Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr/Dr. Sven Kosub
Moritz Maaß/Stefan Eckhardt

Wintersemester 2003/04
Übungsblatt 5
5. Dezember 2003

Effiziente Algorithmen und Datenstrukturen I

Abgabetermin: 12.12.2003 **vor** *der Vorlesung*

Aufgabe 1

Wir betrachten Double Hashing zur Auflösung von Kollisionen. Dazu benutzen wir die Hash-Funktion $h(k,i) = (h_1(k) + ih_2(k)) \bmod m$. Zeigen Sie, dass für einen Schlüssel k genau dann $1/d$-tel der Hashtabelle untersucht wird, wenn d der größte gemeinsame Teiler von $h_2(k)$ und m ist. D.h. insbesondere, dass die gesamte Hashtabelle durchsucht wird, wenn $h_2(k)$ und m relativ prim sind.

Aufgabe 2

Wir betrachten einen binären Suchbaum, aus dem wir einen Schlüssel k löschen wollen. Wir nehmen an, dass der Schlüssel in einem inneren Knoten gespeichert ist, dessen Kinder ebenfalls innere Knoten sind. Geben Sie einen Algorithmus an, der den Schlüssel k entfernt und die Suchbaumeigenschaft wieder herstellt.

Aufgabe 3

Nehmen Sie an, dass in einem Wörterbuch Schlüssel mehrfach auftreten können. Beschreiben Sie einen Algorithmus für die Operation „Finde alle Objekte mit Schlüssel k“ in einem durch binäre Suchbäume implementierten geordneten Wörterbuch. Zeigen Sie, dass Ihr Algorithmus die Laufzeit $O(h+s)$ hat, wobei h die Tiefe des Suchbaumes T und s die Anzahl der gefundenen Elemente ist.

Aufgabe 4

Angenommen, wir halten in einem AVL-Baum eine Referenz auf den linkesten inneren Knoten. Dann kann die Operation „Finde minimalen Schlüssel“ in konstanter Zeit ausgeführt werden. Zeigen Sie, dass sich so eine Referenz in einem AVL-Baum ohne (asymptotischen) Mehraufwand, also mit konstantem Aufwand pro Schritt, aktuell halten lässt. Gehen Sie dabei davon aus, dass keine gleichen Schlüssel eingefügt werden.