April 5, 2021 9:09 AM

Hashing

assume $S \subseteq \{0, 1, \ldots, U-1\}$

membership queries: is $q \in S$?

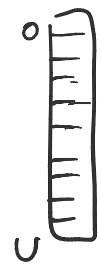
insert

delete

trivial (bit vector):

$O(1)$ time

but space is $O(U)$



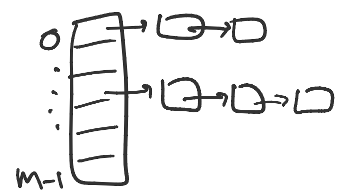
Hashing Method 1:

pick a hash fn $h: \{0, \ldots, U-1\} \to \{0, \ldots, m-1\}$ for some $m \ll U$.

store array $A[0, \ldots, m-1]$

where

$A[i]$ = list of all $x \in S$ with $h(x) = i$



search(y):

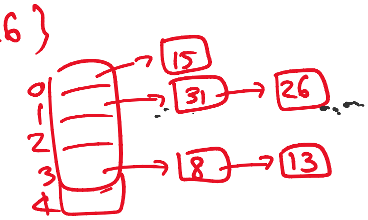
find y in $A[h(y)]$ by linear search

insert $O(1)$

delete $O(1)$

e.g. $S = \{31, 15, 8, 13, 26\}$

$h(x) = x \bmod 5$



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

set $m \approx n \Rightarrow$ $O(n)$ space

$O(1)$ "average" query time

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

But can't assume input is random!

idea - pick a random hash fn from a family

Def Fix prime $p \in [U, 2U]$..
Pick rand $a \in \{1,..,p-1\}$, $b \in \{0,...,p-1\}$.
Define $h_{a,b} : \{0,...,U-1\} \to \{0,..,m-1\}$:

$$\boxed{h_{a,b}(x) = \big((ax+b) \bmod p\big) \bmod m}$$

evaluated in $O(1)$ time

(similar to 2-point sampling)

Prop For any fixed $x, y \in \{0,..,U-1\}$, $(x \neq y)$,

$$\Pr_{a,b}\big[\underbrace{h_{a,b}(x) = h_{a,b}(y)}\big] \le O\left(\tfrac{1}{m}\right).$$

called universal (Carter-Wegman '79)

More strongly, for fixed $i, j \in \{0,..,m-1\}$,

$$\Pr_{a,b}\big[\underbrace{h_{a,b}(x) = i \;\wedge\; h_{a,b}(y) = j}\big] \le O\left(\tfrac{1}{m^2}\right).$$

called strong 2-universal

Pf: Fix $i', j' \in \mathbb{Z}_p$ $(i' \neq j')$.

$$\Pr_{a,b}\left[\begin{array}{c} ax+b \equiv i' \\ \wedge \;\; ay+b \equiv j' \end{array} \pmod p\right] = \frac{1}{p(p-1)}.$$

2x2 linear system
in vars a, b
has unique sol'n

$$a \equiv \frac{i'-j'}{x-y}, \qquad b \equiv i' - ax$$

$$\Rightarrow \Pr_{a,b}\left[\, h_{a,b}(x) = i \wedge h_{a,b}(y) = j \,\right)$$

$$\leq \left(\begin{array}{l} \#\text{ choices} \\ \text{of } i', j' \in \mathbb{Z}_p \; (i' \neq j') \\ \text{with } \; i' \equiv i \mod m \\ \qquad\quad j' \equiv j \mod m \end{array}\right) \cdot \frac{1}{p(p-1)}$$

$$\leq \left\lceil \frac{p}{m} \right\rceil \cdot \left\lceil \frac{p}{m} \right\rceil \cdot \frac{1}{p(p-1)}$$

$$= O\left(\frac{1}{m^2}\right). \qquad \square$$

Rmk: other ex of 2-universal hash families

Dietzfelbinger et al. '97:

$$\underline{h_a} : \{0, .., 2^w - 1\} \to \{0, .., 2^\ell - 1\}$$

$$h_a(x) = \left\lfloor \frac{(a \cdot x) \bmod 2^w}{2^{w-\ell}} \right\rfloor.$$

Patrascu-Thorup '10:

tabulation hashing
(using XORs & tables of rand values)

⋮

Analysis of query time:

for fixed query value $\underline{y}$,

$E[\;\#$ elems of S that collide with $y]$

$$E[\#\text{ elems } x \in \text{ of } S \text{ that } \underset{\substack{\Updownarrow \\ h_{a,b}(x) = h_{a,b}(y)}}{\text{collide}} \text{ with } y]$$

$$= E_{a,b}\Big[\sum_{x \in S - \{y\}} [h_{a,b}(x) = h_{a,b}(y)]\Big]$$

$[E] = \begin{cases} 1 & \text{if } E \text{ true} \\ 0 & \text{else} \end{cases}$

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

$$\underset{\text{by universality}}{\le} O\left(n \cdot \frac{1}{m}\right).$$

Set $m \approx n \Rightarrow$ $\boxed{O(1)}$ expected query time

$\boxed{O(1)}$ insert/delete

$\boxed{O(n)}$ space

assume oblivious adversary
(query values indep. of rand. choices made by algm)

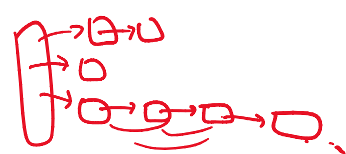
But worst-case query time ??

Assume static ...

Alternative analysis:

$$E[\text{total } \# \text{ of colliding pairs } (x,y) \in S \times S]$$

$$= E\Big(\sum_{x,y \in S} (h_{a,b}(x) = h_{a,b}(y))\Big)$$



$$\le O\left(n^2 \cdot \frac{1}{m}\right) = O\left(\frac{1}{c}\right).$$

(related to "birthday paradox")

Set $m = cn^2$ ← (related to "birthday paradox")

$\Rightarrow \Pr[\text{total \# of colliding pairs} \geq 1]$

$\leq O\left(\frac{1}{c}\right)$ by Markov's ineq.

$\Rightarrow$ with prob. $\Omega(1)$, no collision ← perfect hashing

repeat till success

$\Rightarrow$ $\boxed{O(1)}$ worst-case query time

$\boxed{O(n^2)}$ space

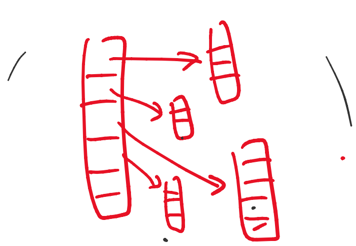
$\boxed{O(n)}$ expected preproc time

But can space be reduced back to $O(n)$?

Final Hashing Method (Fredman, Komlós, Szemerédi '84)

idea - bootstrap

store each bucket $A[i]$ in the data structure with $O(|A[i]|^2)$ space & $O(1)$ worst-case query time



2-level hash table

expected space

$$O\left(m + E\left[\sum_i |A[i]|^2\right]\right)$$

$$= O\left(m + E[\text{total \# colliding pairs}]\right)$$

$$= O\left(m + \frac{n^2}{m}\right)$$

Set $m = n \Rightarrow O(n)$

repeat till success

$$\Rightarrow \boxed{O(n)} \text{ worst-case space}$$

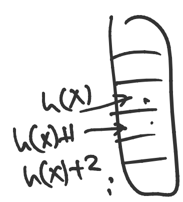
$$\boxed{O(1)} \text{ worst-case query time}$$

$$\boxed{O(n)} \text{ expected preproc time}$$

Rmk - made dynamic by Dietzelbinger et al. '94 with $O(1)$ expected update time

Rmks - alternatives with single table (open addressing)

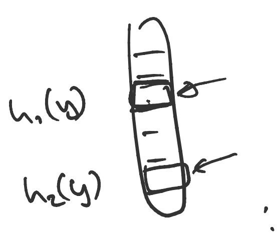
linear probing



assume $m \geq (1+\varepsilon)n$

cuckoo hashing

search(y)



use two hash fns h_1, h_2