Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Dr. Riko Jacob
Michael Schnupp

Wintersemester 2007/2008
Übungsblatt 3
2. November 2007

Algorithmen für die Speicherhierarchie

Abgabetermin: 06.11.2007 **vor** *der Übung*

Datenstrukturen

Aufgabe 1 (2 Punkte)

In der Vorlesung wurde gezeigt, wie man einen Stack effizient im I/O-Modell implementieren kann. Dazu wurde ein Puffer der Größe $2B$ im Speicher verwendet.
Warum ist ein Puffer der Größe B problematisch?
Wäre ein Puffer von $1\frac{1}{2}B$ möglich? (Begründen Sie.)

Aufgabe 2 (5 Punkte)

B-Bäume haben die Eigenschaft, dass alle Blätter genau die Entfernung h zur Wurzel haben. Man kann sie effizient implementieren, wenn man versucht in jeder Ebene einen Verzweigungsfaktor von $\Theta(B)$ zu erzwingen. Dazu betrachten wir die folgende Invariante: Ein Knoten mit Abstand $i < h$ zur Blattebene hat zwischen $(\frac{B}{8})^i$ und $4(\frac{B}{8})^i$ Blätter in seinem Teilbaum. Unterhalb der Wurzel ($i = h$), d.h. im ganzen Baum befinden sich höchstens $4(\frac{B}{8})^i$ Blätter. (Wir gehen davon aus, dass stets $B/8 \geq 4$.)
Zeigen Sie, dass aus dieser Invariante folgende Punkte folgen:

- Jeder Knoten hat höchstens $B/2$ Kinder.
- Die Höhe des Baumes h ist höchstens $1 + \lceil \log_{B/8} N \rceil$.
- Jeder Knoten (außer Wurzel und Blätter) hat mindestens $B/32$ Kinder.

Aufgabe 3 (3 Punkte)

Wir haben N Elemente auf der Platte und möchten ein bestimmtes Element finden. Dabei gehen wir davon aus, dass wir Elemente nur vergleichen können und der Hauptspeicher zu Beginn leer ist.
Zeigen Sie dass, unabhängig von der verwendeten Datenstruktur stets

$$\Omega(\log_B N)$$

I/Os benötigt werden.