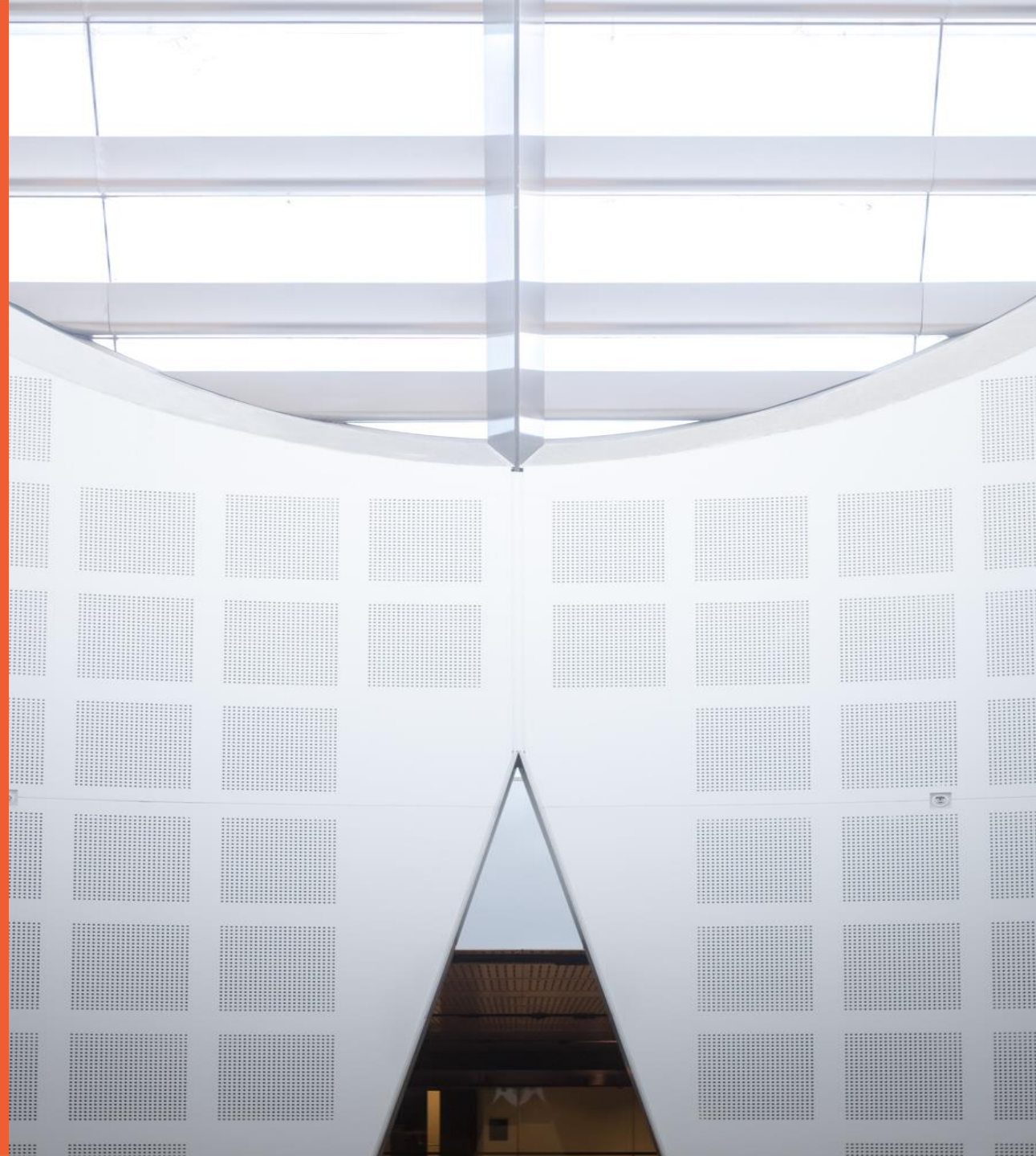
COMPx270: Randomised and Advanced Algorithms

Lecture 6: Hashing and Friends

Clément Canonne

School of Computer Science

THE UNIVERSITY OF SYDNEY



A question ❓

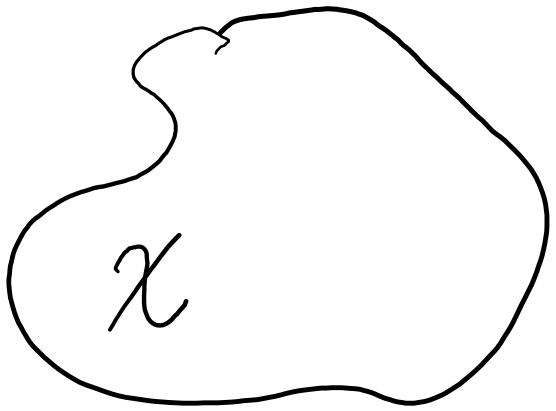
A SID is of the form 450687816 (9 digits, each between 0 and 9). How many distinct SIDs are there?

10^9

How many distinct students have there even been at U Syd?

$\leq 250 \cdot 40{,}000 = 10^7$

Dictionaries (Maps, Associative arrays...)



$m = 10^9$

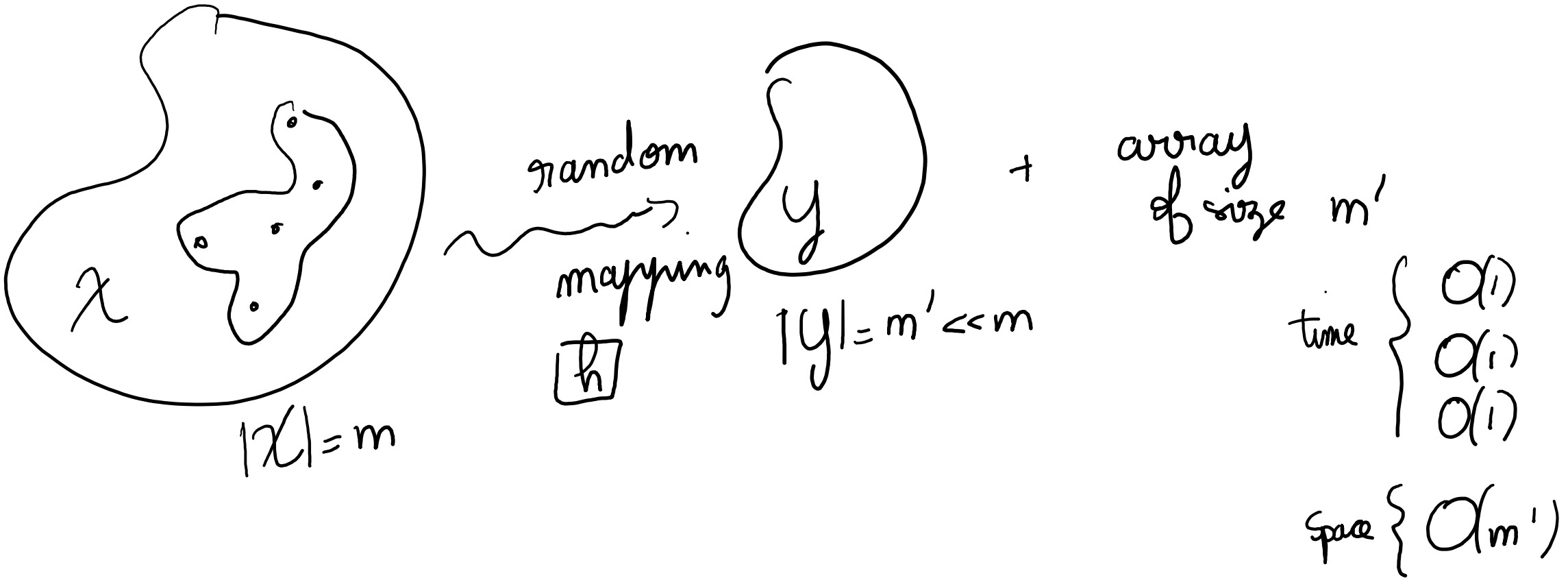
$S \subseteq \mathcal{X}$

$|S| << |\mathcal{X}|$

n $\quad$ m

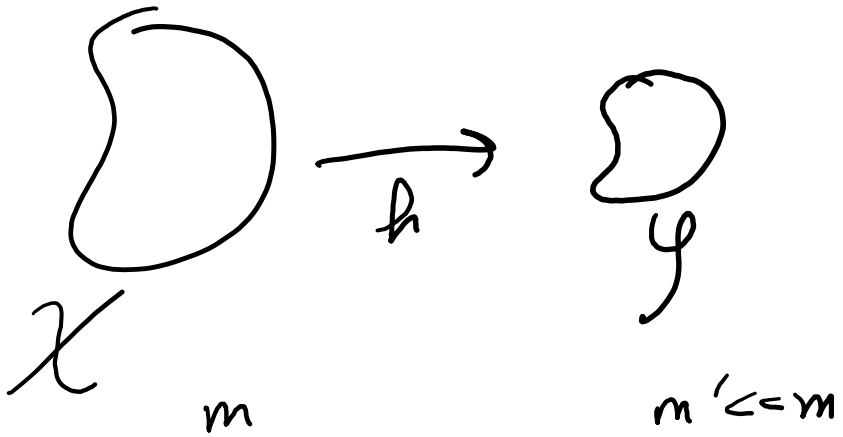
	LIST	ARRAY	BST
INSERT	~~$O(1)$~~/$O(n)$	$O(1)$	$O(\log n)$
LOOKUP	$O(n)$	$O(1)$	$O(\log n)$
REMOVE	$O(n)$	$O(1)$	$O(\log n)$
SPACE	$O(n)$ √	$O(m)$ x	$O(n)$ √

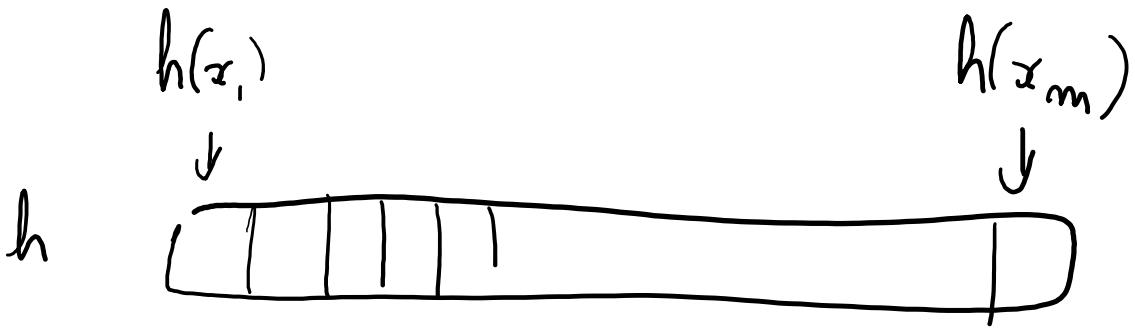
Hash tables


x
|x| = m
random
mapping
h
y
|y| = m' << m
+
array
of size m'
time
O(1)
O(1)
O(1)
space
O(m')

Hash tables: what is random?

- Dataset?
- h itself → $O(m \log m')$ space ✗
- $\mathcal{H}$: hash family → $O(\log |\mathcal{H}|)$

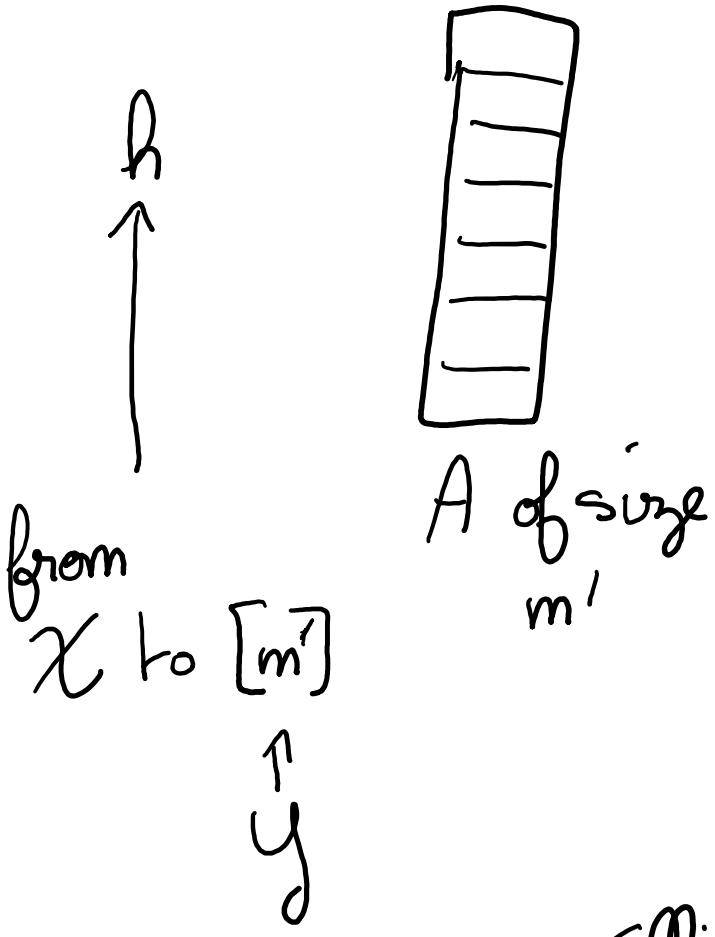




S: n points

universal hash family $\forall x \neq x' \quad \Pr_{h \sim \mathcal{H}}[h(x) = h(x')] \leq \frac{1}{m}$

Hash tables: the data structure



$m \gg m', \gg n$

INSERT(x): $A[h(x)] \leftarrow 1$

LOOKUP $A[h(x)] == 1$?

REMOVE $A[h(x)] \leftarrow 0$

SPACE: $O(\log |\mathcal{H}| + m')$

Can we set $m' = O(n)$?

Collision $x \neq x'$

$h(x) = h(x')$

INSERT(x')

LOOKUP(x)

REMOVE(x)

LOOKUP(x')

Hash tables: no collisions?

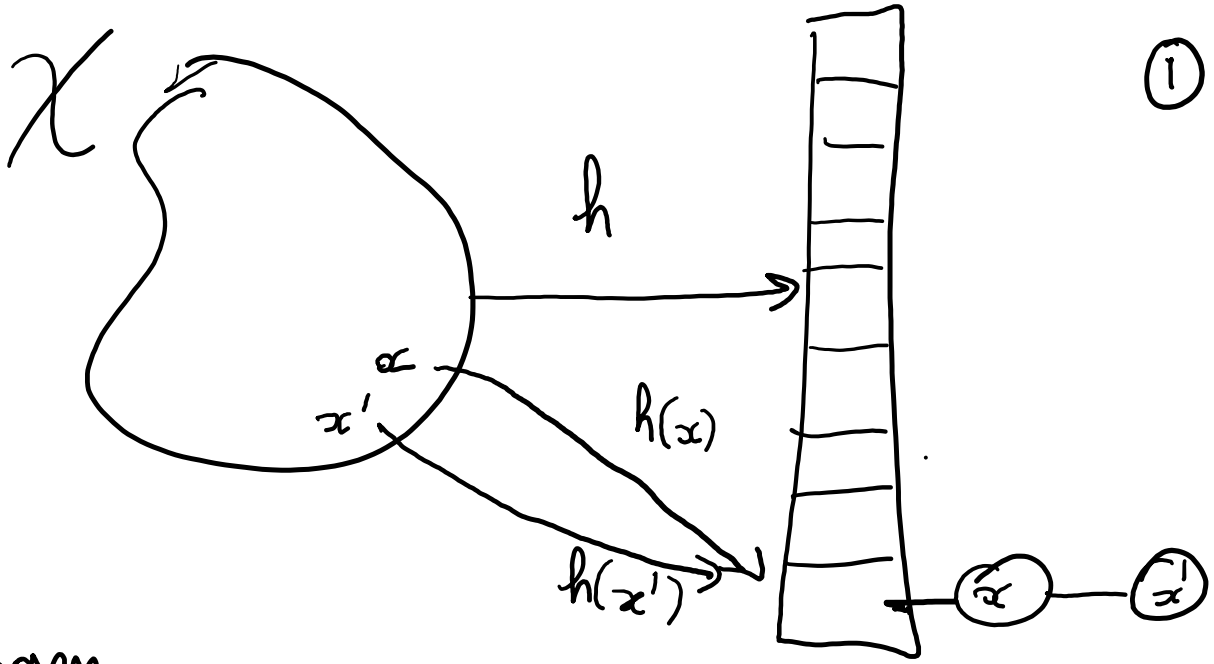
- $m' = \Omega(n^2)$ (bad?)
- If $m' = O(n)$

max # collisions in some bucket $= \Theta\left(\frac{\log n}{\log \log n}\right)$

Hash tables: ~~no~~ collisions

Load factor

$\alpha = \frac{n}{m'}$



① Chaining

$|A| = m' = O(n)$

(h chosen once and for all from "good" hf)

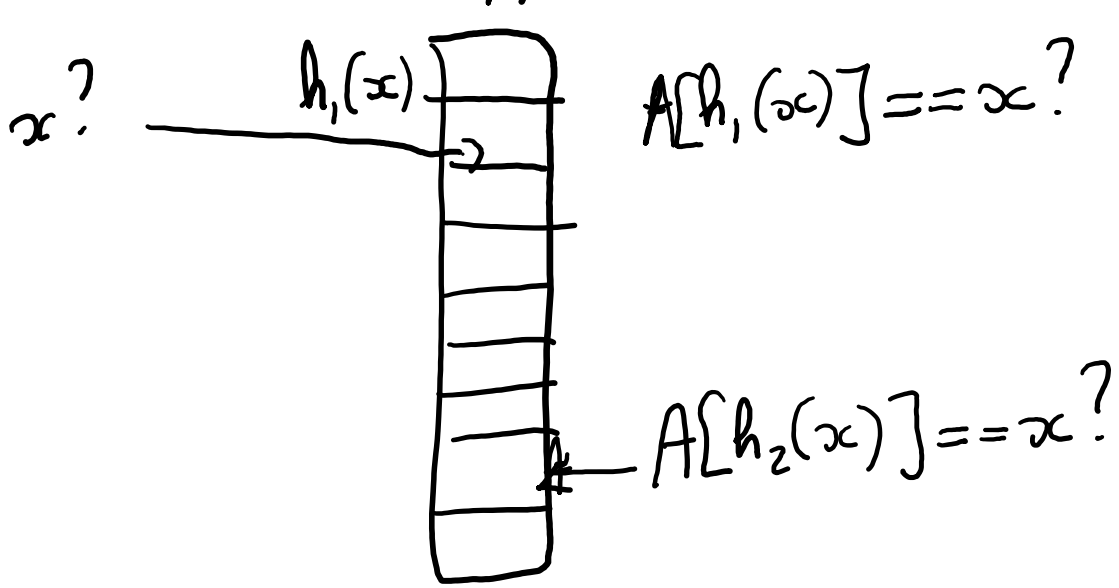
Expected time complexity of INSERT LOOKUP REMOVE $O(1+\alpha)$

Worst-case will happen $O(n)$ $O\left(\frac{\log n}{\log\log n}\right)$

Hash tables: collisions

$\alpha = \frac{n}{m'}$

$h_1, -, h_T$ hash functions



$\forall x, (h_1(x), -, h_T(x))$ is a permutation of $[m']$

$T = m'$

Open addressing ($\alpha \leq 1$)

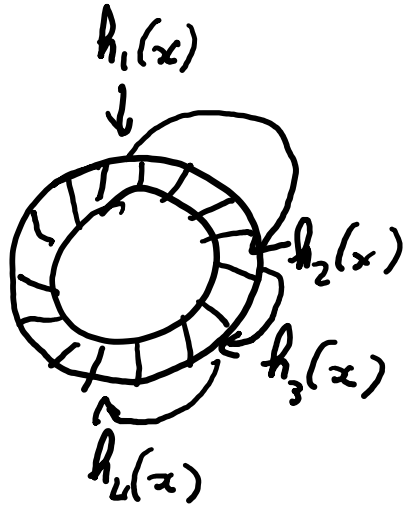
- Query complexity can be bad: $O(\alpha n)$
- Space: $O(m' \log m + m')$

Handling collisions: separate chaining

Handling collisions: open addressing

$h_1, \ldots, h_{m'}$

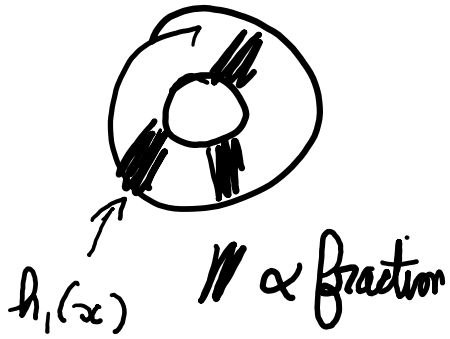
```
INSERT(x)
    For 1 ≤ t ≤ m'
        if A[h_t(x)] == x                    Return
        if A[h_t(x)] == ∅  or  A[h_t(x)] == ⊥
            A[h_t(x)] ← x
            return
```




```
REMOVE(x)
    For 1 ≤ t ≤ m'
        if A[h_t(x)] == x        A[h_t(x)] ← ⊥ ; return
        if A[h_t(x)] == ∅        return
```

Handling collisions: open addressing

Assuming [stuff], expected time complexity of LOOKUP is $O\left(\frac{1}{1-\alpha}\right)$



[stuff] $\forall x$, $(h_1(x), \ldots, h_m(x))$ is a u.a.r permutation of $[m']$

$$\mathbb{E}[T_{n,m'}] = O(1) + \alpha \cdot \mathbb{E}[T_{n-1,m'-1}]$$

$$\leq O(1) + \alpha \mathbb{E}[T_{n,m'}]$$

"hardware"

$$\rightarrow \mathbb{E}[T_{n,m'}] \leq \frac{O(1)}{1-\alpha}$$

Handling collisions: open addressing (linear probing)

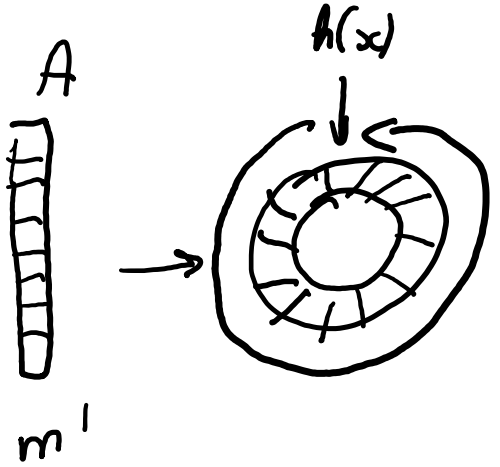
$h_1, \ldots, h_{m'}$ are one hash function in disguise

$$h_1(x) = h(x)$$

$$h_2(x) = h(x) + 1 \quad [m']$$

$$\downarrow$$

$$h_{m'}(x) = h(x) + m' - 1 \quad [m']$$



Theorem Under some assumptions, expected time complexity of LOOKUP is $O\left(\frac{1}{(1-\alpha)^2}\right)$

Handling collisions: open addressing (cuckoo hashing)

h_1, A_1

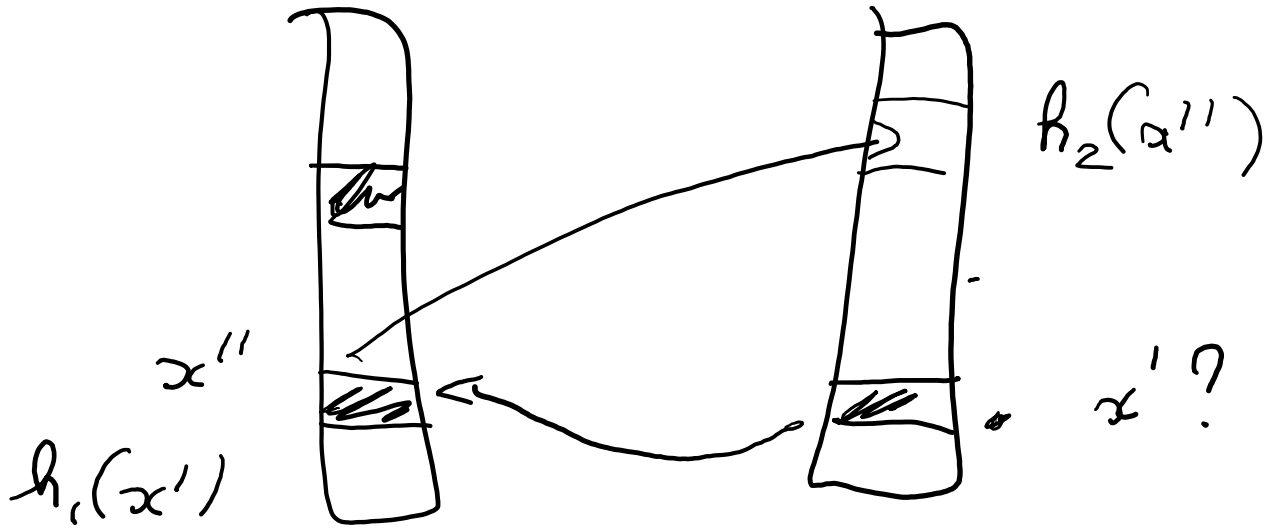
h_2, A_2

$x \longrightarrow A_1[h_1(x)]$ or $A_2[h_2(x)]$

LOOKUP } O(1) worst-case

REMOVE }

INSERT } O(1) expected



Hash tables: summary

Can we do better? Bloom filters!

Bloom filter: faster (O(1) for real)
less space (cst factors)

But LOOKUP is sometimes wrong (false positives)