Every hash table contains three pieces:
1. A **hash function**, $f(k)$. The hash function transforms a key from the keyspace into a small integer.
2. An **array**.
3. A **mystery** third element.

## A Perfect Hash Function

(Angrove, CS 241)
(Beckman, CS 421)
(Cunningham, CS 210)
(Davis, CS 101)
(Evans, CS 126)
(Fagen-Ulmschneider, CS 225)
(Gunter, CS 422)
(Herman, CS 233)

...characteristics of this function?

## A Second 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>

...characteristics of this function?

All hash functions will consist of two parts:
- A **hash**:
- A **compression**:

Characteristics of a good hash function:
1. Computation Time:
2. Deterministic:
3. SUHA:

Towards a general-purpose hashing function:
It is easy to create a general-purpose hashing function when the keyspace is proportional to the table size:
- **Ex:** Professors at CS@Illinois
- **Ex:** Anything you can reason about every possible value

It is difficult to create a general-purpose hashing function when the keyspace is large:
My 40-character strategy:

Alice was beginning to get very tired of
sitting by her sister on the bank, and
of having nothing to do: once or twice s
he had peeped into the book her sister w
as reading, but it had no pictures or co
versations in it, ‘and what is the use
of a book,’ thought Alice ‘without pictu
res or conversations?’ So she was consi
dering in her own mind (as well as she c
ould, for the hot day made her feel very
sleepy and stupid), whether the pleasur
es of making a daisy-chain would be worth
the trouble of getting up and picking t
he 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 it
self, ‘Oh dear! Oh dear! I shall be late
!’ (when she thought it over afterwards,
it occurred to her that she ought to ha
...what is a naïve hashing strategy for this input?

...characteristics of this function?

What is an example of bad input data on this hash function?

---

Reflections on Hashing
We are starting the study of general-purpose hash functions. There
are many other types of hashes for specific uses (ex: cryptographic
hash functions).

Even if we build a good hash function, it is not perfect. What happens
when the function isn’t always a bijection?

Collision Handling Strategy #1: Separate Chaining

Example: \( S = \{ 16, 8, 4, 13, 29, 11, 22 \} \), \(|S| = n\)
\( h(k) = k \% 7, \quad |Array| = N \)

<table>
<thead>
<tr>
<th>[0]</th>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
<th>[4]</th>
<th>[5]</th>
<th>[6]</th>
<th>[7]</th>
</tr>
</thead>
</table>

Load Factor:

Running time of Separate Chaining:

<table>
<thead>
<tr>
<th></th>
<th>Worst Case</th>
<th>SUHA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remove/Find</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CS 225 – Things To Be Doing:

1. Programming Exam B is ongoing
2. MP5 has been released; EC+7 deadline is Monday back from break
3. lab_btree released this week; due Tuesday, March 27th at 11:59pm
   (That’s the Tuesday evening after spring break)
4. Daily POTDs are ongoing!