Last time: A set $\mathcal{H}$ of hash functions $\{0, 1, \ldots, U-1\} \rightarrow \{0, 1, \ldots, n\}$ is universal if, for distinct $x, y \in \mathcal{U}$,

$$\Pr[h(x) \neq h(y)] \leq \frac{1}{n}.$$ 

If $h \in \mathcal{H}$, $\mathcal{H}$ is "all functions", as in SUHA 7.

Comment: "$c$-almost-universal" has $\leq \frac{c}{n}$ in place of $\leq \frac{1}{n}$.

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 some nice hashing apps. We'll see more today, but first, let's construct them."

**Example 1:** [Not the most efficient, but simple proof.]

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

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

**Solution:** Suppose $A = \begin{bmatrix} 0 & \cdots & 0 \\ 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$ is a matrix of $b$ bits.

Define $h_A(x) = \begin{bmatrix} A \\ x \end{bmatrix} \mod 2$ (in each entry)

$l$-bit string $\mapsto$ hash slot in $0, \ldots, n-1$

Ex: $i^{th}$ bit of $h_A(x)$ is $a[i] \cdot x \mod 2$ (in i'th row of $A$).

$\mathcal{H} = \{ h_A : A \in \{0, 1\}^{l \times u} \}$; Picking $h \in \mathcal{H}$ = picking $A$. $h_A(x)$ computable in $O(1)$.
Claim: This is universal.
Proof: Fix any $x, y \in \{0,1\}^l$. Need to show $\Pr_{h \in \mathcal{H}} \left[ h(x) = h(y) \right] \leq \frac{1}{n}$

$$\iff \Pr_{A} \left[ Ax = Ay \mod 2 \right] \leq \frac{1}{n}$$

$$\iff \Pr_{A} \left[ A (x-y) = \left[ \begin{smallmatrix} \alpha \end{smallmatrix} \right] \mod 2 \right] \leq \frac{1}{n}.$$ 

$$\iff \Pr_{A} \left[ A z = \left[ \begin{smallmatrix} \alpha \end{smallmatrix} \right] \mod 2 \right] \leq \left( \frac{1}{2} \right)^l$$

where $z = x-y \mod 2$

$$\neq \left[ \begin{smallmatrix} \theta \end{smallmatrix} \right] \iff x \neq y.$$ 

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

$$\left[ \begin{smallmatrix} a_1 & a_2 & \ldots & a_n \\ 1 & 1 & \ldots & 1 \end{smallmatrix} \right] \left[ \begin{smallmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{smallmatrix} \right] \equiv \left[ \begin{smallmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{smallmatrix} \right] \mod 2.$$ 

$A z = z_i [a_i] + z_i [a_i] + \ldots + 1 [a_i] + \ldots + z_i [a_i]$

where $a_i$'s all random.

Imagine all $a_i$ except $a_i$ picked first, then $a_i$ picked.

$$A z = \left[ \begin{smallmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{smallmatrix} \right] + A [a_i] \equiv \left[ \begin{smallmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{smallmatrix} \right] \mod 2.$$

Probability: $\left( \frac{1}{2} \right)^l$ exactly over remaining choice of $a_i$. 

Pros: Easy proof!

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

Con 2: Evaluating $h_A(x) = O(l)$ word operations
[O(1) is possible]

Con 3: $h_A(0) = 0$ always! 
[Kinda weird! "Each 1 item goes to rand, place" fails.]

$\square$
Cheap fix for Case 3: Also pick random $b \in \{0,1\}^k$,
let $h_{A,b}(x) = Ax + b \mod 2$

Still universal $\begin{align*}
[h_{A,b}(x) & = h_{A,b}(y) \iff Ax + b = Ay + b \mod 2 \\
& \iff Ax = Ay \mod 2],
\end{align*}$

so same proof. $\Box$

"Uniformity": For all $x \in \mathcal{U}$, for all $j < n$, $Pr[h(x) = j] = \frac{1}{n}$.
Because: $h_{A,b}(x) = j \iff \begin{bmatrix} A & b \end{bmatrix} x + b = j \mod 2$

Pick $A$ first, $b$ last. $\Rightarrow \begin{bmatrix} b \end{bmatrix} + b = \begin{bmatrix} j \end{bmatrix} \mod 2$

Prob is $(\frac{1}{n})$ again $\frac{1}{n}$. $\Box$

Fix for Cases 1 & 2:
Carter & Wegman's method:
- Pick any prime number $p$ between $\mathcal{U}$ & $2\mathcal{U}$
- Pick $p$ doesn't have to be random $\Box$

To choose $h$:
choose $a \in \mathbb{Z}_{p^k}$ // $\approx n$ bits
choose $b \in \mathbb{Z}_{p}$ // $\approx n$ bits

at random.

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

Note: can't skip the mod $p$ part! exercise

This is universal.

Exercise: This is universal.

Pro 1: Storing $a,b$ : $\approx 2n$ bits
Pro 2: Eval $h_{a,b}(x)$ : $O(1)$ (4?) machine ops.
More options:

- Pick odd \( a \in \{1, 3, 5, 7, \ldots, n-1\} \) at random
- \( h_a(x) := \left( a \cdot x \mod 2^n \right) \div 2^{n-1} \)

Pro 1: Storing \( x \) as \( l \) bits:

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

Con: Not universal. But is "2-almost-universal." Proof: exercise

Universal + Uniform: For any two items \( x \neq y \),

the 2 bins \( h(x), h(y) \) are a completely random pair.

Higher analogue: "3-wise independent hash family"

For any \( \binom{3}{1} \) 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: 3 efficient-to-eval \( k \)-wise indep hash families

where you store \( O(k \cdot n) \) 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/ SHTA. Catch is...?
... for a static hash table:
All $m$ items given in advance, no insert/dels.
If 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 $C_{ij} := \{1 \text{ if } h(x_i) = h(x_j)\}$, $C = \sum_{1 \leq i < j \leq n} C_{ij} = \# \text{ colliding pairs},$
$\mathbb{E}[C_{ij}] = \frac{1}{n} \quad \text{(universality)}, \quad \mathbb{E}[C] \leq \frac{(m)^{1/2}}{n} \quad \text{[in. of exp.]}
\leq \frac{m^2}{2^n} = \frac{m}{2} \quad \text{(for } n = m\text{)}$

Q: If $\binom{n}{2}$ bins have load $4$s, what will $C$ be at least?

Markov: $\Pr[C \geq 20 \cdot \frac{m}{2}] \leq \frac{1}{20} \quad ; \quad C \leq 10m \text{ except w. prob } \leq 5\%$

[Is that good? Bad? Last time we said $C = 0 \Rightarrow$ no collisions]
What could $C \leq 10m$ imply?]
Say we get...

\[
\begin{array}{ccc}
\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{array} & \begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{array} & \begin{array}{c}
0 \\
\end{array}
\end{array}
\]

What's $C$? \( \binom{5}{2} \) colliding pairs \( \binom{4}{2} \) \( \binom{3}{2} \) \( \binom{2}{2} \) \( \binom{1}{2} = 0 \)

\[
C = \sum_{i=0}^{n-1} \binom{\text{Li}}{2} \quad ; \quad \text{Li := load of } i^{th} \text{ bin.}
\]

Note: \( \binom{\text{Li}}{2} = \frac{\text{Li} \cdot (\text{Li} - 1)}{2} \) \( \Rightarrow \) If some $\text{Li} > 5\sqrt{m}$, then $\text{Li}^2 > \frac{25m}{2} \Rightarrow 10m$

\[
\Rightarrow C > 10m, \text{ contrary to previous, except with prob } \leq 5\% \quad \text{max load } \leq 5\sqrt{m}
\]
Not totally awesome. For \( m = n \), w.h.p. the max load is \( O(\log m / \log \log m) \) under SUHA, STM is way bigger.

But... except with prob \( \leq 5\% \),

\[
\sum_{i=0}^{m} \left( \frac{L_i}{2} \right) \geq \sum_{i:L_i \geq 2} \frac{L_i^2}{4} \quad \therefore \quad \frac{L_i^2}{4} \geq \frac{L_i^2}{4} \text{ when } L_i \geq 2
\]

\[
\sum_{i:L_i \geq 2} L_i^2 \leq 40m. * *
\]

**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 the time.

```
* \( m \) slots
```

```
" WARNING "
```

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 \( \rightarrow 10L_i^2 \) slots, then \( \geq 95\% \) of the time, max load is 1.

So do this for each level-1 slot with load \( \geq 1 \).

Again, after hashing the \( L_i \) items, check load is \( \leq 1 \). If not, rehash, but \( \leq 5\% \) 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

\( \rightarrow m \log(\log m) \) bits to store these.)