Binary Search -- A Fundamental Algorithm

Binary search is a clever, though common sense way to search an ordered set of items. Queries are made, called *probes*, asking whether the desired item is smaller or larger. If the probe is chosen in the middle of the sequence, 1/2 of the possibilities must be eliminated with any answer. Now the details...

Reminder … Algorithm vs Program

- The process just described on the title slide -- suitably embellished -- is the binary search *algorithm* … the idea given abstractly
- A *program* for binary search -- your goal -- will encode the algorithm for a specific situation, in a specific language, with specific assumptions

Today’s Topics: Analyze the binary search algorithm  
Review the Day Finder application  
Reason through the logic of using binary search in the Day Finder context
An Example

- Use binary search to locate a letter in the alphabet

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

After M?
An Example

- Use binary search to locate a letter in the alphabet

\[
\begin{array}{cccccccccccccccccccccccccccc}
\end{array}
\]

After M? N

\[
\begin{array}{cccccccccccccccccccccccccccc}
\end{array}
\]

After G?
An Example

- Use binary search to locate a letter in the alphabet
  
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  
  After M? N
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  
  After G? Y
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  
  After J? Y
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  
  After L?
An Example

- Use binary search to locate a letter in the alphabet
  
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  After M? N
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  After G? Y
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  After J? Y
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  After L? N
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  After K?

The letter is L
Analyzing Properties Of Probe

- Though “before” and “after” questions could be used, adopting one and staying with it simplifies the effort.
- Using “after” questions … the probe should be:
  - For odd-length ranges, the middle item
  - For even-length ranges, last item in first half
- Stop when there is only one item left.

We always eliminate about 1/2 of the items, with best case 1/2+1 and worst case 1/2-1.

Algorithm Analysis

- Understanding the problem …
  + Inputs: The end points, \((lo, hi)\), of an ordered sequence
    Answers to a series of questions
  + Outputs: A selected item
  + How the inputs are transformed to the outputs:
    The questions of the series have the form
    “Is the desired item after item \(x\)?”
    so that the \(x^{th}\) item is chosen to be midway in the interval
    If the reply is yes, the new interval \((next\ after\ x, hi)\)
    If the reply is no, the new interval is \((lo, x)\)
    The output is the item when the interval contains only a single item.
Example -- End Points and Probe

- Find a number in the range 1 to 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
After __ ?

Example -- End Points and Probe

- Find a number in the range 1 to 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
After 10 Y
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
After __ ?
Example -- End Points and Probe

Find a number in the range 1 to 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
After 10
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
After 15 N
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
After ___ ?
Example -- End Points and Probe

❖ Find a number in the range 1 to 20

<table>
<thead>
<tr>
<th>After 10</th>
<th>After 15</th>
<th>After 13</th>
<th>After 14 Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20</td>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20</td>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20</td>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20</td>
</tr>
</tbody>
</table>

The number is 15

Analysis of End Points

❖ Find a number in the range 1 to 20

<table>
<thead>
<tr>
<th>Probe = 10, Range Start = 1, Range End = 20 [1,20]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20</td>
</tr>
</tbody>
</table>

One end of the range is changed after each answer
The new end-point is either the probe or probe+1
Choice of probe or probe+1 depends on N or Y ans
Analysis of Probe

- Find a number in the range 1 to 20. The range is inclusive.

```
1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
```

<table>
<thead>
<tr>
<th>Probe</th>
<th>[Start, End]</th>
<th>(Start + End) / 2</th>
<th>(Start + End) \div 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>[1, 20]</td>
<td>10.5</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>[11, 20]</td>
<td>15.5</td>
<td>15</td>
</tr>
<tr>
<td>13</td>
<td>[11, 15]</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>[14, 15]</td>
<td>14.5</td>
<td>14</td>
</tr>
</tbody>
</table>

```
[15, 15]
```

Probe is found by adding end points and dividing by 2.

Integer divide (\) drops fractional digits, giving the right probe value.

---

Month Extension Technique

- The “complicating” problem with searching for a birthday in a sign, is that the signs span parts of two months.
- Not to worry … logically extend the starting month.

```
Day Find
```

Enter Your Sign, Please

<table>
<thead>
<tr>
<th>Aries</th>
<th>Leo</th>
<th>Sagittarius</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taurus</td>
<td>Virgo</td>
<td>Capricorn</td>
</tr>
<tr>
<td>Gemini</td>
<td>Libra</td>
<td>Aquarius</td>
</tr>
<tr>
<td>Cancer</td>
<td>Scorpio</td>
<td>Pisces</td>
</tr>
</tbody>
</table>

You were born between June 21 and July 22.

Were you born after July 6?

<table>
<thead>
<tr>
<th>Y</th>
<th>N</th>
</tr>
</thead>
</table>

Days in June: 30
Day in July: 22
Day in exJune: 52

The interval to be searched is 21 through 52.
**FIT 100** Visualize The Extended Month

Think of the Zodiac sign as starting at its start day (21) and extending to the end day (22) + number of days in loMo (30)

- Any date that is less than or equal to the last day of loMo is in the loMo
- Any date that is more than the last day of loMo is in hiMo, and is too large by the number of days in loMo

<table>
<thead>
<tr>
<th>June</th>
<th>July</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
</tr>
<tr>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>...</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Cancer</td>
<td></td>
</tr>
</tbody>
</table>

**FIT 100** Transforming Probe To A Date

- The interval is: [21-52]
- The probe for the interval is: \((21+52)/2 = 36\)
- What day is June 36?
  - The month is the next month, July
  - The day is 36 reduced by the number of days in June, 36-30 = 6
Overall Data Flow ...

- What are the new variables needed
  - loEnd, hiEnd, midPt and lastDay
- Where do the initial values come from?
  - After the Zodiac computation, loEnd and hiEnd can be set
  - Once interval set, probe (midPt) can be computed
- How are these values updated?
  - In the yes and no button event handlers
  - In the case of “yes,” which end moves?
    + loEnd moves up past the midPt
  - In the case of “no”, which end moves?
    + hiEnd moves down to the midPt
- When does the questioning terminate?
  - When the end points are equal

Structure Of Solution

Declarations

Private Sub optAri
Private Sub optTau
Private Sub optGem
Private Sub optCan
Private Sub optLeo
Private Sub optVir
Private Sub optLib
Private Sub optSco
Private Sub optSag
Private Sub optCap
Private Sub optAqu
Private Sub optPis
Private Sub cmdOK
Private Sub cmd Yes
Private Sub cmd No

Inherit from Zodiac

- initialize, make first guess
- revise interval, make guess
- revise interval, make guess

© University of Washington, 2001