Datenstrukturen & Algorithmen, WS 1999/2000

Vorlesungsprüfung

26. Mai 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. Im Geburtenregister einer Stadt sind die Geburten der letzten 100 Jahre verzeichnet. Das Geburtenregister soll nun so organisiert werden, daß Anfragen der Art

   Sind Person X und Person Y miteinander verwandt?

   schnell beantwortet werden können. Dabei gelten zwei Personen als verwandt, wenn Sie durch eine ununterbrochene Linie von Elternteil→Kind und Kind→Elternteil Beziehungen verbunden werden können. Überlegen Sie, welchem in der Vorlesung besprochenem Problem diese Fragestellung entspricht.

   (a) Geben Sie eine Datenstruktur an, mit deren Hilfe eine Vielzahl solcher Anfragen möglichst schnell beantwortet werden können. [2 Punkte]

   (b) Skizzieren Sie einen Algorithmus, der diese Datenstruktur möglichst schnell aufbaut. [2 Punkte]

   (c) Was ist der Gesamtaufwand Ihres Verfahrens, wenn das Geburtenregister n Geburten enthält und m Anfragen beantwortet werden müssen? [1 Punkt]

2. Geben Sie einen Algorithmus (Pseudocode) an, der den Wert $value(S)$ eines Infix-Ausdruckes S berechnet. Ist z.B. $S = 26 + 5 \times (30 - 10)$ so soll der Algorithmus 126 ausgeben. Der Infixausdruck S ist in Form eines linearen Feldes $S[1..n]$ gegeben, wobei ein Element $S[i]$ ein Zeichen der Menge $\{+, -, \times, (,), 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ sein kann. Ein syntaktisch korrekter Infix-Ausdruck ist von folgender Form:

$$\begin{aligned} \mathit{Infixausdruck} &:= \mathit{Summe} \\ \mathit{Summe} &:= \mathit{Term}[\{+|-\}\,\mathit{Term}]^* \\ \mathit{Term} &:= \mathit{Faktor}|\mathit{Faktor} \times \mathit{Faktor} \\ \mathit{Faktor} &:= \mathit{Zahl}|(\mathit{Summe}) \\ \mathit{Zahl} &:= \mathit{Ziffer}[\mathit{Ziffer}]^* \\ \mathit{Ziffer} &:= 0|1|2|3|4|5|6|7|8|9 \end{aligned}$$

   Sie können davon ausgehen, daß der vorliegende Infix-Ausdruck syntaktisch korrekt ist. [5 Punkte]

3. Betrachten Sie eine Halde $A[1..n]$ mit n Elementen (Definition wie im Skriptum, d.h. $A[1]$ enthält das Maximum).

   (a) An welcher Position i ($1 \leq i \leq n$) befindet sich das *kleinste* Element der Halde? [1 Punkt]

   (b) Geben Sie einen Algorithmus an, der ein beliebiges lineares Feld $A[1..n]$ in eine Halde $A[1..n]$ umwandelt. Der Algorithmus sollte eine Laufzeit von $O(n)$ aufweisen. [2 Punkte]

   (c) Zeigen Sie, daß der von Ihnen angegebene Algorithmus auch tatsächlich eine Laufzeit von $O(n)$ besitzt. [2 Punkte]

4. AVL-Bäume sind binäre Sortierbäume, sodaß für *jeden* Knoten K im Baum folgende Bedingung gilt:

   > Die Höhen des linken und des rechten von K ausgehenden Teilbaumes unterscheiden sich höchstens um 1.

   Zum Beispiel sind

   AVL-Bäume, die folgenden Bäume aber nicht:

   Berechnen Sie die maximale Höhe $H(n)$ eines AVL-Baumes mit n Knoten.

   (a) Geben Sie eine Rekursionsformel für die *minimale* Anzahl von Knoten, $N(h)$, in einem AVL-Baum der Höhe h an. [2 Punkte]

   (b) Welcher Rekursionsformel aus der Vorlesung sieht diese Rekursionsformel ähnlich? [1 Punkt]

   (c) Erstellen Sie eine Rekursionsformel für $F(h) := N(h) + 1$. Sie können nun das Ergebnis aus der Vorlesung direkt für die Berechnung von $F(h)$ und somit auch für die Berechnung von $N(h)$ benützen. [1 Punkt]

   (d) Berechnen Sie nun aus $N(h)$ die maximale Höhe $H(n)$. [1 Punkt]