

Übungsserie 5: Lösungen

Aufgabe 1: Map-Implementation

Es soll eine einfache *Map* (*MapImpl.java* resp. *map_impl.py*) implementiert werden, welche in der Lage ist, *Entries* aufzunehmen.

Als Basis-Datenstruktur für die Map wird der Einfachheit halber eine doppelt-verkettete Liste verwendet (Java: *java.util.LinkedList*, Python: *list*).

Lösung:

Siehe „*MapImpl.java*“ resp. „*map_impl.py*“

Aufgabe 2: Hash-Tables

Es sollen in untenstehenden Hash-Tabellen die folgenden Zahlen so eingetragen werden wie sie mit der *put()*- Operation eingefügt würden:

12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5

a) Kollisionen werden mittels *Seperate Chaining* behandelt.

Hash-Funktion $h(i) = (2 \cdot i + 5) \bmod 11$

h(i)	Key Chaining
0	
1	20
2	
3	
4	16, 5
5	44, 88, 11
6	94, 39
7	12, 23
8	
9	13
10	

- b) Kollisionen werden mittels *Linear Probing* behandelt.
Hash-Funktion $h(i) = (2 \cdot i + 5) \bmod 11$

h(i)	Linear Probing
0	11
1	39
2	20
3	5
4	16
5	44
6	88
7	12
8	23
9	13
10	94

- c) Kollisionen werden mittels Double Hashing behandelt.
Erste Hash-Funktion $h(i) = (2 \cdot i + 5) \bmod 11$
Zweite Hash-Funktion $d(i) = 7 - i \bmod 7$

h(i)	Double Hashing
0	11
1	23
2	20
3	16
4	39
5	44
6	94
7	12
8	88
9	13
10	5