Institut für Informatik
der Universität München
Prof. M. Hofmann
Dr. Andreas Abel
Dipl. Inf. Roland Axelsson

SS 2009

23. April 2009

Übungen zur Vorlesung
Algorithmen und Datenstrukturen

Blatt 1

Aufgabe P-1: Beweisen Sie (log sei der Logarithmus zur Basis 2):

$$\text{a) } \frac{n^2+1}{2n} = \Theta(n) \qquad \text{b) } \sqrt{n} = \Omega((\log n)^2) \qquad \text{c) } n2^n = O(n^n)$$

Aufgabe P-2: Beschreiben Sie einen Algorithmus zur binären Suche in einem sortierten Array in Pseudocode, und führen Sie eine detaillierte Analyse von dessen Laufzeit mit Hilfe der *Master*-Methode durch.

Aufgabe P-3: Der *Karatsuba-Algorithmus* zur Multiplikation zweier (großer) Binärzahlen zählt zu den divide-and-conquer Algorithmen und funktioniert wie folgt:

Seien $x = x_{2n-1} \dots x_0$ und $y = y_{2n-1} \dots y_0$ zwei Binärzahlen der Stelligkeit $2n$ (durch führende Nullen werden die ursprünglichen Faktoren zur Stelligkeit $2n$ aufgefüllt). Man zerteilt nun x und y in der Mitte in zwei Hälften x_l, x_r, sowie y_l, y_r (geht in konstanter Zeit). Die Gleichung

$$x \cdot y = c_0 \cdot 2^{2n} + (c_2 - c_1 - c_0) \cdot 2^n + c_1 \tag{1}$$

erlaubt es nun, die drei Produkte

$$\begin{aligned} c_0 &= x_l \cdot y_l \\ c_1 &= x_r \cdot y_r \\ c_2 &= (x_l + x_r) \cdot (y_l + y_r) \end{aligned}$$

rekursiv mit Hilfe derselben Gleichheitsbeziehung zu berechnen. Die Multiplikation zweier $2n$-stelliger Zahlen wird also zurückgeführt auf drei Multiplikationen von je zwei n bzw. $n+1$-stelligen Zahlen sowie vier Additionen $2n$-stelliger Zahlen. Die für die Addition benötigte Zeit ist $\mathcal{O}(n)$.

- Geben Sie in Pseudocode das Standardverfahren zur Multiplikation zweier Zahlen an (wie in der Schule gelehrt) und bestimmen Sie dessen Zeitkomplexität in $\mathcal{O}$-Notation.

- Beweisen sie die Gleichheitsbeziehung unter (1). Tipp: Schreiben Sie x und y durch x_l, x_r bzw. y_l, y_r ausgedrückt auf und multiplizieren Sie das entstandene Produkt aus.
- Geben Sie die Methode von Karatsuba als rekursiven Algorithmus in Pseudocode an und bestimmen sie dessen Laufzeit mit der Master-Methode.

Aufgabe H-1: Seien $f, g : \mathbb{N} \to \mathbb{R}^+$ monotone Funktionen. Beweisen oder widerlegen Sie die folgenden Behauptungen:

- $f(n) + g(n) = \Theta(\max(f(n), g(n)))$
- Wenn $f(n) = O(g(n))$ ist, so auch $2^{f(n)} = O(2^{g(n)})$.
- $f(n) = \Theta(f(\frac{n}{3}))$
- $(f(n) + g(n))^2 = O(f(n)^2) + O(g(n)^2)$

Aufgabe H-2: Führen Sie eine detaillierte Analyse der Laufzeit, wie in der Vorlesung am Beispiel INSERTION-SORT vorgeführt, für die *merge*-Routine, die beim Sortieren durch Mischen (MERGE-SORT) verwendet wird, durch.

Aufgabe H-3: Bestimmen Sie das asymptotische Wachstum der Lösungen $T(n)$ der folgenden Rekursionsgleichungen mit der *Master*-Methode, wo dies möglich ist:

- $T(n) = 9T(\frac{n}{3}) + n$
- $T(n) = 3\,T(\frac{3n}{4}) + (n^2 + n)^2$
- $T(n) = 2\,T(\frac{n}{4}) + \sqrt{n \log n}$
- $T(n) = 32T(\frac{n}{4}) + n^2\sqrt{n}$

Aufgabe H-4: Entwerfen Sie einen Algorithmus für das folgende Problem: für eine Menge M von reellen Zahlen und eine weitere reelle Zahl r ist zu entscheiden, ob sich r als Summe $r = s+t$ zweier (nicht notwendigerweise verschiedener) Elemente $s, t \in M$ schreiben lässt. Ihr Algorithmus sollte Laufzeit $\Theta(n \log n)$ haben. Zeigen Sie dies.

Abgabe bis Donnerstag, 30. April, 12 Uhr c.t. in einem der dafür vorgesehenen Briefkasten in der Oettingen- oder Theresienstraße. Bearbeitung entweder alleine oder (bevorzugt) in Gruppen von mehreren Personen.