

Übungen zu Objektorientierte Programmierung 2

frieder.loch@ost.ch

Woche 12: Hashing

Aufgabe 1: Dynamisches Hashing

Implementieren Sie die *put* Funktion der dynamischen Hashmap, analog zur Vorlesung. Verwenden Sie die *getIndex* Methode, um den Index zu erhalten.

V *put*(K key, V value)

Aufgabe 2: Hashtabelle

In die Hash-Tabellen werden folgende Zahlen der Reihe nach eingefügt: 12,44,13,88,23,94,11,39,20,16,5.

Separate Chaining

Die Hash-Funktion ist wie folgt:

$$h(k) = (2k + 5) \bmod 11$$

Tragen Sie die Zahlen in die Hash-Tabelle ein.

d(k)	Separate Chaining
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Linear Probing

Die Hash-Funktionen sind wie folgt:

$$h(k) = (2k + 5) \bmod 11$$

$$p(k) = (h(k) + j) \bmod 11$$

wobei:

$$j = 0, 1, 2, 3, \dots \text{ (Anzahl Kollisionen)}$$

Tragen Sie die Zahlen in die Hash-Tabelle ein.

d(k)	Linear Probing
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Quadratic Probing

Die Hash-Funktionen sind wie folgt:

$$h(k) = (2k + 5) \bmod 11$$

$$p(k) = (h(k) + j^2) \bmod 11$$

wobei:

$$j = 0, 1, 2, 3, \dots \text{ (Anzahl Kollisionen)}$$

Tragen Sie die Zahlen in die Hash-Tabelle ein.

d(k)	Quadratic Probing
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Double Hashing

Die Hash-Funktionen sind wie folgt:

$$h(k) = (2k + 5) \bmod 11$$

$$d(k) = (h(k) + j(7 - k \bmod 7)) \bmod 11$$

wobei:

$$j = 0, 1, 2, 3, \dots \text{ (Anzahl Kollisionen)}$$

Tragen Sie die Zahlen in die Hash-Tabelle ein.

d(k)	Double Hashing
0	
1	
2	
3	
4	

d(k)	Double Hashing
5	
6	
7	
8	
9	
10	