Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Hanjo Täubig

Sommersemester 2006
Übungsblatt 9
30. Juni 2006

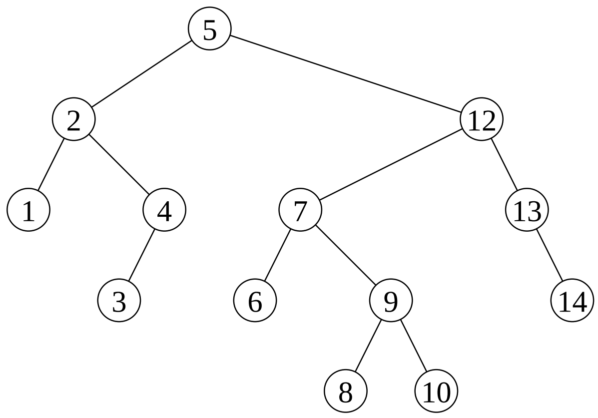
Informatik IV

Abgabetermin: 7. Juli 2006 **vor** *der Vorlesung*

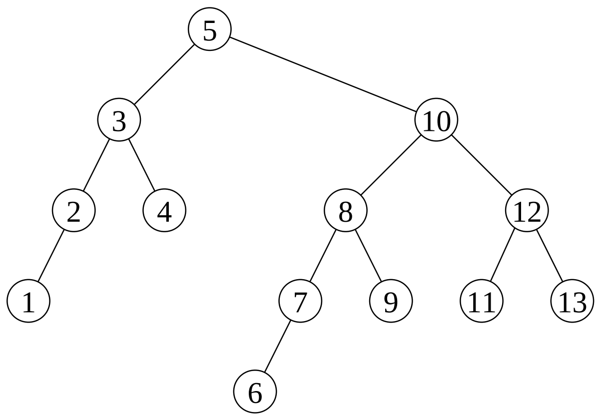
Aufgabe 1 (10 Punkte)

Geben Sie die einzelnen Schritte für folgende Operationen an:

a) Fügen Sie den Schlüssel 11 in den folgenden AVL-Baum ein:



b) Löschen Sie den Schlüssel 4 aus folgendem AVL-Baum:



Aufgabe 2 (10 Punkte)

Fügen Sie in einen anfangs leeren $(2, 3)$-Baum nacheinander die Schlüssel 10,17,13,6,8,3,1 ein. Entfernen Sie danach die Schlüssel 8 und 13. Stellen Sie die einzelnen Schritte dar.

Aufgabe 3 (10 Punkte)

Der Level eines Knotens in einem Baum $\mathcal{T}$ ist definiert als seine Distanz vom Wurzelknoten (also die Anzahl der Kanten auf dem Pfad zur Wurzel). Die externe Pfadlänge $E(\mathcal{T})$ des Baums sei definiert als die Summe der Levels aller Bätter. Die interne Pfadlänge $I(\mathcal{T})$ des Baums sei definiert als die Summe der Levels aller internen Knoten (Nicht-Bätter).

a) Beweisen Sie, dass in jedem Baum mit k internen Knoten, in dem jeder interne Knoten genau t Kinder hat, genau $(t-1)k+1$ Blätter existieren.

b) Bestimmen Sie die Maximalwerte für $E(\mathcal{T})$ und $I(\mathcal{T})$ über allen Binärbäumen $\mathcal{T}$ mit k internen Knoten.

Aufgabe 4 (10 Punkte)

Sei A ein Array mit n ganzen Zahlen. Wir nennen eine Zahl ein *dominierendes Element* in A, falls sie öfter als $\frac{n}{2}$-mal in A auftaucht.
Entwerfen Sie einen möglichst effizienten Algorithmus, der entscheidet, ob A ein dominierendes Element enthält, und der es ausgibt, falls es existiert.
Analysieren Sie die Laufzeit!
Hinweis: Es geht besser als Sortieren…