



9. ÜBUNG ZUR VORLESUNG PROGRAMMIERUNG II, SS 2004

Aufgabe 1: Hashfunktion (10 Punkte)

Gegeben sei die Menge von Hashfunktionen

$$H^{fool} = \{h_{(a,b)} : a, b \in 0 \dots m-1\}$$

mit $h_{(a,b)}(x) = ax + b \mod m$. Und N die Anzahl aller Schlüssel.

Zeigen Sie, dass es eine Menge M von $\lfloor \frac{N}{m} \rfloor$ Schlüsseln gibt, so dass gilt:

$$\forall x, y \in M : \forall h_{(a,b)} \in H^{fool} : h_{(a,b)}(x) = h_{(a,b)}(y)$$

auch wenn m prim ist.

Aufgabe 2: Löschen eines Elements (5 Punkte)

In der Vorlesung wurde auf das Problem beim Löschen eines Eintrags aus einer Hashtabelle bei Verwendung von offener Adressierung hingewiesen. Dabei wurde die Variante der Markierung des Elements vorgestellt.

Erklären Sie kurz die Problematik, die beim Löschen auftreten kann und finden Sie eine weitere Möglichkeit das Problem bei linear probing zu umgehen.

Aufgabe 3: Anwendung Hashfunktion (5 Punkte)

Es gibt Firmen, die jeden Tag Millionen von Dateien von ihren tausenden Servern liefern. Diese Dateien werden für andere Firmen ausgeliefert, die für jeden Download bezahlen müssen. Die Server produzieren dabei Log-Dateien, welche für jeden Download eine Zeile mit der Information über Preis und Kunden enthalten.

Beschreiben Sie einen Algorithmus, der die Log-Dateien einmal am Tag einliest und die Kosten pro Teilnehmer (einige Tausend) ausgibt. Dies soll mit maximaler Effizienz erfolgen.

Aufgabe 4: Kuckuck-Hashing (10 Punkte)

Eine weitere Variante des Hashing stellt das Kuckuck-Hashing dar. Man verwendet dabei zwei Arrays t_1 und t_2 , in denen Daten mit zwei unterschiedlichen Hashfunktionen $h_1(x)$ und $h_2(x)$ gespeichert werden. Ein neues Element wird entweder in Array t_1 mit $h_1(x)$ oder in Array t_2 mit $h_2(x)$ abgelegt. Will man ein neues Element einfügen, aber die Zelle in t_1 ist schon belegt, verhält sich der Algorithmus wie ein Kuckuck: Das aktuelle Element

wird durch das neue Element ersetzt. Das alte Element nimmt den Platz im Array t2 ein, den die zweite Hashfunktion bestimmt. Eventuell wird damit wieder ein Element ersetzt, welches dann zurück in das Array t1 wandert. Dieser Prozess geht solange weiter, bis eine leere Zelle gefunden wurde.

- i) Schreiben Sie die Insert-Funktion in Pseudocode oder C++ und geben Sie diese schriftlich ab.
- ii) Kuckuck-Hashing hat ein Problem. Überlegen Sie sich welches und geben Sie ein Beispiel dafür an.

Aufgabe 5: Binärer Suchbaum (5 Punkte)

Beweisen Sie induktiv, dass die aus der Vorlesung bekannte Funktion `tree_walk(TreeNode *x)` korrekt ist, also, dass die Funktion auf die Wurzel angewendet den Baum korrekt sortiert ausgibt.

Aufgabe 6: Kommutativität (5 Punkte)

Ist das Löschen im binären Suchbaum kommutativ, also führt das Löschen von x und dann y in einem binären Suchbaum zum gleichen Baum, wie das Löschen von y und dann x? Begründen Sie warum, oder geben Sie ein Gegenbeispiel!

Abgabe: 25.06.2004 (Vor der Vorlesung)