

Selbsttest 9

1) Hashtabellen

Die Länge einer Hashtabelle sollte immer eine Primzahl sein, muss aber nicht

Hashtabellen sind für häufiges Einfügen und wieder Löschen von Elementen gedacht

Hashtabellen mit Überlauf Listen können auch als Prioritätslisten verwendet werden

Der worst case beim Einfügen in eine Hashtabelle mit linearem Probieren ist nicht schneller, als der worst case beim Einfügen in eine sortierte lineare Liste.

Ja	Nein
X	
	X
	X
X	

2) Quadratisches Hashen

Ist eine Hashtabelle mit einer Länge von 11 Elementen für das quadratische Hashen geeignet? In welcher Reihenfolge werden die Elemente bei quadratischer Kollisionsstrategie besucht, wenn der die Hashfunktion den Wert 0 liefert und alle Plätze belegt sind?

```
void insert(String key, Data data) {
    int a = hash(key); int d = -tabSize;
    while(tab[a] besetzt) {
        d = d + 2; a = (a + Math.abs(d)) % tabSize;
    }
    tab[a].key = key; tab[a].data = data;
}
```

Lösung:

Ja, 11 ist eine Primzahl und für $j = 2$ gilt: $j * 4 + 3 = 2 * 4 + 3 = 8 + 3 = 11$

<i>a</i>	<i>d</i>
0	-11
9	-9
5	-7
10	-5
2	-3
3	-1

<i>a</i>	<i>d</i>
4	+1
7	+3
1	+5
8	+7
6	+9

3) Hashtabellen: Berechnung eines Hashcodes

Für eine bestimmte Klasse soll eine Methode geschrieben werden, die den Hashcode eines Objektes berechnet. Welche Anforderungen gibt es an diesen Hashcode?

Wie könnte diese Methode für folgende Klasse aussehen?

```
public class Fahrzeug {
    private int anzahlRaeder; // 0-15
    private char fuehrerscheinklasse; // A, B, C, D, E, F, G
    private boolean hatPickerl; // true/false
    private String kennzeichen; // str.hashCode();
    private int baujahr; // 1900-2155

    public int calcHashCode() {
        ...
    }
}
```

```
    return ...;
}
}
```

Lösung:

```
public int calcHashCode() {
    int code = anzahlRaeder; // 0-15 = 4 Bit
    code |= (fuehrenscheinKlasse - 'A') << 4; // 0-7 = 3 Bit
    code |= (hatPickerl ? 1 : 0) << 7; // 0 oder 1 = 1 Bit
    code |= (baujahr - 1900) << 8; // 1900-2155 = 8 Bit
    code |= str.hashCode() << 16;
    return code;
}
```

Die Idee dahinter ist, dass sich die (Zahlen-)Bereiche der einzelnen Felder nicht überlappen (man kann sich das auch mit den einzelnen Bits der Zahlen und Shift-Operationen vorstellen. Die Multiplikation macht es aber einfacher, außerdem werden so die Bits noch mal ein bisschen "durcheinandergewürfelt").

Ziel ist es, soweit als möglich alle erlaubten Daten(-kombinationen) auf einen eigenen Hash-Wert abzubilden.

4) Hashtabellen - Diskussion

Was sind die Vorteile von Hashtabellen? Welche Operationen sind schnell durchzuführen?
Welche Operationen sind für Hashtabellen weniger geeignet?