



**1. Begriffserklärung** (leicht) (schriftlich, 4 Punkte)

Bei den Hashingverfahren haben Sie einige neue Begriffe kennengelernt. Erklären Sie folgende Begriffspaare und grenzen Sie die Begriffe gegeneinander ab.

- *geschlossenes Hashing*  $\longleftrightarrow$  *offenes Hashing*
- *lineare Sondierung*  $\longleftrightarrow$  *quadratische Sondierung*
- *Divisionsverfahren*  $\longleftrightarrow$  *Multiplikationsverfahren*
- *Primärkollision*  $\longleftrightarrow$  *Sekundärkollision*

**2. Hashing I** (leicht–mittel) (schriftlich, 2+2+2 Punkte)

In eine Hashtabelle mit 11 Einträgen (Indizes 0 bis 10) sollen die Wörter „CLAUS“, „DRASKOCZY“, „LEWANDOWSKI“, „SCHMID“, „GRUBER“, „GRIEPENTROG“, „FISCHER“, „PHILIPP“ und „BURKOVSKI“ in der angegebenen Reihenfolge eingefügt werden. Für die verwendete Hashfunktion sind den Buchstaben A bis Z die Werte  $b(A)=1$  bis  $b(Z)=26$  zugeordnet; gewichten Sie den ersten Buchstaben mit 1 und den vierten Buchstaben mit Gewicht 3. Die Hashfunktion sei also  $f(a_1a_2a_3\dots) = ((b(a_1) + 3 \cdot b(a_4)) \bmod 11)$  (Beispiel: der Hashwert des Wortes „FUSSBALL“ ist damit  $((6 + 3 \cdot 19) \bmod 11) = 8$ ). Kollisionen sollen mit dem Verfahren der linearen Sondierung aufgelöst werden. Die Schrittweite ist 1.

- Geben Sie die ausgefüllte Tabelle, sowie für jeden Eintrag alle die im Zuge der Kollisionsauflösung berechneten Indizes an.
- Perfektes Hashing: modifizieren Sie die Gewichte in der Hashfunktion so, dass für die obigen 9 Elemente in einer Hashtabelle mit 11 Einträgen keine Kollision stattfindet.
- Finden Sie auch so eine perfekte Hashfunktion, wenn die Hashtabelle nur 9 Einträge hat?

**3. Hashing II** (leicht) (schriftlich, 2 Punkte)

Die Teilaufgabe c) der Aufgabe 2 ist nur mit viel Glück durch manuelles Probieren zu lösen. Daher wollen Sie ein kleines Programm schreiben, der alle Möglichkeiten durchprobiert.

Aber wieviele Möglichkeiten für die Hashfunktion (Divisionsverfahren) gibt es, wenn Sie maximal die ersten vier Buchstaben des Wortes verwenden dürfen? Geben Sie im eClaus-System nicht nur die Lösung, sondern auch eine Begründung der Lösung an!

- 4. Hashing-Programm I** (leicht) (schriftlich, 4 Punkte)  
Schreiben Sie ein Ada-Programm, das eine Zeichenkette vom Benutzer einliest und diese Zeichenkette in einer Hashtabelle speichert (jede Eingabe soll dabei in Großbuchstaben umgewandelt werden).  
Verwenden Sie das geschlossene Hashing mit linearen Listen. Dabei soll die Hashtabelle nur aus Zeigern auf Elemente bestehen (zu Anfang alles Nullzeiger). Überlegen Sie sich selbst eine Hashfunktion und begründen Sie Ihre Entscheidung.
- 5. Hashing-Programm II** (leicht) (schriftlich, 3 Punkte)  
Erweitern Sie Ihr Programm aus Aufgabe 4 mit der Option, Elemente aus der Hashtabelle zu löschen. Dabei soll das Programm dem Benutzer mitteilen, ob die zu löschende Zeichenkette in der Hashtabelle enthalten war oder nicht.  
Wieso ist das Löschen einfacher zu implementieren, wenn die Hashtabelle nur aus Zeigern besteht (und nicht aus Elementen mit angehängter Liste)?
- 6. Löschen beim linearen Sondieren** (mittel) (schriftlich, 4 Punkte)  
Im Vorlesungsskript, Abschnitt 9.3.12, wurde kurz die Idee des Löschens beim linearen Sondieren vorgestellt.  
Überlegen Sie sich den Löschalgorithmus im Detail und formulieren Sie ihn in Pseudocode. Geben Sie zusätzlich zum Pseudocode auch ein (nicht triviales) Beispiel im eClaus-System ab.