CS 225
Data Structures
Oct. 27 – Hashing
Hashing

Goals:
We want to define a **keyspace**, a (mathematical) description of the keys for a set of data.

...use a function to map the **keyspace** into a small set of integers.
Hashing
A Hash Table based Dictionary

Client Code:

```
1 Dictionary<KeyType, ValueType> d;
2 d[k] = v;
```

A **Hash Table** consists of three things:

1.

2.

3.
A Perfect Hash Function

\[(\text{Angrave, CS 241}) \quad \text{(Beckman, CS 421)} \quad \text{(Cunningham, CS 210)} \quad \text{(Davis, CS 101)} \quad \text{(Evans, CS 126)} \quad \text{(Fagen-Ulmschneider, CS 225)} \quad \text{(Gunter, CS 422)} \quad \text{(Herman, CS 233)}\]

Hash function

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A Perfect Hash Function

Keyspace:
Rolling 5 dice!

Hash function

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></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></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>
Hash Function

Our hash function consists of two parts:

• A hash:

• A compression:

Choosing a good hash function is tricky…

• Don’t create your own (yet*)

• Very smart people have created very bad hash functions
Hash Function

Characteristics of a good hash function:

1. Computation Time:

2. Deterministic:

3. Satisfy the SUHA:
General Purpose Hash Function

Easy to create if: $|\text{KeySpace}| \sim N$

Difficult to Create:
Hash Function

*Given:* Easy to create a hash function of strings of length 8.
Idea: Map 40 character things to length 8:

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, ‘and what is the use of a book,’ thought Alice ‘without pictures or conversations?’ So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her. There was nothing so very remarkable in that; nor did Alice think it so very much out of the way to hear the Rabbit say to itself, ‘Oh dear! Oh dear! I shall be late!’ (when she thought it over afterwards, it occurred to her that she ought to ha
Idea: Map 40 character things to length 8:

https://en.wikipedia.org/wiki/Main_Page
https://en.wikipedia.org/wiki/2017_Austr
https://en.wikipedia.org/wiki/19th_Natio
https://en.wikipedia.org/wiki/Japanese_g
Hash Function

In CS 225, we focus on general purpose hash functions.

Other hash functions exist with different properties (eg: cryptographic hash functions)
A Hash Table based Dictionary

Client Code:

```
1 Dictionary<KeyType, ValueType> d;
2 d[k] = v;
```

A Hash Table consists of three things:
1. 
2. 
3. 
Exam 8 (programming exam) is coming up Monday:
More Info: https://courses.engr.illinois.edu/cs225/fa2017/exams/

MP5: Available now!
Extra Credit +7 deadline: Monday, Oct. 30

Lab: lab_btree
Due Sunday, Oct. 29 at 11:59pm

POTD
Every Monday-Friday – Worth +1 Extra Credit /problem (up to +40 total)