enter

SHREE
DATE: / /
PAGE NO.:

We want to show =

$$\Rightarrow \quad 1 - \frac{1}{2^k} + B_k \geq \frac{3}{2}$$

$$\Rightarrow \quad B_k \geq \frac{1}{2} + \frac{1}{2^k}$$

Hence, we need to prove =

$$1 - \left(1 - \frac{1}{k}\right)^k \geq \frac{1}{2} + \frac{1}{2^k}$$

and that's all

Other techniques for MAXSAT is semi-definite programming = 0.78 performance ratio

Hashing

Hash table = $|T|$
Hash function - $h(x)$

S is subset of universe U

$|S| > |T|$

$|U| >> |T|$

so map can never be one to one.

When x, y in U have $h(x) = h(y)$
↳ collision

Teacher's Signature

Resolving Collisions

① Chaining
└ Singly linked list

assume = simple uniform hashing — i.e. the hash function h is such that any element of U is equally likely to hash into any of the slots in T independent of other elements

uniform hashing → difficult

i.e.

$$Pr(h(u) = t) = \frac{1}{|T|}$$

where $0 \le t \le |T| - 1$

SIMPLE UNIFORM HASHING

$m = |T|$ = cells in table

n elements $= |S|$

$n_i \rightarrow$ no. of elements mapped to index i of table

$$n_i = |\{h(x) = i\}|$$

$\therefore n = \sum n_i$

Now, if we were to search for element in hash table

SHREE
DATE : / /
PAGE NO. :

→ So searching, inserting, deleting time will depend on length of LL.

→ What is the average value of n_i

$$E[n_i] = \sum E[n_{ij}]$$

where $n_{ij} = \begin{cases} 1 & \text{if } j^{th} \text{ element goes to } i^{th} \\ 0 & \text{o/w} \end{cases}$

$$E[n_i] = \sum Pr(n_{ij} = 1) \times 1$$

$$= \sum \frac{1}{|T|} \quad \leftarrow (\because \text{simple uniform hashing})$$

$$= \sum \frac{1}{m}$$

$$\boxed{E[n_i] = \frac{n}{m}}$$

and $\boxed{\frac{n}{m} = \text{load factor denoted } \alpha}$

→ Find time req. to search

successful → runs till element is found

unsuccess → run along entire list

Pr (traverse j elements before finding elem) $= O(1 + \frac{n}{m})$

$= O(1 + \frac{n}{m})$

Teacher's Signature

Universal Hashing

So, if S is chosen once h is fixed — you can always find a bad S — cl. adversial choice of S.

So, what if we choose h once S is given. Is this possible?

Lecture 11: 18th February 2021

Consider a family of hash functions

$$H = \{h_1, h_2 \dots h_r\}$$

So, given a S, we will choose one of these r func only —

So, we need to understand what kind of properties should H hold to solve collisions problem.

We define a universal family of hash functions where —

a family H is UNIVERSAL

for every pair of distinct keys from universe, k and l,

hash func with $h(k) = h(l)$ is at most $\frac{|H|}{m}$.

Teacher's Signature

So, we evaluate $h_1, h_2 \ldots$ for $k \& l$

	k	l
h_1	k_1	l_1
h_2	k_2	l_2
h_3	k_3	l_3
⋮		
h_r	k_r	l_r

and check how many $h(k) = h(l)$

$$\#\ \text{st. } h(k) = h(l) \leq \frac{|H|}{m}$$

Proof: This can be found —

Under uniform hash assumption =

$$Pr(h(k) = i) = \frac{1}{m}$$

$$Pr(h(k) = h(l)) = \frac{1}{m}$$

So if we choose a h UAR from H having 'r' h.

$$\text{So, } Pr(\text{any } h \ni h_i = h_j) \leq \frac{|H|}{m}$$

Here, we are basing our calculation on prob. space given by choice of Hash function.

So, If this ineq. holds, this is analogous to the uniform hash assumption

Teacher's Signature

So,

$Pr(h(k) = h(l)) = Pr(i_1 = i_2)$

where h is chosen UAR

↳ Prof choosing two like indices from T at random

Q → Do universal family of hash f^n exist?

Ans = YES.

let p be a large prime.
pick $p > m$. where $m = |T|$

The universe is $[0, p-1]$

let $Z_p = \{0, 1, \dots p-1\}$ = modulo of p

and

$Z_p^* = \{1, 2 \dots p-1\}$ (excluding 0 from Z_p)

Definition

→ Hash family H, will consist of all f^n of kind

$$h_{ab}(k) = ((ak+b) \bmod p) \bmod m$$

∵ hash value has to be $[0, m-1]$

where $k \in U = \{0, 1, \dots p-1\}$

And ∵ we dont want $a = 0$ so, a in Z_p^* and b in Z_p

Teacher's Signature

SHREE
DATE: / /
PAGE NO.:

⇒ There are $p(p-1)$ functions in this family H.

→ **Theorem:** The class H of function defined earlier is a universal class of hash funcn

Proof

Pick distinct k & l from U
then prob.
$h(k) = h(l)$ is at most $\frac{1}{m}$
for h chosen UAR from H.

$h_{ab}(k) = ((ak+b) \bmod p) \bmod m$
$= r \bmod m$

$h_{ab}(l) = ((al+b) \bmod p) \bmod m$
$= s \bmod m$

So, can $r = s$? (NO) ($r = (ak+b) \bmod p$; $s = (al+b) \bmod p$)

Why? $r - s = a(k-l) \bmod p$

∵ $k \neq l$ and $a \neq 0$
So $a(k-l) \not\equiv 0 \bmod p$

∵ p is greater than $k \& l$
so, $(k-l)$ cannot be a multiple of p

Teacher's Signature

⇒ $r \neq s$

⇒ if $r \neq s$, ⇒ for a given (a, b), we can find a distinct pair (r, s)

⇒ for every pair of r, s among the $p(p-1)$ possible pairs, it can be shown that (r, s) is mapped 1-1 to the pair (a, b).

⇒ a choice of a, b will give rise to a unique choice of r, s.

So we can pretend instead of picking a, b UAR we picked r, s from $p(p-1)$ possible pairs UAR.

Going back to our problem, what is the possibility that even tho $r \neq s$ but

$$h_{ab}(k) = h_{ab}(l)$$

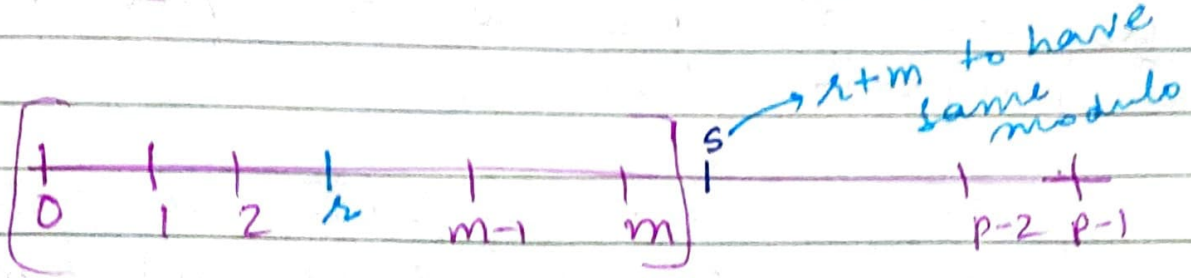
$$r \bmod m = s \bmod m \quad \text{—(1)}$$

(1) is possible if s is of the form

$$r + 1m, r + 2m, \ldots r + i \ldots$$

Teacher's Signature

$(r+m) \bmod m = r \bmod m$



To find how many time this is possible $h(k) = h(l)$.

we look at the size of set to which s belongs i.e.

$\{r+m, r+2m \text{ ---} \}$

AP ← we want to find a_n

$a_n = a + (n-1)d$

$p-1 = r+m + (n-1)m$

$\frac{p-1-r-m+m}{m} = n$

$n = \frac{p-r-1}{m} \leq \frac{p-1}{m}$

So, we can say that, bad values of s are $\leq \frac{p-1}{m}$

Total choices of s are $p-1$ or p

So $B \leq \frac{1}{m}$

Hence proved.

Teacher's Signature

Q How can we use this family to improve hashing.

Standard hashing based solutions suggest that we should spend $O(a)$ time for each search.

Can we design $O(1)$ worst case solution?

Let us look at a lemma:

Lemma: Let n keys — store in hash table — using h fn drawn UAR from universal family of m fns. The expected # collision is ${}^{n}C_2 \frac{1}{m}$

Proof: P for every pair of keys — $k \neq l$

$$X_{kl} = \begin{cases} 1 & \text{if } h(k) = h(l) \\ 0 & \text{o/w} \end{cases}$$

$$E[X_{kl}] = P_r[X_{kl} = 1]$$
$$= P_r(h(k) = h(l))$$
$$= \frac{1}{m} \text{ (proved in last proof)}$$

$X =$ # collisions

SHREE
DATE: / /
PAGE NO.:

$$X = \sum_{\substack{k,l \\ s.t.\ k \neq l}} X_{kl}$$

$$E[X] = \sum_{\substack{k,l \\ s.t.\ k \neq l}} E[X_{k,l}]$$

$$E[X] = {}^{n}C_2 \times \frac{1}{m}$$

(choosing k & l)

HP

Ideally ~~This means~~, $|T| = O(n)$ where n is no. of keys so no waste of space.

And usually $m = O(n)$

$\Rightarrow E[X] = \Theta(n)$

as $E[X] = {}^{n}C_2 \times \frac{1}{m}$

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

$(\because m = O(n))$

$$= \frac{n(n-1)}{2} \times \frac{1}{n}$$

$$= \Theta(n)$$

$\Rightarrow$ lot of collisions if $|T| = O(n)$

Teacher's Signature

→ But if $|T| = m = n^2$.
that is very large table.

$$E[X] = {}^{n}C_2 \times \frac{1}{m}$$

$$= \frac{n(n-1)}{2} \times \frac{1}{n^2}$$

$$\leq \frac{1}{2}$$

So, to get few collisions $|T|$ should be quadratic in the no. of elements to be hashed. But this is wasteful and an overkill in terms of space.

Lemma: Hash n keys — h fn UAR —
let n_i refer to no. of collisions at index i.

$$E[\Sigma_i \, n_i^2] < 2n$$

Proof:
Approach 1 -

$$= E[\Sigma_i \, n_i^2]$$

$$= \sum_i E[n_i^2] \qquad [n \geq n_i \geq 0]$$

$$= \sum_i \sum_{j=0}^{n} j^2 \, [Pr(n_i = j)] \quad (Pr \text{ that } j \text{ things go to cell } i)$$

$$Pr(n_i = j) = {}^{n}C_j \left(\frac{1}{m}\right)^j \left(\frac{m-1}{m}\right)^{n-j}$$

$\left(\frac{1}{m}\right)^j$: Prob of j things going to cell i

$\left(\frac{m-1}{m}\right)^{n-j}$: prob of rest of things

$$\Rightarrow E\left[\sum_i n_i^2\right] = \sum_i \sum_{j=0}^{n} j^2 \, Pr(n_i = j)$$

$$= \sum_i \sum_{j=0}^{n} j^2 \cdot {}^{n}C_j \left(\frac{1}{m}\right)^j \cdot \left(\frac{m-1}{m}\right)^{n-j}$$

$\because$ all n_i are identical

$$= n \sum j^2 \quad n \quad n \quad n$$

$$\vDash O(n)$$

another approach

$E\left[\sum_i n_i^2\right] \leq 2n$ ← TRICK

for any non-negative integer x, $x^2 = x + 2 \cdot {}^{x}C_2$

$$E\left[\sum_i n_i^2\right] = E\left[\sum_i \left(n_i + 2 \cdot {}^{n_i}C_2\right)\right]$$

$\because$ each n_i is non negative

$$= E\left[\sum_i n_i\right] + 2\, E\left[\sum_i {}^{n_i}C_2\right]$$

even though each n_i is a RV; we're only hashing n elements

So, $\sum_i n_i = n$

$$= E[n] + 2\, E\left[\sum_i {}^{n_i}C_2\right]$$

$$= n + 2E\left[\sum_i {}^{n_i}C_2\right] \quad \text{— (i)}$$

now, ~~let's look at this~~

Teacher's Signature

SHREE
DATE: / /
PAGE NO.:

$n_i C_2$ is the pairs of elements that collide at index i

$\sum_i {}^{n_i}C_2$ = no. of colliding pairs in entire table
= no. of collisions

So, $E[\sum_i {}^{n_i}C_2]$ is just the expected no. of collisions.

Which means:

$$E[\sum_i {}^{n_i}C_2] = E[\text{no. of collisions}]$$
$$= {}^{n}C_2 \times \frac{1}{m}$$

(here $m = n$)

$$= \frac{n(n-1)}{2} \times \frac{1}{n}$$
$$= \frac{n-1}{2} \quad \text{— (ii)}$$

Subst (ii) in (i):

$$E[\sum_i n_i^2] = n + 2 \times \frac{(n-1)}{2}$$
$$= n + n - 1$$
$$= 2n - 1$$
$$< 2n$$

Teacher's Signature

Date
Page

Lecture 12: 16th February 2021

We proved lemma 2
using which we get the idea that

~~total # collisions at index i is n_i^2~~
~~of~~ that to avoid collision at index i we must increase size of table by n_i^2 where n_i is # collisions at index i.

So, if we look at the two lemmas

① $E[\#\text{collisions}] = \binom{n}{2}\frac{1}{m}$

② $E\left[\sum_{i=0}^{n-1} n_i^2\right] \leq 2n$ [here $m = n$]

So, no matter what h is used, when $m = n$, we expect collisions but we can get rid of these collisions. How?

Acc. to lemma 1 —

$$|T| = n^2$$

So for each set of elements that are colliding we use another hash fn with $|T_i| = n_i^2$

~~So, total space used~~
and we know by lemma 2 that $E[\Sigma n_i^2] \leq 2n$

So total space used is $n + 2n = 3n$

(n: main hash table; $2n$: extra hash tables for collisions)

So,

1. Pick $m = n$
2. for each set of colliding keys
   - hash them to a bigger table of size n_i^2 individually.

Total Space $\leq 3n$

Example :-

1. $a = 13$, $b = 8$
   $m = 11$, $p = 53$

   find collisions

2. At each i
   $m = n_i^2$; same p
   pick a new hash function
   k

→ **Space : $O(n)$**

→ What is the time per query?

Time : $O(1)$

But, what if there are collisions in secondary hash table as well.

we know, expected no. of collision = $\frac{1}{2}$

using Markov,

P(at least one collision for a hash fn chosen UAR)

$= P(\text{collision} \geq 1)$

$= P(\text{collision} \geq \mu \times c)$

$= P(\text{collision} \geq \frac{1}{2} \times c)$

$\Rightarrow c = 2$

$\Rightarrow P(\text{collision} \geq 1\mu) \leq \frac{1}{2}$ (at most $\frac{1}{2}$)

Hence,

dont try one more level of hashing but ~~the~~ make another choice for h fn (try a couple of times till a 'good' hash fn is chosen).

└ why couple of times

because prob. of bad hash fun is $\frac{1}{2}$ so, there is a good chance in one of two/three times you'll get a bad/good hash function.

~~as, Prob(collision ≥ 1) $\geq \frac{1}{2}$~~

~~Prob(collision ≤ 1) $\leq \frac{1}{2}$~~

as, Prob(coll ≥ 1) $\leq \frac{1}{2}$

$\Rightarrow$ Prob (zero collision) $> \frac{1}{2}$ } So couple of times means

in 2 or less tries you'll get good hash function.

QUIZ 1 DISCUSSION (SKIPPED)

Lecture 13: 19th February 2020

Parallel Computing

Computer Architecture

- increasing clock frequency but memory performance does not scale up then processors sits idle (Memory Wall) not able to
  - if ↑se clock frequency, ↑ catch up with instruction level parallelism (ILP Wall)
    - dissipated power will become much more significant (Power Wall)

MW + PW + ILP = Brick Wall

The PRAM Model (Parallel RAM)

① → RAM Model or von Neumann Model