ALG 4.2
Universal Hash Functions:

Hash Function

\[ f : A \rightarrow B \]

Keys indices

f has conflict at \( x, y \in A \) if \( x \neq y \) but \( f(x) = f(y) \)

\[ \sigma_f(x, y) = \begin{cases} 
1 & \text{if } x \neq y \text{ and } f(x) = f(y) \\
0 & \text{else}
\end{cases} \]
If $H$ is a set of hash functions,

$$\sigma_H(x, y) = \sum_{f \in H} \sigma_f(x, y)$$

for set of keys $S$,

$$\sigma_H(x, S) = \sum_{f \in H} \sum_{y \in S} \sigma_f(x, y)$$
H is a *universal* set of hash functions

if $\sigma_H(x, y) \leq \frac{|H|}{|B|}$ for all $x, y \in A$

i.e. no pair of keys $x, y$ are mapped into the same index by $> \frac{1}{|B|}$ of all functions in $H$

\[
\sigma_H(x, y) > |H| \left( \frac{1}{|B|} - \frac{1}{|A|} \right)
\]

**proof**

let $a = |A|, b = |B|$

By counting, we can show

\[
\sigma_f(A, A) \geq b \left( \frac{a}{b} - 1 \right)^2 \geq \frac{a^2}{b} - a
\]
Thus
\[ \sigma_H(A, A) \geq a^2 |H| \left( \frac{1}{b} - \frac{1}{a} \right) \]

By the pigeon hole principle
\[ \exists \ x, y \in A \ \text{s.t.} \ \sigma_H(x, y) \geq |H| \left( \frac{1}{b} - \frac{1}{a} \right) \]

*note*

in most applications, \(|A| >> |B|\)
and then any universal class has asymptotically a minimum number of conflicts

**Proposition 2:**

Let \( x \in A, S \subseteq A \)

For \( f \) chosen randomly from a universal class \( H \) of hash functions, the expected number of collisions is

\[ \sigma_f(x, S) \leq \frac{|S|}{|B|} \]

**proof**

\[
E(\sigma_f(x, S)) = \frac{1}{|H|} \sum_{f \in H} \sigma_f(x, S)
\]

\[ = \frac{1}{|H|} \sum_{y \in S} \sigma_H(x, y) \text{ by definition} \]

\[ \leq \frac{1}{|H|} \sum_{y \in S} \frac{|H|}{|B|} \text{ by definition of universal} \]

\[ = \frac{|S|}{|B|} \]
**application**

*associative memory storage* of \(|S|\) keys onto \(|B|\) linked lists.

Given key \(x \in A\), store \(x\) in list \(f(x)\)

Proposition 2 implies each list has expected

\[
\text{length} \leq \frac{|S|}{|B|} = O(1) \text{ if } |B| \geq |S|
\]

Gives \(0(1)\) time for STORE, RETRIEVE, and DELETE operations

**Proposition 3**

Let \(R\) be a sequence of requests with \(k\) insertion operations into an associative memory.

If \(f\) is chosen at random from set of universal 2 class \(H\), the expected

*total cost of all \(k\) searches is*

\[
\leq |R| \left(1 + \frac{k}{|B|}\right).
\]

**proof**

There are \(|R|\) total search ops, and each takes by Proposition 2 expected

\[
\text{time} \leq 1 + \frac{k}{|B|}.
\]

**note**

if \(|B| \geq k\), then expected total

\[
\text{time is } O(|R|).
\]
Bounds on distribution of $\sigma_f(x,S)$

**Proposition 4**

Let $x \in \mathbb{A}$, $S \subset \mathbb{A}$

Let $\mu =$ expected value of $\sigma_f(x,S)$

For $f$ chosen randomly from universal set of functions $H$,

$$\text{Prob} (\sigma_f(x,S) > t \cdot \mu) < \frac{1}{t}$$

**proof**

immediate from Markov bound

**improved** bounds on probability:

$$\text{prob} \leq \frac{11}{t^4} \text{ for universal hash funs. } H_2, H_3$$

(using 2nd and 4th moments of prob. distribution.)

$H =$ universal set of hash functions.

$E_1 =$ Expected cost of *random* set of $k$ requests using a *worst case* function $f$ in $H$

(*random input*)

$E_2 =$ Expected cost of *worst case* set of $k$ requests using a *random* function $f$ in $H$

(*randomized algorithm*)
Prop 5 \( E_1 \geq (1 - \varepsilon) E_2 \) where \( \varepsilon = \frac{|B|}{|A|} \)

**proof**

Let \( a = |A|, b = |B| \).

Prop 2 implies \( E_2 \leq 1 + \frac{|S|}{b} \).

Suppose \( S \) is chosen randomly. for \( x, y \in S \),

\[
E(\sigma_f(x, y)) = \frac{1}{a} \sigma_f(A, A)
\]

\[
\geq \frac{1}{2} a^2 \left( \frac{1}{b} - \frac{1}{a} \right)
\]

by Prop 1

\[
\geq \left( \frac{1}{b} - \frac{1}{a} \right)
\]

So \( E_1 \geq 1 + E(\sigma_f(x, S)) \)

\[
\geq 1 + |S| \left( \frac{1}{b} - \frac{1}{a} \right)
\]

---

**Example of Universal 2 Class**

Set of Keys Table

Let \( A = \{0, 1, \ldots, a-1\} \) Set of Keys

\( B = \{0, 1, \ldots, b-1\} \) Table

Let \( p \) be a prime \( \geq a \)

\( Z_p = \{0, 1, \ldots, p-1\} = \text{number field mod } p \)

**define** \( g : Z_p \to B \) s.t.

\( g(x) = x \mod b \)

**define** for \( n, m \in Z_p \) with \( m \neq 0 \),

\( h_{n,m} : A \to Z_p \)

with \( h_{n,m}(x) = (mx+n) \mod p \)

**define** \( f_{n,m} : A \to B \) s.t.

\( f_{n,m}(x) = g(h_{n,m}(x)) \)

\( H_1 = \{ f_{m,n} \mid m, n \in Z_p, m \neq 0 \} \)

**Claim:** \( H_1 \) is universal 2
**Lemma**

for distinct $x, y \in A$,

$$\sigma_H(x, y) = \sigma_g(Z_p, Z_p)$$

**proof**

$$\sigma_g(Z_p, Z_p) = |\{(r,s) \mid r,s \in Z_p, r \neq s, g(r) = g(s)\}|$$

Observe that the linear equations:

$$xm \equiv r \pmod{p}$$
$$ym \equiv s \pmod{p}$$

have *unique* solutions in $Z_p$

So $(r,s) = (h_{m,n}(x), h_{m,n}(y))$ then

$$(f_{m,n}(x) = f_{m,n}(y) \text{ if and only if } g(r) = g(s))$$

$\sigma_H(x,y)$ is the number of such pairs in

$(r,s) \in \sigma_g(Z_p, Z_p)$

**Theorem**

$H_1$ is universal

**proof**

Let $n_i = |\{t \in Z_p \mid g(t) = i\}| $

By definition of $g(x) = x \pmod{b}$,

$$\Rightarrow n_i \leq \frac{p-1}{b} + 1$$

For any given $r$, the number of $s$ where $s \neq r$ and $g(r) = g(s)$ is

$$\sigma_g(r, Z_p) \leq \frac{p-1}{b}.$$ 

But there are $p$ choices of $r$,

so $p \cdot \left(\frac{(p-1)}{b}\right) \geq \sigma_g(Z_p, Z_p)$

But $H_1(x,y)$ by Lemma

(Also note $\sigma_{H_1}(x,x) = 0$)

Hence $\sigma_{H_1}(x,y) \leq \frac{|H_1|}{b}$ since $|H_1| = p(p-1)$

so $H_1$ is universal.
Universal Hash Fns on Long keys

Given class of hash functions $H$, define hash functions $J = \{ h_{f,g} \mid f, g \in H \}$ where $h_{f,g}(x_1, x_2) = f(x_1) \oplus g(x_2)$

exclusive or

**Theorem** Suppose $B = \{0, 1, \ldots, b = 1\}$ where $b$ is a power of 2. Suppose this class of fns $A \rightarrow B$

$\exists$ real $r \forall i \in B \forall x_1, y_1 \in A$, $x_1 \neq y_1$

$\Rightarrow |\{ f \in H \mid f(x_1) \oplus f(y_1) = i \}| \leq r|H|$

Then $\forall x, y \in (A \times A)$, $x \neq y$

$|\{ h \in J \mid h(x) \oplus h(y) = i \}| \leq r|H|$

**Proof** for $x = (x_1, x_2)$, $y = (y_1, y_2)$ in $A \times A$

$i \in B$ then $|\{ h \in J \mid h(x) \oplus h(y) = i \}|$

$= |\{ f, g \in H \mid f(x_1) \oplus g(x_2) \oplus f(y_1) \oplus g(y_2) = i \}|$

$= \sum_{y \in H} |\{ f \in H \mid f(x_1) \oplus f(y_1) = i \oplus g(x_2) \oplus g(y_2) \}|$

$\leq |\{ f \in H \mid f(x_1) \oplus f(y_1) = i \}| \leq r|H|$

**example** $H_1$ with $m = 0$ gives $J$ with $r = \frac{1}{|B|}$ universal!
Universal Hashing \textit{without Multiplication}

\[ A = \text{set of } d \text{ digit numbers base } \alpha \text{ so, } |A| = \alpha^d \]

\[ B = \text{set of binary numbers length } j \]

\[ M = \text{arrays of length } d \cdot \alpha, \text{ with elements in } B \]

\[ \forall m \in M \text{ let } m(k) = \text{kth element of array } m \]

\[ \forall x \in A \text{ let } x_k = \text{kth digit of } x \text{ base } \alpha \]

\text{definition } f_m(x) = m(x_1+1) \oplus m(x_1+x_2+2) \oplus \ldots \oplus m \left( \sum_{k=1}^{d} x_k + k \right)

\[ H_2 = \{ f_m | m \in M \} \text{ is universal } \]

\text{proof for } x, y \in A,

Let \( f_m(x) = r_1 \oplus r_2 \oplus \ldots \oplus r_s \) rows of m

\[ f_m(y) = r_{s+1} \oplus \ldots \oplus r_t \]

Then \( f_m(x) = f_m(y) \iff r_1 \oplus \ldots \oplus r_t = 0 \)

But if \( x \neq y \Rightarrow \exists k \text{ s.t. } r_k \text{ in only one of } f_m(x), f_m(y) \)

So \( f_m(x) = f_m(y) \iff r_k = \bigoplus_{i \neq k} r_i \)

But there are only \(|B|\) possibilities for row \( r_k \)

so \( x, y \) will collide for \( \frac{1}{|B|} \) of fns \( f_m \in H_2 \)

\[ \text{Hence } H_2 \text{ is universal } \]
Analysis of Hashing
for Uniform Random Hash fn

load factor $\alpha = \frac{\text{# of keys hashed}}{\text{# of indices in Hash Table}}$

Hashing with Chaining
keep list of conflicts at each index

length is binomial variable

expected length = $\alpha$

Expected Time Cost per hash = $O(1 + \alpha)$

By Chernoff Bounds, with high likelyhood
time cost per hash $\leq O(\alpha \log(\# \text{ keys}))$
Open Address Hashing
(With Uniform Random Hash fn)

Resolve conflicts by applying another hash function

\[ \alpha = \text{load factor} = \text{prob. of occupied hash address} \]

# rehashes as geometric variable

expected hash time = \[ \frac{1}{1 - \alpha} = 1 + \alpha + \alpha^2 + \ldots \]