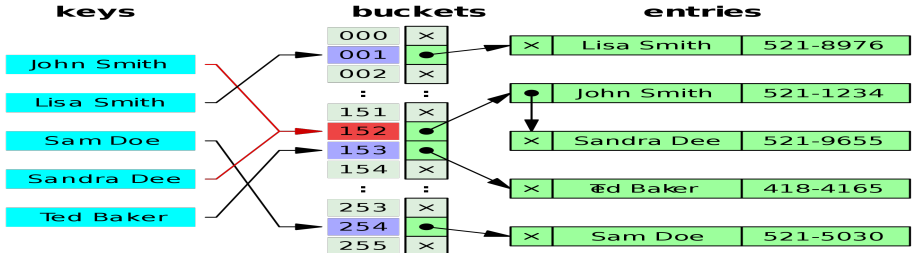


# Algorithmen I - Tutorium 4

Sebastian Schmidt – *isibboi@gmail.com*

Arbeitsgruppe Kryptographie und Sicherheit





Elementarereignisse  $\Omega$

Ereignisse: Teilmengen von  $\Omega$

$p_x$  = Wahrscheinlichkeit von  $x \in \Omega$ .  $\sum_x p_x = 1$  !

Gleichverteilung:  $p_x = \frac{1}{|\Omega|}$

$\mathbb{P}[\mathcal{E}] = \sum_{x \in \mathcal{E}} p_x$

Zufallsvariable (ZV)  $X_0 : \Omega \rightarrow \mathbb{R}$

0-1-Zufallsvariable (Indikator-ZV)  $I : \Omega \rightarrow \{0, 1\}$

Erwartungswert  $E[X] = \sum_{y \in \Omega} p_y X(y)$

Linearität des Erwartungswerts:  $E[X + Y] = E[X] + E[Y]$

Hash-Beispiel

Hash-Funktionen  $\{0..m-1\}^{\text{Key}}$

$\mathcal{E}_{42} = \{h \in \Omega : h(4) = h(2)\}$

$p_h = m^{-|\text{Key}|}$

$\mathbb{P}[\mathcal{E}_{42}] = \frac{1}{m}$

$X = |\{e \in M : h(e) = 0\}|$

$E[X] = \frac{|M|}{m}$

- Hashtabelle mit  $h_n(x) := x \bmod n$ 
  - Wie groß sollte die Hashtabelle sein?
  - Gegeben  $\{36, 78, 50, 1, 92, 15, 43, 99, 64\}$ . Füge diese Zahlen in eine Hashtabelle mit Hashfunktion  $h_5$  und  $h_7$  ein.
- Gebt Beispiele für gute und schlechte Hashfunktionen
- Laufzeiten: insert, find und remove

- Hashtabelle mit  $h_n(x) := x \bmod n$ 
  - Wie groß sollte die Hashtabelle sein?
    - Gegeben  $\{36, 78, 50, 1, 92, 15, 43, 99, 64\}$ . Füge diese Zahlen in eine Hashtabelle mit Hashfunktion  $h_5$  und  $h_7$  ein.
  - Gebt Beispiele für gute und schlechte Hashfunktionen
  - Laufzeiten: insert, find und remove

- Hashtabelle mit  $h_n(x) := x \bmod n$ 
  - Wie groß sollte die Hashtabelle sein?
  - Gegeben  $\{36, 78, 50, 1, 92, 15, 43, 99, 64\}$ . Füge diese Zahlen in eine Hashtabelle mit Hashfunktion  $h_5$  und  $h_7$  ein.
- Gebt Beispiele für gute und schlechte Hashfunktionen
- Laufzeiten: insert, find und remove

- Hashtabelle mit  $h_n(x) := x \bmod n$ 
  - Wie groß sollte die Hashtabelle sein?
  - Gegeben  $\{36, 78, 50, 1, 92, 15, 43, 99, 64\}$ . Füge diese Zahlen in eine Hashtabelle mit Hashfunktion  $h_5$  und  $h_7$  ein.
- Gebt Beispiele für gute und schlechte Hashfunktionen
- Laufzeiten: `insert`, `find` und `remove`

- Hashtabelle mit  $h_n(x) := x \bmod n$ 
  - Wie groß sollte die Hashtabelle sein?
  - Gegeben  $\{36, 78, 50, 1, 92, 15, 43, 99, 64\}$ . Füge diese Zahlen in eine Hashtabelle mit Hashfunktion  $h_5$  und  $h_7$  ein.
- Gebt Beispiele für gute und schlechte Hashfunktionen
- Laufzeiten: insert, find und remove

# Anwendungsbeispiele für Hashtabellen

Hashtabellen sind besser als Bäume, wenn man die erwartete Laufzeit betrachtet.

- Fallen euch konkrete Beispiele oder Gegenbeispiele ein?



# Anwendungsbeispiele für Hashtabellen

Hashtabellen sind besser als Bäume, wenn man die erwartete Laufzeit betrachtet.

- Fallen euch konkrete Beispiele oder Gegenbeispiele ein?

Seien  $m, n \in \mathbb{N}$ .  $n$  ist die Anzahl der Elemente, die in eine Hashtabelle der Größe  $m$  eingefügt werden.

Sei  $U$  ein Universum mit  $|U| \geq mn$ .

Zeige, dass eine Teilmenge  $M \subset U$  existiert mit  $|M| \geq n$ , sodass alle Elemente aus  $M$  dem selben Slot der Hashtabelle zugeordnet werden.

*„Nach dem dritten Glas Bier behauptet ein Kommilitone, man könne Hashing mit verketteten Listen entscheidend verbessern, indem man die verketteten Listen stets sortiert halte.“*

- Wie ändert sich dadurch die Worst-Case Laufzeit von `insert`, `find` und `remove`?