



2018-06-19

Hashfunktionen  
└ Hashlisten

## Hashliste

Eine Datenstruktur welche die Adresse von Datensätzen anhand einer Hashfunktion berechnet.

## Hashfunktion

$$h : \mathcal{K} \rightarrow \{0, \dots, m - 1\}$$

## Synonyme

$$h(k) = h(k')$$

## Hashliste

Eine Datenstruktur welche die Adresse von Datensätzen anhand einer Hashfunktion berechnet.

## Hashfunktion

$$h : \mathcal{K} \rightarrow \{0, \dots, m - 1\}$$

## Synonyme

$$h(k) = h(k')$$

Warum? -> effiziente Datenstrukturen, Verschlüsselung von Informationen

Abbildung von variabler Länge auf feste Länge

Hashfunktionen sind surjektiv

$$k \in \mathcal{K}$$

2018-06-19

Hashfunktionen  
└ Hashlisten

## Divisions-Rest-Methode

$$h(k) = k \bmod m$$

Beispiel für  $m = 5$ 

$k$	10	20	30	33	9
$h(k)$	0	0	0	3	4

Divisions-Rest-Methode

$$h(k) = k \bmod m$$

Beispiel für  $m = 5$ 

$k$	10	20	30	33	9
$h(k)$	0	0	0	3	4

## externe Kollisionsauflösung

Speichern der Kollisionen in einer verketteten Liste.

 $m = 5$ 

0	1	2	3	4
10			33	9
20				
30				

## externe Kollisionsauflösung

Speichern der Kollisionen in der Hashliste.

0	1	2	3	4
10	20	30	33	9

2018-06-19

## Hashfunktionen

## Hashlisten

Hier lineares sondieren wie in der Vorlesung

$k$	10	20	30	33	9
$h(k)$	0	0	0	3	4

externe Kollisionsauflösung

Speichern der Kollisionen in einer verketteten Liste.

 $m = 5$ 

0	1	2	3	4
10			33	9
20				
30				

externe Kollisionsauflösung

Speichern der Kollisionen in der Hashliste.

0	1	2	3	4
10	20	30	33	9

## Sondierungsfunktion

Erzeugt eine Permutation aller Adressen abhängig vom Schlüssel.

Lineares Sondieren:

$h(k), h(k) - 1, h(k) - 2, \dots, 0, m - 1, \dots, h(k) + 1$

0	1	2	3	4
10	9	33	30	20

Quadratisches Sondieren:

$h(k), h(k) - 1, h(k) + 1, h(k) - 4, h(k) + 4, \dots$

0	1	2	3	4
10	30	9	33	20

2018-06-19

## Hashfunktionen

### Hashlisten

#### Sondierungsfunktion

Erzeugt eine Permutation aller Adressen abhängig vom Schlüssel.

Lineares Sondieren:

$h(k), h(k) - 1, h(k) - 2, \dots, 0, m - 1, \dots, h(k) + 1$

0	1	2	3	4
10	9	33	30	20

Quadratisches Sondieren:

$h(k), h(k) - 1, h(k) + 1, h(k) - 4, h(k) + 4, \dots$

0	1	2	3	4
10	30	9	33	20

$k$	10	20	30	33	9
$h(k)$	0	0	0	3	4

Idealerweise Suchen in  $O(1)$

Umso voller umso mehr Kollisionen->umso langsamer->mehr Speicher

# Kryptographische Hashfunktionen

## Hashfunktion

$$h : \mathcal{S} \rightarrow \{0, 1\}^n$$

Anforderung an eine Kryptographische Hashfunktionen:

- 1 Die Laufzeit der Hashfunktion ist gering
- 2 Preimage-resistance: Es ist schwer die Daten  $k$  für einen gegebenen Hash  $y$  zu finden das gilt  $h(k) = y$
- 3 2nd-preimage resistance: Es ist schwer für gegebene Daten  $k$  ein  $k'$  zu finden sodass gilt  $h(k) = h(k')$
- 4 Collision resistant: Es ist schwer zwei Daten  $k, k'$  zu finden sodass gilt  $h(k) = h(k')$

## Hashfunktionen

### └ Kryptographische Hashfunktionen

### └ Kryptographische Hashfunktionen

Schwer hier->hohe laufzeit  
Widerspricht 1) -> abwiegen

#### Hashfunktion

$$h : \mathcal{S} \rightarrow \{0, 1\}^n$$

Anforderung an eine Kryptographische Hashfunktionen:

- Die Laufzeit der Hashfunktion ist gering
- Preimage-resistance: Es ist schwer die Daten  $k$  für einen gegebenen Hash  $y$  zu finden das gilt  $h(k) = y$
- 2nd-preimage resistance: Es ist schwer für gegebene Daten  $k$  ein  $k'$  zu finden sodass gilt  $h(k) = h(k')$
- Collision resistant: Es ist schwer zwei Daten  $k, k'$  zu finden sodass gilt  $h(k) = h(k')$

Speichern von Passwörtern in einer Datenbank mithilfe einer Hashfunktion und eines Saltes.

$$p_{out} = h(p_{in} + salt)$$

2018-06-19

# Hashfunktionen

## └ Kryptographische Hashfunktionen

Speichern von Passwörtern in einer Datenbank mithilfe einer Hashfunktion und eines Saltes.

$$p_{out} = h(p_{in} + salt)$$

2018-06-19

# Hashfunktionen

## Fazit

- ◆ Ermöglichen Suchen in konstanter Zeit
  - ◆ Ermöglichen sicheres Speichern von Passwörtern
- Vielen Dank fürs Zuhören

- 1 Ermöglichen Suchen in konstanter Zeit
- 2 Ermöglichen sicheres Speichern von Passwörtern

Vielen Dank fürs Zuhören

Besonders im Kryptographischen bereich weiterentwicklung da wettrüsten