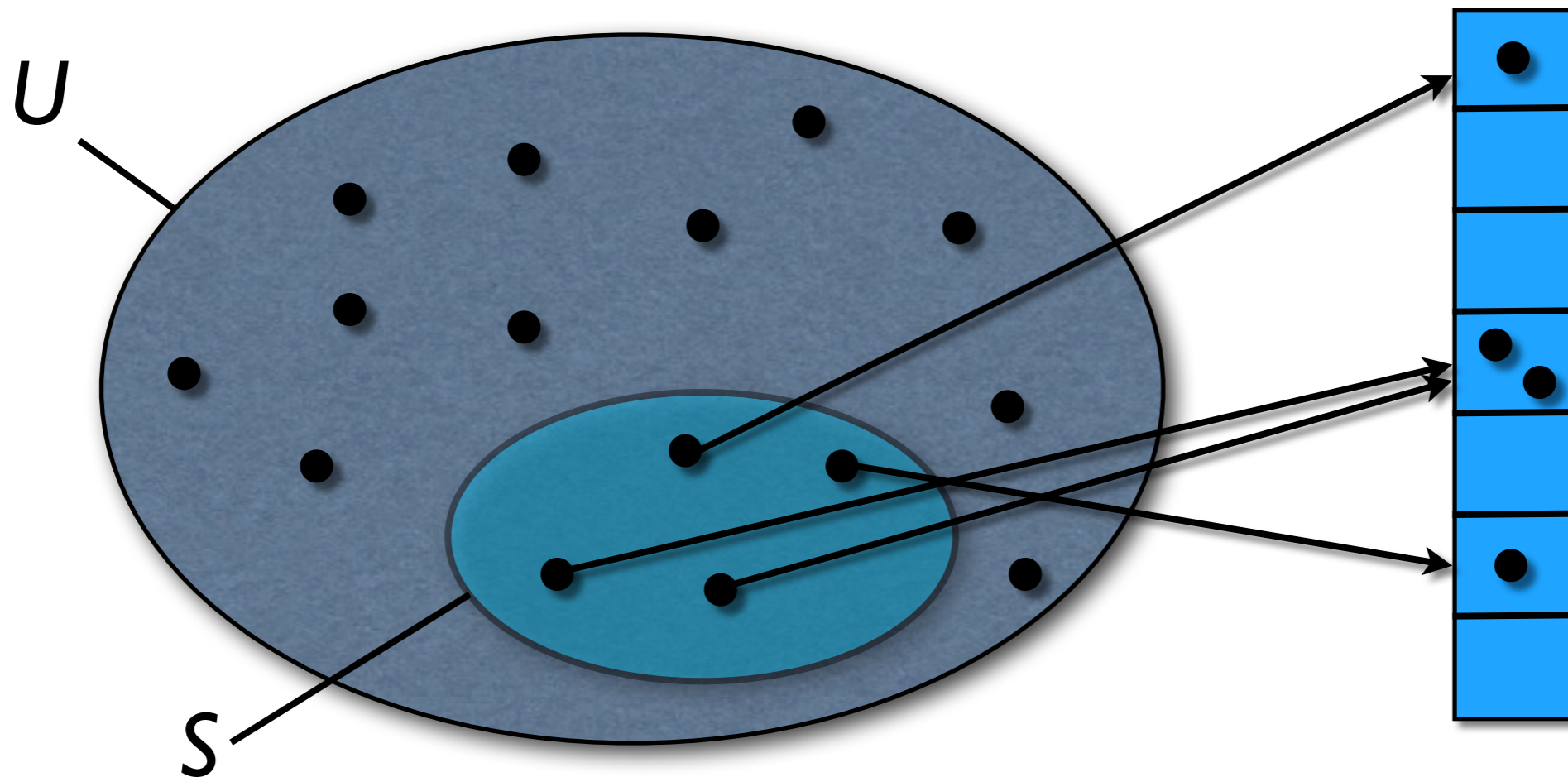


Lecture 2: Hashing

Johannes Fischer

Hashing

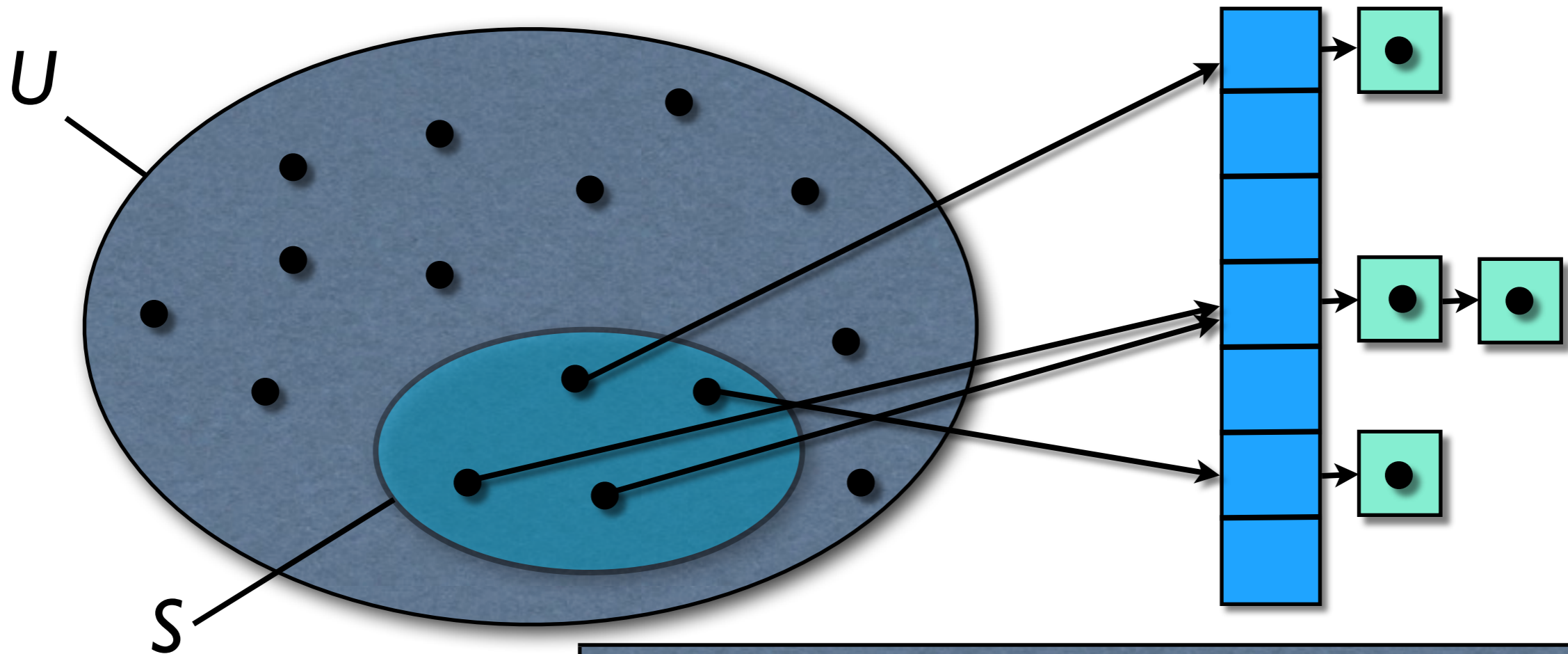
- set S of n objects from universe $U=[0,u-1]$
- operations: insert/delete/search



Baseline Algorithms

	array $A[0, u-1]$	linked list	balanced search tree
search	$O(1)$	$O(n)$	$O(\lg n)$
insert	$O(1)$	$O(1)$	$O(\lg n)$
delete	$O(1)$	$O(n)$	$O(\lg n)$
space	$O(u)$	$O(n)$	$O(n)$

Hashing with Chaining



hashing with chaining	
search	$O(1)$ expected
insert	$O(1)$ amortized
delete	$O(1)$ expected, amortized
space	$O(n)$

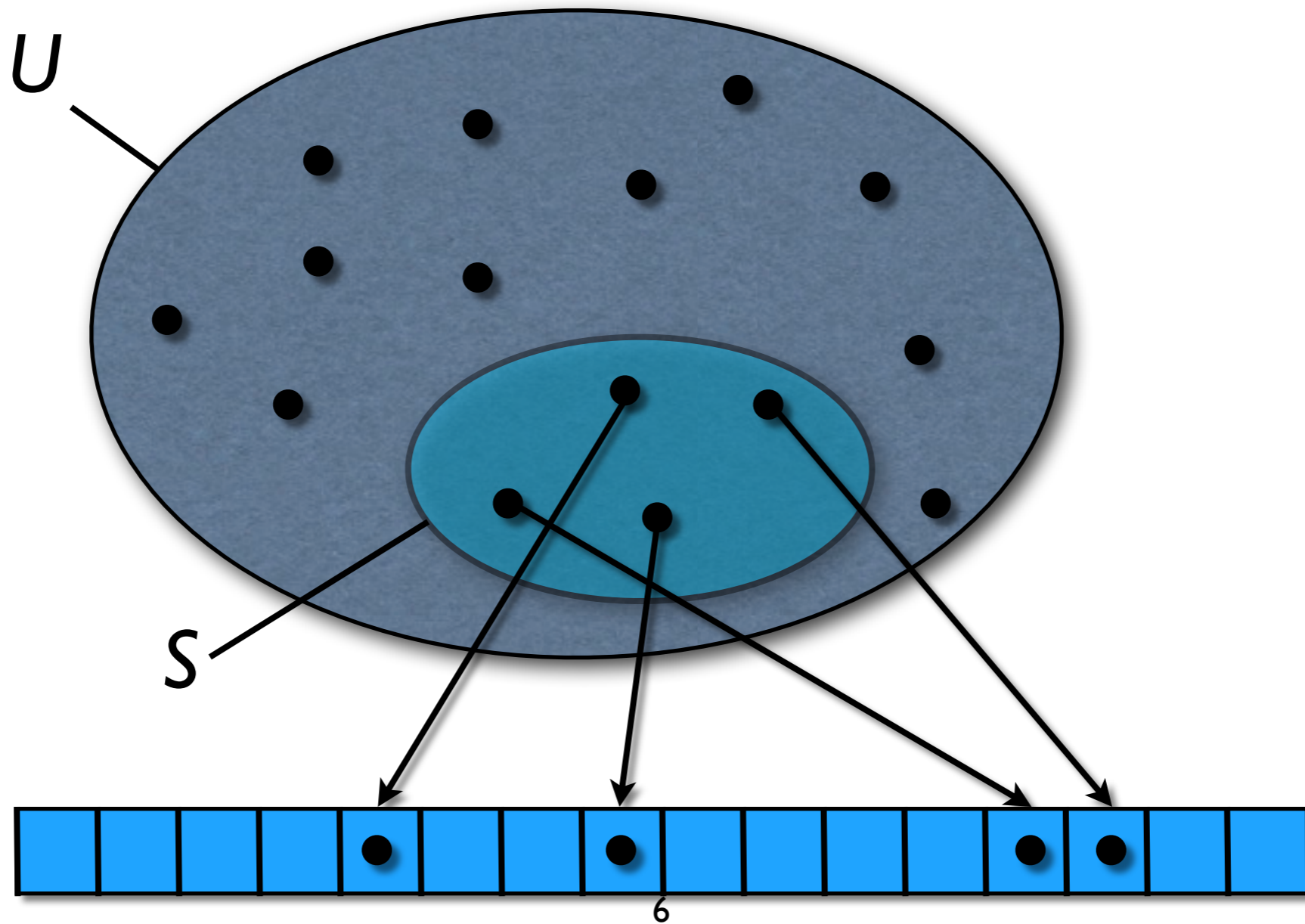
Perfect Hashing

- S static (insert all items at beginning)
 - ▶ can achieve $O(1)$ **worst case** search
- Fredman, Komlós, Szemerédi [J.ACM 1984]

perfect hashing	
search	$O(1)$ w.c.
construction	$O(n)$ exp.
space	$O(n)$ w.c.

Idea

- Table of size $n^2 \Rightarrow$ collisions **unlikely**

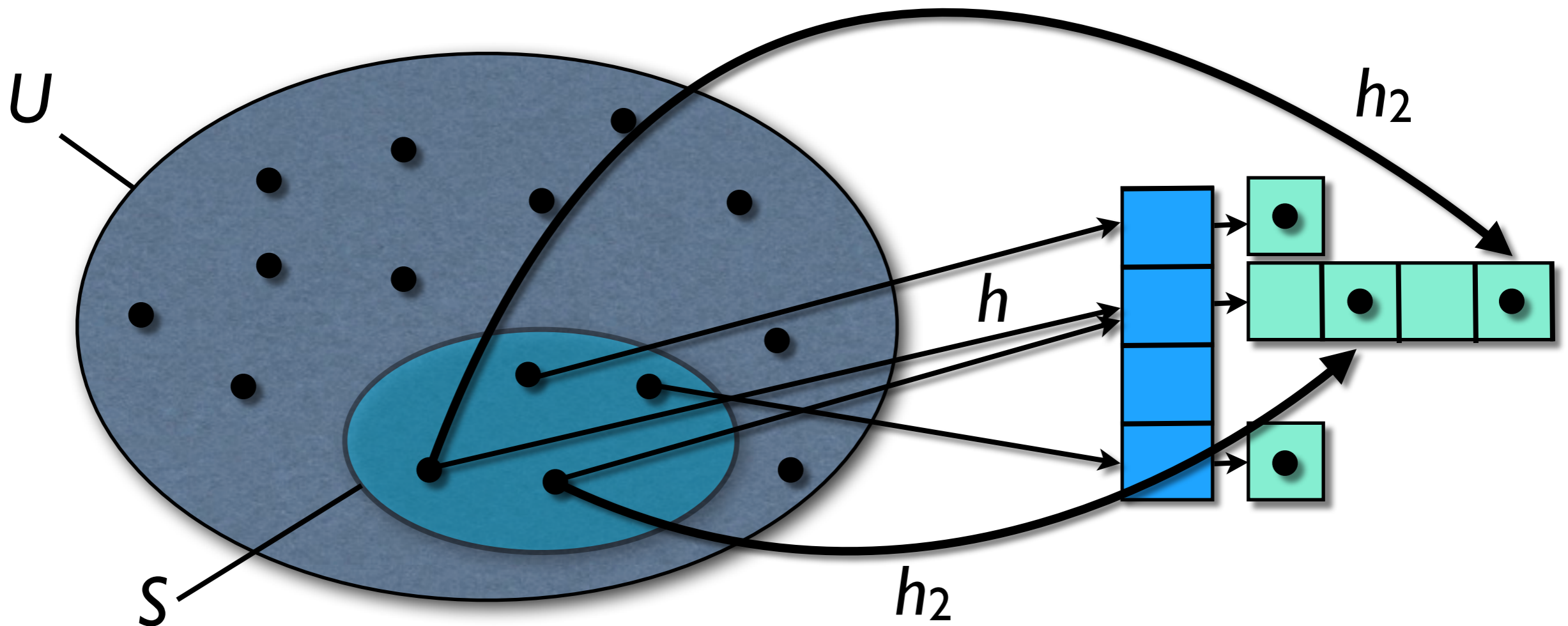


Idea

- Formally:
 - ▶ storing n keys
 - ▶ table of size $m=n^2$
 - ▶ universal hash function h ("truly random")
- ➔ Prob[single collision] $< 1/2$
- **Proof:** Prob[$h(x)=h(y)$] = $1/m$ for $x \neq y$
 - \Rightarrow Exp[#collisions] = #pairs \cdot Prob[$h(x)=h(y)$]
 - $= (n^2 - n)/2 \cdot 1/m = 1/2 - 1/2n < 1/2$
 - \Rightarrow Prob[#collisions ≥ 1] \leq Exp[#collisions] $< 1/2$ ■

Using the n^2 -idea

- start as in hashing with chaining ($m=n$)
- bucket i with n_i items: n^2 -idea with $m_i = n_i^2$
 - ▶ need n additional hash functions h_i



Space Usage

- Claim: $\text{Exp}[\sum_{1 \leq i \leq m} n_i^2] \leq 2n$

- **Proof:**
$$\begin{aligned} \sum n_i^2 &= \sum \left(2 \binom{n_i}{2} + n_i \right) \\ &= 2 \sum \binom{n_i}{2} + \sum n_i \\ &= 2 \sum \binom{n_i}{2} + n \end{aligned}$$

$$\Rightarrow \text{Exp}[\sum n_i^2] = n + 2 \text{Exp}[\sum \binom{n_i}{2}]$$

$$\text{Exp}[\sum \binom{n_i}{2}] = \binom{n}{2} \cdot \frac{1}{m} = \frac{n-1}{2}$$

$$\Rightarrow \text{Exp}[\sum n_i^2] = n + 2 \frac{n-1}{2} = 2n - 1 < 2n \blacksquare$$

The Final Picture

- try first hash function until $\sum n_i^2 = \Theta(n)$

- with Markov's inequality:

$$\text{Prob}[\sum n_i^2 \geq 4n] \leq 1/2$$

➡ on **average** less than 2 trials needed

- same true for the n secondary hash functions until collision free

➡ **expected** running time $O(n)$

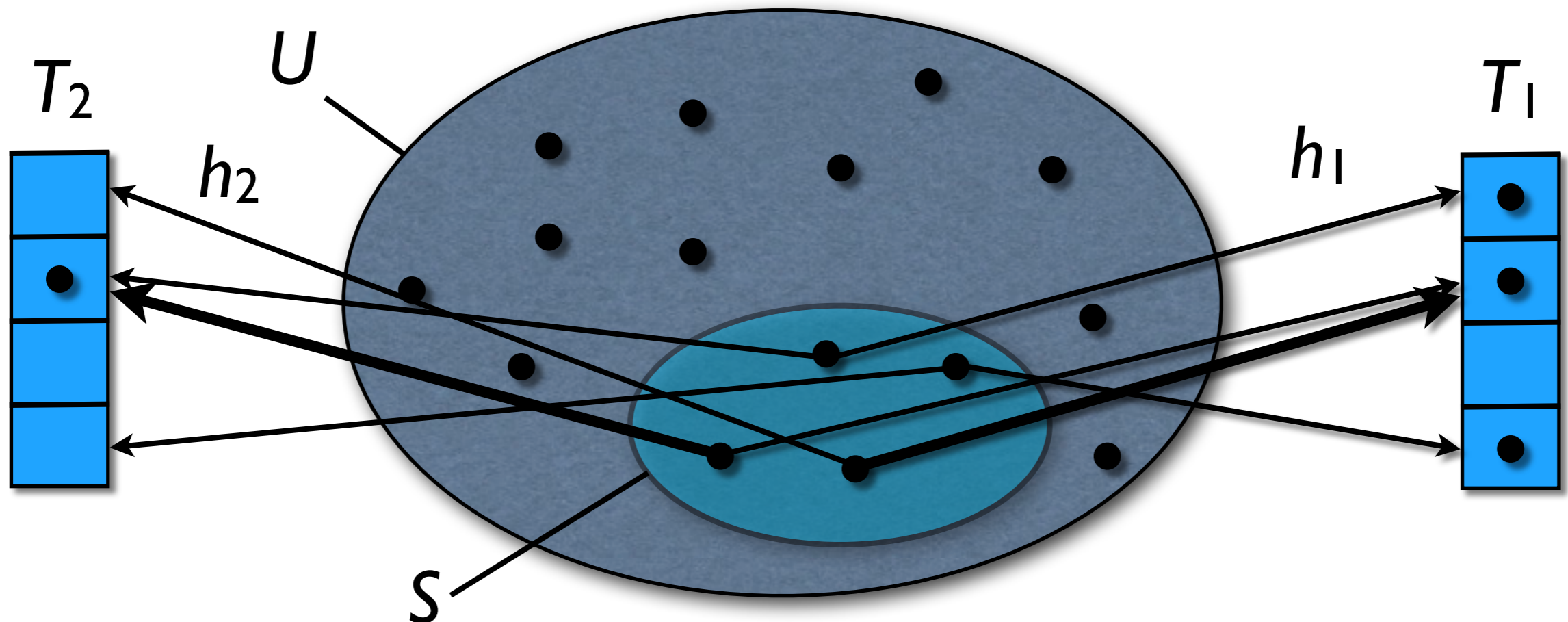
Cuckoo Hashing

- Performance of perfect hashing, but **dynamic**
- Pagh, Rodler [J.Alg. 2004]
- current state of the art for $O(1)$ w.c. search

cuckoo hashing	
search	$O(1)$ w.c.
insert	$O(1)$ exp., amort.
delete	$O(1)$ exp., amort.
space	$O(n)$ w.c.

Idea

- 2 tables T_1 and T_2 (both of size $\Theta(n)$)
- 2 hash functions h_1 and h_2
 - ▶ x either at $T_1[h_1(x)]$ or $T_2[h_2(x)]$



Simple Procedures

- table sizes $m \approx 2n$

function search(x):

if ($T_1[h_1(x)] = x$ **or** $T_2[h_2(x)] = x$) **return** true
otherwise return false

function delete(x):

if ($T_1[h_1(x)] = x$) $T_1[h_1(x)] \leftarrow \perp; n--$

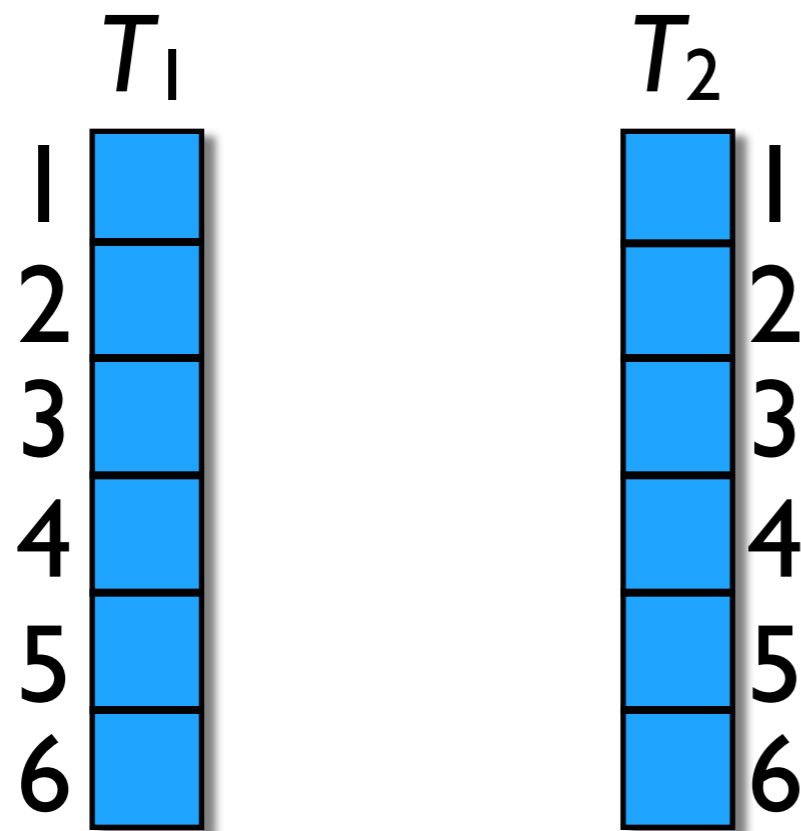
if ($T_2[h_2(x)] = x$) $T_2[h_2(x)] \leftarrow \perp; n--$

if ($n < m/8$) rehash($m/2$)

Insertion

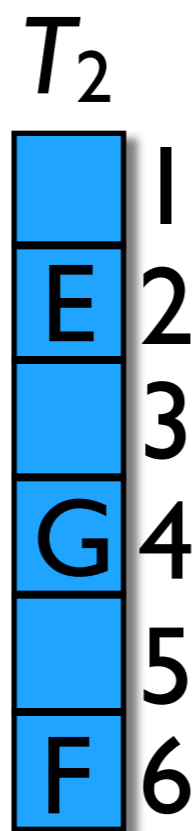
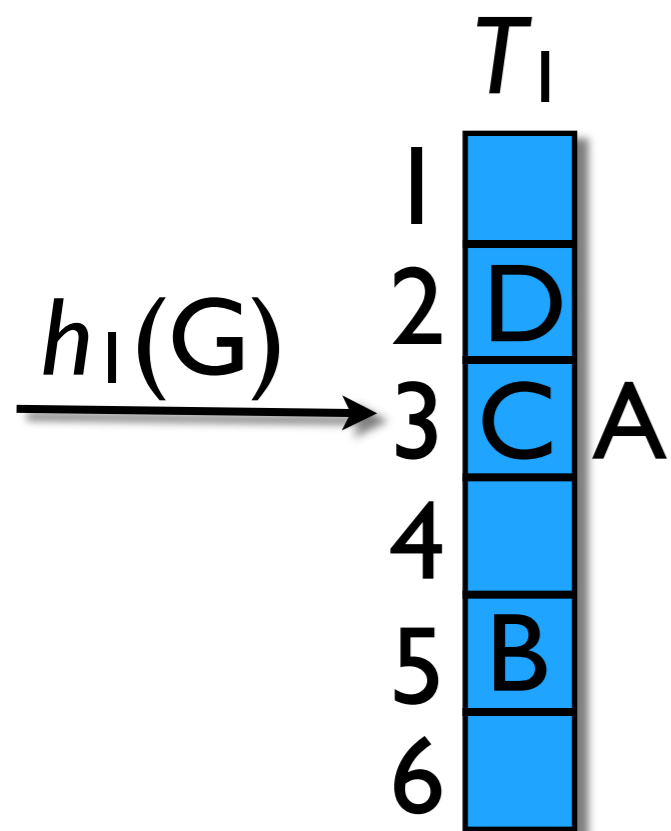
```
function insert(x):  
  if (search(x)) return  
   $k \leftarrow 1$   
  repeat maxLoop times:  
    swap  $x$  with  $T_k[h_k(x)]$   
    if ( $x = \perp$ )  
       $n++$ ; if ( $n > m/2$ ) rehash( $2m$ )  
    return  
   $k \leftarrow 3 - k$   
  rehash( $m$ ); insert( $x$ )
```

Example



x	$h_1(x)$	$h_2(x)$
A	3	2
B	5	2
C	3	6
D	2	4
E	5	2
F	2	6
G	3	4

Example



x	$h_1(x)$	$h_2(x)$
A	3	2
B	5	2
C	3	6
D	2	4
E	5	2
F	2	6
G	3	4

insert(G)

FAIL

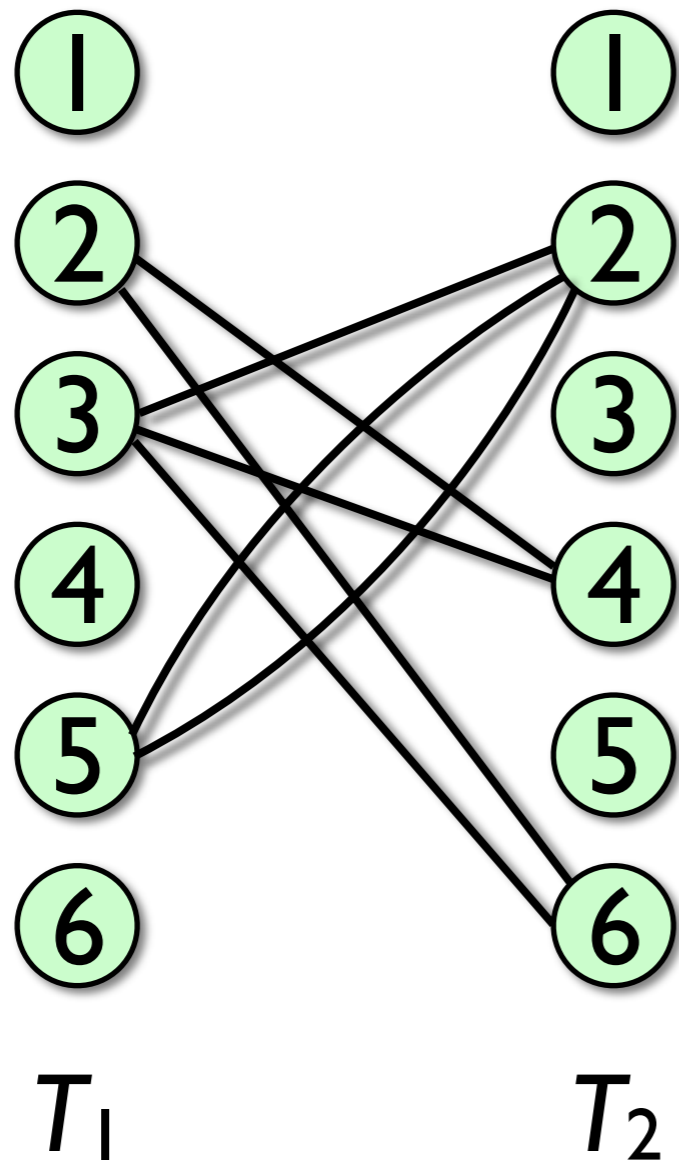
(c,k) -Universal Hash Functions

- recall **universal** h.f. $h: [1,u] \rightarrow [1,m]$
 $\text{Prob}[h(x)=h(y)] < 1/m$ for $x \neq y$
- **(c,k) -universal** h.f. $h: [1,u] \rightarrow [1,m]$
 $\text{Prob}[h(x_1)=y_1, \dots, h(x_k)=y_k] \leq c/m^k$
for fixed $x_i \neq x_j$, fixed y_1, \dots, y_k , and random h
- cuckoo hashing:
 - ▶ two $(1, \lg n)$ -universal h.f.
 - ▶ computed in $O(1)$ time and $O(n^\epsilon)$ space

Cuckoo Graph

- bipartite graph
 - ▶ vertices $\hat{=}$ cells of hash tables
 - ▶ edges $\hat{=}$ possibilities where elements hash
- Formally:
 - $V = \{T_k[x] : 1 \leq x \leq 2n, 1 \leq k \leq 2\} \Rightarrow |V|=4n$
 - $E = \{(T_1[h_1(x)], T_2[h_2(x)]) : x \in S\} \Rightarrow |E|=n$

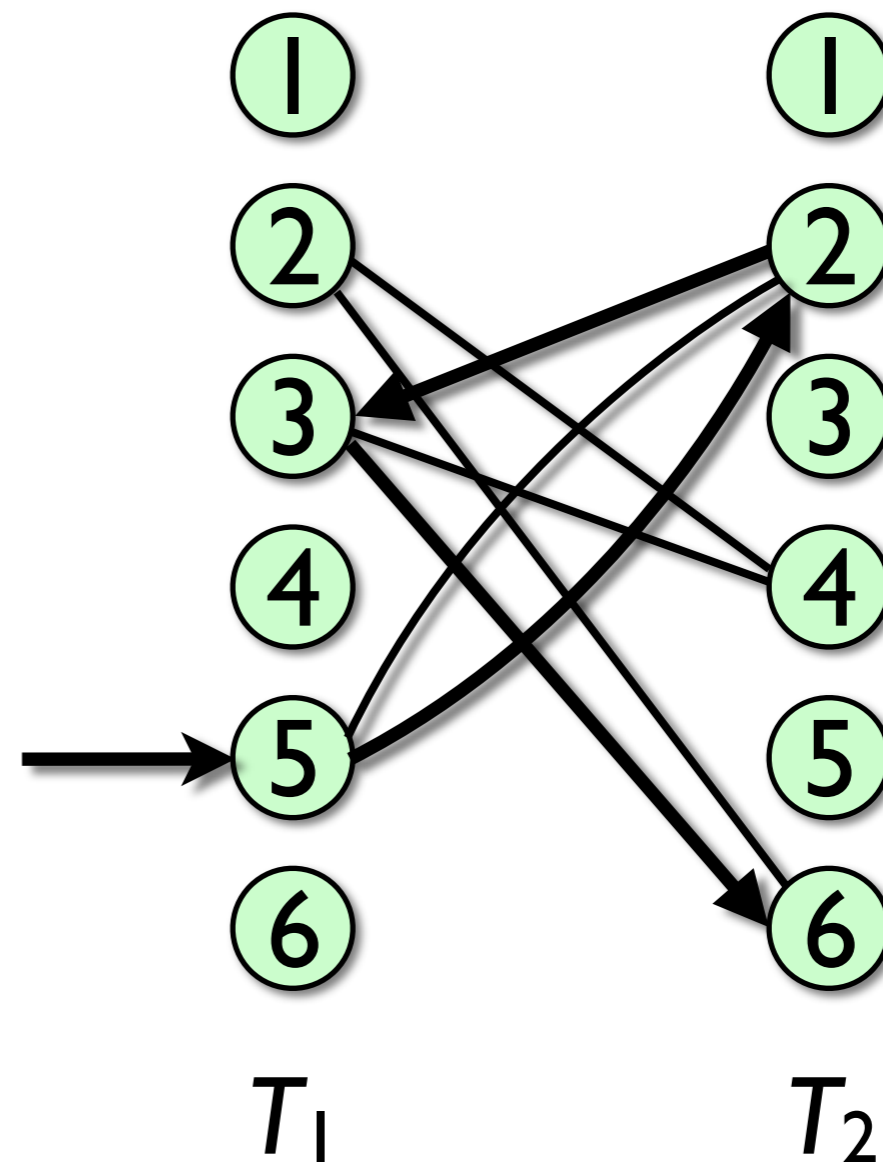
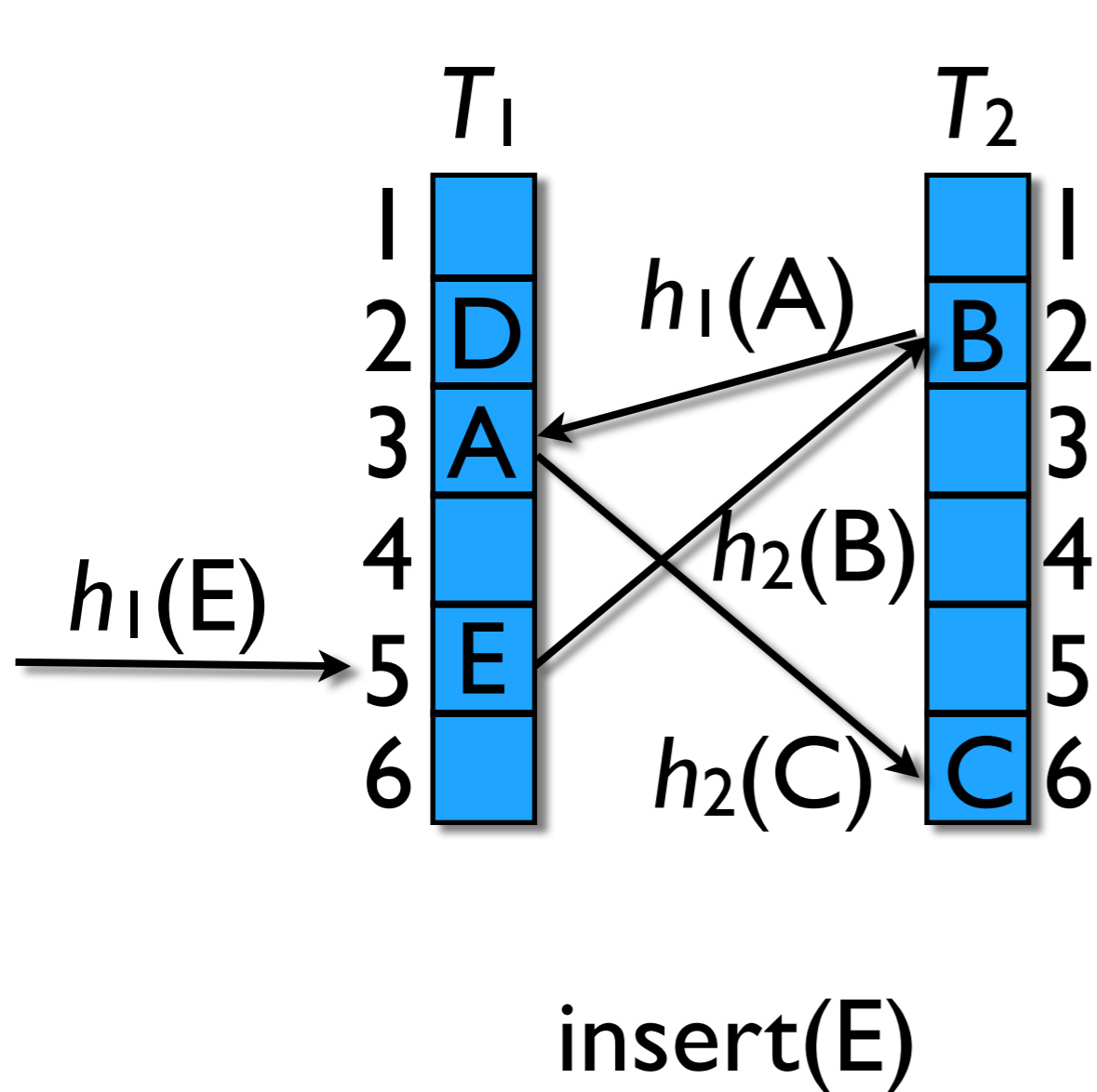
Cuckoo Graph



x	$h_1(x)$	$h_2(x)$
A	3	2
B	5	2
C	3	6
D	2	4
E	5	2
F	2	6
G	3	4

- insertions \Leftrightarrow walk in cuckoo graph

Example



Analysis

- \Rightarrow analyze

probability
of walks of length
maxLoop

▶ cause rehash!

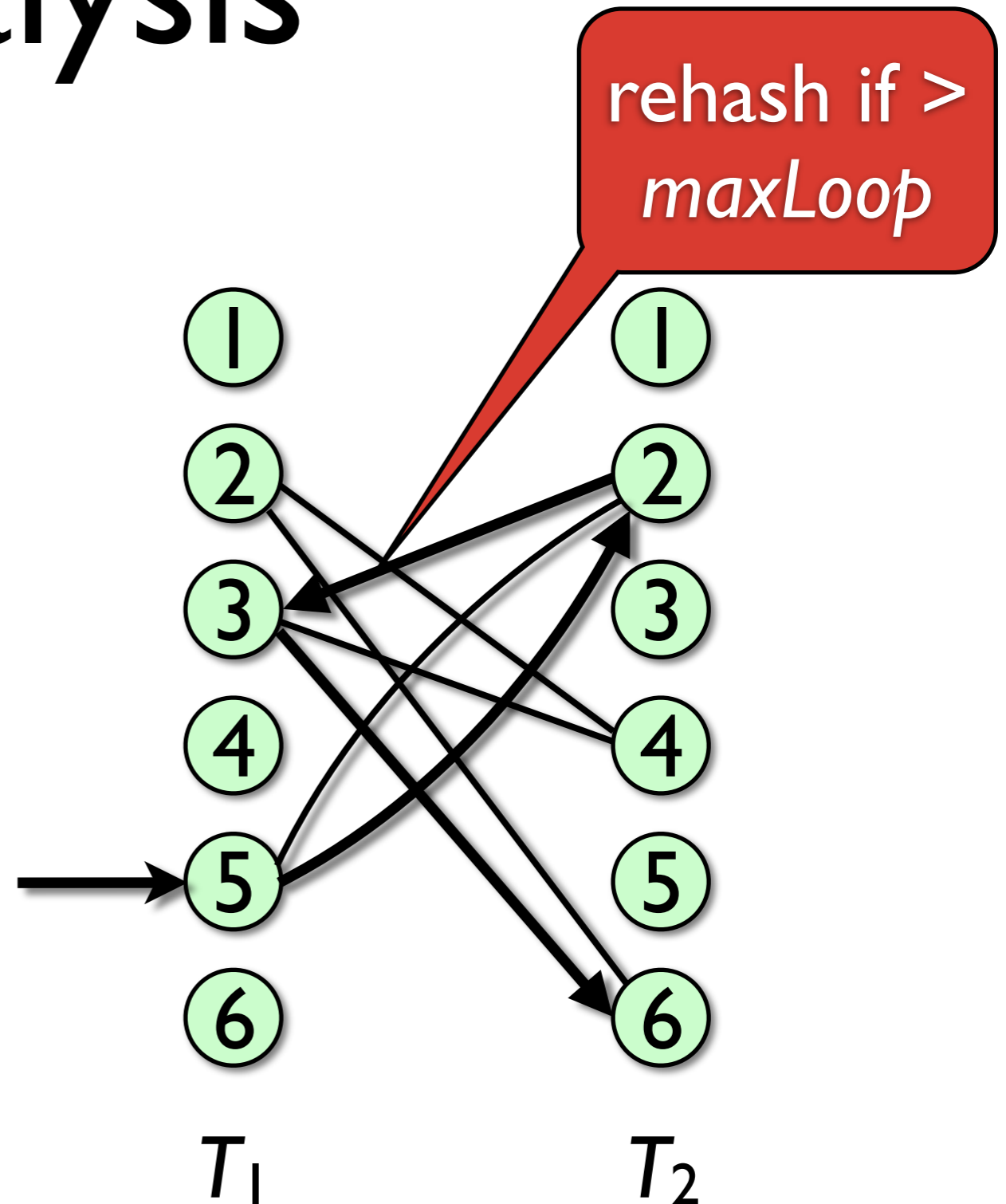
- distinguish 3 cases:

I. no cycle

II. 1 cycle

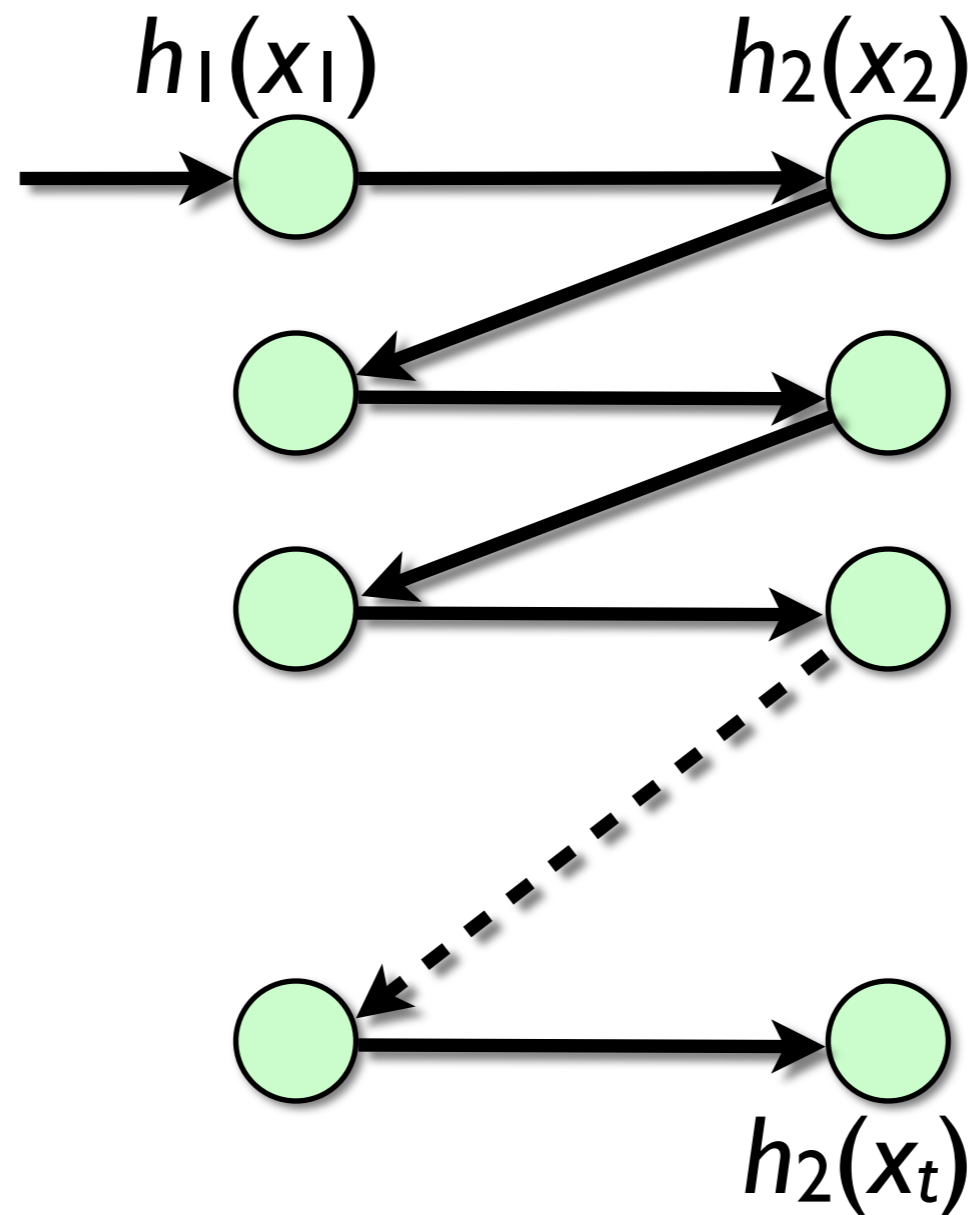
III. 2 cycles

- fix $maxLoop = 6 \lg n$



I: No Cycles

look at walk $h_1(x_1), h_2(x_2), h_1(x_3), \dots, h_{1/2}(x_t)$
with all vertices **different** ($x_i \neq x_j$)



I: No Cycles

$\text{Prob}[T_1[h_1(x_1)] \text{ occupied}]$

$$\leq \sum_{y_1 \in S, y_1 \neq x_1} \text{Prob}[h_1(x_1) = h_1(y_1)]$$

$$\leq (n-1) \frac{1}{m} \leq \frac{1}{2}$$

$\text{Prob}[T_1[h_1(x_1)] \text{ and } T_2[h_2(x_2)] \text{ occupied}]$

$$\leq \sum_{y_1, y_2 \in S} \text{Prob}[h_1(x_1) = h_1(y_1), h_2(x_2) = h_2(y_2)]$$

$$\leq \sum_{y_1, y_2 \in S} \text{Prob}[h_1(x_1) = h_1(y_1)] \cdot \text{Prob}[h_2(x_2) = h_2(y_2)]$$

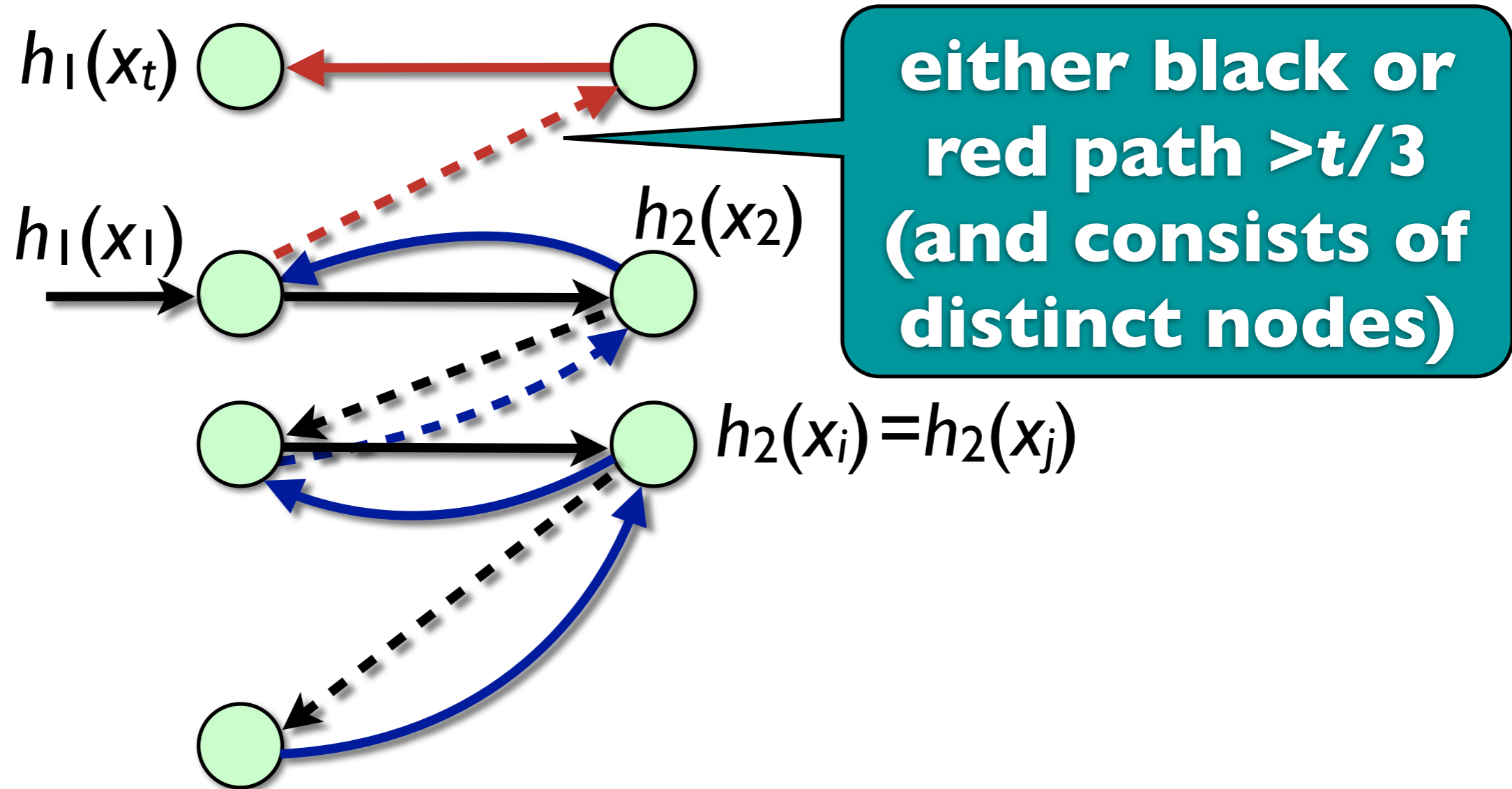
$$\leq n^2 \cdot \frac{1}{m^2} = \frac{1}{4}$$

I: No Cycles

- Prob[t cells occupied] $\leq 1/2^t = 1/n^6$
 \Rightarrow rehash occurs with probability $\leq 1/n^2$
- Expected running time:
Exp[#cells occupied] $\leq \sum_{1 \leq t \leq 6 \lg n} t \cdot 2^{-t}$
 $\leq \sum_{t \geq 1} t \cdot 2^{-t}$
(= 2) = $O(1)$

II: One Cycle

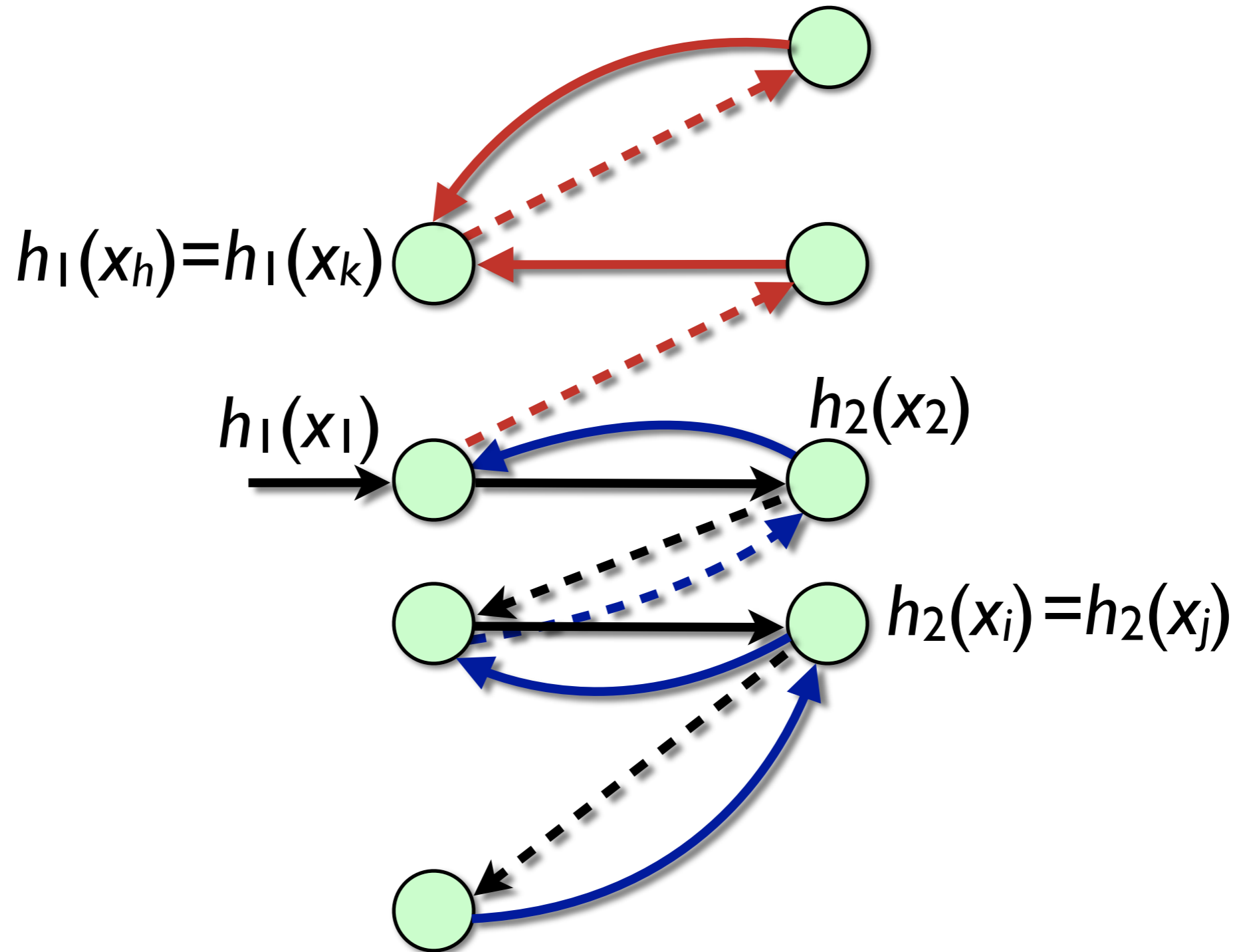
look at walk $h_1(x_1), h_2(x_2), h_1(x_3), \dots, h_{1/2}(x_t)$
 with cycle from $h_{1/2}(x_j)$ to $h_{1/2}(x_i)$ with $i < j$



II: One Cycle

- Prob[rehash]
< Prob[$t/3$ **different** cells occupied]
 $\leq 2^{-t/3} = 1/n^2$
 \Rightarrow rehash occurs with probability $\leq 1/n^2$
- Expected running time again $O(1)$

III: Two Cycles



III: Two Cycles

- Analysis by **counting** # length- t 2-cycle walks in (arbitrary) cuckoo graphs **starting at $h_1(x_1)$** ($N(n,m,t)$)
- walk $h_1(x_1), h_2(x_2), h_1(x_3), \dots, h_{1/2}(x_t)$ with $h_{1/2}(x_t)$ forming 2nd loop
- $N(n,m,t) \leq t^3 n^{t-1} m^{t-1}$
 - ▶ t^3 possible ways of forming 2 loops
 - ▶ x_i may be any $s \in S$ ($i \geq 2$) $\Rightarrow n^{t-1}$ choices
 - ▶ $h_{1/2}(x_i)$ may be any $h \in [1, m]$ ($i \geq 2$) $\Rightarrow m^{t-1}$ choices

III: Two Cycles

- probability of each possibility is $\leq m^{-2t}$:

$$\begin{aligned} & \text{Prob}[h_1(x_1)=i_1 \wedge h_2(x_1)=j_1 \wedge \dots \wedge h_1(x_t)=i_t \wedge h_2(x_t)=j_t] \\ &= \text{Prob}[h_1(x_1)=i_1 \wedge \dots \wedge h_1(x_t)=i_t] \cdot \text{Prob}[h_2(x_1)=j_1 \wedge \dots \wedge h_2(x_t)=j_t] \\ &= m^{-t} \qquad \qquad \qquad \cdot m^{-t} \\ &= m^{-2t} \end{aligned}$$

III: Two Cycles

- \Rightarrow probability of case (3) (=rehash) at most

$$\begin{aligned} & \sum_{t=3}^{6 \lg n} \frac{t^3 n^{t-1} m^{t-1}}{m^{2t}} \\ &= \sum_{t=3}^{6 \lg n} \frac{t^3 n^{t-1}}{m^{t+1}} \\ &= \frac{1}{mn} \sum_{t=3}^{6 \lg n} t^3 \left(\frac{n}{m}\right)^t \\ &\leq \frac{1}{2n^2} \sum_{t \geq 1} t^3 \left(\frac{1}{2}\right)^t = o\left(\frac{1}{n^2}\right) \end{aligned}$$

Wrapping Up

- Prob[**I** insert loops *maxLoop* times] = $O(n^{-2})$
- \Rightarrow Prob[**n** inserts cause rehash(*m*)] = $O(n^{-1})$
- \Rightarrow Prob[rehash **successful**] = $1 - O(n^{-1})$
- \Rightarrow Exp[# **trials** for rehash] = $O(1)$
- \Rightarrow Exp[**time** for rehash] = $O(n)$
- \Rightarrow $O(1)$ amortized insert. time (exp.)

```
function insert(x):  
    if (search(x)) return  
     $k \leftarrow 1$   
    repeat maxLoop times:  
        swap x with  $T_k[h_k(x)]$   
        if ( $x = \perp$ )  
             $n++$ ; if ( $n > m/2$ ) rehash( $2m$ )  
        return  
     $k \leftarrow 3 - k$   
    rehash(m); insert(x)
```