

7. Übung:

## Algorithmen und Datenstrukturen

Sommersemester 2008

28. Mai 2008

---

Abgabe bis Freitag, 06. Juni 2008, vor der Übung

**Aufgabe 7.1:** (Punkte 4)

Geben Sie die asymptotischen Laufzeiten der Mengenoperationen *lookup*, *insert*, *delete* und *union* (Vereinigung) für die Implementierungen Feld (Bitvektor), geordnete Liste und ungeordnete Liste an und begründen Sie.

**Aufgabe 7.2:** (Punkte 4)

Demonstrieren Sie Lineares Sondieren auf einer anfangs leeren Hashtafel der Größe  $m = 8$  und  $h_i(x) = ((x \bmod 8) + i) \bmod 8$  am Beispiel der Folge von Operationen: Insert(3), Insert(12), Insert(27), Insert(28), Insert(35), Delete(12), Delete(35), Insert(35). Gelöschte Felder werden mit *Deleted* markiert.

**Aufgabe 7.3:** (Punkte 6)

Implementieren Sie eine Hash-Tabelle als Klasse. Geben Sie Pseudocode für die Wörterbuchoperationen *Lookup*, *Insert* und *Delete* an. Verwenden Sie Lineares Sondieren, d.h. die Folge von Hashfunktionen

$$h_i(x) = ((x \bmod m) + i) \bmod m$$

für *Hashing mit offener Adressierung* auf einer Tafel der Größe  $s$ . Markieren Sie gelöschte Felder mit *Deleted*.

**Aufgabe 7.4:** (Punkte 3/3)

Das Kuckuck-Hashing ist eine weitere Hashing Variante.

Das Verfahren verwendet zwei Arrays  $t_1$  und  $t_2$  und zwei unterschiedliche Hash Funktionen  $h_1$  und  $h_2$ .

- Ein Element wird an Stelle  $p_1$ , berechnet durch  $h_1$ , in  $t_1$  eingefügt.
- Ist  $p_1$  bereits besetzt wird das dort vorhandene Element entnommen und an Stelle  $p_2$ , berechnet durch  $h_2$ , in  $t_2$  eingefügt.

- c) Ist  $p_2$  bereits besetzt wird das dort vorhandene Element entnommen und mit a) fortgefahren.
- Implementieren Sie Kuckuck-Hashing in einer *insert* Funktion.
  - Welches Problem kann auftreten? Geben Sie ein Beispiel an. Hierzu müssen Sie sich zwei Hash Funktionen überlegen.