Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Dr. Riko Jacob
Gero Greiner

Sommersemester 2009
Übungsblatt 4

Algorithmen für die Speicherhierarchie

Abgabetermin: 20.5.2009 **vor** *der Vorlesung*

Aufgabe 1

In der Vorlesung wurde die Handhabung eines Stacks mit maximaler Größe N und zwei reservierten Blöcken im internen Speicher vorgestellt.

a) Zeigen Sie, dass $\mathcal{O}(1/B)$ eine obere Schranke für die amortisierte Anzahl der I/Os pro Stack-Operation ist.

b) Was gilt für die Anzahl der I/Os pro Stack-Operation, wenn nur ein Block im Hauptspeicher benutzt werden darf.

Aufgabe 2

Beschreiben Sie eine Implementierung eines Stacks, so dass die amortisierte Schranke von $\mathcal{O}(1/B)$ I/Os pro Stack-Operation gilt und die Größe des Stacks im externen Speicher dennoch variabel ist.

Aufgabe 3

Für die Verwendung von B-Bäumen im I/O-Modell kann folgende Balance-Invariante verwendet werden. Sei h die Tiefe des B-Baumes und für einen Knoten v mit Tiefe j sei $i := h - j$ der Level von v (d.h. alle Blätter haben Level 0, die Wurzel soll Level h besitzen). Weiterhin wird mit dem Gewicht von v die Anzahl der Blätter in dem Unterbaum mit Wurzel v bezeichnet. Dann darf das Gewicht eines Knotens v mit Level $i \leq h$ höchstens $4(B/8)^i$ betragen und für $i < h$ mindestens $(B/8)^i$. Zeigen Sie, dass dies die folgenden drei Aussagen impliziert:

a) Jeder Knoten hat höchstens $B/2$ Kinder.

b) Die Tiefe des B-Baums beträgt höchstens $1 + \lceil \log_{B/8} N \rceil$.

c) Jeder Knoten außer der Wurzel hat mindestens $B/32$ Kinder.

d) Gibt es eine passende Wahl der Parameter a und b des klassischen B-Baumes, welche äquivalent zur der definierten Invariante ist?

Aufgabe 4

a) Wie groß muss der interne Speicher mindestens sein, damit sich die Terme $\mathcal{O}(\log_B(N/M))$ und $\mathcal{O}(\log_B N)$ asymptotisch unterscheiden.

b) Eine einfache untere Schranke für die Suche in einem binären Baum wäre $\mathcal{O}((\log N)/B)$. Ist dies realisierbar?