



Helmut Schauer  
Educational Engineering Lab  
Department for Informatics  
University of Zurich



# Algorithmen und Datenstrukturen FS 2008

## Hashtabellen



Helmut Schauer  
Educational Engineering Lab  
Department for Informatics  
University of Zurich



## Dictionary Problem

**Mit einem Schlüssel (key) versehene Eintragungen sollen so gespeichert werden, dass sie aufgrund ihres Schlüssels effizient wiedergefunden werden können.**



Helmut Schauer  
Educational Engineering Lab  
Department for Informatics  
University of Zurich



## Schnittstelle

**boolean isEmpty()**

**int size()**

**void clear()**

**Object put(Object key, Object value)**

**Object get(Object key)**

**boolean containsKey(Object key)**

**Object remove(Object key)**



Helmut Schauer  
Educational Engineering Lab  
Department for Informatics  
University of Zurich



## Offene Hashverfahren

0		
1		
i	"Max"	"12345"
N-1		

$h(\text{key}) \dots$  Hashfunktion  $0 \leq h(\text{key}) < N$

$k(\text{key}) \dots$  Kollisionsfunktion  $0 < k(\text{key})$

```
put("Max", "12345");
```

```
i = h("Max")
```



Helmut Schauer  
Educational Engineering Lab  
Department for Informatics  
University of Zurich



## Eintragen von key und value

```
Object put(Object key, Object value) {  
    int i = hash(key);  
    int k = coll(key);  
    while (tab[i].isTaken())  
        i = (i+k)%N;  
    if (tab[i].isEmpty()) {  
        if (n>=N-1) return null; // table full  
        else n++;  
    }  
    tab[i] = new Entry(key,value);  
    return tab[i];  
}
```



Helmut Schauer  
Educational Engineering Lab  
Department for Informatics  
University of Zurich



## Aufsuchen von key

```
Object get(Object key) {  
    int i = hash(key);  
    int k = coll(key);  
    while (!tab[i].isEmpty()&!tab[i].key.equals(key))  
        i = (i+k)%N;  
    if (tab[i].isEmpty())  
        return null; // not found  
    else  
        return tab[i].value; // found  
}
```



Helmut Schauer  
Educational Engineering Lab  
Department for Informatics  
University of Zurich



## Hashfunktion

$$\text{hash}(\text{key}) = \text{key} \% N$$

## Kollisionsbehandlung

**linear:**

$$\text{coll}(\text{key}) = 1$$

**double hash:**

$$\text{coll}(\text{key}) = 1 + \text{key} \% (N - 2)$$

$N$  prim  $\Rightarrow N$  und  $\text{coll}(\text{key})$  teilerfremd



Helmut Schauer  
Educational Engineering Lab  
Department for Informatics  
University of Zurich



## Komplexität für offene Hashverfahren

**Belegungsgrad**       $\alpha = n/N$        $0 \leq \alpha < 1$

**Einfügen:**       $E(\alpha) = \sum \alpha^i = 1/(1-\alpha) \in O(1)$

**Aufsuchen:**       $A(\alpha) = 1/\alpha \int 1/(1-x) dx = 1/\alpha \ln(1/(1-\alpha)) \in O(1)$

**zB: 80% gefüllte Hashtabelle**

$$\alpha = 0.8$$

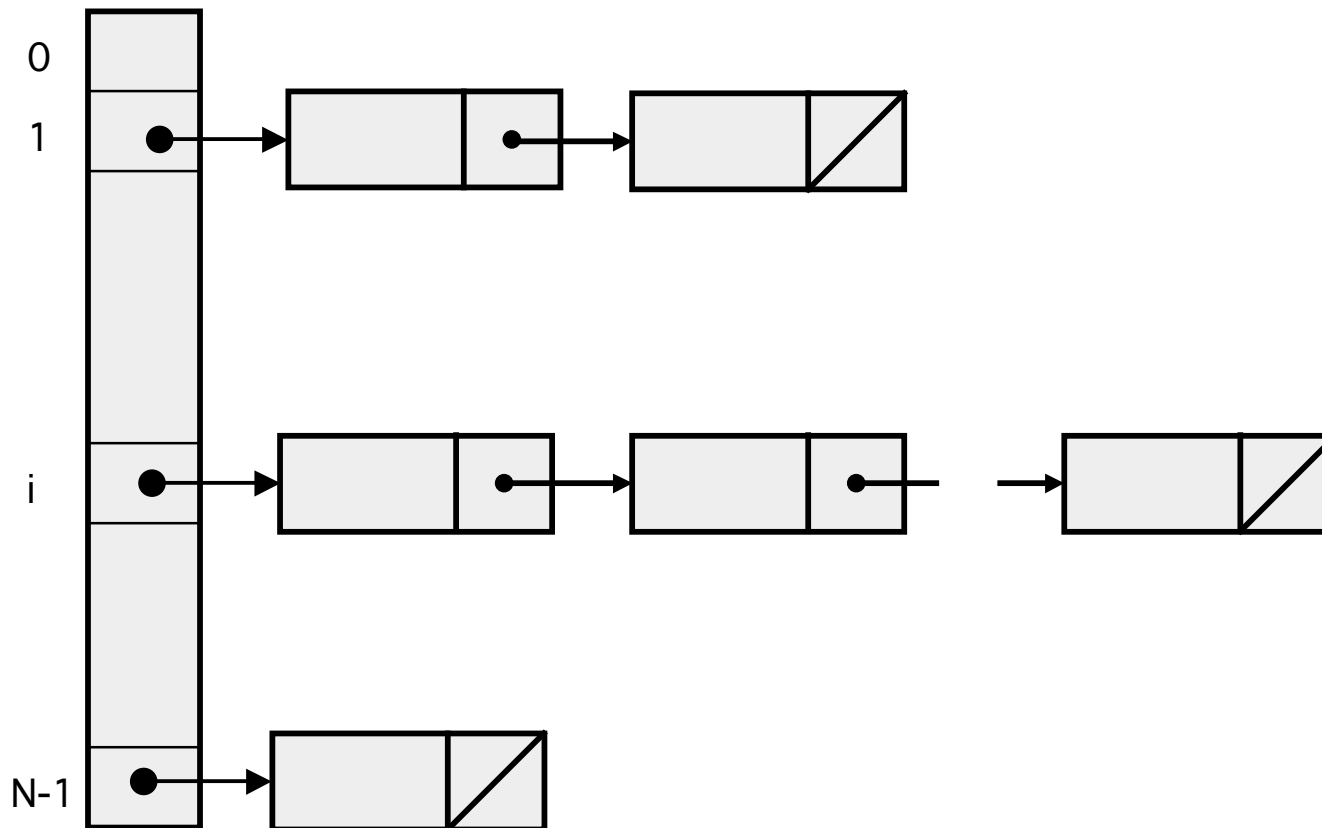
$$E(0.8) = 5$$

$$A(0.8) \approx 2$$





## Hashverfahren mit Verkettung





Helmut Schauer  
Educational Engineering Lab  
Department for Informatics  
University of Zurich



## Komplexität für Hashverfahren mit Verkettung

**Belegungsgrad**       $\alpha = n/N$        $0 \leq \alpha$

**Einfügen:**       $E(\alpha) = 1 \in O(1)$

**Aufsuchen:**       $A(\alpha) = 1 + \alpha/2 \in O(1)$