Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr/ Dr. Jens Ernst
Johannes Nowak

Wintersemester 2005/06
Übungsblatt 7
7. Dezember 2005

Grundlegende Algorithmen

Abgabetermin: 14.12.2005 **nach** *der Vorlesung*

Aufgabe 1 (10 Punkte)

a) Sei T ein Binärbaum. Wir definieren die Knotentiefe d_v eines Knotens in T als die Länge des Pfades (Anzahl der Kanten) von der Wurzel von T zu v. Die **Pfadlänge** von T ist die Summe aller Knotentiefen d_v der Knoten in T. Geben Sie einen **linearen** Algorithmus zur Berechnung der Pfadlänge von T an.
(*Hinweis*: Sie können davon ausgehen, dass die Kinder eines Knotens v über Pointer $left(v)$ und $right(v)$ in konstanter Zeit erreicht werden können.)

b) Die **interne Pfadlänge** $I(T)$ von T ist die Summe der Tiefen aller inneren Knoten von T. Die **externe Pfadlänge** $E(T)$ von T ist die Summe der Tiefen aller externen Knoten (Blätter) von T. Zeigen Sie, dass für einen echten Binärbaum auf n inneren Knoten gilt

$$E(T) = I(T) + 2n.$$

(*Hinweis*: In einem echten Binärbaum hat jeder Knoten entweder genau zwei Kinder (innere Knoten) oder ist ein Blatt.)

Aufgabe 2 (10 Punkte)

Gegeben sei ein Heap, der mittels der in der Vorlesung vorgefürten Methode linearisiert in einem Array A gespeichert ist. Element $A[1]$ enthält somit die Wurzel. Gegeben sei weiterhin ein Schlüsselwert x. Geben Sie einen Algorithmus an, der mit möglichst wenig Schlüsselvergleichen feststellt, ob x im Heap gespeichert ist oder nicht.
(*Hinweis:* vollständiges Durchsuchen von A kommt nicht in Betracht)

Aufgabe 3 (10 Punkte)

Seien H_1 und H_2 zwei Heaps gleicher Tiefe, und sei H_1 darüber hinaus ein vollständiger Binärbaum. Wie kann man mit möglichst wenig Aufwand einen Heap H erzeugen, der alle Schlüsselwerte von H_1 und H_2 enthält?