Today: Hashing - II
- Universal Hashing
- Analysis assuming universal hash family
- Construction of universal hash families
- Perfect hashing

-------------------

Additions on 10/15/2006

1. WE DID NOT COVER PERFECT HASHING IN LECTURE. THIS MATERIAL IS INCLUDED IN THESE NOTES AS "OPTIONAL READING" ONLY.

2. THE UNIVERSAL HASHING SCHEME IN THESE NOTES TAKE $O(\log n)$ TIME TO COMPUTE. THE TEXT DESCRIBES A MORE EFFICIENT ALGORITHM, TAKING $O(1)$ TIME TO COMPUTE. THIS IS THE FORM OF THE RESULT THAT IS MOST USEFUL TO REMEMBER.
Recall from last lecture

**Dictionary**:
- \( \text{INSERT}(k, s) \) \( \quad \) \( k \in \{0...u-1\} \)
- \( \text{SEARCH}(k, s) \) \( \{ |s| = n \) \)
- \( \text{DELETE}(k, s) \)

**Hashing + Chaining**:

- Expected time for **unsuccessful** \( \text{SEARCH} \) \( (k \text{ uniform in } U) \)
  \[ = \Theta(1 + \alpha) \] where \( \alpha = \frac{n}{m} \)

- Expected time for **successful** \( \text{SEARCH} \) \( (k \text{ uniform in } S) \)
  \[ = \Theta(1 + \alpha) \] also.

- Space complexity = \( n \left( 1 + \frac{1}{\alpha} \right) \)

**Good News**: linear Space; \( \Theta(1) \) running time on average.

**Bad News**: Needs strong assumption on distribution of \( \text{SEARCH} \) queries.

But this was necessary? or not?
Today: • Will allow worst possible sequence of INSERT, DELETE, SEARCH operations
  • Will pick h at random
  • Will get same complexities!

Plan: • Define UNIVERSAL hash family $H_{u,m,n}$
  • INITIALIZE $(S, U, m, n)$
    - Pick $h \in H_{u,m,n}$ at random
    - Initialize $L_i = \emptyset \forall i$
  • INSERT/SEARCH/DELETE $(R, S)$
    $i \leftarrow h(k)$;
    Insert/Search/Delete $k$ in $L_i$.
  • Analyze above based only on $h \in H$ random
Crux of analysis from last lecture:

\[ P_y \left[ h(k_1) = h(k_2) \text{ } | \text{ } k_1 \neq k_2 \right] = \frac{1}{m} \]

Can't use this today since \( k_1, k_2 \) never random.

Plan today: Replace with

for every \( k_1 \neq k_2 \)

\[ P_y \left[ h(k_1) = h(k_2) \text{ } | \text{ } h \in \mathcal{X} \right] = \frac{1}{m} \]

It is said to be universal if it satisfies this property.

**Questions**

1. Is **universal** useful?

2. Can we construct **universal** families efficiently?

   Answer to both: YES!!
**Part I**  
Univonality is Good

**Context:** Suppose

\[
(\sigma_1, k_1)
\]

\[
(\sigma_2, k_2)
\]

\[
\vdots
\]

\[
(\sigma_k, k_k)
\]

sequence of \( k \) updates

to \( S \) with

\[
\sigma_i \in \{\text{INSERT, DELETE, SEARCH}\}
\]

\( k_i \) keys.

At the end we have a set \( S \), \(|S| = n\).

Now consider a \textbf{SEARCH} \((k, S)\) query

(No assumption on distribution of \( \sigma_1, \sigma_2, \ldots, \sigma_k \)

\( k_1, \ldots, k_k \))

**Theorem:** if \( h \in \mathcal{H} \) is chosen uniformly from

\textbf{UNIVERSAL} family \( \mathcal{H} \), then

\[
E[\text{\# collisions with } k \text{ in } S] = O(1 + \frac{n}{m})
\]
Proof: let $S = \{ k_1, ..., k_n \}$; random

let $I_1, ..., I_n$ be indicator variables

with $I_j = 1$ if $h(k_j) = h(k)$

$= 0$ o.w.

$\# \text{ collisions} = \sum_{j=1}^{n} I_j$

$E[\# \text{ collisions}] = E[\sum_{j=1}^{n} I_j]$

$= \sum_{j=1}^{n} E[I_j]$

Case 1: $R \& S$

Then $E[I_j] = \frac{1}{m}$ (by Universal property)

$\Rightarrow \sum_{j=1}^{n} E[I_j] = \frac{n}{m} = O(1 + \frac{n}{m})$

Case 2: $R \& S \& N$ say $k = k'$

Then $E[\# \text{ collisions}] = 1 + \sum_{j=2}^{n} E[I_j]$

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

$= O(1 + \frac{n}{m})$
Conclusion: Goodness of Universal

\[ \Rightarrow \text{ If } \mathcal{H} \text{ is Universal then } \]

Expected time for \text{INSERT/DELETE/SEARCH} is \( O(1+\alpha) \) where

\[ \alpha = \text{Load Factor} = \frac{n}{m} = \frac{151}{151} \].

Important: Above does NOT assume any distribution on keys.

\[ \Rightarrow \]

Moving On:

Do Universal hash families even exist?

Answer: YES...

\( \mathcal{H} = \{ f : \{0, \ldots, m-1\} \rightarrow \{0, m-1\} \} \) is Universal for all such functions.

... but this is useless; can't compute/remember.
II. CONSTRUCTIVE UNIVERSAL Hash Families

CONSTRUCTION

- Let $m$ be a prime.
- $U = \text{power of } m = m^{y+1}$.

- So $\text{key} = r = <\mathbf{r}_0, \mathbf{r}_1, ..., \mathbf{r}_v>$.

- $X = \{ h_\mathbf{a} \mid \mathbf{a} = <a_0, a_1, ..., a_r> \in \mathbb{Z}_0, ..., m^{-1/2y+1} \}$

  where

  \[ h_\mathbf{a}(r) = \sum_{i=0}^{y} a_i \cdot r_i \pmod{m} \]

- $|X| = m^{y+1} = |U|$

  Storing $h$ is comparable to storing $r$.

Theorem: Above family $X$ is UNIVERSAL.
Proof: Need to prove for every $k \neq k'$

$$\Pr \left[ \sum_{i=0}^{\gamma} a_i k_i \equiv \sum_{i=0}^{\gamma} a_i k'_i \pmod{m} \right] = \frac{1}{m}$$

w.l.o.g. assume $k = k'$

then

$$\Pr \left[ \sum_{i=0}^{\gamma} a_i k_i \equiv \sum_{i=0}^{\gamma} a_i k'_i \pmod{m} \right] = \Pr \left[ \sum_{i=0}^{\gamma} a_i (k_i - k'_i) = 0 \right]$$

$$= E \left[ \Pr \left[ \sum_{i=0}^{\gamma} a_i (k_i - k'_i) = 0 \pmod{m} \right] \right]$$

$$= E \left[ \Pr \left[ a_{\gamma} = - (k_{\gamma} - k'_{\gamma}) \sum_{i=0}^{\gamma-1} a_i (k_i - k'_i) \right] \right]$$

Recall number theory

$$\forall b \not\equiv 0 \pmod{m} \exists b^{-1} \; s.t. \; b \cdot b^{-1} = 1$$

$$= E \left[ \frac{1}{m} \right] = \frac{1}{m} \quad \square$$
Conclude

- **INITIALIZE** $(S, m, y)$
  - Set $a = \langle a_0, \ldots, a_r \rangle$ where $a_i \in \{0, \ldots, m \}$ chosen uniformly
  - $\text{LIST}[j] \leftarrow \emptyset \ \forall \ j \in \{0, \ldots, m-1\}$

- **INSERT/DELETE/SEARCH** $(r = \langle r_0, \ldots, r_s \rangle, S)$
  
  $j = \sum a_i \cdot r_i \mod m$

  INSERT/DELETE/SEARCH $(r, \text{LIST}[j])$

- Space complexity $= n + m$ ($n = \text{max size of } S$)

- Expected Time complexity of Ins/Search/Delete $= O\left(1 + \frac{n}{m}\right) = O\left(1 + \frac{1}{d}\right)$

  $d = 1 \Rightarrow \text{linear space}$

  Expected constant time
[The following material was not covered in lecture.]

**Perfect Hashing:**

**Static Dictionary Problem**

- Want to construct one hash function \( h \), that does not collide on \( S \); and uses linear space.

**Goals:** Worst case \( \Theta(1) \) time for search

- Space \( \Theta(|S|) \).

---

**Perfect Hashing [FKS]**

Idea: two level universal hashing

- Given \( S = \{k_1, ... , k_n\} \)

- **Stage 1:** Pick \( h_i : \{0, ... , m-1\} \rightarrow \{0, ... , m-1\} \) at random where \( m \approx n \) prime.

  Let \( \mathcal{C}_j = \{ i \mid \text{for all } i \in \{0, ... , n-1\}, h_i(k_j) = j \} \)
if \( 0 \leq l^2 \leq 10n \)

o.k., else retry Stage 2;

- Stage 2: For \( m_i \) prime \( \leq l^2 \)

  (Stage 2i) pick \( h_{2,i} : \{0 \ldots u-1\} \rightarrow \{0 \ldots m_i - 1\} \)

  let \( S_i = \{ k_i | h_1(k_i) = j \} \)

  if \( \exists k, k' \in S_i \) s.t.

    \( h_{2,i}(k) = h_{2,i}(k') \)

    then

    retry (Stage 2i)

  else continue;

- Stage 3:

  SEARCH \((k, S)\)

  . \( j \leftarrow h_1(k) \)

  . \( j_2 \leftarrow h_{2,i}(k) \)

  . Return \((j, j_2)\) if \( T[j][j_2] = k \)

  . \( \phi \) o.w.

  \( T[j] \leftarrow \text{array of length } m_i \)
Claim 1. \( P_v \left[ \exists k, k' \in S \text{ s.t. } h_{2, i}(k) = h_{2, j}(k') \right] \)

\[
\frac{1}{2} \leq \frac{1}{2}
\]

Proof: \( \leq \leq P_v \left[ \text{ } h_{2, i}(k) = h_{2, j}(k') \text{ } \right] \)

\[
\leq \left( l_i \right) \cdot \frac{1}{2} \leq \frac{1}{2}
\]

Claim 2. \( P_v \left[ \sum l_j \geq 10 n \right] \leq \frac{1}{2} \)

Proof: Note that

\[
E\left[ \sum_{j=1}^{m-1} l_j^2 \right] = E\left[ \sum_{i=1}^{n} \sum_{i_1=1}^{n} I_{i, i_2} \right]
\]

where \( I_{i, i_2} = 1 \text{ if } h(k_{i_1}) = h(k_{i_2}) \text{ otherwise } = 0 \)

\[
= E\left[ \sum_{i} I_{i, i} \right] + E\left[ \sum_{i_1 \neq i_2} I_{i_1, i_2} \right]
\]
\[ n + \sum_{i_1 i_2} \mathbb{E} [I_{i_1 i_2}] \]

\[ = n + n(n-1) \cdot \frac{1}{m} \]

\[ \leq 2n \quad \text{if} \quad n \geq m \]

\[ P_\epsilon \left[ \sum \xi_i^2 > 10n \right] \leq \frac{1}{5} \quad \text{by Markov's} \]

**Conclude:** Each stage expected to run twice before finding right hash.