Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Dr. Hanjo Täubig
Tobias Lieber

Sommersemester 2011
Übungsblatt 1
19. August 2011

Grundlagen: Algorithmen und Datenstrukturen

Abgabetermin: In der jeweiligen Tutorübung

Hausaufgabe 1

Betrachten Sie den Algorithmus 1 zur Bestimmung eines minimalen Elements in einem Integer-Array.

Algorithmus 1: Min

Input : int $A[]$, int n
int $i = 1$
int $min = A[0]$
while $i < n$ **do**
  if $A[i] < min$ **then**
    $min = A[i]$
  $i = i + 1$
return min

Bestimmen Sie die exakte Worst-Case-Laufzeit des Algorithmus, indem Sie die exakte Anzahl der Rechenoperationen, wie sie in der Vorlesung definiert wurden (Vergleiche, Zuweisungen, Additionen, ...), in Abhängigkeit der Eingabegröße berechnen (keine asymptotische Abschätzungen durch Landau-Symbole).
Berechnen Sie nun auch die Worst-Case Laufzeit mit Hilfe der Landau-Symbole.

Hausaufgabe 2

In welcher Relation bezüglich der Landau-Symbole stehen die Funktionen $\operatorname{ld} n = \log_2 n$ und $\ln n = \log_e n$? Beweisen Sie ihre Aussage.

Aufgabe 1

Beweisen Sie folgende Aussagen durch Induktion für alle $n \in \mathbb{N} := \{1, 2, 3, 4, 5, \dots\}$:

1. $\sum_{i=0}^{n} i^3 = \frac{n^2(n+1)^2}{4}$,
2. $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$,
3. $19|(5 \cdot 2^{3n+1} + 3^{3n+2})$ (In Worten: 19 teilt $(5 \cdot 2^{3n+1} + 3^{3n+2})$),

wobei F_n die n-te Fibonaccizahl nach der rekursiven Definition $F_n = F_{n-1} + F_{n-2}$ mit den Anfangswerten $F_0 = 0$ und $F_1 = 1$ ist.

Aufgabe 2

Ordnen Sie die Funktionen $\sqrt{\operatorname{ld}(n+1)}$, n^2, $(\operatorname{ld} n)^{\operatorname{ld} n}$, 2^n, n, 10^{100}, $\sqrt{n}$, $n^{\operatorname{ld}\operatorname{ld} n}$, $\operatorname{ld}\sqrt{n+1}$, $n \log n$ nach Ihrem Wachstum.

Aufgabe 3

Wir betrachten nochmals den Minimum-Algorithmus (siehe Algorithmus 1). Nachdem wir bereits wissen, dass die Worst-Case-Laufzeit in $\mathcal{O}(n)$ liegt, wollen wir dieses Mal die *genaue* Average-Case-Laufzeit berechnen (nicht die asymptotische Average-Case-Laufzeit).
Sie dürfen davon ausgehen, dass jede Eingabe der Länge n aus n unterschiedlichen Zahlen besteht.

Aufgabe 4

Beweisen Sie die in der Vorlesung angegebene Regel:
Für alle Funktionen $f(n), g(n)$ mit $\exists n_0 \in \mathbb{N} : \forall n > n_0 : f(n), g(n) > 0$ gilt:

$$\mathcal{O}(f(n) + g(n)) = \mathcal{O}(f(n)) + \mathcal{O}(g(n))$$

wobei $\mathcal{O}(f(n)) + \mathcal{O}(g(n)) := \{h_f(n) + h_g(n) \mid h_f(n) \in \mathcal{O}(f(n)) \ \wedge \ h_g(n) \in \mathcal{O}(g(n))\}$ (Hier ist also die Gleicheit der beiden Mengen zu zeigen).