Datenstrukturen & Algorithmen, WS 1999/2000

Vorlesungsprüfung

28. März 2000

Es sind keinerlei Unterlagen und Hilfsmittel zugelassen. Zeit: 90 min

Notenschlüssel: Genügend ab 11 Punkten, Befriedigend ab 13 Punkten, Gut ab 15 Punkten, Sehr gut ab 17 Punkten. [20 Punkte möglich]

1. In einem doppelt indizierten Feld $\texttt{A[1..}n\texttt{,1..}n\texttt{]}$ sind ganze Zahlen so gespeichert, daß

   $$\texttt{A[}i\texttt{,}j\texttt{]} \leq \texttt{A[}i\texttt{,}j+1\texttt{]} \text{ für alle } i = 1, \ldots, n,\ j = 1, \ldots, n-1$$

   und

   $$\texttt{A[}i\texttt{,}j\texttt{]} \leq \texttt{A[}i+1\texttt{,}j\texttt{]} \text{ für alle } i = 1, \ldots, n-1,\ j = 1, \ldots, n.$$

   (a) Geben Sie einen möglichst effizienten Algorithmus an (Pseudo-Code), der überprüft, ob eine Zahl x in dem Feld `A` vorkommt.
   Ein einfacher Algorithmus hat Aufwand $O(n \log n)$ [2 Punkte]. Für einen optimalen Algorithmus mit Aufwand $O(n)$ erhalten Sie [4 Punkte].
   Hinweis für einen optimalen Algorithmus: Beginnen Sie die Suche bei $\texttt{A[1,}n\texttt{]}$. Schliessen sie nach Vergleich mit der gesuchten Zahl x einen Teil des Feldes aus, in dem die Zahl nicht vorkommen kann.

   (b) Zeigen Sie, daß der Aufwand $O(n)$ optimal ist. Überlegen Sie dazu, in welcher Beziehung zueinander die Elemente $\texttt{A[1,}n\texttt{]}$, $\texttt{A[2,}n-1\texttt{]}$, $\texttt{A[3,}n-2\texttt{]}$,..., $\texttt{A[}n\texttt{,1]}$ stehen. [1 Punkt]

2. Betrachten Sie einen rekursiven Algorithmus, der zuerst eine binäre Suche in einem Feld $\texttt{A[1..}n\texttt{]}$ durchführt und danach eine Hälfte des Feldes rekursiv weiter bearbeitet. Für die Zeitkomplexität dieses Algorithmus ergibt sich die Rekursionsgleichung

   $$T(n) = T\left(\frac{n}{2}\right) + b \cdot \log n\ ,\quad T(1) = c$$

   mit positiven Konstanten b und c.

   (a) Wie groß (in O-Notation) ist die Zeitkomplexität des Algorithmus? [2 Punkte]

   (b) Beweisen Sie ihre Antwort aus a). [3 Punkte]

3. Betrachten Sie die Zeichen `a,b,c,d,e,f,g,h` die mit den Häufigkeiten

z	a	b	c	d	e	f	g	h
$f(z)$	8	9	10	11	12	13	14	15

   in einem Text vorkommen.

   (a) Wieviele Bits (*ohne* der Codetabelle) benötigt man bei optimaler Codierung um diesen Text zu übertragen? Begründen Sie Ihre Antwort und geben Sie das Verfahren zur optimalen Codierung an. [3 Punkte]

(b) Betrachtet man die Häufigkeit der Zeichen `a,b,c,d,e,f,g,h` genauer, so erkennt man, daß $f(\mathtt{a}) \leq f(\mathtt{b}) \leq \cdots \leq f(\mathtt{h})$ und $2 \cdot f(\mathtt{a}) > f(\mathtt{h})$. Betrachten Sie nun den allgemeinen Fall für $n = 2^k$ (mit $k > 0$) Zeichen $z_1, \ldots, z_n$ mit $f(z_1) \leq \cdots \leq f(z_n)$ und $2 \cdot f(z_1) > f(z_n)$.
Zeigen Sie, daß in diesem Fall der optimale Code ein Blockcode mit $\lceil \log_2 n \rceil$ Bits pro Zeichen ist. [2 Punkte]

4. (a) Beschreiben Sie (möglichst prägnant) die Struktur von 2-4-Bäumen. [2 Punkte]

   (b) Welchen Vorteil haben 2-4-Bäume gegenüber natürlichen Bäumen bei der Verwendung als Suchbäume? [2 Punkte]

   (c) Welchen Nachteil haben 2-4-Bäume gegenüber natürlichen Bäumen? [1 Punkt]