Übung: Hashing und Hash Tables

Hashfunktionen

1) Berechnen Sie den jeweiligen Hash-Wert des Keys $k = 42$, indem Sie diesen auf die Divisionsrest-Methode und die Multiplikations-Methode mit folgenden Parametern anwenden:

Divisionsrest-Methode: $m = 100$
Multiplikations-Methode: $m = 100$, $a = 0.618$

2) Bestimmen Sie einen Schlüssel q mit $q \neq 42$ und $h(q) = h(42)$, wobei h der Divisionsrest-Methode entspricht.

Implementation einer Hashtable

3) Implementieren Sie nach dem Beispiel aus der Präsentation eine Hash Table in Python oder Java, die intern folgende Hashfunktion verwendet:

$$h(k) = ((a \cdot k + b) \bmod p) \bmod m$$

Dabei sollen a, b zufällig aus $\{0, 1, \ldots, p-1\}$ gewählt werden. Für p wählen Sie eine Primzahl $> m$.
Hinweis: Table Doubling muss nicht implementiert werden.

4) Zu welcher Art von Hashing gehört die in Aufgabe 3 implementierte Hashfunktion.

Kollisionsbehandlung

Gegeben sind eine Hashtable mit der Größe 23 und die beiden Hashfunktionen:

$$h_1(k) = Quersumme(k) \text{ und } h_2(k) = k \bmod 23$$

5) Fügen Sie folgende Werte: 12, 99, 111, 76, 23, 30 jeweils mit

- Hashing mit Verkettung
- linearem Sondieren

in jeweils eine Tabelle ein (insgesamt 4 Tabellen). Geben Sie die nicht-leeren Teile der Tabelle nach jedem Schritt an.
`Beispiel:`

12 einfügen

$h(k)$	Inhalt
3	12

99 einfügen

$h(k)$	Inhalt
3	12
18	99

usw. ...

Double Hashing

Gegeben sind die beiden Hashfunktionen:

$$h_1(k) = Quersumme(k) \text{ und } h_2(k) = 2k + 1$$

Die Größe m der Hashtable beträgt 13.

6) Geben Sie die aus dem Double Hashing resultierende Hashfunktion $h_3(k, i)$ an.

7) Führen Sie mit dieser Hashfunktion h_3 folgende Operationen aus und tragen Sie dementsprechend die Werte in den Hashtable ein:

1. Fügen Sie folgende Werte in den Array ein: 453, 66, 39, 10, 1.
2. Löschen Sie die 10.
3. Fügen Sie die 0 ein.

$h(k)$	Inhalt
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

Lösungen

Hashfunktionen

1) Berechnen Sie den jeweiligen Hash-Wert des Keys $k = 42$, indem Sie diesen auf die Divisionsrest-Methode und die Multiplikations-Methode mit folgenden Parametern anwenden:

Divisionsrest-Methode: $m = 100$
Multiplikations-Methode: $m = 100$, $a = 0.618$

$h_1(42) = 42$
$h_2(42) = 95$

2) Bestimmen Sie einen Schlüssel q mit $q \neq 42$ und $h(q) = h(42)$, wobei h der Divisionsrest-Methode entspricht.

Eine Zahl, bei der die Einerstelle 2 und die Zehnerstelle 4 ist.
z.B 142; 457348537498542

Implementation einer Hashtable

3) Implementieren Sie nach dem Beispiel aus der Präsentation eine Hash Table in Python oder Java, die intern folgende Hashfunktion verwendet:

$$h(k) = ((a \cdot k + b) \bmod p) \bmod m$$

Dabei sollen a, b zufällig aus $\{0, 1, \ldots, p-1\}$ gewählt werden. Für p wählen Sie eine Primzahl $> m$.
Hinweis: Table Doubling muss nicht implementiert werden.

Java:

```java
private static int hash(Key key) {
    int p = 1009;
    Random random = new Random();

    int a = random.nextInt(p);
    int b = random.nextInt(p);

    return ((a * key.hashCode() + b) % p) % m;
}
```

Python:

```python
@staticmethod
def __hash(key):
    p = 1009
    a = random.randrange(p)
    b = random.randrange(p)

    return ((a * HashTable.__hash(key) + b) % p) % m
```

4) Zu welcher Art von Hashing gehört die in Aufgabe 3 implementierte Hashfunktion.

Universelles Hashing

Kollisionsbehandlung

Gegeben sind eine Hashtable mit der Größe 23 und die beiden Hashfunktionen:

$$h_1(k) = Quersumme(k) \text{ und } h_2(k) = k \bmod 23$$

5) Fügen Sie folgende Werte: 12, 99, 111, 76, 23, 30 jeweils mit

- Hashing mit Verkettung
- linearem Sondieren

in jeweils eine Tabelle ein (insgesamt 4 Tabellen). Geben Sie die nicht-leeren Teile der Tabelle nach jedem Schritt an.

Verkettung:

12 einfügen

$h_1(k)$	Inhalt
3	12

99 einfügen

3	12
18	99

111 einfügen

3	12 → 111
18	99

76 einfügen

3	12 → 111
13	76
18	99

23 einfügen

3	12 → 111
5	23
13	76
18	99

30 einfügen

3	12 → 111 → 30
5	23
13	76
18	99

12 einfügen

$h_2(k)$	Inhalt
12	12

99 einfügen

7	99
12	12

111 einfügen

7	99
12	12
19	111

76 einfügen

7	76 → 99
12	12
19	111

23 einfügen

0	23
7	99 → 76
12	12
19	111

30 einfügen

0	23
7	99 → 76 → 30
12	12
19	111

lineares Sondieren:

12 einfügen

$h_1(k)$	Inhalt
3	12

99 einfügen

3	12
18	99

12 einfügen

$h_2(k)$	Inhalt
12	12

99 einfügen

7	99
12	12

111 einfügen

3	12
4	111
18	99

76 einfügen

3	12
4	111
13	76
18	99

23 einfügen

3	12
4	111
5	23
13	76
18	99

30 einfügen

3	12
4	111
5	23
6	30
13	76
18	99

111 einfügen

7	99
12	12
19	111

76 einfügen

7	99
8	76
12	12
19	111

23 einfügen

0	23
7	99
8	76 → 76
12	12
19	111

30 einfügen

0	23
7	99
8	76
9	30
12	12
19	111

Double Hashing

Gegeben sind die beiden Hashfunktionen:

$$h_1(k) = Quersumme(k) \text{ und } h_2(k) = 2k + 1$$

Die Länge m des Arrays beträgt 13.

6) Geben Sie die daraus resultierende Hashfunktion $h_3(k, i)$ an.

$$h_3(k, i) = (Quersumme(k) + i(2k + 1)) \bmod 13$$

7) Führen Sie mit dieser Hashfunktion h_3 folgende Operationen aus und tragen Sie dementsprechend die Werte in den Hashtable ein:

1. Fügen Sie folgende Werte in den Array ein: 453, 66, 39, 10, 1.
2. Löschen Sie die 10.
3. Fügen Sie die 0 ein.

$h(k)$	Inhalt
0	39
1	~~10~~ ~~DEL~~ 0
2	66
3	
4	1
5	
6	
7	
8	
9	
10	
11	
12	453

- 453: 12
- 66: 12→0
- 39: 12→0
- 10: 1
- 1: 1→4
- `DEL`: 1
- 0: 0→1