Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Christian Scheideler
Stefan Eckhardt

Wintersemester 2005/06
Übungsblatt 4
19. November 2005

Effiziente Algorithmen und Datenstrukturen I

Abgabetermin: 18.11.2004 **vor** *der Zentralübung*

Aufgabe 1 (10 Punkte)

Ser $rep_s : \{0,1\}^s \to [0 \dots 2^s - 1]$ (mit $s \in \mathbb{N}$) die Funktion, die einen binären Vektor der Länge s auf die Zahl aus $\mathbb{Z}$ abbildet, die entsteht, wenn man den Vektor als Binärzahl interprätiert, d.h. $rep_s(a_{s-1}, \dots, a_0) = \sum_{i=0}^{s-1} a_i s^i$. Sei $N = 2^r$. Die Grundmenge $[0 \dots N-1]$ soll in einer Hash-Tabelle der Größe $m = 2^s$ gespeichert werden. Die Hashfunktion $h_A : [0 \dots N-1] \to [0 \dots m-1]$ ist für eine $s \times r$ Matrix A mit Einträgen aus $\{0,1\}$ definiert als $h_A(x) = rep_s(A \cdot rep_r^{-1}(x))$. d.h. $h_A(x)$ berechnet das Matrixprodukt (über den Körper $\mathbb{Z}_2$) von A und dem Ziffernvektor der Binärdarstellung von x und interpretiert das Ergebnis als Binärzahl. Die Menge $\mathcal{H} = \{h_A | A \in \{0,1\}^{s \times r}\}$ beinhalte alle diese Funktionen.

1. Zeigen Sie, dass $\mathcal{H}$ eine universelle Familie von Hashfunktionen ist.
   Hinweis: Seien $x, y \in [0 \dots N-1]$ mit $x \neq y$. Dann gilt $h_A(x) = h_A(y)$ genau dann, wenn $A(\vec{x} - \vec{y}) = 0$ (mit $\vec{x} = rep^{-1}(x)$ und $\vec{y} = rep^{-1}(y)$) Da die Vektoren der Dimension r einen Vektorraum über $\mathbb{Z}_2$ bilden, kann man aus $A(\vec{x} - \vec{y}) = 0$ schließen, dass alle Zeilen von A orthogonal zu $\vec{x} - \vec{y}$ sind.

2. Sein $S \subseteq [0 \dots N-1]$ $|S| = n$ und $x \in [0 \dots N-1]$. Berechnen Sie mittels des Ergebnisses aus Teilaufgabe a) den Erwartungswert
   $$\mu = \frac{\sum_{h \in \mathcal{H}} \delta_h(x, S)}{|\mathcal{H}|}$$
   für die erwartete Anzahl der Kollisionen $\delta_h(x, S)$, die beim Einfügen von x in eine Hash- Tabelle (in der bereits die Menge S gespeichert ist) auftreten, wenn die zufällig gewählte Hashfunktion $h \in \mathcal{H}$ verwendet wird.

Aufgabe 2 (10 Punkte)

Wir betrachten Double Hashing zur Auflösung von Kollisionen. Dazu benutzen wir die Hash-Funktion $h(k, i) = (h_1(k) + i \cdot h_2(k)) \bmod m$. 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 3 (10 Punkte)

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.