Bergische Universität GH Wuppertal
Fachbereich Mathematik
Prof. Dr. A. Frommer, Holger Arndt

Informatik II: Algorithmen und Datenstrukturen

SS 2002

6. Übungsblatt

Aufgabe 18 (Fibonacci-Zahlen, 6T)

Die Fibonacci-Zahlen f_i, $i = 0, 1, \ldots$ sind folgendermaßen definiert:
$f_0 = 1$, $f_1 = 1$ und $f_i = f_{i-1} + f_{i-2}$ für $i \geq 2$.

a) Geben Sie $f_2, f_3, \ldots, f_8$ an.

b) Der folgende rekursive Algorithmus bestimmt f_n:

Eingabe: $n \in \mathbb{N}_0$
Ausgabe: f_n
Algorithmus fib_rek(n)
falls $n \leq 1$
    fib_rek(n) = 1
sonst
    fib_rek(n) = fib_rek($n-1$) + fib_rek($n-2$)

Zeigen Sie: Für den Zeitaufwand $T(n)$ von fib_rek gilt

$$T(n) = \Omega(f_n)\,.$$

Es ist bekannt (das brauchen Sie nicht zu zeigen):

$$f_i = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^{i+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{i+1} \right) .$$

Bestimmen Sie damit eine Zahl $\alpha > 1$, so dass

$$T(n) = \Omega(\alpha^n)\,.$$

c) Entwerfen Sie einen alternativen Algorithmus zur Berechnung von f_n mit Zeitaufwand $\mathcal{O}(n)$.

Aufgabe 19 (Aufwandsformel, 6T)

Beweisen Sie: Lässt sich der asymptotische Zeitaufwand nach der folgenden Rekursionsformel

$$T(n) = \begin{cases} \mathcal{O}(1) & \text{für } n = 1 \\ k \cdot T(n-1) + \mathcal{O}(n^\alpha) & \text{für } n > 1 \end{cases}$$

mit festem $k \in \mathbb{N}$, $\alpha \geq 0$ berechnen, so erhält man als explizite Formel

$$T(n) = \begin{cases} \mathcal{O}(n^{\alpha+1}) & \text{für } k = 1 \\ \mathcal{O}(k^n \cdot n^\alpha) & \text{für } k > 1\,. \end{cases}$$

Hinweis: Sie sollten auf ein Zwischenergebnis der Form

$$T(n) = k^{n-1} \cdot T(1) + (k^{n-2} + k^{n-3} + \ldots + 1) \cdot \mathcal{O}(n^\alpha)$$

stoßen.

Aufgabe 20 (einfache Aufwandsbestimmung, 6T)

Betrachtet wird der Divide-and-Conquer-Algorithmus *Maximale Teilsumme* aus der Vorlesung.

a) Beweisen Sie, dass für den Zeitaufwand die Beziehung

$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}(n)$$

gilt, wenn n eine Potenz von 2 ist (und die Bestimmung von i und j wie im zugehörigen C-Programm durchgeführt wird).

b) Zeigen Sie, dass für den Gesamtaufwand

$$T(n) = \mathcal{O}(n \log n)$$

gilt.

Abgabe: Mi., 29.05.2002, 14 Uhr