10—Hashing I

Chapter 8.3, 8.5

CSE326 Spring 2002

April 24, 2002

--- Easy Dictionary ---

- $n$ numbers in range 1 \ldots N
- How long to Find?
- How long to Insert?
- How long to Remove?
- Why don’t we use this?

--- Complicating the Issue ---

$h : K \rightarrow T$

$h(k) = k$
Shrinking the Hash Table

What Do We Want From $h$?

Comprehensive Collision Insurance

- $m < N$ so $h(x) = h(y)$ for some $x \neq y \in K$
- $x$ and $y$ collide
- How do we resolve the collision?
  * Chaining
  * Open Addressing
Chains of Love

Separate Chaining

- $T$ is a table of buckets
- How many probes to find $c$, $d$, $e$?

How to Insert?

How to Delete?
**Important Quantities**

- Load Factor: \( \alpha = \frac{n}{m} \)
- \( S(\alpha) \): Expected \# probes for successful search
- \( U(\alpha) \): Expected \# probes for unsuccessful search

\[ U(\alpha) = 1 + \alpha \]

\[ U(\alpha) = E[1 + \text{chain search}] = 1 + E[\text{chain length}] = 1 + \alpha \]
Calculating $S$

$S(n) \approx 2 + \frac{n}{2}$

What should $n$ and $m$ be for constant-time operations?

---

Separate Chaining: Summary

- Good performance if bucket lists aren’t too long
  - Both theoretically and practically
  - Much depends on the hash function
- Same space overhead as binary tree
  - key + pointer or two
  - Still too much for many applications

---

Coalesced Chaining

Store buckets *internally* in the table
Coalesced Chaining

- Allocate new cells from top of table
- Insertion order:
  g a d b c f

Coalesced Chaining Imperfect

How to Delete?

Deleting

Add deleted field to entry
--- Searching After a Delete ---

Search for c
Skip over deleted = 1 entries

--- Inserting After a Delete ---

Watch deleted field when inserting

--- Delete Problems ---

- What happens to searches with lots of deleted items?
- What happens to inserts?
More Coalesced Problems

What happens when table gets full?

Open Addressing

How much space for Separate Chained Table? Coalesced Chained?
- r-bit records (key + info), p-bit pointers

Open Addressing

- Table is array of keys
- Need to set probe sequence to resolve collisions
  - Linear Probing
  - Quadratic Probing
  - Double Hashing

h(k) = k mod 10
Linear Probing

- First try \( h(k) \)...
- \( \ldots \) then \( (h(k) + 1) \mod m \)
- \( \ldots \) then \( (h(k) + 2) \mod m \)
- \( \ldots \) in general, \( (h(k) + i) \mod m \)

Problem

Once a primary cluster gets started, it tends to grow and slow things down

Problem Solution

- This is a problem of the probe sequence
- We want:
  - Random probe sequence (so no clusters form)
  - Deterministic probe sequence (so we can find colliding keys)
Quadratic Probing

- First try $h(k)$...
- ... then $(h(k) + 1^2) \mod m$
- ... then $(h(k) + 2^2) \mod m$
- ... in general, $(h(k) + i^2) \mod m$

Oops

Should be easy to insert 82, the table is only half full...

The Problem with Squaring

<table>
<thead>
<tr>
<th>$i$</th>
<th>$i \mod 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>

Maybe 10 was a bad choice for the table size?
Oops, We Did It Again

<table>
<thead>
<tr>
<th>$i$</th>
<th>$i \mod 15$</th>
<th>$i$</th>
<th>$i \mod 12$</th>
<th>$i$</th>
<th>$i \mod 7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>4</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It's Not Just Us

- Just like regular math, $x^2 \equiv (-x)^2 \pmod{m}$

One set of #'s modulo 10

- $-6 \mod 10 = 4 \mod 10$
- $-5 \mod 10 = 5 \mod 10$
- $-4 \mod 10 = 6 \mod 10$
- $-3 \mod 10 = 7 \mod 10$
- $-2 \mod 10 = 8 \mod 10$
- $-1 \mod 10 = 9 \mod 10$

Another set of #'s modulo 10

- $\cdots -8 \equiv 2 \equiv 12 \cdots \pmod{10}$

The World is Against Squares

<table>
<thead>
<tr>
<th>$i$</th>
<th>$i \mod 10$</th>
<th>$i$</th>
<th>$i \mod 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

A quadratic probe sequence will always touch only half the table.
Double Hashing

\[ h(k) = k \mod 11 \]
\[ h_2(k) = \frac{k}{11} \mod 11 \]

• Probe sequence is \( h(k) + i \cdot h_2(k) \)

• 37 mod 11 = 4

• Probe increment \( \lfloor \frac{37}{11} \rfloor \mod 11 = 3 \)

Probe sequence is \( h(k) + i \cdot h_2(k) \)

\[ 37 \mod 11 = 4 \]

Probe increment \( \lfloor \frac{37}{11} \rfloor \mod 11 = 3 \)

Double Hashing

\[ h(k) = k \mod 11 \]
\[ h_2(k) = \frac{k}{11} \mod 11 \]

• 205 mod 11 = 7

• Probe increment is \( \lfloor \frac{205}{11} \rfloor \mod 11 = 7 \)

205 mod 11 = 7

Probe increment is \( \lfloor \frac{205}{11} \rfloor \mod 11 = 7 \)