Entwurf und Analyse von Algorithmen

WS 1999/2000
Prof. Dr. K. Madlener
Übungsblatt 3

Aufgabe 3.1. a) Gegeben sei die Rekursionsgleichung

$$T(n) = \begin{cases} 0 & \text{falls } n = 0 \\ 3 \cdot T(\lfloor n/2 \rfloor) + n & \text{sonst} \end{cases}$$

für alle $n \in \mathbb{N}$. Lösen Sie diese Gleichung exakt für alle $n = 2^k$, $k \in \mathbb{N}$.

b) Gegeben sei die Rekursionsgleichung

$$T(n) = \begin{cases} 0 & \text{falls } n = 0 \\ 5 & \text{falls } n = 1 \\ 3 \cdot T(n-1) + 4 \cdot T(n-2) & \text{sonst} \end{cases}$$

für alle $n \in \mathbb{N}$. Lösen Sie diese Gleichung exakt für alle $n \in \mathbb{N}$.

Hinweis: Ermitteln Sie zunächst einige Anfangswerte, kommen Sie zu einer Vermutung und beweisen Sie diese durch Induktion.

c) Gegeben sei die Rekursionsgleichung

$$T(n) = \begin{cases} 1 & \text{falls } n = 1 \\ 5 \cdot T(n/2) + n \cdot \log_2 n & \text{sonst} \end{cases}$$

für alle $n \in \mathbb{N}$. Lösen Sie diese Gleichung exakt für alle $n = 2^k$, $k \in \mathbb{N}$. Geben Sie auch die Lösung möglichst einfach mit Hilfe der Θ-Notation an.

Hinweis: $\sum_{i=1}^{k} i \cdot x^i = \frac{x + x^{k+1}(kx - k - 1)}{(x-1)^2}$ für $x \in \mathbb{R}_0^+ \setminus \{1\}$.

Aufgabe 3.2. a) Sei H die aus der Vorlesung bekannte Entropie.
Zeigen Sie: Sind $(p_1, \ldots, p_n)$ und $(q_1, \ldots, q_n)$ Wahrscheinlichkeitsverteilungen, so gilt $H(p_1, \ldots, p_n) \leq -\sum_{i=1}^{n} p_i \log_2 q_i$, wobei Gleichheit genau dann gilt, wenn $(p_1, \ldots, p_n) = (q_1, \ldots, q_n)$.

b) Zeigen Sie die Kraft'sche Ungleichung: Sind $t_1, \ldots, t_n$ die Blatt-Tiefen aller Blätter in einem Binärbaum, so gilt $\sum_{i=1}^{n} 2^{-t_i} \leq 1$.

Aufgabe 3.3. Betrachten Sie folgenden probabilistischen Algorithmus A (probabilistisches InsertionSort). Sei $a : \{1, \ldots, n\} \to \{1, \ldots, n\}$ eine Permutation der Zahlen $\{1, \ldots, n\}$ mit $n \in \mathbb{N}$ mit $n > 2$. Wir wollen diese Permutation als einen Array der Größe n auffassen.

```
for k := 2 to n do
    repeat
        random i in [1..k]
    until sorted(a_1 ... a_{i-1}, a_k, a_i, ... a_{k-1})
    (a_1, ..., a_n) := (a_1 ... a_{i-1}, a_k, a_i, ... a_n)
end
```

a) Machen Sie eine average-case Analyse von obigen Algorithmus, d.h. bestimmen Sie $T_A^*(n) = \frac{1}{|\{x:|x|=n\}|} \sum_{|x|=n} T_A'(x)$

b) Realisieren Sie Algorithmus A in einer Programmiersprache Ihrer Wahl (etwa Java). Sie können dazu den auf unserer Vorlesungsseite angegebenen Rahmen verwenden. Die Bearbeitung dieses Aufgabenteils ist freiwillig.

Aufgabe 3.4. Wir betrachten folgendes Problem: Gegeben seien drei $n \times n$-Matrizen A, B und C mit Einträgen aus $\mathbb{Z}$. Wir wollen überprüfen, ob $AB = C$ (Matrixmultiplikation). Nach der naiven Methode geht das in $O(n^3)$ Schritten (wenn man die Addition und Multiplikation in $\mathbb{Z}$ als einen Schritt zählt). Betrachte nun folgenden probabilistischen Algorithmus (wobei X ein n-Vektor mit Einträgen aus $\{0, 1\}$ sei):

```
function Freivalds(A, B, C, n)
  for j := 1 to n do X_j := 0 oder 1 mit Wahrscheinlichkeit je 1/2 end
  if (XA)B = XC then
                                return true
                          else
                                return false
```

Zeigen Sie, dass dieser Algorithmus mit mindestens Wahrscheinlichkeit 1/2 die richtige Antwort auf die Frage „Ist $AB = C$?“ gibt. Zeigen Sie weiter, dass dieser Algorithmus nur $\Theta(n^2)$ Schritte benötigt (für die Matrixmultiplikation ist ein solch effizienter Algorithmus bisher nicht bekannt). Geben Sie abschließend einen Algorithmus für eine Funktion *Freivalds1* (A, B, C, n, ϵ) an, der nur mit einer Wahrscheinlichkeit von höchstens ϵ die falsche Antwort auf obige Frage gibt. Sehen Sie ein Problem mit diesem Algorithmus, wenn das ϵ „sehr klein“ ist?

Informationen zur Vorlesung:
http://www-madlener.informatik.uni-kl.de/ag-madlener/lectures/eaa/ws1999-2000/eaa.html