Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr

WS 2002/03
Übungsblatt 3
1. November 2002

Effiziente Algorithmen und Datenstrukturen I

Letzter Abgabetermin: 11. November 2002 (vor der Übung)

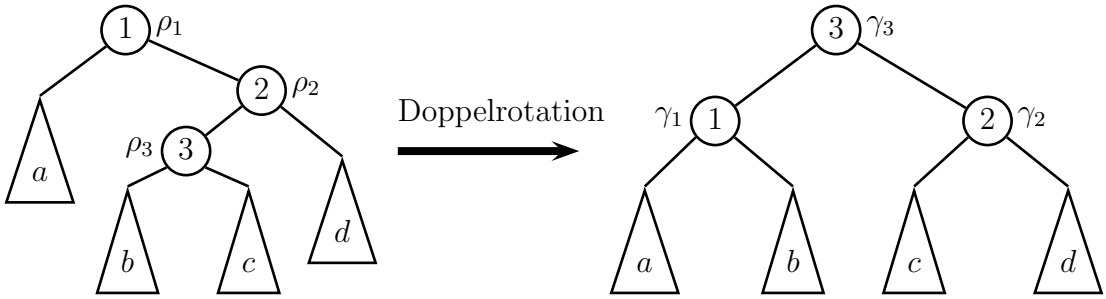
Aufgabe 1

In gewichtsbalancierten Binärbäumen ist das Gewicht $\rho(v)$ eines inneren Knotens v definiert als

$$\rho(v) = \frac{|T_l(v)|}{|T(v)|} = 1 - \frac{|T_r(v)|}{|T(v)|}.$$

Hierbei bezeichen $T(v)$ den Unterbaum mit Wurzel v, $T_l(v)$ bzw. $T_r(v)$ den linken bzw. rechten Unterbaum von v und $|T|$ die Anzahl der Blätter in einem Baum T.

Nach einer Doppelrotation in einem Binärbaum können die neuen Gewichte γ_i aus den alten Gewichten ρ_i berechnet werden (i bezeichne hierbei den Knotenindex, siehe nachfolgende Abbildung).



Geben Sie einen formalen Beweis für die Richtigkeit der folgenden drei Gleichungen an:

$$\gamma_1 = \frac{\rho_1}{\rho_1 + (1-\rho_1)\rho_2\rho_3} \qquad \gamma_2 = \frac{\rho_2(1-\rho 3)}{1-\rho 2\rho_3} \qquad \gamma_3 = \rho_1 + (1-\rho_1)\rho_2\rho_3.$$

Aufgabe 2

In der Vorlesung wurden AVL-Bäume als externe Suchbäume definiert, d.h. die gespeicherten Schlüssel sind nur in den Blättern abgelegt. Beschreiben Sie für diese Art der Realisierung die DELETE Operation.

Hinweis: Wird die Beschreibung mittels eines Programms in Pseudo-Code angegeben, ist dieses entsprechend zu kommentieren. Programme ohne Kommentar werden nicht akzeptiert.

Aufgabe 3

Im Gegensatz zu Aufgabe 2 soll nun ein AVL-Baum durch einen internen Suchbaum realisiert werden, D.h. alle in den inneren Knoten gespeicherten Werte sind nun ebenfalls Schlüssel, die im Baum gespeichert sind. Geben Sie für diese Art der Realisierung die INSERT Operation an. Sollte der Schlüssel bereits im Baum gespeichert sein, so soll dieser nicht verändert werden.

Hinweis: Wird die Beschreibung mittels eines Programms in Pseudo-Code angegeben, ist dieses entsprechend zu kommentieren. Programme ohne Kommentar werden nicht akzeptiert.

Aufgabe 4

Sei π eine Permutation auf den Zahlen $1, 2, \ldots, n$. Der Baum $T[\pi]$ ist definiert als der (natürliche) binäre Suchbaum, der entsteht, wenn man die Schlüssel $\pi(1), \pi(2), \ldots, \pi(n)$ in dieser Reihenfolge in einen anfangs leeren Baum einfügt. Zum Einfügen eines neuen Schlüssels wandert man im aktuellen Baum so lange nach unten (entsprechend der Suchbaumregel), bis man auf einen leeren Unterbaum stößt. An dieser Stelle wird ein neuer Knoten mit dem neuen Schlüssel eingefügt.

Die interne Pfadlänge $P(T[\pi])$ der Baumes $T[\pi]$ sei definiert als die Summe aller Tiefen $t(v)$ sämtlicher Knoten v des Baumes, d.h.

$$P(T[\pi]) = \sum_{v \text{ Knoten von } T[\pi]} t(v)$$

Zeigen Sie, dass $\sum_\pi \frac{P(T[\pi])}{n!} = \mathcal{O}(n \log n)$, d.h. die erwartete interne Pfadlänge eines zufällig erzeugten Baumes ist $\mathcal{O}(n \log n)$.

Hinweis: Das Prinzip aus der Analyse des QUICKSORT Algorithmus könnte auch hier sinnvoll sein.