Prof. G. Zachmann
David Mainzer

TU Clausthal
Institut für Informatik
2. Juni 2010

Sommersemester 2010

Übungen zu Informatik II - Blatt 8

Abgabe in der Übung am 8. 06 / 9. 06. 2010

Bitte beachten Sie, dass die Programmieraufgaben von Ihnen in der Übung vorgeführt und erklärt werden müssen. Zusätzlich senden Sie die Lösung, unter Angabe ihres Namens, an **dm@tu-clausthal.de**.

Aufgabe 1 (Hashtabellen in Python, *0.5+4.5+2 Punkte*)

Schreiben Sie ein Programm, welches Hashing mit $h_{a,m}(x) = x \mod m$ implementiert. Für die Kollisionsbehandlung soll *Open Addressing* mit linearer Sondierung ($s(j,k)=j$) verwendet werden. Implementieren Sie dazu:

a) eine Klasse *TableEntry*, welche die Datenstruktur für einen Eintrag der Hashtabelle enthält,

b) eine Klasse *HashTable*, welche die Hashtabelle selbst implementiert. Die Klasse soll die Funktion *h(self,key)*, *s(self,j,k)*, *insert(self,key)* und *delete(self,key)* enthalten.

Schreiben Sie ein Testprogramm, welches die Hashtabelle mit gleichverteilten Zufallszahlen zu 50% füllt und danach $2 \cdot m$ mal abwechselnd ein neues Element in die Hashtabelle einfügt und ein Element aus der Hashtabelle löscht, d.h. $4m$(*insert(), delete()*). Auch hier sollen je ein zufälliger Werte eingefügt und ein zufällig gewähltes Element, das in der Hashtabelle enthalten ist, gelöscht werden. Geben Sie die Anzahl der Kollisionen für *insert* und *delete* aus. Führen Sie die Tests für m = 11, 101 und 1009 durch. Mitteln Sie je 10 Werte (Anzahl an Kollisionen) für m = 1009.

Aufgabe 2 (Worst-Case-Betrachtung, *2 Punkte*)

Wie viele Schritte werden im schlechtesten Fall benötigt um in eine anfangs leere Hashtabelle n Schlüssel einzufügen, wenn für die Kollisionsbehandlung *Chaining* verwendet wird? Wie viele Schritte benötigt man um nach jedem der n eingefügten Schlüssel einmal zu suchen?

Aufgabe 3 (Divisions-Rest-Methode, *1+2 Punkte*)

Betrachten Sie die Hashfunktion $h_{a,m}(x) = (a \cdot x) \mod m$, mit m prim.

a) Zeigen Sie, daß $\{h_{a,m} | 0 \leq a < m\} = \{h_{a,m} | a \in \mathbb{Z}\}$.

b) Zeigen Sie weiterhin, daß $h_{a,m}(x)$, $x = 0, 1, 2, 3, \ldots$ mit $a \neq c \cdot m, c \in \mathbb{Z}$, Periode m hat und jede Periode eine Permutation von $(0, 1, \ldots, m-1)$ darstellt. D.h. solch ein h ist eine gute Hash-Funktion, falls die Keys alle in $[0, m-1]$ liegen, oder gleichverteilt aus $\mathbb{Z}$ kommen, aber eine schlechte Hash-Funktion, falls die Keys alle in der Nähe von $b \cdot \mathbb{Z}$ liegen.

Aufgabe 4 (Quadratisches Sondieren, *2 Punkte*)

Zeigen Sie, dass die mittlere Anzahl von Hashtabellenplätzen, die bei einer erfolgreichen Suche (mit gleicher Wahrscheinlichkeit für alle Schlüssel) inspiziert werden, bei Hashing mit linearem Sondieren nicht von der Reihenfolge abhängt, in der die Schlüssel in die anfangs leere Hashtabelle eingefügt worden sind. Gilt die entsprechende Aussage auch für quadratisches Sondieren?