Today’s announcements:

MP6 available, due 11/18, 11:59p. EC due 11/11, 11:59p

Party planning: Need to buy snacks... 8 total, 5 kinds to choose from \{__, __, __, __, __\}. How many different snack configurations can I offer?

* * | * | | * * * * | *

Hash functions:

• Computed in O(1) time.
• Deterministic.
• Satisfy the SUHA.
Hashing: summary and the plan for the day...

client code

```
dict<ktype, vtype> d;

ex: insert is d[k] = v;
```
Hash Functions

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>...</th>
<th>N-1</th>
</tr>
</thead>
</table>

Easy, if $|\text{KeySpace}| \sim N$

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>...</th>
<th>N-1</th>
</tr>
</thead>
</table>
Hashing Strings (an example)

Given: 8 character strings are easy to hash

The idea: Select 8 random positions from long strings and hash that substring.

A bunch of strings:

Lookyhere, Huck, being rich ain't going
No! Oh, good-licks; are you in real dead
Just as dead earnest as I'm sitting here
nto the gang if you ain't respectable, y
Can't let me in, Tom? Didn't you let me
Yes, but that's different. A robber is m
irate is -- as a general thing. In most
Now, Tom, hain't you always ben friendly
ut, would you, Tom? You wouldn't do that
Huck, I wouldn't want to, and I DON'T wa
ay? Why, they'd say, 'Mph! Tom Sawyer's
t!' They'd mean you, Huck. You wouldn't
uck was silent for some time, engaged in
Well, I'll go back to the widder for a m
can come to stand it, if you'll let me
All right, Huck, it's a whiz! Come along
Will you, Tom -- now will you? That's go
Hashing Strings (an example)

Given: 8 character strings are easy to hash

The idea: Select 8 random positions from long strings and hash that substring.

A bunch of strings:

http://en.wikipedia.org/wiki/Le%C5%9Bna_Grobla
http://en.wikipedia.org/wiki/Blow_the_Man_Down
http://en.wikipedia.org/wiki/Swen_K%C3%B6nig
http://en.wikipedia.org/wiki/2/7th_Cavalry_Commando_Regiment_(Australia)
http://en.wikipedia.org/wiki/Alice_High_School
http://en.wikipedia.org/wiki/RFA_Sir_Bedivere_(L3004)
http://en.wikipedia.org/wiki/Birthright_(band)
http://en.wikipedia.org/wiki/Vachon
http://en.wikipedia.org/wiki/McCarthy_%26_Stone
http://en.wikipedia.org/wiki/Newfoundland_Irish
http://en.wikipedia.org/wiki/Bona_Sijabat
Collision handling - Separate Chaining: (an example of open hashing)

\[ S = \{ 16, 8, 4, 13, 29, 11, 22 \} \quad |S| = n \quad h(k) = k \mod 7 \]
Collision Handling - Probe based hashing: (example of closed hashing)

$S = \{16, 8, 4, 13, 29, 11, 22\}$  \quad |S| = n  \quad h(k) = k \% 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...
Probe based hashing – 2 problems...

Removals:

<table>
<thead>
<tr>
<th>0</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>13</td>
</tr>
</tbody>
</table>

Clustering:

http://groups.engin.umd.umich.edu/CIS/course.des/cis350/hashing/WEB/HashApplet.htm
Probe based hashing: (double hashing)

\[ S = \{16, 8, 4, 13, 29, 11, 22\} \quad |S| = n \quad H(k,i) = h_1(k) + ih_2(k) \]

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...
Hash table performance: expected # of probes for Find(key) under SUHA

Linear probing -
  successful: \(\frac{1}{2} \left( 1 + \frac{1}{1-\alpha} \right)\)
  unsuccessful: \(\frac{1}{2} \left( 1 + \frac{1}{1-\alpha} \right)^2\)

Double hashing -
  successful: \(\frac{1}{\alpha} \ln \frac{1}{1-\alpha}\)
  unsuccessful: \(\frac{1}{1-\alpha}\)

Separate chaining -
  successful: \(1 + \frac{\alpha}{2}\)
  unsuccessful: \(1 + \alpha\)

Do not memorize these!

Observe:
- As \(\alpha\) increases, running times increase…
- If \(\alpha\) is held constant then running times are constant…

Which is better?
- Big records –
- Structure speed –
Hash table performance: expected # of probes for Find(key) under SUHA

Linear probing -

successful: \( \frac{1}{2} \left( 1 + \frac{1}{1-\alpha} \right) \)
unsuccessful: \( \frac{1}{2} \left( 1 + \frac{1}{1-\alpha} \right)^2 \)

Double hashing -

successful: \( \frac{1}{\alpha / \ln \left( 1 - \alpha \right)} \)
unsuccessful: \( \frac{1}{1 - \alpha} \)
What’s left???

Running times of dictionary algorithms are a function of load factor, _____ , but we hoped for __________ running times.

hmmmm....

What structures do hash tables replace for us?