

Aufgabe 1

$$27 \bmod 7 = 6$$

$$49 \bmod 7 = 0$$

$$73 \bmod 7 = 3$$

$$10 \bmod 7 = 3 \text{ (Kollision)} \rightarrow 4$$

$$13 \bmod 7 = 6 \text{ (Kollision)} \rightarrow 0 \text{ (Kollision)} \rightarrow 1$$

Bucket	Key
0	49
1	13
2	
3	73
4	10
5	
6	27

Aufgabe 2

$$17 \bmod 7 = 3 \text{ (} \rightarrow [17])$$

$$36 \bmod 7 = 1 \text{ (} \rightarrow [36])$$

$$43 \bmod 7 = 1 \text{ (Kollision } \rightarrow [36, 43])$$

$$39 \bmod 7 = 4 \text{ (} \rightarrow [39])$$

$$88 \bmod 7 = 4 \text{ (Kollision } \rightarrow [39, 88])$$

$$21 \bmod 7 = 0 \text{ (} \rightarrow [21])$$

Bucket	Key
0	[21]
1	[36, 43]
2	
3	[17]
4	[39, 88]
5	
6	

Aufgabe 3

Vorteil: Alle Behälter werden ausgefüllt

Nachteil: Das lineare Sondieren kann zu Anhäufungen von Schlüsseln führen, was das Hashing verlangsamt.

Aufgabe 4

Vorteil: Funktioniert auch dann, wenn die Hashtabelle bereits (über)voll ist.

Nachteil: Die Verwendung einer zusätzlichen Datenstruktur (Liste) verlangsamt das Verfahren.

Aufgabe 5

Beim Suchen von Schlüsseln in einer Hashtabelle benötigt man bei einer guten Hashfunktion (wenige Kollisionen) konstante Rechenzeit zum Auffinden eines Werts.

Im Gegensatz dazu benötigt das Suchen nach Werten in einer Liste der Länge n im Mittel $n/2$ Vergleiche.