Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr

WS 2002/03
Übungsblatt 5
15. November 2002

Effiziente Algorithmen und Datenstrukturen I

Letzter Abgabetermin: 25.November 2002 (vor der Übung)

Aufgabe 1

In dieser Aufgabe soll eine Folge S von Zahlen aus $[0..999]$ in eine Hash-Tabelle der Größe n eingetragen werden. Hierbei wird offenes Hashing mit den drei folgenden Hashfunktionen verwendet.

$$\begin{aligned} h_1(x,i) &= (\ x+i\ ) \mod n \\ h_2(x,i) &= (\ x+i^2\ ) \mod n \\ h_3(x,i) &= (\ x+\ i\cdot(1+(x \mod (n-1)))\ ) \mod n \end{aligned}$$

Geben Sie für die folgenden Belegungen von n und S, jeweils für alle Hashfunktionen, die resultierende Hash-Tabelle sowie die Anzahl der auftretenden Kollisionen beim Einfügen von S in eine anfangs leere Hash-Tabelle an. Begründen Sie, warum die Funktion h_2 als Hashfunktion ungeeignet ist.

n	S
10	253,401,5,693,75,134,713
15	914,811,2,87,898,933,222,563,719, 781,256,938
20	56,39,61,371,287,896,603,290,678, 856,600,778,671,500,998,644,418

Aufgabe 2

Im Folgenden werde für das offene Hashing ein Array T der Länge m betrachtet, in dem $n < m$ Werte gespeichert sind. Es gelte die Annahme, dass die Sequenz $[h(x,i)]$ mit $i = 0,1,\ldots,m-1$ eine zufällige Permutation über der Menge $\{0,1,\ldots,m-1\}$ ist. Die Kosten für die Operation $\textsc{Insert}(x)$ sei $1+\min\{i \mid T[h(x,i)]$ noch nicht besetzt$\}$.

Zeigen Sie folgende Aussage:

Die erwarteten Kosten für eine INSERT*-Operation in eine Hash-Tabelle mit Füllgrad* $\beta = \frac{n}{m} < 1$ *ist* $\frac{m+1}{m-n+1} \approx \frac{1}{1-\beta}$.

Aufgabe 3

Es gelten die Voraussetzungen aus Aufgabe 2. Zeigen Sie nun folgende Behauptung.

> *Die erwarteten Kosten für eine erfolgreiche* SEARCH-*Operation in einer Hash-Tabelle mit Füllgrad* $\beta = \frac{n}{m}$ *ist* $\mathcal{O}\left(\frac{1}{\beta}\log(\frac{1}{1-\beta})\right)$.

Hinweis: Die Aussage aus Aufgabe 2 kann zur Lösung dieser Aufgabe verwendet werden.

Aufgabe 4

Sei $\text{rep}_s : \{0,1\}^s \to [0 \mathrel{..} 2^s - 1]$ (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 interpretiert, d.h. $\text{rep}_s(a_{s-1}, \ldots, a_0) = \sum_{i=0}^{s-1} a_i s^i$.

Sei $N = 2^r$. Die Grundmenge $[0 \mathrel{..} m-1]$ soll in einer Hash-Tabelle der Größe $m = 2^s$ gespeichert werden sollen. Die Hashfunktion $h_A : [0 \mathrel{..} N-1] \to [0 \mathrel{..} m-1]$ ist, für eine $s \times r$ Matrix A mit Einträgen aus {0,1}, definiert als $h_A(x) = \text{rep}_s(A \cdot \text{rep}_r^{-1}(x))$, 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 $\mathcal{H} = \{h_A \mid A \in \{0,1\}^{s \times r}\}$ beinhalte alle diese Funktionen.

In der Vorlesung wurde eine Menge $\mathcal{F}$ von Hashfunktionen $f : [0 \mathrel{..} N-1] \to [0 \mathrel{..} m-1]$ als universell definiert, wenn für alle Werte $x, y \in [0 \mathrel{..} N-1]$ mit $x \neq y$ gilt:

$$|\{f \in \mathcal{F} \mid f(x) = f(y)\}| \;\leq\; \frac{|\mathcal{F}|}{m}$$

a) Zeigen Sie, dass H eine universelle Menge von Hashfunktionen ist.

   Hinweis: Seien $x, y \in [0 \mathrel{..} N-1]$ mit $x \neq y$. Dann gilt $h_A(x) = h_A(y)$ genau dann, wenn $A(\vec{x} - \vec{y}) = 0$ (mit $\vec{x} = \text{rep}_r^{-1}(x)$ und $\vec{y} = \text{rep}_r^{-1}(y)$). 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.

b) Sei $S \subseteq [0 \mathrel{..} N-1]$, $|S| = n$ und $x \in [0 \mathrel{..} N-1]$. 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.