Fall. Drücken Sie die Komplexität jeweils in Θ-Notation aus. Begründen Sie ihre Antwort. (6P.)

(b) Schreiben Sie ein Programm *BinäreSuche(A,v,low,high)* in Pseudocode für die **binäre Suche**. Gehen Sie davon aus, dass die Folge A sortiert ist und Sie v mit dem mittleren Element der Folge vergleichen können und nach diesem Vergleich die Hälfte der Folge nicht weiter betrachten brauchen. Die binäre Suche ist ein Algorithmus, der diese Prozedur wiederholt anwendet, wobei der verbleibende Teil der Folge jeweils halbiert wird. Gestartet wird der Aufruf mit *BinäreSuche(A,v,1,n)*. Untersuchen Sie den Algorithmus im schlechtesten Fall. Drücken Sie die Komplexität in Θ-Notation aus und begründen Sie ihre Antwort. (12P.)

Abgabe des Übungsblattes bis Mittwoch, 27.4.2011 – 12 Uhr durch Einwurf in den entsprechenden Briefkasten im Gebäude 50.34, 1. UG.