Chapter 7, Hash tables:

▶ General introduction; separate chaining (Friday, Nov 18)
▶ Open addressing (Monday before Thanksgiving)
▶ Hash table performance (Today)
▶ (Begin Chapter 8, Strings (Wednesday))

Today:

▶ Elements of hashtable performance
▶ Clustering and chaining in open addressing
▶ The mathematics of hash functions
▶ Perfect hashing
Coming up:
Do **Open Addressing** project (suggested by Friday, Dec 2)

Due **Today, Nov 28** (end of day) (recommended to have been done before break)
Read Section 7.3
Do Exercises 7.(4,5,7,8)
Take quiz (on Section 7.3 etc)

Due **Wed, Nov 30** (end of day)
Read Section 8.1
Do Exercises 8.(4 & 5)

Due **Thurs, Dec 1**
Take quiz (on Section 8.1)

Due **Fri, Dec 2**
Do Exercises 8.(7, 14, 20)
Read Section 8.2
Find: Search the data structure for a given key
Insert: Add a new key to the data structure
Delete: Get rid of a key and fix up the data structure

containsKey() Find
get() Find
put() Find + insert
remove() Find + delete
<table>
<thead>
<tr>
<th>Data Structure</th>
<th>Find</th>
<th>Insert</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsorted array</td>
<td>$\Theta(n)$</td>
<td>$\Theta(1)$ [(\Theta(n))]</td>
<td>$\Theta(n)$</td>
</tr>
<tr>
<td>Sorted array</td>
<td>$\Theta(lg \ n)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(n)$</td>
</tr>
<tr>
<td>Linked list</td>
<td>$\Theta(n)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(1)$</td>
</tr>
<tr>
<td>Balanced BST</td>
<td>$\Theta(lg \ n)$</td>
<td>$\Theta(1)$ [(\Theta(lg \ n))]</td>
<td>$\Theta(1)$ [(\Theta(lg \ n))]</td>
</tr>
<tr>
<td>What we want</td>
<td>$\Theta(1)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(1)$</td>
</tr>
</tbody>
</table>
\[
\begin{align*}
\text{rehash} & \quad \rightarrow \\
O(1) & \quad c_0 \\
O(1) & \quad c_0 \\
O(1) & \quad c_0 \\
\cdots & \quad \cdots \\
O(1) & \quad c_0 \\
O(n) & \quad c_1 + c_2 n \\
O(1) & \quad c_0 \\
\cdots & \quad \cdots \\
O(1) & \quad c_0
\end{align*}
\]

\[
T(n) = (n - 1)c_0 + c_1 + c_2 n = (c_0 + c_2)n + c_1 - c_0 = \Theta(n)
\]
\[
\frac{(n+1) + n + (n-1) + \cdots + 3 + 2 + 1 + \cdots + 1}{m}
\]

\[
= \frac{m + n + (n-1) + \cdots + 2 + 1}{m}
\]

the initial \( m \) accounting for the last probe in each case

\[
= \frac{m}{m} + \frac{(n+1) \cdot \frac{n}{2}}{m}
\]

as an arithmetic series

\[
\approx 1 + \frac{(n+1) \cdot \frac{n}{2}}{2 \cdot n}
\]

since \( m \) is about \( 2 \cdot n \)

\[
= 1 + \frac{n + 1}{4}
\]

by cancellation
\[\frac{[s_0 + 1 + s_0 + (s_0 - 1) + \cdots + 2] + \cdots + 1 + \cdots 1}{m} = 1 + \frac{\sum_{i=0}^{\gamma-1} \sum_{j=1}^{s_i} j}{m}\]
What is the probability that a miss $k$ requires at least $i$ probes?

\[
h(k) \uparrow \quad \cdots \quad \uparrow \quad h(k) + i - 1 \\
\quad h(k) + 1 \quad \quad h(k) + i - 2
\]

**Conditional probability**

$P(X \mid Y)$: What is the probability of event $X$ in light of event $Y$?

\[
P(X \land Y) = P(X) \cdot P(X \mid Y)
\]

\[
P(X_0 \land X_1 \land \cdots \land X_{N-1}) = P(X_0) \cdot P(X_1 \mid X_0) \cdot P(X_1 \mid X_0 \land X_1) \cdots P(X_{N-1} \mid X_0 \land \cdots \land X_{N-2})
\]
\[ P(T[h(k) + 1] \neq \text{null} \mid T[h(k)] \neq \text{null}) = \frac{n - 1}{m - 1} \]

The probability that a miss requires at least \( i \) probes:

\[
\frac{n}{m} \cdot \frac{n - 1}{m - 1} \cdots \frac{n - i + 2}{m - i + 2} 
\leq \left(\frac{n}{m}\right)^{i-1} \quad \text{since } n < m
\]

\[
\leq \alpha^{i-1} \quad \text{by substitution}
\]
\[
\sum_{i=1}^{m} i \cdot P(\text{it takes } i \text{ probes}) = \sum_{i=1}^{m} i \cdot \left( P(\text{it takes at least } i \text{ probes}) - P(\text{it takes at least } i + 1 \text{ probes}) \right) \\
= \sum_{i=1}^{m} P(\text{it takes at least } i \text{ probes}) \quad \text{by telescoping} \\
\leq \sum_{i=1}^{m} \alpha^{i-1} \quad \text{by the previous result} \\
\leq \sum_{i=1}^{\infty} \alpha^{i-1} \quad \text{since } m < \infty \\
= \sum_{i=0}^{\infty} \alpha^{i} \quad \text{by a change of variable} \\
= \frac{1}{1 - \alpha} \quad \text{by geometric series}
\]
Is the following assumption true for linear probing?

\[ P(T[h(k) + 1] \neq \text{null} \mid T[h(k)] \neq \text{null}) = \frac{n-1}{m-1} \]

In general, is the following assumption true for a probing strategy?

\[ P(T[\sigma(k,1)] \neq \text{null} \mid T[\sigma(k,0)] \neq \text{null}) = \frac{n-1}{m-1} \]

What is the difference between

Each array index is equally likely to be the hash of a given key. vs Each array position is equally likely to be occupied.
Linear probing is biased towards clustering:

<table>
<thead>
<tr>
<th>x</th>
<th>Number of buckets with exactly x previous buckets filled</th>
<th>Number of filled buckets with exactly x previous buckets filled</th>
<th>Probability that a bucket is filled if exactly x previous buckets are filled</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>97</td>
<td>48</td>
<td>.495</td>
</tr>
<tr>
<td>1</td>
<td>48</td>
<td>22</td>
<td>.458</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>12</td>
<td>.545</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>7</td>
<td>.583</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>4</td>
<td>.571</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>.75</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>2</td>
<td>.667</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Expected number of probes for a miss in a hashtable using linear probing (from Knuth):

\[
\frac{1}{2} \cdot \left(1 + \frac{1}{(1 - \alpha)^2}\right)
\]
After \( n \) calls to \texttt{put()} with unique keys, no removals, consider \textbf{average chain length} over all keys (low is good), \textbf{percent of keys that are in their ideal location} (high is good), and \textbf{length of the longest chain} (low is good).

<table>
<thead>
<tr>
<th></th>
<th>Linear probing</th>
<th>Quadratic probing</th>
<th>Double hashing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( n )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Surnames</td>
<td>1000</td>
<td>2.092 64.7% 31</td>
<td>1.421 75.8% 9</td>
</tr>
<tr>
<td>Mountains</td>
<td>1360</td>
<td>1.568 73.8% 17</td>
<td>1.729 65.8% 11</td>
</tr>
<tr>
<td>Mountains (height)</td>
<td>1360</td>
<td>1.932 75.1% 99</td>
<td>1.882 68.9% 18</td>
</tr>
<tr>
<td>Chemicals</td>
<td>663</td>
<td>1.517 75.0% 16</td>
<td>1.729 65.5% 10</td>
</tr>
<tr>
<td>Chemicals (symbol)</td>
<td>663</td>
<td>1.885 71.0% 20</td>
<td>1.837 66.4% 13</td>
</tr>
<tr>
<td>Books</td>
<td>718</td>
<td>1.419 76.7% 8</td>
<td>1.659 70.0% 11</td>
</tr>
<tr>
<td>Books (ISBN)</td>
<td>718</td>
<td>1.542 74.4% 21</td>
<td>1.670 67.8% 15</td>
</tr>
<tr>
<td>Random strings</td>
<td>5000</td>
<td>1.544 77.6% 49</td>
<td>1.735 69.9% 37</td>
</tr>
<tr>
<td>Random strings</td>
<td>5000</td>
<td>1.531 77.1% 35</td>
<td>1.729 69.8% 28</td>
</tr>
<tr>
<td>Random strings</td>
<td>5000</td>
<td>1.643 77.5% 76</td>
<td>1.754 68.6% 29</td>
</tr>
</tbody>
</table>
Hash functions should distribute the keys *uniformly* and *independently*.

**Uniformity:**

\[
P(h(k) = i) = \frac{1}{m}
\]

**Independence:**

\[
P(h(k_1) = i) = P(h(k_1) = i \mid h(k_2) = j)
\]
Why do we talk about integer hashes?
Division method:

\[ h(k) = k \mod m \]

Middle square method (see code)

Multiplicative method:

\[ h(k) = \lfloor m(k \cdot a - \lfloor k \cdot a \rfloor) \rfloor \]

“Universal” hash (later...)
ASCII sum:

\[ h(k) = \left( \sum_{i=0}^{n-1} s[i] \right) \]

String polynomial:

\[ h(k) = (k[0] \cdot b^{n-1} + k[1] \cdot b^{n-2} + \cdots + k[n-2] \cdot b + k[n-1]) \mod m \]

Carter-Wegman:

\[ h(k) = (h_0(k[0]) + h_1(k[1]) + \cdots + h_{n-1}(k[n-1])) \mod m \]

\[ = \left( \sum_{i=0}^{n-1} h_i(k[i]) \right) \mod m \]
### Area codes ($n = 303$)

<table>
<thead>
<tr>
<th>Method</th>
<th>Average penalty</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Division</td>
<td>0.673</td>
<td>0.808</td>
</tr>
<tr>
<td>Mid square</td>
<td>1.090</td>
<td>1.640</td>
</tr>
<tr>
<td>Multiplicative</td>
<td>0.508</td>
<td>0.478</td>
</tr>
<tr>
<td>Fibonacci</td>
<td>0.617</td>
<td>0.696</td>
</tr>
<tr>
<td>Universal</td>
<td>0.578</td>
<td>0.617</td>
</tr>
</tbody>
</table>

### Book ISBNs ($n = 718$)

<table>
<thead>
<tr>
<th>Method</th>
<th>Average penalty</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Division</td>
<td>0.618</td>
<td>1.050</td>
</tr>
<tr>
<td>Mid square</td>
<td>0.812</td>
<td>1.480</td>
</tr>
<tr>
<td>Multiplicative</td>
<td>0.565</td>
<td>0.954</td>
</tr>
<tr>
<td>Fibonacci</td>
<td>0.544</td>
<td>0.873</td>
</tr>
<tr>
<td>Universal</td>
<td>0.667</td>
<td>1.150</td>
</tr>
<tr>
<td>Method</td>
<td>Average penalty</td>
<td>Variance</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>----------</td>
</tr>
<tr>
<td><strong>Randomly generated from ([0, 1000)) ((n = 150))</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Division</td>
<td>1.36</td>
<td>.958</td>
</tr>
<tr>
<td>Mid square</td>
<td>1.86</td>
<td>1.96</td>
</tr>
<tr>
<td>Multiplicative</td>
<td>1.34</td>
<td>.919</td>
</tr>
<tr>
<td>Fibonacci</td>
<td>1.41</td>
<td>1.07</td>
</tr>
<tr>
<td>Universal</td>
<td>1.39</td>
<td>1.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Average penalty</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Randomly generated from ([0, 1000)) ((n = 400))</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Division</td>
<td>.518</td>
<td>1.16</td>
</tr>
<tr>
<td>Mid square</td>
<td>1.73</td>
<td>3.68</td>
</tr>
<tr>
<td>Multiplicative</td>
<td>.405</td>
<td>.930</td>
</tr>
<tr>
<td>Fibonacci</td>
<td>.448</td>
<td>.980</td>
</tr>
<tr>
<td>Universal</td>
<td>.488</td>
<td>1.08</td>
</tr>
<tr>
<td></td>
<td>Average penalty</td>
<td>Variance</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------</td>
<td>----------</td>
</tr>
<tr>
<td><strong>Chemicals</strong> ($n = 663$)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCII sum</td>
<td>.505</td>
<td>1.00</td>
</tr>
<tr>
<td>String polynomial</td>
<td>.424</td>
<td>.805</td>
</tr>
<tr>
<td>Carter-Wegman</td>
<td>.800</td>
<td>1.63</td>
</tr>
<tr>
<td><strong>Books</strong> ($n = 718$)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCII sum</td>
<td>.818</td>
<td>1.51</td>
</tr>
<tr>
<td>String polynomial</td>
<td>.745</td>
<td>1.30</td>
</tr>
<tr>
<td>Carter-Wegman</td>
<td>2.06</td>
<td>4.08</td>
</tr>
<tr>
<td>Method</td>
<td>ASCII sum</td>
<td>String polynomial</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------</td>
<td>-------------------</td>
</tr>
<tr>
<td><strong>Randomly generated strings</strong> ((n = 150))</td>
<td>1.32</td>
<td>1.43</td>
</tr>
<tr>
<td>Average penalty</td>
<td>.879</td>
<td>1.09</td>
</tr>
<tr>
<td><strong>Randomly generated strings</strong> ((n = 400))</td>
<td>.515</td>
<td>.425</td>
</tr>
<tr>
<td>Variance</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A hashing scheme must reduce the occurrence of collisions and “deal” with them when they happen.

- **Separate chaining**, where $m < n$, deals with collisions by chaining keys together in a bucket.
- **Open addressing**, where $n < m$, deals with collisions by finding an alternate location.
- **Perfect hashing** deals with collisions by preventing them altogether.

This topic is parallel with the **optimal BST problem**: What if we knew the keys ahead of time? What if we got to choose the hash function based on what keys we have?
Let $H$ stand for a class of hash functions (a set of hash functions defined by some formula).

Let $m$ be the number of buckets.

$H$ is universal if

$$\forall k, \ell \in \text{Keys}, \quad |\{h \in H \mid h(k) = h(\ell)\}| \leq \frac{|H|}{m}$$
$\mathcal{H}$ is universal if

$$\forall k, \ell \in \text{Keys}, \ |\{h \in \mathcal{H} | h(k) = h(\ell)\}| \leq \frac{|\mathcal{H}|}{m}$$

One particular family of classes of hash functions, given $p$, a prime number greater than all keys, and $m$, the number of buckets, is denoted $\mathcal{H}_{mp}$:

$$\mathcal{H}_{mp} = \{ \ h_{ab}(k) = ((ak + b) \mod p) \mod m \ | \ a \in [1, p) \text{ and } b \in [0, p) \}$$
Theorem $H_{pm}$ is universal.

Proof. Suppose $p$ and $m$ as specified earlier. Suppose $k, \ell \in \text{Keys}$, and $h_{ab} \in H_{pm}$ (which implies supposing that $a \in [1, p)$ and $b \in [0, p)$).

Let $r = (a \cdot k + b) \mod p$ and $s = (a \cdot \ell + b) \mod p$

Subtracting gives us

$$r - s \equiv (a \cdot k + b) - (a \cdot \ell + b) \mod p$$

$$\equiv a \cdot (k - \ell) \mod p$$

Now $a$ cannot be 0 because $a \in [1, p)$. Similarly $k - \ell$ cannot be 0, since $k \neq \ell$. Hence $a \cdot (k - \ell) \neq 0$.

Since $p$ is prime and greater than $a$, $k$, and $\ell$, it cannot be a factor of $a \cdot (k - \ell)$. In other words, $a \cdot (k - \ell) \mod p \neq 0$. By substitution, $r - s \neq 0$, and so $r \neq s$.

By another substitution, $(a \cdot k + b) \mod p \neq (a \cdot \ell + b) \mod p$. 
Define the following function, given \( k \) and \( \ell \), which maps from \((a, b)\) pairs to \((r, s)\) pairs (formally, \([1, p) \times [0, p) \rightarrow [1, p) \times [0, p)\)):

\[
\phi_{k\ell}(a, b) = ((a \cdot k + b) \mod p, (a \cdot \ell + b) \mod p)
\]

Now consider the inverse of that function.

\[
\phi_{k\ell}^{-1}(r, s) = (((r - s) \cdot (k - \ell)^{-1}) \mod p, (r - ak) \mod p)
\]

\[
= (a, b)
\]

The existence of \( \phi^{-1} \) implies that \( \phi \) is a one-to-one correspondence. Hence for each \((a, b)\) pair, there is a unique \((r, s)\) pair. Since the pair \((a, b)\) specifies a hash function, that means that for each hash function in the family \( \mathcal{H}_{pm} \), there is a unique \((r, s)\) pair.
There are $p - 1$ possible choices for $a$ and $p$ choices for $b$, so there are $p \cdot (p - 1)$ hash functions in family $\mathcal{H}_{pm}$. Likewise there are $p$ choices for $r$, and for each $r$ there are $p - 1$ choices for $s$ (since $s \neq r$). Thus we can partition the set $\mathcal{H}_{pm}$ into $p$ subsets by $r$ value, each subset having $p - 1$ hash functions. For a given $r$, at most one out of every $m$ can have an $s$ that is equivalent to $r \mod m$, in other words, at most $\frac{p-1}{m}$ hash functions. Now sum that for all $p$ of the subsets of $\mathcal{H}_{pm}$, and we find that the number of hash functions for which $k$ and $\ell$ collide are

$$p \cdot \frac{p - 1}{m} = \frac{p \cdot (p - 1)}{m} = \frac{|\mathcal{H}_{pm}|}{m}$$

Therefore $\mathcal{H}_{pm}$ is universal by definition. □
Theorem [Probability of any collisions.] If Keys is a set of keys, \( m = |\text{Keys}|^2 \), \( p \) is a prime greater than all keys, and \( h \in \mathcal{H}_p \), then the probability that any two distinct keys collide in \( h \) is less than \( \frac{1}{2} \).

**Proof.** Suppose we have a set Keys, \( m = |\text{Keys}|^2 \), \( p \) is a prime greater than all keys, and \( h \in \mathcal{H}_p \).

Consider the number of pairs of unique keys. The number of pairs of keys is

\[
\binom{n}{2} = \frac{n!}{2! \cdot (n - 2)!} = \frac{n!}{2 \cdot (n - 2)!} = \frac{n \cdot (n - 1) \cdot (n - 2)!}{2 \cdot (n - 2)!} = \frac{n \cdot (n - 1)}{2}
\]
Since $\mathcal{H}_{pm}$ is universal, each pair collides with probability $\frac{1}{m}$. Multiply that by the number of pairs, and the expected number of collisions is

$$\frac{n \cdot (n-1)}{2} \cdot \frac{1}{m} < \frac{n^2}{2} \cdot \frac{1}{m} \quad \text{since} \ n \cdot (n - 1) < n^2$$

$$= \frac{n^2}{2} \cdot \frac{1}{n^2} \quad \text{since} \ m = n^2$$

$$= \frac{1}{2} \quad \text{by cancelling} \ n^2$$

With the expected number of collisions less than one half, the probability there are any collisions is also less than $\frac{1}{2}$. □
\[ h(k) = (93, 0) \in H_{101, 10} \]

\[ h_1(k) = (0, 0) \in H_{101, 0} \]
\[ h_2(k) = (56, 15) \in H_{101, 9} \]
\[ h_3(k) = (47, 22) \in H_{101, 4} \]
\[ h_6(k) = (1, 100) \in H_{101, 4} \]
\[ h_7(k) = (0, 0) \in H_{101, 0} \]
\[ h_8(k) = (0, 0) \in H_{101, 0} \]

\[ h_1(k) = (0, 0) \in H_{101, 0} \]
\[ h_2(k) = (56, 15) \in H_{101, 9} \]
\[ h_3(k) = (47, 22) \in H_{101, 4} \]
\[ h_6(k) = (1, 100) \in H_{101, 4} \]
\[ h_7(k) = (0, 0) \in H_{101, 0} \]
\[ h_8(k) = (0, 0) \in H_{101, 0} \]
Coming up:  
*Do Open Addressing* project (suggested by Friday, Dec 2)

**Due Today, Nov 28 (end of day)** (recommended to have been done before break)  
Read Section 7.3  
Do Exercises 7.(4,5,7,8)  
Take quiz (on Section 7.3 etc)

**Due Wed, Nov 30 (end of day)**  
Read Section 8.1  
Do Exercises 8.(4 & 5)

**Due Thurs, Dec 1**  
Take quiz (on Section 8.1)

**Due Fri, Dec 2**  
Do Exercises 8.(7, 14, 20)  
Read Section 8.2