**Searching A List**

- If there's no order to the list (like the deck of cards)...  
  - best you can do is start at the beginning  
  - This is called sequential or linear search

- Binary search is a simple, common sense way to search through an *ordered* set of items.  
  - Questions, often referred to as queries or probes, are asked to *find if the desired item is smaller or larger.*  
  - If the question is chosen from the middle of the sequence, \( \frac{1}{2} \) the possibilities are eliminated with each answer.  
  - It's a bit like 20 questions, but MUCH more specific.

**How Good is a Particular Algorithm?**

- You might think we can't answer this question without programming a computer and trying it.

- Amazingly, it is possible to make very good comparisons between algorithms without programming them!

- Basic idea: estimate the number of “steps” each algorithm needs to solve the problems.

- This gives us an abstract, mathematical way to compare the speed of different algorithms

**Algorithm vs. Program**

- Remember that an algorithm is an abstraction.

- We can apply it, at least mentally, to a variety of situations, even without a computer

- A program incorporates all the details needed for a computer to perform the algorithm

- A program for search will encode the algorithm for a specific situation, in a specific language, with specific assumptions
Battle of the Algorithms

- Binary Search: Each question allows you to throw out half of the unexamined items (throw half of the phone book away each time)

- Linear Search: Each question lets you tear out only one page, or throw out one card

Do The Math for Searching 200 Items

<table>
<thead>
<tr>
<th></th>
<th>linear</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>step 0</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>step 1</td>
<td>199</td>
<td>100</td>
</tr>
<tr>
<td>step 2</td>
<td>198</td>
<td>50</td>
</tr>
<tr>
<td>step 3</td>
<td>see where it’s going?</td>
<td>197</td>
</tr>
</tbody>
</table>

Bottom Line

- It can be shown mathematically that when a sorted list of N items is to be searched.....

- Linear search needs on average about N/2 steps

- Binary search needs on average about Log₂N steps
  - ☐ No, you don’t have to be able to compute Log₂N!
  - ☐ Just remember this, the bigger N is, the bigger the improvement.
  - ☐ So, the larger the number of things to be searched, the faster binary search becomes

Trade-Offs

- If we know algorithm A has a better formula than algorithm B:

  Would we ever still want to use algorithm B??
**Searching a small set of things: 20**

<table>
<thead>
<tr>
<th></th>
<th>linear</th>
<th>binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>step 0</td>
<td>20 remaining</td>
<td>20</td>
</tr>
<tr>
<td>step 1</td>
<td>19</td>
<td>10</td>
</tr>
<tr>
<td>step 2</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>step 3</td>
<td>17</td>
<td>3</td>
</tr>
</tbody>
</table>

Could you tell the difference in time if a computer does the search?