
Informatik II

<http://www.informatik.uni-freiburg.de/proglang/teaching/aktuell/info2>

Übungsblatt 6

Abgabe: 4.6.2002

Aufgabe 1:

Die Schlüsselreihe 5, 27, 13, 54, 26, 10, 15, 24 soll durch geschlossenes Hashen mit direkter Verkettung in eine Hashtabelle der Größe 7 eingefügt werden. Geben Sie das Ergebnis nach Einfügung mit den folgenden Hashfunktionen an und untersuchen Sie, wieviele Konflikte jeweils bei der Suche nach einem Schlüsselwert maximal auftreten können:

i.) $\text{hash}(x) = x \bmod 7$

ii.) multiplikative Methode mit $\theta = \pi, m = 7$

iii.) multiplikative Methode mit $\theta = \sqrt{2}, m = 7$

Wieviele Schlüsselvergleiche können bei einer Kollision beim geschlossenen Hashing mit direkter Verkettung nach Einfügen von n Schlüsseln maximal auftreten?

Aufgabe 2: Erweitern Sie die Klasse `ChainedHashTable`, um die Hashfunktionen der Aufgabe 1 zu implementieren. Dabei ist die Methode `h` geeignet zu überladen.

Aufgabe 3:

Schreiben Sie ein Programm, das alle Spieler der Fussballweltmeisterschaft (Nachname, Vorname, Land) aus einer Datei einliest und in eine geschlossene Hashtabelle der Größe 1000 unter Benutzung von *Universellem Hashing* einträgt. Verwenden Sie als Schlüssel den Nach- und Vorname eines Spielers.

Demonstrieren Sie Ihr Programm, indem Sie die vorgegebene Datei `ueb6-2.txt`¹ einlesen und von den Spielern "RONALDO", "ZIDANE, Zinedine" und "BECKENBAUER, Franz" das Herkunftsland bestimmen.

Gehen Sie etwa wie folgt vor:

- Schreiben Sie eine Klasse `Player`, deren Objekte Fussballspieler repräsentieren und zusätzlich folgendes Interface implementieren:

```
interface Key public java.math.BigInteger key();
```

Die Methode `key` ist eine injektive Abbildung des (Fussballspieler-)Universums in die natürlichen Zahlen, die durch die Objekte der Klasse `java.math.BigInteger` repräsentiert werden.

¹Die Datei `ueb6-2.txt` ist auf den Vorlesungswebseiten zu finden.

- Schreiben Sie eine Klasse `UniversalDirectChainedHashTable`, die eine Erweiterung der Klasse `ChainedHashTable` ist und die Hashfunktion `h` geeignet überlädt.
- Bekommt der Konstruktor der Klasse `UniversalDirectChainedHashTable` eine Primzahl `p` und eine Hashtabellengrösse `m` übergeben, soll zufällig eine Hashfunktion $h_{a,b} \in \mathcal{H}$ ausgewählt werden.

Die Hashfunktion `h` ist so zu überladen, dass die durch den Konstruktor zufällig ausgewählte Hashfunktion $h_{a,b}$ verwendet wird.

- Schreiben Sie ein Hauptprogramm, das die Spielerdatei einliest, die Spieler mit Hilfe einer der universellen Hashfunktionen in einen `ChainedHashTable` einfügt, und danach für die angegebenen Spieler ihr Herkunftsland bestimmt.

Überlegen Sie sich hierzu, wie gross die Primzahl p gewählt werden muss, und verwenden Sie dann eine passende Primzahl.