

Datenstrukturen und Algorithmen – Informatik I

4. Übung

Abgabe der Lösungen: 24. Mai 2004

Hinweis: Die Implementierungsaufgaben sind in Java oder C/C++ zu lösen. Der Quellcode sowie ein dokumentierter Beispiellauf sind beim Betreuer der Kleingruppe abzugeben.

Master-Theorem

Sei $f(n)$ eine Funktion und $T(n)$ definiert durch die Rekursion

$$T(n) = aT\left(\frac{n}{b}\right) + f(n),$$

wobei $a \geq 1$ und $b > 1$ reelle positive Konstanten sind. Statt $T\left(\frac{n}{b}\right)$ kann auch $T\left(\lfloor \frac{n}{b} \rfloor\right)$ oder $T\left(\lceil \frac{n}{b} \rceil\right)$ stehen. Dann läßt sich $T(n)$ in asymptotischer Schreibweise wie folgt ausdrücken:

1. Falls $f(n) \in O(n^{\log_b a - \varepsilon})$, $\varepsilon > 0$, dann ist $T(n) \in \Theta(n^{\log_b a})$.
2. Falls $f(n) \in \Theta(n^{\log_b a})$, dann ist $T(n) \in \Theta(n^{\log_b a} \cdot \log_b n)$
3. Falls $f(n) \in \Omega(n^{\log_b a + \varepsilon})$, $\varepsilon > 0$, und $af(n/b) \leq cf(n)$ mit $c < 1$ und genügend großem n , dann ist $T(n) \in \Theta(f(n))$.

Aufgabe 16: Rekursionsgleichungen

2+2+2 Punkte

Schätzen Sie folgende Rekursionsgleichungen möglichst genau ab. Formulieren Sie Ihre Antwort mit Hilfe der O - bzw. Θ -Notation.

- a) $T(n) = 3T(\lfloor \frac{n}{3} \rfloor + 28) + n$
- b) $T(n) = T(\frac{2n}{3}) + 1$
- c) $T(n) = 3T(\frac{n}{4}) + n \log n$

Aufgabe 17: Rekursionsgleichungen, Substitutionsmethode

2+2 Punkte

- a) Schätzen Sie die folgende Rekursionsgleichung möglichst genau ab. Benutzen Sie die O -Notation.

$$\begin{aligned} T(1) &= 1 \\ T(n) &= 3T(\lfloor \sqrt[3]{n} \rfloor) + \log_3 n. \end{aligned}$$

Tip: Wählen Sie eine geeignete Funktion $f(n)$ und definieren Sie die neue Variable $m := f(n)$. Ersetzen Sie dann in der Rekursionsgleichung n durch $f^{-1}(m)$. Lösen Sie die neue Rekursionsgleichung $S(m) = T(f^{-1}(m))$.

- b) Lösen Sie folgende Rekursionsgleichung exakt. *Tip:* Berechnen Sie die ersten Werte $T(n)$.

$$\begin{aligned} T(0) &= \alpha, \quad T(1) = \beta \\ T(n) &= \frac{1 + T(n-1)}{T(n-2)}. \end{aligned}$$

Aufgabe 18: Implementierungsaufgabe: Bitte mischen**3+1 Punkte**

- a) Implementieren Sie den MergeSort Algorithmus, der eine Zahlenfolge von n Zahlen sortiert.
- b) Testen Sie Ihre Implementierung an der folgenden Sequenz:
12 19 79 9 4 78 20 7 75 35 40 0 24 32 311 5 74 43 31 105 97 36 12 201

Aufgabe 19: Implementierungsaufgabe: Rucksackproblem**5+1 Punkte**

Ein Dieb mit einem Rucksack der Größe $M = 117$ bricht in einen Laden ein, knackt den Safe und findet dort $N = 6$ verschiedene Arten von Gegenständen (jeweils beliebig oft). Die Gegenstände haben folgende Eigenschaften:

Gegenstand	0	1	2	3	4	5
Wert	11	16	21	28	9	3
Größe	8	12	14	20	7	2

Der Dieb möchte natürlich Gegenstände mit so hohem Wert wie möglich stehlen, kann aber nur seinen Rucksack füllen und in jeder Hand einen zusätzlichen Gegenstand tragen. Um dieses Problem zu lösen, könnte er alle Kombinationsmöglichkeiten ausprobieren, dies ist jedoch zu aufwendig. Glücklicherweise steht neben dem Safe ein Computer. Er schreibt ein Programm, das ihm mittels dynamischer Programmierung die optimale Füllung (d.h. diejenige maximalen Werts) des Rucksacks berechnet.

Dabei geht er folgendermaßen vor:

```
for n=1...N:
  betrachte die Gegenstände E1...En
  for m=1...M:
    fülle mit den Gegenständen E1...En einen Rucksack der Größe m,
    benutze dazu bereits vorher berechnete Ergebnisse.
    Tausche ggfs. den zuletzt eingepackten Gegenstand aus.
```

Verfolge die Entscheidungen zurück, angefangen bei dem zuletzt eingepackte Teil, um die ideale Füllung auszugeben.

- a) Implementieren Sie dieses Programm und lösen Sie damit das gegebene Problem.
- b) In welchen anderen praktischen Anwendungen könnte dieses Problem relevant sein?