Today’s announcements:

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

Hashing - using “hash tables” to implement _________

Suppose we have the following info…

<table>
<thead>
<tr>
<th>Locker Number</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>Jay Hathaway</td>
</tr>
<tr>
<td>92</td>
<td>Linda Stencel</td>
</tr>
<tr>
<td>330</td>
<td>Bonnie Cook</td>
</tr>
<tr>
<td>46</td>
<td>Rick Brown</td>
</tr>
<tr>
<td>124</td>
<td>Kim Petersen</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

…and we want to be able to retrieve a name, given a locker number.

Now suppose our keys are not so nicely described…

Course Number -> Schedule info
Color -> BMP
Vertex -> Set of incident edges
Flight number -> arrival information
URL -> html page
dice roll -> payoff amt

dice roll -> payoff amt
Some general vocabulary

A *dictionary* is a structure supporting the following:

```c
void insert(kType & k, dType & d)
void remove(kType & k)
dType find(kType & k)
```

An *associative array* is a dictionary with a particular interface—overloads the [] operator for insert and find:

```c
myDictionary["Miguel"] = 22;
dType d = myDictionary["Miguel"];```
Hashing:

(defn) Keyspace — a (mathematical) description of the keys for a set of data.

Goal: use a function to map the keyspace into a small set of integers.

What’s fuzzy about this goal?
Basic Idea: we seek a mapping, $h(k)$
A perfect hash function:

(Ann, black cat)
(Ben, HP)
(Cory, spy)
(David, bball player)
(Ellen, butterfly)
(Finn, cereal killer)
(Gus, ghost)
(Harmony, bee)

A contrived example:

*these keys have a fabulous hash fn.*

a. each key hashes to a different int
b. collection of keys hash to a seq of ints
A perfect hash function?

Roll 5 dice:

Is this a good hash function?
A Hash Table consists of:

- An array
- A hash function
- A collision resolution strategy

KeySpace

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

keys used:

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

Easy, if $|\text{KeySpace}| \sim N$
Hash Functions:
• Consist of 2 parts:
  – A Hash: Function mapping a key to an integer \( i \)
  – A compression: function mapping \( i \) into the array cells 0 to N-1.
• Choosing a hash function is tricky...
  – Don’t create your own (yet)
  – Smart people can produce poor hash functions (what’s a bad hash function?)
    • Knuth’s multiplicative hash in “the Art of Computer Programming”

• Characteristics:
  • Computed in _________ time.
  • Deterministic.
  • Satisfy the SUHA.