Technische Universität München
Institut für Informatik
Prof. Dr. J. Esparza/Dr. S. Gerke

WS 2000/01
Übungsblatt 3
7. November 2000

Effiziente Algorithmen und Datenstrukturen

1. Kontrollblatt

(Abgabetermin: Dienstag, den 14. November 2000, in der Vorlesung)

Wenn Sie einen Schein erwerben wollen, schreiben Sie bitte Ihren Namen und Ihre Matrikelnummer leserlich auf Ihre Lösungen.

Aufgabe 1

Zeigen Sie, daß es keine Konstante $c > 1$ gibt, so daß gilt:

> Sei T ein $(2,4)$-Baum mit n Blättern, und sei v ein Knoten in T der Tiefe d. Dann ist die Anzahl der Blätter in v's Unterbaum von oben (bzw. von unten) durch n/c^d beschränkt.

(Tip: Betrachten Sie einen $(2,4)$ Baum, dessen Wurzel zwei Kinder u, v hat und dessen Unterbaum, der an dem Knoten u gewurzelt ist, sowenig Blätter wie möglich besitzt und dessen Unterbaum, der an dem Knoten v gewurzelt ist, soviele Blätter wie möglich besitzt. Betrachten Sie insbesondere den Fall $d = 1$.)

Aufgabe 2

Entwerfen Sie eine Datenstruktur für zweidimensionale Daten (d.h. aus einem Universum $U = V \times V$ mit linearer Ordnung auf V), so daß folgende Operationen in $O(\log n)$ (wobei n die Anzahl der Elemente in der Datenstruktur ist) ausgeführt werden können:
einfüge$((a, b))$,*suche*$((a, b))$.
Sei $N_1(a)$ die Anzahl der Elemente mit erster Komponente a und $N_2(b)$ die Anzahl der Elemente mit zweiter Komponente b. Die folgenden Operationen sollen in der Laufzeit $O(\log n + N_1(a) \log n)$ bzw. $O(\log n + N_2(b) \log n)$ ausgeführt werden können:
streiche$((a, *))$ (d.h. Streichen aller Elemente mit erster Komponente a), *streiche*$((*, b))$.

Aufgabe 3

Zeigen Sie: Die Tiefe eines Rot-Schwarz-Baumes mit n Blättern ist $O(\log n)$.
(Tip: Sei h die Tiefe des höchsten Blattes. Zeigen Sie, daß die Tiefe des Baumes höchstens $2h$ ist und daß $h \leq \lceil \log n \rceil$)