

Aufgabe 1

- (a) $17 \bmod 11 = 6$
 $36 \bmod 11 = 3$
 $43 \bmod 11 = 10$
 $39 \bmod 11 = 6$ (Kollision) $\rightarrow 7$
 $88 \bmod 11 = 0$
 $21 \bmod 11 = 10$ (Kollision) $\rightarrow 0$ (Kollision) $\rightarrow 1$

Bucket	Key
0	88
1	21
2	
3	36
4	
5	
6	17
7	39
8	
9	
10	43

Lastfaktor $\alpha = 6/11$

Da die Auslastung (Anteil der besetzten Behälter) zu gross ist (gut wäre $\frac{1}{8} \leq \alpha \leq \frac{1}{2}$) sollte man hier die Grösse der Hashtabelle verdoppeln.

Aufgabe 2

- $17 \bmod 7 = 3$ ($\rightarrow [17]$)
 $36 \bmod 7 = 1$ ($\rightarrow [36]$)
 $43 \bmod 7 = 1$ (Kollision $\rightarrow [36, 43]$)
 $39 \bmod 7 = 4$ ($\rightarrow [39]$)
 $88 \bmod 7 = 4$ (Kollision $\rightarrow [39, 88]$)
 $21 \bmod 7 = 0$ ($\rightarrow [21]$)

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

Lastfaktor $\alpha = 6/4 = 3/2 = 1.5$

Da die Auslastung (Anzahl Elemente pro Liste) zu klein ist (gut wäre $2 \leq \alpha \leq 8$), sollte man die Grösse der Hashtabelle halbieren

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 ($O(1)$).

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

Aufgabe 6

- Addiere die ASCII- oder Unicode-Werte der Zeichen.

Problem: viele Zeichenketten werden so auf die gleiche Zahl abgebildet (abc und cba)

- Addiere die mit ihrem Stellenwert gewichteten ASCII-Werte der Zeichen.

Wenn das i -te Zeichen mit der ASCII-Nummer a_i 7 Bit benötigt (128 Werte), so kann man mit

$$a_0 \cdot 128^0 + a_1 \cdot 128^1 + a_2 \cdot 128^2 + \dots + a_n \cdot 128^n$$

Jeder Zeichenkette $a_0a_1a_2 \dots a_n$ eine eindeutige Zahl zuordnen.