Lab_hash is optional

Release date: December 1st

Due date: December 8th
ICES Evaluations

You are strongly encouraged to provide course feedback

go.illinois.edu/ices-online
Learning Objectives

Motivate and define a hash table

Discuss what a ‘good’ hash function looks like

Identify a key weakness of the hash table

Introduce strategies to ‘correct’ this weakness
## Insertion (Separate Chaining)

### Table:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>B+</td>
<td>2</td>
</tr>
<tr>
<td>Anna</td>
<td>A-</td>
<td>4</td>
</tr>
<tr>
<td>Alice</td>
<td>A+</td>
<td>4</td>
</tr>
<tr>
<td>Brett</td>
<td>A-</td>
<td>2</td>
</tr>
<tr>
<td>Greg</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>Sue</td>
<td>B</td>
<td>7</td>
</tr>
<tr>
<td>Ali</td>
<td>B+</td>
<td>4</td>
</tr>
<tr>
<td>Laura</td>
<td>A</td>
<td>7</td>
</tr>
<tr>
<td>Lily</td>
<td>B+</td>
<td>7</td>
</tr>
</tbody>
</table>

### Diagram:

- **Greg** (A) - 0
- **Brett** (A-) - 2
- **Betty** (B) - 7
- **Ali** (B+) - 4
- **Alice** (A+) - 4
- **Anna** (A-) - 2
- **Bob** (B+) - 2
- **Sue** (B) - 7
- **Laura** (A) - 7
- **Lily** (B+) - 7
Simple Uniform Hashing Assumption

Given table of size $m$, a simple uniform hash, $h$, implies

$$\forall k_1, k_2 \in U \text{ where } k_1 \neq k_2 , \ Pr(h[k_1] = h[k_2]) = \frac{1}{m}$$

**Uniform:** keys are equally likely to hash to any position

**Independent:** key hash values are independent of other keys
Separate Chaining Under SUHA

Under SUHA, a hash table of size $m$ and $n$ elements:

- find runs in: __________.
- insert runs in: __________.
- remove runs in: __________.
Open vs Closed Hashing

Addressing hash collisions depends on your storage structure.

- **Open Hashing**: store \( k,v \) pairs externally

- **Closed Hashing**: store \( k,v \) pairs in the hash table
Collision Handling: Probe-based Hashing

S = {1, 8, 15}

h(k) = k \% 7

|S| = n

|Array| = m
Collision Handling: Linear Probing

|S| = n
|Array| = m

$h(k) = k \% 7$

$h(k, i) = (k + i) \% 7$

Try $h(k) = (k + 0) \% 7$, if full...
Try $h(k) = (k + 1) \% 7$, if full...
Try $h(k) = (k + 2) \% 7$, if full...
Try ...

$S = \{ 16, 8, 4, 13, 29, 11, 22 \}$
Collision Handling: Linear Probing

\[ S = \{ 16, 8, 4, 13, 29, 11, 22 \} \]

\[ h(k, i) = (k + i) \% 7 \]

\[ |S| = n \]

\[ |Array| = m \]

\_find(29)

\_remove(16)
A Problem w/ Linear Probing

**Primary clustering:**

<table>
<thead>
<tr>
<th></th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1'</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1''</td>
</tr>
<tr>
<td>5</td>
<td>3'</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

**Remedy:**
Collision Handling: Quadratic Probing

$S = \{ 16, 8, 4, 13, 29, 11, 22 \}$

$|S| = n$

$h(k) = k \mod 7$

$|Array| = m$

$h(k, i) = (k + i*i) \mod 7$

Try $h(k) = (k + 0) \mod 7$, if full...

Try $h(k) = (k + 1*1) \mod 7$, if full...

Try $h(k) = (k + 2*2) \mod 7$, if full...

Try ...
A Problem w/ Quadratic Probing

**Secondary clustering:**

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0'</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0''</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0''''</td>
</tr>
</tbody>
</table>

**Description:**

**Remedy:**
Collision Handling: Double Hashing

S = \{ 16, 8, 4, 13, 29, 11, 22 \} \quad |S| = n
h_1(k) = k \mod 7 \quad |\text{Array}| = m
h_2(k) = 5 - (k \mod 5)

h(k, i) = (h_1(k) + i*h_2(k)) \mod 7

Try h(k) = (k + 0*h_2(k)) \mod 7, if full...
Try h(k) = (k + 1*h_2(k)) \mod 7, if full...
Try h(k) = (k + 2*h_2(k)) \mod 7, if full...
Try ...
Running Times

(Don’t memorize these equations, no need.)

(Expectation under SUHA)

Open Hashing:

insert: __________.

find/ remove: __________.

Closed Hashing:

insert: __________.

find/ remove: __________.
Running Times  
(Don’t memorize these equations, no need.)

The expected number of probes for find(key) under SUHA

**Linear Probing:**
- Successful: $\frac{1}{2}(1 + \frac{1}{1-\alpha})$
- Unsuccessful: $\frac{1}{2}(1 + \frac{1}{1-\alpha})^2$

**Double Hashing:**
- Successful: $\frac{1}{\alpha} \times \ln(\frac{1}{1-\alpha})$
- Unsuccessful: $\frac{1}{1-\alpha}$

**Separate Chaining:**
- Successful: $1 + \frac{\alpha}{2}$
- Unsuccessful: $1 + \alpha$

Instead, observe:

- As $\alpha$ increases:
- If $\alpha$ is constant:
Running Times

The expected number of probes for find(key) under SUHA

Linear Probing:
• Successful: $\frac{1}{2}(1 + \frac{1}{1-\alpha})$
• Unsuccessful: $\frac{1}{2}(1 + \frac{1}{1-\alpha})^2$

Double Hashing:
• Successful: $\frac{1}{\alpha} \times \ln(\frac{1}{1-\alpha})$
• Unsuccessful: $\frac{1}{1-\alpha}$
ReHashing

What if the array “fills”?
Which collision resolution strategy is better?

- Big Records:

- Structure Speed:

What structure do hash tables implement?

What constraint exists on hashing that doesn’t exist with BSTs?

Why talk about BSTs at all?
# Running Times

<table>
<thead>
<tr>
<th></th>
<th>Hash Table</th>
<th>AVL</th>
<th>Linked List</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Find</strong></td>
<td>Amortized:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Worst Case:</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Insert</strong></td>
<td>Amortized:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Worst Case:</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Storage Space</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bonus Slides
Hash Function (Division Method)

Hash of form: $h(k) = k \% m$

Pro:

Con:
Hash Function (Multiplication Method)

Hash of form: \( h(k) = \lfloor m(kA \mod 1) \rfloor, \ 0 \leq A \leq 1 \)

Pro:

Con:
Hash Function (Universal Hash Family)

Hash of form: $h_{ab}(k) = ((ak + b) \mod p) \mod m$, $a, b \in Z_p^*, Z_p$

$\forall k_1 \neq k_2$, $Pr_{a,b}(h_{ab}[k_1] = h_{ab}[k_2]) \leq \frac{1}{m}$

Pro:

Con: