Randomized Algo.

def A dictionary is a data structure over $\mathcal{U} = \{0, \dots, N-1\}$ for storing set $S \subseteq \mathcal{U}$ of keys with associated value.

Supports $\begin{cases} \text{insert}(x, y): \text{ add } x \text{ to } S \text{ with value } y \\ \text{lookup}(z): \text{ decide if } z \in S, \text{ if so, return value.} \end{cases}$

The complexity measured wrt $n = |S|$.

wrt $\downarrow$ with respect to

The space complexity is the # of integers used to store S.

Q :

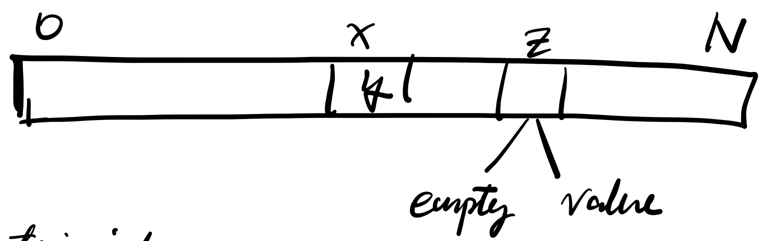
What type of integers are we dealing with ?

Convention : all integers are in $O(\log n)$.

$\Downarrow$

$N \le \mathrm{poly}(n)$

Example array



insert (x, y) $\rightarrow$ trivial

lookup (z) $\rightarrow$ return $\begin{cases} \text{empty} \\ \text{value} \end{cases}$

parameters : space : $N = |U|$ $\Rightarrow$ size of universe

insert : $O(1)$

lookup : $O(1)$

often $|S| \ll |U|$

$|S|$ $\downarrow$ actual net id

$|U|$ $\downarrow$ all possible net id

Example linked list

insert(x,y) $O(1)$

lookup(z) $O(|S|)$ $\Longrightarrow$ bad.

space $O(|S|) = O(n)$

Q better?

Want: space $O(n)$

insert $O(1)$

lookup $O(1)$

Yes, randomization.

but expected run time.

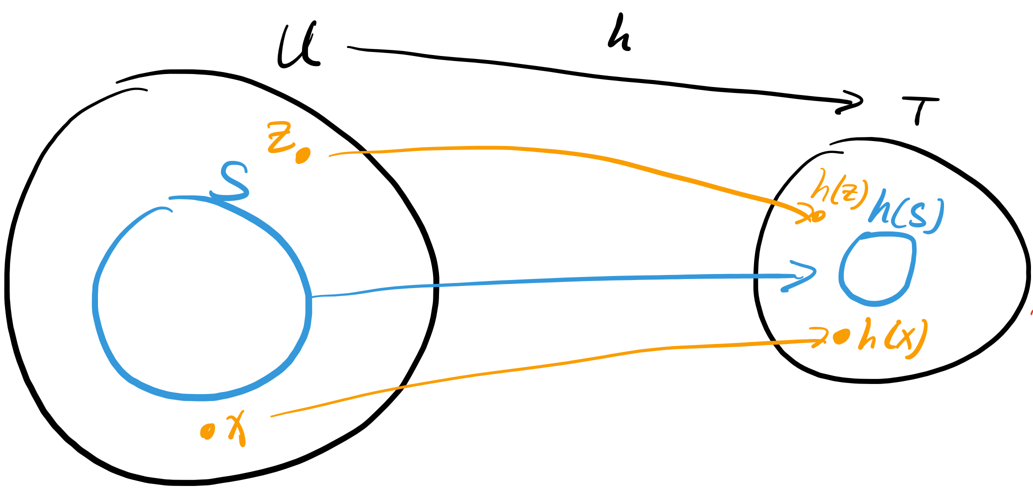
Idea: hashing

function $h : U \to T$

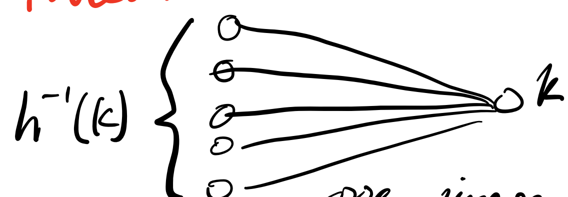
with $|T| \simeq |S|$

$\Downarrow$

array is affordable.



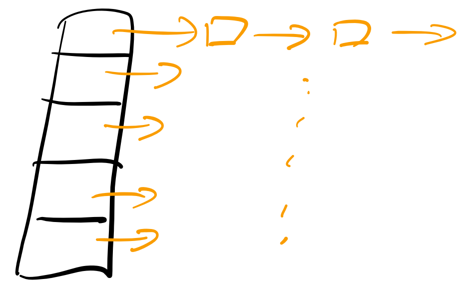
Problem: collision



pre-image can be huge

def Hash table w/ chain is:

- hash function h
- Array of size m of linked list.



Insert(x,y) insert (x,y) to $L[h(x)]$

lookup(z) = lookup z in $L[h(x)]$

prop insert(x,y) takes

- 1 eval of h
- $O(1)$ ops

prop lookup(z) takes

- 1 eval of h
- $O(|L[h(z)]|)$

def: The load of hash func h on a set S at key k is $|L[k]|$

prop: space complexity is

$$O(|T|) + O(|S|)$$

Idea: randomly choose h

prop $S \subseteq U$ $\quad h: U \to T$ random function. Any $z \in U$

$$\mathbb{E}[\,|L[h(z)]|\,] \le 1 + \frac{|S|}{|T|}$$

$$= \Theta(1) \text{ if } |T| = \Theta(|S|)$$

Pf: $$|L[h(z)]| = \sum_{x \in S} \mathbb{1}[h(x) = h(z)]$$

$$= \mathbb{1}[z \in S] + \sum_{\substack{x \in S \\ x \neq z}} \mathbb{1}[h(x) = h(z)]$$

$$\mathbb{E}[\,|L[h(z)]|\,] = \mathbb{E}[\mathbb{1}[z \in S]] + \sum_{\substack{x \in S \\ x \neq S}} \mathbb{E}\Big[\mathbb{1}[h(x) = h(z)]\Big]$$

$$= \mathbb{1}[z \in S] + \underbrace{\sum}_{|S|} \underbrace{\Pr(h(x) = h(z))}_{\frac{1}{|T|}}$$

$$\le 1 + \frac{|S|}{|T|}$$

Q Does this work?

A: No. Storing $h: U \to T$ is expensive. Takes $|U|$ to store everything used.

Idea: choose h pseudo random.

- enough randomness so that $\mathbb{E}[\text{load}]$ small
- not too much randomness to avoid

def <u>universal hash function</u> is a collection of hash functions

$$\mathcal{H} = \{h : U \to T\} \text{ s.t. } x \neq y \in U, \ \Pr[h(x) = h(y)] = \frac{1}{|T|}$$

prop Let p be a prime

$$H : \mathbb{Z}_p^k \times \mathbb{Z}_p^k \to \mathbb{Z}_p$$

given by $H(x,b) = \sum_{i=1}^{k} x_i b_i$

claim: this is universal hash function.

hash family: $\mathcal{H} = \left\{ h : \mathbb{Z}_p^k \to \mathbb{Z}_p \,\middle|\, h(x) = H(x,b),\ b \in \mathbb{Z}_p^k \right\}$

Each $h \in \mathcal{H}$ - can be stored in $O(k)$

- can be evaluated in $O(k)$

Pf: h is given by $b \in \mathbb{Z}_p^k \Rightarrow k$ integers $\Rightarrow O(k)$

$h(x) = \sum_{i=1}^{k} x_i b_i$ is simply k multiplication and addition $\Rightarrow O(k)$

universality:

lemma: $\mu_x = \mathbb{Z}_p \to \mathbb{Z}_p$ multiplication by x map.

is injective when $x \neq 0$.

pf

$$\mu_x(y) = \mu_x(z) \iff xy = xz$$

$$\iff x(y-z) = 0$$

$$\iff p \mid x(y-z)$$

$$\iff p \mid y-z \quad \text{as } p \nmid x$$

$$\iff y = z$$

lemma: u_x is a bijection Pf $|\mathbb{Z}_p| = |\mathbb{Z}_p|$

lemma:

$x \neq 0$, y is uniform over $\mathbb{Z}_p$

$\Rightarrow$ $x \cdot y$ is uniform over $\mathbb{Z}_p$

Pf: $$\Pr[\underbrace{x \cdot y}_{u_x} = z] = \Pr[y = \underbrace{u_x^{-1}(z)}_{\text{fixed}}] \Rightarrow \frac{1}{p}$$

lemma: $\underbrace{X \text{ over } \mathbb{Z}_p, \ Y}_{\text{independent random variable}}$ uniform over $\mathbb{Z}_p$ $\Rightarrow$ $X+Y$ uniform.

Pf: $$\Pr[X+Y=z]$$

$$= \sum_x \underbrace{\Pr[\underbrace{x+Y=z}_{Y=z-x} \mid X=x]}_{\Pr[Y=z x] = \frac{1}{p}} \cdot \Pr[X=x]$$

because X, Y independent.

$$= \frac{1}{p} \cdot \sum_{x} \Pr[X = x] = \frac{1}{p}$$

Now, back to universality of hash function

lemma: $x \neq y \in \mathbb{Z}_p^k$, $\Pr_{b \in \mathbb{Z}_p^k}[H(x,b) = H(y,b)] = \frac{1}{p}$

Pf: $\exists i_0$ s.t. $x_{i_0} \neq y_{i_0}$

$$\Pr[H(x,b) = H(y,b)]$$

$$= \Pr\left[\sum x_i b_i = \sum y_i b_i\right]$$

$$= \Pr\left[\sum_{i=1}^{k} b_i (x_i - y_i) = 0\right]$$

$$= \Pr\Big[\underbrace{b_{i_0}}_{A} \underbrace{(x_{i_0} - y_{i_0})}_{\text{non zero}} + \underbrace{\sum_{i \neq i_0} b_i (x_i - y_i)}_{B} = 0\Big]$$

independent random variables

A is uniform $\Rightarrow$ by lemma, $A \cdot (x_{i_0} - y_{i_0})$ uniform

B, A independent $\Rightarrow$ $\underbrace{A+B}_{Z}$ uniform.

$\Pr[\underline{Z=0}] = \frac{1}{p}$ $\blacksquare$

uniform over $\mathbb{Z}_p$

Fact for any n, in $O(n)$ deterministic time

we can construct prime p, $n \le p \le n$

Theorem $S \subseteq \mathcal{U}$, $|S| = n$, $|\mathcal{U}| = N \le \text{poly}(n)$

one can in $O(n)$ deterministic time construct hash function family

$\mathcal{H} : \mathcal{U} \to T$ where

- $|T| = \le O(n)$
- choose $h \in \mathcal{H}$ takes $O(1)$ space to store
- evaluation takes $O(1)$ time
- $\forall z \in \mathcal{U}$, $\mathbb{E}[|L[h(z)]|] \le O(1)$

Pf: choose $p \in [n, 2n]$, pick k s.t. $p^k \ge |\mathcal{U}|$ $\Rightarrow k \le O(1)$