

Frank Hoffmann
Klaus Kriegel
Romain Grunert
Ludmilla Scharf
André Schulz

Ws 2007/08

8. Dezember 2007
Abgabe: 17. Dezember 2007
vor der Vorlesung

8. Übung zur Vorlesung Höhere Algorithmik

1. Aufgabe (5 Punkte).

Gegeben ist eine Folge von aufsteigend sortierten Zahlen $S = (a_1, \dots, a_n)$. S soll in einem Suchbaum T abgespeichert werden. Die Werte aus S sollen dabei in inneren Knoten stehen und Blätter sollen erfolgloses Suchen repräsentieren. Ferner sei t_i die Tiefe des Knoten, der a_i speichert und $t_{(i,i+1)}$ die Tiefe des Blattes, welches das Intervall (a_i, a_{i+1}) repräsentiert. Analog dazu sind $t_{(-\infty, a_1)}$ und $t_{(a_n, \infty)}$ definiert.

Uns ist zusätzlich bekannt, welche Anfragen wie oft gestellt werden. Sei

$$\begin{aligned} \forall 0 < i \leq n : \quad p_i &= \text{Anzahl der Anfragen nach } a_i, \\ q_0 &= \text{Anzahl der Anfragen nach } a, \text{ mit } a < a_1, \\ q_n &= \text{Anzahl der Anfragen nach } a, \text{ mit } a_n < a, \\ \forall 1 < i < n : \quad q_i &= \text{Anzahl der Anfragen nach } a, \text{ mit } a_i < a < a_{i+1}. \end{aligned}$$

Die Kosten aller Anfragen berechnet sich als:

$$c(T) := \left(\sum_{i=1}^n p_i (t_i + 1) \right) + \left(\sum_{j=1}^{n-1} q_j t_{(j,j+1)} \right) + q_0 t_{(-\infty, a_1)} + q_n t_{(a_n, \infty)}.$$

Berechnen Sie einen Baum in $O(n^3)$ der $c(T)$ minimiert. Benutzen Sie dynamisches Programmieren.

2. Aufgabe (5 Punkte).

Wir betrachten Binärbäume mit Knoten $\{v_1, \dots, v_n\}$. Sei t_i die Tiefe von v_i . Wir definieren die durchschnittliche Tiefe $d(T)$ eines Baumes T als

$$d(T) := \frac{1}{n} \sum_{i=1}^n t_i.$$

Angenommen $d(T) = \Theta(\log n)$, wie groß kann dann die Tiefe eines einzelnen Blattes maximal werden?

3. Aufgabe (5 Punkte).

Wir betrachten eine Menge S von Binärzahlen. Ein Binärbaum heißt Radixbaum für S , wenn folgende Eigenschaften erfüllt sind:

- (a) Die ausgehenden Kanten sind jeweils mit 0 und 1 beschriftet.

- (b) Ein Knoten trägt als Beschriftung einen String aus $\{0,1\}^*$. Die Beschriftung entspricht der Konkatination der Kantenbeschriftungen von der Wurzel zum Knoten selbst.
- (c) Jedes Element aus S ist im Baum als Knotenbeschriftung vorhanden.
- (d) Jeder Knoten dessen Beschriftung ein Element aus S enthält, ist zusätzlich markiert.

Zeigen Sie, dass Sie die Strings aus S lexikografisch in linearer Zeit Sortieren können, wenn S als Radixbaum gegeben ist.

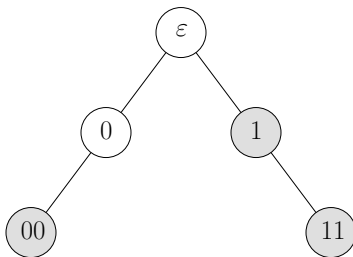


Abbildung 1: Radixbaum für die Menge $\{1, 00, 11\}$.

4. Aufgabe (5 Punkte).

Zum Hashing mit Sondierung benutzen wir die Hashfunktion

$$h(x, i) := (x + i^2) \pmod{m},$$

wobei die Grösse der Hashtabelle m eine Primzahl ist. Zeigen Sie, dass für jedes x nur die Hälfte aller Zellen in der Hashtabelle von $h(x, \cdot)$ adressiert werden kann.

Wir betrachten nun als verbesserte Variante die Hashfunktion

$$h^*(x, i) := \begin{cases} x + (i/2)^2 \pmod{m} & i \text{ ist gerade,} \\ x - ((i+1)/2)^2 \pmod{m} & i \text{ ist ungerade.} \end{cases}$$

Die Tabellengrösse m ist wiederum Primzahl und zusätzlich gelte: $m \pmod{4} = 3$. Zeigen Sie, dass nun jede Zelle aus der Hashtabelle genau einmal in der Sondierungsfolge $h^*(x, i), 0 \leq i < m$ auftritt.

Tipp: Überlegen Sie sich, dass eine Quadratzahl modulo 4 nur 1 oder 0 sein kann!