ecture 8:

Last time: A set $\mathcal{H}$ of hash functions $\{0,1,\ldots,\mathcal{U}-1\} \to \{0,1,\ldots,n-1$

is <u>universal</u> if, $\forall$ distinct $x,y < \mathcal{U}$,

$$\Pr_{h\in\mathcal{H}}\left[h(x) \neq h(y)\right] \le \frac{1}{n}.$$

[← always $= \frac{1}{n}$ if $\mathcal{H}$ is "all functions", as in SUHA]

Comment: "c-almost-universal" has "$\le \frac{c}{n}$" in place of "$\le \frac{1}{n}$".
If $c = 2$, e.g., morally just as good [and often easier to achieve]

[We saw an example last time showing that universal hash families can be as good as SUHA for at least <u>some</u> nice hashing apps. We'll see more today, but first, let's construct them...]

<u>example 1</u>: [Not the most efficient, but simple proof.]

Assume $\mathcal{U} = 2^u$ (e.g.: $u = 64, 128$)

$n = 2^\ell$ (e.g.: $\ell = 16$ (table of size 65536))

Suppose $A = \ell \updownarrow \underbrace{\begin{bmatrix} 0 & 0 & 1 & \cdots \\ 0 & 1 & 1 & \cdots \\ & & & \vdots \\ & & & 1 \end{bmatrix}}_{\longleftarrow\ u\ \longrightarrow}$ is a matrix of bits. [couple thousand bits]

Define $h_A(\overset{\text{u-bit string}}{x}) = \begin{bmatrix} & A & \end{bmatrix}\begin{bmatrix} \\ x \\ \\ \end{bmatrix} = \underset{\uparrow\ \ell\text{-bit string} \,\equiv\, \text{hash slot in } 0\ldots n-1}{\begin{bmatrix} \\ \end{bmatrix}}$ mod 2 (in each entry)

i^{th} bit ~~entry~~ of $h_A(x)$ is $a^{(i)} \cdot x \bmod 2$ [← computable in a few machine instructions, esp. if $a^{(i)}$ fits in a word]

$\uparrow$ i^{th} row of A.

$\mathcal{H} = \{h_A : A \in \{0,1\}^{\ell \times u}\}$. Picking $h \in \mathcal{H}$ $\equiv$ picking A. [$h_A(x)$ computable in $O(\ell)$]

Claim: This is universal.

Proof: Fix any $x, y \in \{0,1\}^\ell$, $x \neq y$. Need to show $\Pr_{h \in \mathcal{H}}[h(x) = h(y)] \leq \frac{1}{n}$

$$\iff \Pr_A [Ax = Ay \mod 2] \leq \frac{1}{n}$$

$$\iff \Pr_A \left[A(x-y) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mod 2\right] \leq \frac{1}{n}.$$

$$\iff \Pr_A \left[Az = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \mod 2\right] \leq \left(\frac{1}{2}\right)^\ell$$

where $z = x - y \mod 2$

$\neq \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \because x \neq y$.

$\Rightarrow$ z has at least one 1, say z_i.

$$\underbrace{\begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_u \\ | & | & & | \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} * \\ * \\ \vdots \\ * \\ \overset{z_i \nearrow}{1} \\ * \\ \vdots \\ * \end{bmatrix}}_{z} \overset{?}{=} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mod 2.$$

$$Az = z_1 \begin{bmatrix} a_1 \end{bmatrix} + z_2 \begin{bmatrix} a_2 \end{bmatrix} + \cdots + \underset{\nearrow z_i}{1} \begin{bmatrix} a_i \end{bmatrix} + \cdots + z_u \begin{bmatrix} a_u \end{bmatrix}$$

where $\begin{bmatrix} a_j \end{bmatrix}$'s all random.

Imagine all a_j's except a_i picked first, then a_i picked.

$$Az = \underbrace{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\text{some length-}\ell\text{ string}} + \begin{bmatrix} | \\ a_i \\ | \end{bmatrix} \overset{?}{=} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mod 2.$$

Probability ⟦over remaining choice of a_i⟧ $= \left(\frac{1}{2}\right)^\ell$ exactly

□

Pros: Easy proof!

Con 1: Storing A: ℓu bits. ⟦Okay, but $2u$ is possible⟧

Con 2: Evaluating $h_A(x) = \Theta(\ell)$ word operations ⟦$O(1)$ is possible⟧

Con 3: $h_A(0) = 0$ always!

⟦Kinda weird! "Each 1 item goes to rand. place" fails.⟧

Cheap fix for Con 3: Also pick random $b \in \{0,1\}^{\ell}$,

let $h_{A,b}(x) = Ax + b \bmod 2$

Still universal ✓ [[$h_{A,b}(x) = h_{A,b}(y) \iff Ax+b = Ay+b \bmod 2$

$\iff Ax = Ay \bmod 2$,

so same proof.]]

"Uniformity": For all $x < \mathcal{U}$, for all $j < n$, $\Pr_{h = h_{A,b}}[h(x) = j] = \frac{1}{n}$.

Because: $h_{A,b}(x) = j \iff \begin{bmatrix} & A & \end{bmatrix}\begin{bmatrix} \\ x \\ \\ \end{bmatrix} + \begin{bmatrix} b \end{bmatrix} = \begin{bmatrix} j \end{bmatrix} \bmod 2$

Pick A first, b last. $\Rightarrow \underset{Ax \uparrow}{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}} + \begin{bmatrix} \\ b \\ \\ \end{bmatrix} = \begin{bmatrix} 0 \\ j \\ \vdots \\ 0 \end{bmatrix} \bmod 2$

$\nwarrow j$

Prob is $(\frac{1}{2})^{\ell}$ again $= \frac{1}{n}$. ✓

~

Fix for Cons 1 & 2:

Carter & Wegman's method:

- ~~Pick~~ Fix any prime number p between $\mathcal{U}$ & $2\mathcal{U}$ [[takes a bit of effort, but it's a one-time cost.]]

  [[doesn't have to be random]]

To choose h: choose $a \in \{1, 2, \dots, p-1\}$ // $\approx u$ bits

$b \in \{0, 1, \dots, p-1\}$ // u bits

at random.

$$h_{a,b}(x) := ((a \cdot x + b) \bmod p) \bmod n.$$

[[Note: can't skip the mod n part! exercise]]

exercise: This is universal.

[kinda annoying, but elementary]]

Pro 1: Storing a, b: $\approx 2u$ bits

Pro 2: Eval. $h_{a,b}(x)$: $O(1)$ (4?) machine ops.

More options:

- Pick odd $a \in \{1, 3, 5, 7, \dots, n-1\}$ at random
- $h_a(x) := \underbrace{(a \cdot x \bmod \mathcal{U})}_{\text{drop all but } u \text{ LSBs}} \ \underbrace{\text{div } \frac{\mathcal{U}}{n}}_{\text{right-shift by } u-\ell \text{ bits}}$

Pro 1: Storing a: ℓ bits ☺

Pro 2: Eval $h_a(x)$: $O(1)$ ops (3?)

Con : Not universal. But is "2-almost-universal." Proof: exercise [not too hard]

* * *

Universal + Uniform: For any two items $x \neq y$, the 2 bins $h(x), h(y)$ are a completely random pair. [each possib. occurs with prob $1/n^2$]

Higher analogue: "3-wise independent hash family"

For any distinct 3 items x_1, x_2, x_3, $(h(x_1), h(x_2), h(x_3))$ is each possibility with prob. $\frac{1}{n^3}$.

Fact:
- ∃ efficient-to-eval k-wise indep. hash families where you store $O(k \cdot u)$ bits. [but a bit elaborate to explain]
- 5-wise indep. is good enough for many SUHA-achievable properties

* * *

One more example where universal (or 2-almost-universal) is "good enough": "Perfect Hashing".

Goal: Worst-case $O(1)$ lookups, space $O(m)$...

⟦sounds awesome, better than w/ SUHA. Catch is...⟧

... for a static hash table:

All m items $x_1 \dots x_m$ given in advance, no insert/dels, just lookups.

⟦Could build an optimal BST or similar, but hashing will suffice⟧

Say we use a universal family with $n=m$. ⟦Optimistic; last time we considered $n \gg m^2$⟧

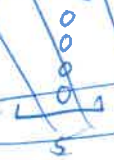
Recall: After hashing, if $C_{ij} = \begin{cases} 1 & \text{if } h(x_i)=h(x_j) \\ 0 & \neq \end{cases}$,

$C = \sum_{1 \le i < j \le n} C_{ij}$ = # colliding pairs,

$\mathbb{E}[C_{ij}] \le \frac{1}{n}$ (universality), $\mathbb{E}[C] \le \binom{m}{2}\frac{1}{n}$ ⟦lin. of expec.⟧

$\le \frac{m^2}{2n} = \frac{m}{2}$ (for $n=m$)

~~Q: If s^{th} bin has load L_s, what will C be at least?~~



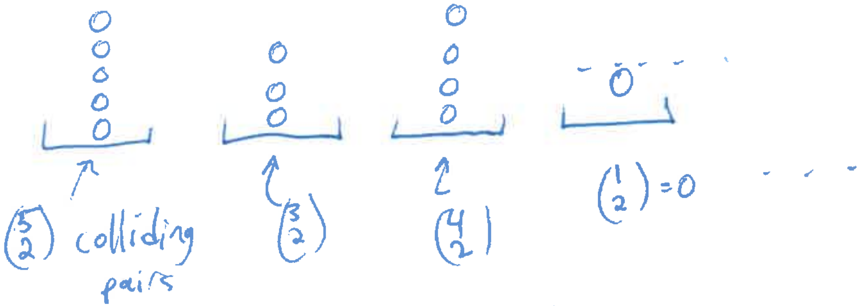
~~$L_s = 4 \to$ contributes $\binom{4}{2}$ to C. $C = \sum_{s=0}^{n-1} \binom{L_s}{2}$~~

Markov: $\Pr[C \ge 20 \cdot \frac{m}{2}] \le \frac{1}{20}$; $\boxed{\therefore C \le 10m \text{ except w. prob} \le 5\%}$ ⊛

⟦Is that... good? Bad? Last time we said $C=0 \Rightarrow$ no collisions

What could $C \le 10m$ imply?⟧

Say we get...



What's C?

$\therefore C = \sum_{i=0}^{n-1} \binom{L_i}{2}$, L_i := load of i^{th} bin.

Note: $\binom{L_i}{2} = \frac{L_i(L_i-1)}{2} \approx \frac{L_i^2}{2}$. $\therefore$ If some $L_i > 5\sqrt{m}$, then $L_i^2 \gtrsim \frac{25m}{2} \gg 10m$

$\Rightarrow C \gg 10m$, contrary to ⊛

$\therefore$ except with prob $\le 5\%$, Max load $\le 5\sqrt{m}$.

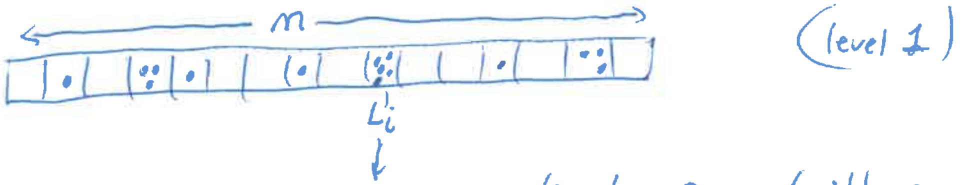
⟦Not totally awesome. For $m=n$, w.h.p. the max load is $O\left(\frac{\log m}{\log\log m}\right)$ under SUHA. STM is way bigger⟧

⟦But...⟧ Moreover, except with prob ≤5%,

$$10m \geq C = \sum_{i=0}^{n-1} \binom{L_i}{2} \geq \sum_{i:L_i\geq 2} \frac{L_i^2}{4}, \quad \because \binom{L_i}{2} \geq \frac{L_i^2}{4} \text{ when } L_i \geq 2$$

$$\Rightarrow \sum_{i:L_i\geq 2} L_i^2 \leq 40m. \quad (**)$$

Idea: Hash with universal family, compute L_i's. Check that (**) holds. ⟦If not, pick a new h and try again. But happens 95% of time⟧



Now for each slot i with $L_i \geq 2$, build a second-level hash table of size $10L_i^2$ with a new random hash function h_i from a universal family

Last time we saw: for universal hashing, if L_i items → $10L_i^2$ slots, then ≥95% of the time, max load is 1.

So do this for each level-1 slot with load ≥1. ⟦Again, after hashing the L_i items, check load is ≤1. If not, rehash, but ≤5% chance.⟧

Space: m (level 1)

$$+ \sum_{i:L_i\geq 2} 10L_i^2 \leq 400m, \text{ by } (**)$$

∴ total space: $O(m)$ ☺

worst-case lookup: 2 hashes.

(Slight bummer: needs up to m different hash functions → $m \cdot O(u)$ bits to store these.)