Recap: Universal Hashing. (from lecture)

A family $\mathcal{H}$ of hash functions mapping keys from $U \rightarrow \{0, 1, 2, \ldots, m-1\}$ is universal if

$$\forall x \neq y \in U : \Pr [h(x) = h(y)] = \frac{1}{m}$$

(can also say: $\forall x \neq y \in U$, $|\{h \in \mathcal{H} \mid h(x) = h(y)\}| = \frac{1}{m}$)

Thm (from class): Choose $h \in \mathcal{H}$ at random, and hash all keys $e \in S$ (where $S \subseteq U$) into $m$ slots.

Given key $x$, $\mathbb{E}[\# \text{ collisions with } x] \leq \frac{|S|}{m}$

from keys $y \neq x$

Then $\mathbb{E}[T_{\text{lookup}}] \leq \Theta(1) + \frac{|S|}{m} = \Theta(1)$, if $|S| = \Theta(m)$

$\# \text{ elts in each slot, since we're doing chaining}$

Worst-case lookup still $\Theta(|S|)$ (All hash to same slot)

Goal for today: a hash table w/ good worst case behavior.
Perfect Hashing

Given \( n \) keys, construct a static hash table size \( \Theta(n) \) such that worst-case lookup is \( \Theta(1) \). Must know all keys ahead of time!

Ex. immutable file storage, CD, DVD

Recall (from lecture): \( E \left[ \text{total \# collisions} \right] \leq \binom{n}{2} \cdot \frac{1}{m} \)

\( x \neq y, h(x) = h(y) \)

\# of slots

\# of keys

\[ I_{xy} = \begin{cases} 1 & \text{if } h(x) = h(y) \\ 0 & \text{otherwise} \end{cases} \]

\[ \Rightarrow E\left[ \text{\# collisions} \right] = E\left[ \sum_{x 
eq y} I_{xy} \right] \]

\[ = \sum_{x 
eq y} E[I_{xy}] = \binom{n}{2} \cdot \frac{1}{m} \]

\# of pairs by def of universal hashing

Remember \( \binom{n}{2} = \frac{n(n-1)}{2} \)

\[ \Rightarrow E[\text{\# collisions}] < \frac{1}{2} \]

when \( m = n^2 \)

Idea: Can take \( \Theta(n^2) \) space and find a hash function with no \( n^2 \) slots on our \( n \) keys

Why? Markov's Inequality: \( P(X \geq c \cdot E[X]) \leq \frac{1}{c} \)

\[ \Rightarrow P(\# \text{collisions} \geq 1) \leq \frac{1}{2} \]

\[ \Rightarrow P(\text{no collisions}) > \frac{1}{2} \]
Since \( P(\text{no collisions}) > \frac{1}{2} \).

Then if \( X = \# \text{ tries till we get } h \in \mathcal{H} \text{ w/ no collisions} \),

\[
\mathbb{E}[X] = 1 \cdot \left( \frac{1}{2} \right) + 2 \cdot \left( \frac{1}{2^2} \right) + 3 \left( \frac{1}{2^3} \right) + \ldots
\]

\[
= \sum_{i=1}^{\infty} \left( \sum_{j=1}^{\infty} \frac{1}{2^i} \right) = \sum_{j=1}^{\infty} \frac{1}{2^j} = \boxed{2}
\]

So, expected \( \# \text{ tries} \) is 2 to get \( h \) : no collisions on hashing \( n \) keys \( \rightarrow \) \( n^2 \) slots.

**BUT** we want \( \Theta(n) \) space!

Better idea: **Two-level Hash Table** w/ universal hashing

![Diagram of two-level hash table](image)

- First level: stores hash fn for second level.
- Second level: stores actual data

**LOOKUP** \((x)\):

\[
i = h(x) \quad \text{index in 1st level}
\]

\[
 j = h_i(x) \quad \text{index in 2nd level}
\]

return \( B_i[j] \)
Properties:
- All keys mapped to \( i \) by \( h \) (1st level) are hashed into 2nd level \( B_i \).
- No collisions in 2nd level:
  - by setting \( m_i = n_i^2 \)
  - \( \# \text{ keys } \times \text{ st. } h(x) = i \)
  - \( m = n \)

Storage Analysis:

\[
E[\text{storage}] = E[\Theta(n) + \sum_{i=0}^{m-1} \Theta(n_i)^2] \\
= \Theta(n) + \Theta[E[\sum_{i=0}^{m-1} n_i^2]]
\]

What is \( E[\sum_{i=0}^{m-1} n_i^2] \)?

Use the property:
\( a^2 = a + 2 \binom{a}{2} \) (Proof: substitute \( \binom{a}{2} = \frac{a(a-1)}{2} \))

Then
\[
E[\sum_{i=0}^{m-1} n_i^2] = E[\sum_{i=0}^{m-1} n_i^2 + 2 \sum_{i=0}^{m-1} \binom{n_i}{2}]
\]
- \( E[\sum_{i=0}^{m-1} n_i] + 2E[\sum_{i=0}^{m-1} \binom{n_i}{2}] \rightarrow \text{Total number of pairs of keys that collide} \)

By universality, any pair \( x, y \) (\( x \neq y \)) has probability \( \frac{1}{m} \) of colliding.

\( \Rightarrow \) expected \# of pairs that collide is at most \( \binom{n}{2} \times \frac{1}{m} = \frac{n-1}{2} \)

So
\[
E[\sum_{i=0}^{m-1} n_i^2] \leq n + 2 \left( \frac{n-1}{2} \right) = 2n
\]
So, \( E[\text{storage}] = \Theta(n) + \Theta(E[\sum_{i=0}^{m-1} n_i^2]) \)

\[ \leq \Theta(n) + \Theta(2n) \]

\[ = \Theta(n) \]

So, expected storage for 2-level scheme is linear.

* but we want actual, not expected

Let try lots of times until we get storage = \( \Theta(n) \)

Therefore, hash table on \( S \) with \( n \) keys \( |S| = n \)

**Construct (set \( S \))**:

repeat:

- Pick \( h \in \mathcal{H} \) at random
- Hash all keys in \( S \) until total storage is \( \Theta(n) \)

for each group \( B_i \):

repeat:

- Pick \( h_i \in \mathcal{H}_i \) at random
  - Family of hash fns, hashing to \( m_i = n_i^2 \) slots
- Hash all keys in \( B_i \) until there are no collisions in \( B_i \).
Aside: Bucket Sort (related to what we were just using)

Given: \( n \) numbers from some distribution (e.g. uniform in \([0, 1)\))

Divide them into \( n \) "buckets" of equal range

For \([0, 1)\) example:

\[
\begin{align*}
0 & \quad \frac{1}{n} & \quad \frac{2}{n} & \quad \cdots & \quad \frac{\lfloor n/2 \rfloor \cdot \frac{1}{n}}{n} \\
& \quad \frac{\lfloor n/2 \rfloor \cdot \frac{1}{n}}{n} & \quad \frac{\lfloor n/2 \rfloor \cdot \frac{1}{n}}{n} & \quad \cdots & \quad \frac{\lfloor n/2 \rfloor \cdot \frac{1}{n}}{n} \\
& \quad \frac{n}{n} & \quad \frac{n}{n} & \quad \cdots & \quad \frac{n}{n} \\
& \quad \frac{n}{n} & \quad \frac{n}{n} & \quad \cdots & \quad \frac{n}{n}
\end{align*}
\]

\( n \) groups of size \( n \).

Put each element \( i \) into its corresponding bucket.

Note: For any \( i, b \), \( P(i \text{ is in bucket } b) = \frac{1}{n} \).

Sort each bucket using insertion sort for \( \Theta(n_b^2) \) time, then concatenate.

Analysis same as perfect hashing: \( E(T(n)) = O(n) \)