Institut für Informatik
Prof. Dr. Heiko Röglin
Tobias Brunsch
Clemens Rösner

Algorithmen und Berechnungskomplexität I
Wintersemester 2013/14
Abgabe: 05.11.2013, 12:30 Uhr

Übungsblatt 3

Aufgabe 3.1

6 Punkte

In der Vorlesung haben wir uns auf die Laufzeitanalyse von BinarySearch konzentriert. Nichtsdestotrotz spielen Korrektheitsbeweise in der Algorithmik eine wichtige Rolle.

Beweisen Sie, dass BinarySearch korrekt ist.

Aufgabe 3.2

2+2+2 Punkte

Bestimmen Sie für die folgenden Rekursionsgleichungen für $T(n)$ mit $T(2) = T(1) = 1$ eine möglichst einfache Funktion $g(n)$ mit $T(n) = \Theta(g(n))$.

(a) $T(n) = 2 \cdot T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n^{1+1/n}$,

(b) $T(n) = 4 \cdot T\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + 2^n$,

(c) $T(n) = 3 \cdot T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \frac{n^2 \log_2(n)}{\sqrt{n}}$.

Aufgabe 3.3

6 Punkte

Wir wollen die Laufzeiten der naiven Matrixmultiplikation und des Strassen-Algorithmus empirisch vergleichen. Dabei stellen wir schnell technische Schwierigkeiten fest: Der Rekursionsschritt im Strassen-Algorithmus funktioniert nur, wenn die Anzahl der Zeilen und Spalten der zu multiplizierenden Matrizen gerade ist. Eine einfache Möglichkeit, das für jeden Rekursionsschritt zu garantieren, ist, die Ausgangsmatrizen vor dem Aufruf solange mit Nullzeilen und -spalten aufzufüllen, bis die Anzahl der Zeilen und Spalten Zweierpotenzen sind.

Implementieren Sie die naive Matrixmultiplikation und den Strassen-Algorithmus und stellen Sie deren Laufzeiten für die Multiplikation zweier $n \times n$-Matrizen graphisch dar. Bis zu welchen Werten n kommen Sie mit Ihren Implementierungen?

Hinweise:

- Brechen Sie die Rekursion im Strassen-Algorithmus bei kleinen Matrizen ab und gehen Sie zur naiven Matrixmultiplikation über.
- Da die Laufzeit für die Matrixmultiplikation bereits für kleine Zahlen n deutlich größer als die Laufzeit zum Auffüllen der Matrix ist, können Sie davon ausgehen, dass die Laufzeit des Strassen-Algorithmus für die Multiplikation zweier $n_1 \times n_1$- bzw. $n_2 \times n_2$-Matrizen mit $n_1, n_2 \in (2^k, 2^{k+1}]$ vergleichbar groß ist. Daher brauchen Sie den Strassen-Algorithmus für dieses Experiment nicht für beide Matrizen ausführen.

Aufgabe 3.4

2+2+2 Punkte

Seien $(a_n, \ldots, a_1)$ und $(b_n, \ldots, b_1)$ die Binärdarstellungen zweier natürlicher Zahlen a und b der Länge n, wobei n eine Zweierpotenz ist. Wir wollen die Binärdarstellung $(c_{2n}, \ldots, c_1)$ des Produktes $c = a \cdot b$ bestimmen.

(a) Geben Sie die Laufzeit der klassischen Schulmethode zur Multiplikation von a und b an.

(b) Geben Sie einen Divide-and-Conquer-Algorithmus zur Multiplikation von a und b an, der mit vier Multiplikationen von Zahlen der Länge $n/2$ auskommt. Welche Laufzeit hat diese Methode?

(c) Geben Sie einen Divide-and-Conquer-Algorithmus zur Multiplikation von a und b an, der mit drei Multiplikationen von Zahlen der Länge $n/2$ auskommt. Welche Laufzeit hat diese Methode?

Aufgabe 3.5

3+3 Zusatzpunkte

(a) Betrachten Sie die Rekursionsgleichung $T(1) = 1$ und $T(n) = 2 \cdot T(\lfloor n/2 \rfloor) + \gamma(n) \cdot n^2$ für $n \geq 2$, wobei $\gamma\colon \mathbb{N} \to \mathbb{N}$ die folgende Funktion ist:

$$\gamma(n) = \begin{cases} 1 & \text{falls } n \text{ gerade}\,, \\ 2 & \text{sonst}\,. \end{cases}$$

Bestimmen Sie eine möglichst einfache Funktion $g(n)$ mit $T(n) = \Theta(g(n))$.

(b) Bestimmen Sie für die Rekursionsgleichung $T(2) = 1$ und $T(n) = 4 \cdot T(\sqrt{n}) + \log_2(n)$ für $n \geq 3$ eine möglichst einfache Funktion $h(n)$ mit $T(n) = \Theta(h(n))$. Ignorieren Sie der Einfachheit halber Rundungsprobleme in der Analyse und beschränken Sie sich auf Zweierpotenzen n.

Aufgabe 3.6

6 Zusatzpunkte

Wir betrachten sogenannte *homogene lineare Differenzengleichungen 2. Ordnung*. Diese haben die Gestalt $R(0) = c_0$, $R(1) = c_1$ und $R(k) = a \cdot R(k-1) + b \cdot R(k-2)$ für $k \geq 2$. Das *charakteristische Polynom* einer solchen Differenzengleichung ist definiert als $p(z) = z^2 - a \cdot z - b$. Besitzt dieses Polynom zwei verschiedene Nullstellen $z_1 \neq z_2$, dann können wir $R(k)$ explizit angeben:

$$R(k) = \lambda \cdot z_1^k + \mu \cdot z_2^k\,.$$

Die Koeffizienten λ und μ hängen dabei von den Anfangsbedingungen $R(0)$ und $R(1)$ ab und können durch Lösen des linearen Gleichungssystems

$$\begin{aligned} \lambda \cdot 1 + \mu \cdot 1 &= c_0 \\ \lambda \cdot z_1 + \mu \cdot z_2 &= c_1 \end{aligned}$$

bestimmt werden.

Lösen Sie die Rekursionsgleichung $T(n) = 2 \cdot T(n/2) + 3 \cdot T(n/4) + n$ mit $T(1) = 0$ und $T(2) = 1$ in Θ-Notation. Ignorieren Sie Rundungsprobleme und konzentrieren Sie sich bei der Analyse auf Zweierpotenzen n.

Hinweis: Entledigen Sie sich zunächst des Summanden n, indem Sie statt $T(n)$ eine Funktion $S(n) = T(n) + h(n)$ für eine geeignete Funktion h betrachten. Bringen Sie die homogene Rekursionsgleichung für $S(n)$ anschließend auf obige Form.