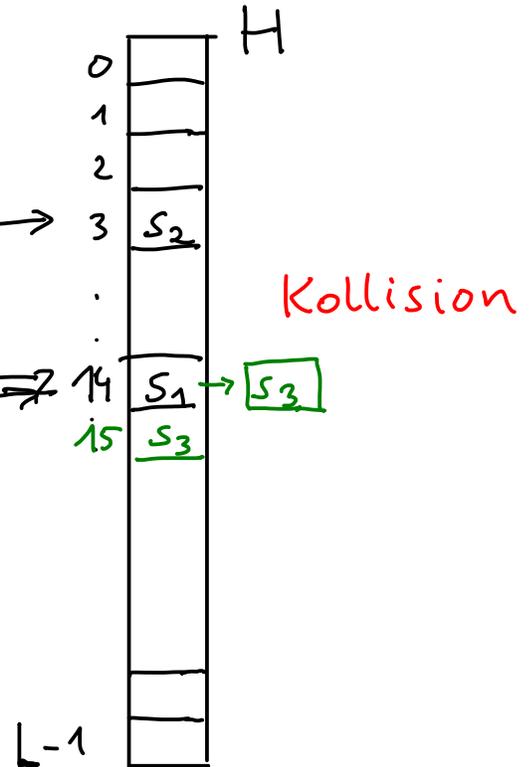


Hashtabellen (gestreute Speicherung)

s_1 $h(s_1) = 14$
Hashfunktion

$h(s_2) = 3$

$h(s_3) = 14$



$h: \{0, 1, \dots, 2^w - 1\} \rightarrow \{0, \dots, L - 1\}$

w = Wortlänge der Schlüssel, z.B. 32

L ist häufig von der Form 2^d

$L > n$

Grundidee: Schlüssel s (und zugehöriger Wert) wird auf $H[h(s)]$ gespeichert.

Kollision: $h(s) = h(s')$

a) Verkettung:

$H[i] \rightarrow$ verkettete Liste aller Schlüssel mit $h(s) = i$

b) „offene Adressierung“:

Es wird nach einer Ersatzposition gesucht.

z.B. $h(s) + 1, h(s) + 2, \dots$ (lineares Sondieren)

SATZ. Bei einer Hashtabelle mit Verkettung
 ist $E[\text{Länge der Liste } H[h(x)]] \leq 1 + 2\alpha$.

↑
 fester Schlüssel
 ↑
 falls x in der Tabelle ist.

Beweis:

$S = \{x_1, x_2, \dots, x_n\}$ = gespeicherte Schlüssel

$E[\text{Länge}] = 1$, falls $x \in S$

$$+ \sum_{y \in S - \{x\}} \underbrace{\Pr[h(y) = h(x)]}_{2/L} \leq 1 + \frac{n \cdot 2}{L} = 1 + 2\alpha.$$

SATZ. Erwartete Laufzeit für Einfügen,
 Löschen, Suchen ist $O(1 + \alpha)$. (= $O(1)$ für
 konstantes α)

Offene Adressierung

mit linearem Sondieren.

$$h(x), h(x)+1, h(x)+2, \dots \pmod{L}$$

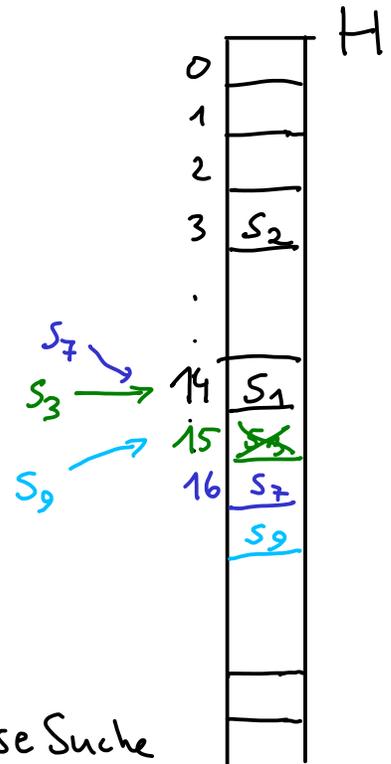
Beim Entfernen darf man nicht
 einfach löschen, sondern man
 muss die Einträge als gelöscht
 markieren.

(Sonst werden andere Einträge nicht
 gefunden.)

Erwartete Laufzeit $O\left(\frac{1}{(1-\alpha)^2}\right)$ erfolglose Suche

!
 $\alpha < 0.7$

$O\left(\frac{1}{1-\alpha}\right)$ für vorhandenen Schlüssel



Kuckuckshashing

2 Hashfunktionen h_1, h_2

Schlüssel x ist
in $H_1[h_1(x)]$ oder $H_2[h_2(x)]$
gespeichert.

$O(1)$ für Suchen (und Löschen)
im schlimmsten Fall.

Einfügen: Kette von Austauschen,
im Erwartungswert $O(1)$.

Im schlimmsten Fall (Pr. $< \frac{1}{n}$) muss die gesamte Tabelle
mit neuen Hashfunktionen h_1, h_2 aufgebaut werden.
 $O(n)$ Zeit.

