Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Matthias Baumgart

Wintersemester 2006/2007
Übungsblatt 3
3. November 2006

Effiziente Algorithmen und Datenstrukturen I

Abgabetermin: 10.11.2006 **vor** *der Vorlesung*

Aufgabe 1 (10 Punkte)

Es seien T_1 und T_2 zwei (a, b)-Bäume mit n_1 bzw. n_2 Knoten, so dass für alle $x \in T_1$ und $y \in T_2$ gilt: $\text{key}(x) < \text{key}(y)$. Entwerfen Sie eine Prozedur CONCATENATE, die T_1 und T_2 zu einem neuen (a, b)-Baum verschmilzt und deren Laufzeit $O(\log(n_1 + n_2))$ ist.

Aufgabe 2 (10 Punkte)

(a) Zeigen Sie, dass die Tiefe eines Rot-Schwarz-Baumes mit n Blättern $O(\log n)$ ist.

(b) Beschreiben Sie die DELETE-Operation in Rot-Schwarz-Bäumen. Achten Sie hierbei darauf, dass der Baum auch nach der Operation ein Binärbaum sein muss. Zeigen Sie, dass die Laufzeit einer DELETE-Operation $O(\log n)$ ist.

Hinweis: Durch geeignete Fallunterscheidung kann der Baum so traversiert und modifiziert werden (vom zu entfernenden Blatt bis maximal zur Wurzel und wieder zurück), dass das Blatt mittels eines elementaren Schrittes entfernt werden kann.

Aufgabe 3 (10 Punkte)

Fügen Sie die Schlüssel $1, 2, \ldots, 9$ in der Reihenfolge

$$(1, 9, 5, 7, 8, 6, 3, 2, 4)$$

in einen anfänglich leeren AVL-Baum ein. Entfernen Sie anschließend den Schlüssel 9.

Zeichnen Sie jeweils den resultierenden Baum (einschließlich notwendiger Rotationen).

Aufgabe 4 (10 Punkte)

Angenommen, wir halten in einem AVL-Baum eine Referenz auf den linkesten inneren Knoten. Dann kann die Operation „Finde minimalen Schlüssel“ in konstanter Zeit ausgeführt werden. Zeigen Sie, dass sich eine solche Referenz in einem AVL-Baum ohne (asymptotischen) Mehraufwand, also mit konstantem Aufwand pro Schritt, aktuell halten lässt. Gehen Sie davon aus, dass keine gleichen Schlüssel eingefügt werden.