Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Matthias Baumgart

Wintersemester 2007/2008
Übungsblatt 4
9. November 2007

Effiziente Algorithmen und Datenstrukturen I

Abgabetermin: 16.11.2007 **vor** *der Vorlesung*

Aufgabe 1 (10 Punkte)

In der Vorlesung wurde die durchschnittliche Anzahl der Sondierungen bei erfolgreicher Suche mit Double Hashing mit

$$\frac{1}{\alpha} \ln \frac{1}{1-\alpha}$$

angegeben, wobei wir voraussetzen, dass jede Hashposition gleich wahrscheinlich ist und auf jeden Schlüssel mit gleicher Wahrscheinlichkeit zugegriffen wird.

Leiten Sie diese Formel her.

Aufgabe 2 (10 Punkte)

Beim *Robin Hood Hashing* wird versucht, die für eine erfolgreiche Suche maximal benötigten Vergleiche möglichst gering zu halten. Bei einer Kollision zweier Elemente wird nun nicht mehr automatisch das zuletzt eingefügte Element weiter bewegt, sondern das Element, das bisher weniger Sondierungsschritte benötigt hat.

Beispiel: Ein Element k_1 wurde im j_1-ten Sondierungsschritt an Position $h(j_1, k_1)$ gespeichert. Kollidiert nun ein Element k_2 im j_2-ten Sondierungsschritt mit k_1, d.h., es gilt $h(j_1, k_1) = h(j_2, k_2)$, dann gehen wir wie folgt vor. Gilt $j_1 < j_2$, dann wird k_2 an Position $h(j_2, k_2)$ gespeichert und für k_1 eine neue Position gesucht (entsprechend seiner Sondierungsfolge). Anderenfalls lässt man k_1 gespeichert und sucht für k_2 weiter.

Fügen Sie die Schlüssel $10, 22, 31, 4, 15, 65, 28, 39, 50$ in eine Hashtabelle der Länge $n = 11$ ein, wobei Sie eine Kollision durch die Robin Hood Strategie auflösen. Die Sondierungsfolge sei gegeben durch $h(j,k) = (h_1(k) - jh_2(k)) \bmod n$, wobei $h_1(k) = k \bmod n$ und $h_2(k) = 1 + (k \bmod (n-1))$.

Aufgabe 3 (10 Punkte)

Die 32-Bit Version des Fowler-Noll-Vo Hashs berechnet sich wie folgt:

$hash = 2166136261$
for each *byte_of_data* **do**
  $hash = (hash \cdot 16777619) \bmod 2^{32}$
  $hash = hash \oplus byte\_of\_data$
return *hash*

Bei einer 32-Bit Eingabe wird diese in 4 Blöcke zu je einem Byte aufgeteilt. Diese Blöcke werden nacheinander innerhalb der Laufschleife mit den 8 niedrigwertigsten Bits von *hash* durch ein XOR ($\oplus$) verknüpft.

Überprüfen Sie mindestens 1000 verschiedene 32-Bit Zahlen hinsichtlich des Lawineneffektes (die Änderung eines Bits in der Eingabe bewirkt eine Änderung von mindestens der Hälfte der Bits der Ausgabe). Verändern Sie dazu jedes der 32 Bits dieser Zahlen und berechnen Sie, wie viele Bits des Hashwertes sich dadurch ändern.

Stellen Sie Ihre Ergebnisse graphisch dar.

Aufgabe 4 (10 Punkte)

Es sei $U = \{0, 1, \dots, n-1\}$ für eine Primzahl n. Wir definieren $\mathcal{H} = \{h_\alpha \colon \alpha \in U^{r+1}\}$, wobei

$$h_\alpha(x) = \sum_{i=0}^{r} \alpha_i x^i \bmod n \qquad \text{für } x \in U\ .$$

Zeigen Sie, dass $\mathcal{H}$ universell ist.