

# AlgoDat Tutorium Blatt 6

Alex B.

Mai 2024

## 1 Hashtabellen

- Abspeichern von Daten so, dass Zugriff, Löschen und einfügen möglichst schnell funktionieren (im Mittel  $O(1)$ )
- Wertemenge  $U$  ist sehr viel größer als Schlüsselmenge  $K$
- Hashfunktion: Funktion, welche einen Schlüssel auf einen Hashwert abbildet
- Ziel einer Hashfunktion: Kollisionen vermeiden, bzw. auflösen

## 2 Aufgaben

1. Wie könnte man eine eventuelle Kollision auflösen?
2. Zeige an einer Hashtabelle mit 9 Slots, wie nacheinander die Werte 7, 23, 16, 14, 42, 69, 27, 79 und 5 mithilfe der Hashfunktion  $h(k)=k \bmod m$  mit  $m = 9$  eingefügt werden. Kollisionen werden durch Verkettung aufgelöst.
3. Auf welche Slots werden die Schlüssel 47, 48, 49, 50, 51 und 52 in einer Hashtabelle mit Größe  $m=1000$  mithilfe der Hashfunktion  $h(k)=\lfloor m * (k * A \bmod(1)) \rfloor$  mit  $A = \frac{\sqrt{5}-1}{2}$  abgebildet?
4. Eine Hashtabelle der Länge  $m=7$  mit offener Adressierung habe die Hilfsfunktion  $h'(k)=k$ . Zeige das Einfügen der Schlüssel 11, 21, 34, 7, 15, 29 und 16 bei folgenden Sondierverfahren
  - a ) linearem Sondieren
  - b ) quadratischem Sondieren mit  $c_1=1$  und  $c_2=5$
  - c ) doppeltem Hashing mit  $h_1(k) = k$  und  $h_2(k) = 1 + (k \bmod(m-2))$

### 3 Zusatzaufgabe Hoare-Kalkül

Schreibe ein Programm, welches zwei Arrays mit ganzen Zahlen erhält. Am Ende des Programms soll in einer Zählvariable stehen, wie viele Elemente in den Arrays den gleichen Wert hatten. Eine andere Zählvariable soll zählen, wie viele Elemente in Array 1 größer waren als in Array 2. Eine dritte Zählvariable soll zählen, wie viele Elemente in Array 1 kleiner waren als in Array 2. Beweise anschließend die Korrektheit deines Programmes mit dem Hoare-Kalkül.