

Problem ("Dictionary") Given set  $S$  of  $n$  integers in  $\{0, 1, \dots, U-1\}$

build data structure st. we can quickly answer

membership queries: (given  $y$ , is  $y \in S$ ?)  
Static case

e.g. Sorted array  $\Rightarrow$   $\begin{cases} O(\log n) & \text{query time} \\ O(n \log n) & \text{preprocessing time} \\ O(n) & \text{space} \end{cases}$

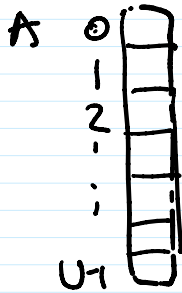
Solve more general Problem: predecessor search

("dynamic": insert/delete)

$O(\log n)$  update time by balanced search trees

better?

Method 0 "bit vector"



$O(1)$  query time  
 $O(U)$  space

("RAM model of computation")

Method 1 hash table, with chaining

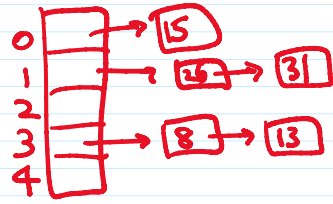
Pick hash fn  $h: \{0, \dots, U-1\} \rightarrow \{0, \dots, m-1\}$   
for some  $m \ll U$ .

Store array  $A[0, \dots, m-1]$

where  $A[i] =$  list of all  $x \in S$  with  $h(x) = i$   
 $\rightarrow$  "bucket" / "bin"

e.g.  $h(x) = x \bmod m$

$\{8, 13, 15, 26, 31\} \quad m=5$



query( $y$ ):

search the list  $A[h(y)]$

if input is rand. unif distributed,  
each bucket has  $\sim \frac{n}{m}$  elements "on average"

Set  $m \approx n \Rightarrow O(n)$  space / preproc time  
 $O(1)$  "average" query time

but can't assume input is random

key idea - [Carter, Wegman '77]

randomize the choice of hash fn  $h$ .

Ex Assume  $U$  is prime.

Pick rand.  $a \in \{1, \dots, U-1\}$   
 $b \in \{0, \dots, U-1\}$

Define  $h_{ab}: \{0, \dots, U-1\} \rightarrow \{0, \dots, m-1\}$ :

$$h_{ab}(x) = \left( \underbrace{(ax+b) \bmod U}_{O(1) \text{ time}} \right) \bmod m$$

$\uparrow$   
 $O(1)$  time

Property For any fixed  $x, y \in \{0, \dots, U-1\}$ ,  $x \neq y$ ,

$$\Pr_{a,b} [ h_{ab}(x) = h_{ab}(y) ] \leq O\left(\frac{1}{m}\right)$$

called universality  $\rightarrow$

Called universality  $\rightarrow$

$a, b \in \{0, \dots, U-1\}$

$\rightarrow$  More strongly, for any fixed  $x, y \in \{0, \dots, U-1\}, x \neq y$ ,  
& fixed  $i, j \in \{0, \dots, m-1\}$ ,

Called 2-universality  $\rightarrow$   $\Pr_{a,b} \left( \underbrace{h_{a,b}(x)} = i \wedge \underbrace{h_{a,b}(y)} = j \right) \leq O\left(\frac{1}{m^2}\right)$

Pf: For  $s, t \in \{0, \dots, U-1\}$ ,

$$\left\{ \begin{array}{l} \Pr_{a,b} \left[ \begin{array}{l} ax + b \equiv s \pmod{U} \\ \wedge ay + b \equiv t \pmod{U} \end{array} \right] \\ = \frac{1}{U(U-1)} \end{array} \right. \quad \leftarrow \begin{array}{l} 2 \times 2 \text{ linear eqns} \\ \text{over } \mathbb{Z}_U \\ \text{have unique root} \\ \text{in vars } a, b \end{array}$$

$$\Rightarrow \Pr_{a,b} \left( h_{a,b}(x) = i \ \& \ h_{a,b}(y) = j \right) \quad \left( \begin{array}{l} a \equiv \frac{s-t}{x-y} \pmod{U} \\ b \equiv s - ax \pmod{U} \end{array} \right)$$

$$= \sum_{\substack{s, t \in \{0, \dots, U-1\} \\ s \pmod{m} = i \\ t \pmod{m} = j}} \Pr_{a,b} \left( \begin{array}{l} (ax + b) \pmod{U} = \underline{s} \\ \wedge (ay + b) \pmod{U} = \underline{t} \end{array} \right)$$

$$\leq O\left(\frac{U}{m} \cdot \frac{U}{m} \frac{1}{U(U-1)}\right) = O\left(\frac{1}{m^2}\right) \quad \square$$

Space  $O(m+n)$

Expected query time: for fixed  $y$ ,

$$\begin{aligned} & E \left[ \# x \in S - \{y\} \text{ with } h_{a,b}(x) = h_{a,b}(y) \right] \\ & \leq E \left[ \sum_{x \in S - \{y\}} I_x \right] \quad I_x = \begin{cases} 1 & \text{if } h_{a,b}(x) = h_{a,b}(y) \\ 0 & \text{else} \end{cases} \\ & = \sum_{x \in S - \{y\}} E[I_x] \\ & = \sum_{x \in S - \{y\}} \Pr \left( \underline{h_{a,b}(x) = h_{a,b}(y)} \right) \end{aligned}$$

$$= \sum_{x \in S - \{y\}} \Pr(\underline{h_{a,b}(x) = h_{a,b}(y)})$$

$$\leq O\left(n \cdot \frac{1}{m}\right)$$

Set  $m \approx n \Rightarrow$

- $O(1)$  expedited query time
- $O(n)$  space
- $O(n)$  preproc time

Q: can we get worst-case query time?  
 yes (in static case)