Binary Search -- A Fundamental Algorithm

FIT 100

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
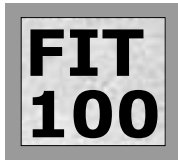
Algorithm Analysis

❖ Understanding the problem …

➕ Inputs: The end points, (*lo*, *hi*), of an ordered sequence

Answers to an series of questions

➕ Outputs: A selected item

➕ How the inputs are transformed to the outputs:

A series of questions is posed of the form

"Is the desired item after item x?"

so that the xth 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
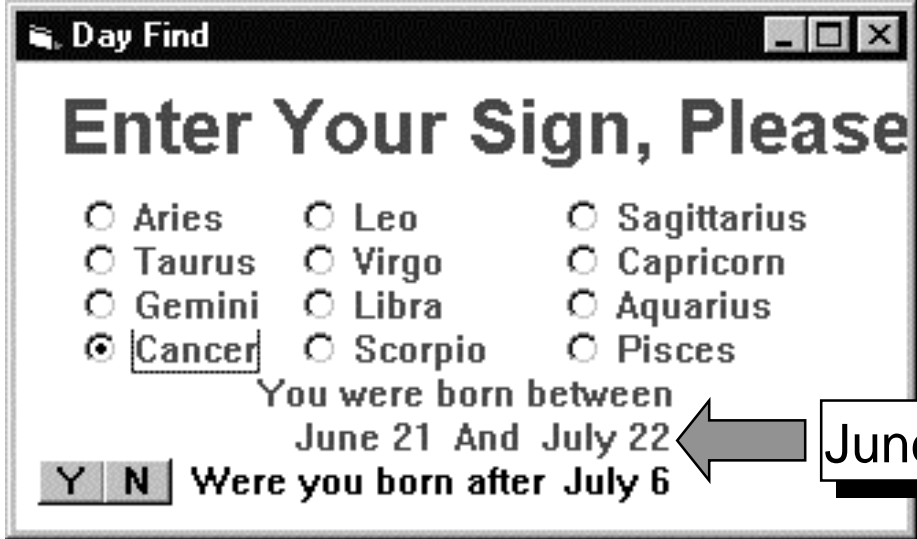
Analyzing Properties Of Solution

- ❖ End points … inclusive
- ❖ Before/after questions … stay with one form
- ❖ Probing odd-length and even-length intervals (\)
- ❖ New interval's endpoints … one is kept, one changes
- ❖ Termination … when is it over?

```
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? 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
The letter is L
```

Guessing Days In A Sign

❖ 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

○ Aries      ○ Leo        ○ Sagittarius
○ Taurus     ○ Virgo      ○ Capricorn
○ Gemini     ○ Libra      ○ Aquarius
⊙ Cancer     ○ Scorpio    ○ Pisces
        You were born between
        June 21  And  July 22
Y  N  Were you born after  July 6
```

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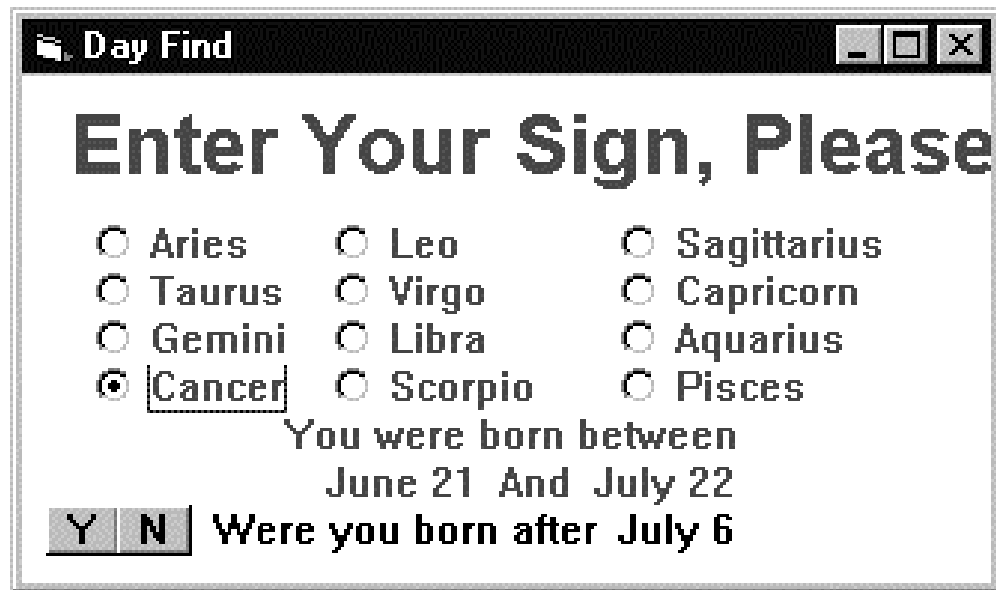
June 52

The interval to be searched is 21 through 52

Transforming Probe To A Date

- ❖ The size of the interval is: (52 - 21) = 31
- ❖ The midpoint of the interval is: 31 \ 2   = 15
- ❖ The probe, low end + midpoint: 21 + 15   = 36
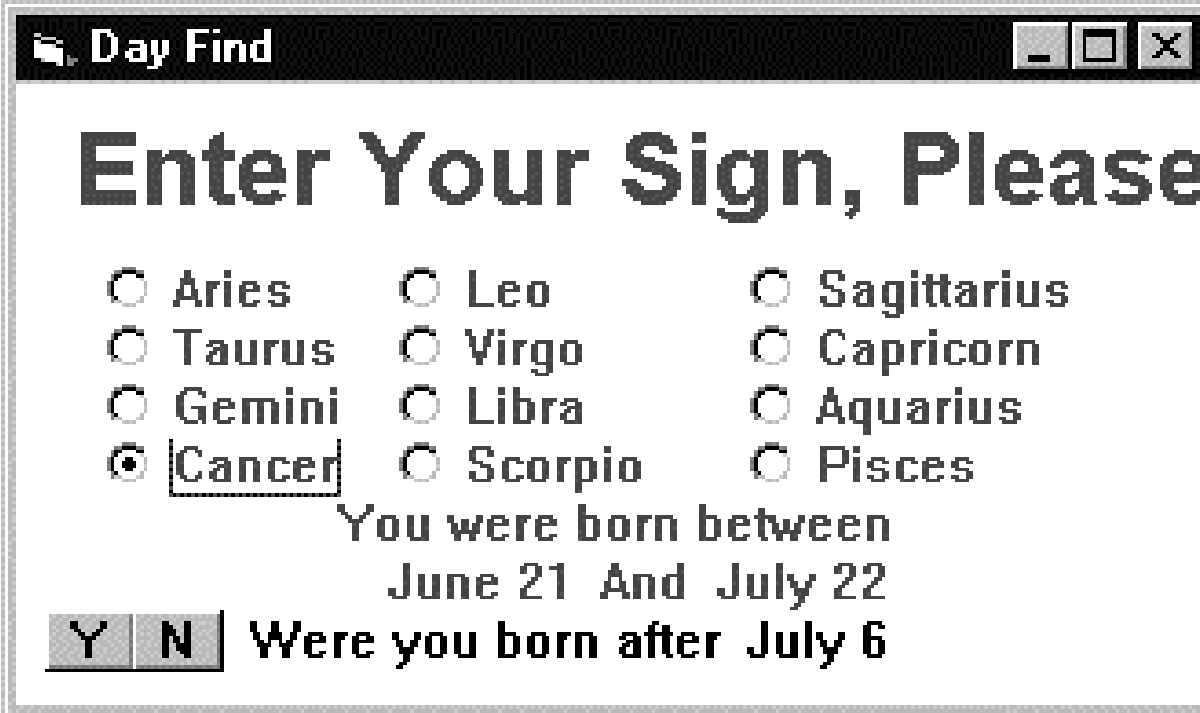- ❖ What day is June 36?

Day Find

Enter Your Sign, Please

○ Aries   ○ Leo   ○ Sagittarius
○ Taurus   ○ Virgo   ○ Capricorn
○ Gemini   ○ Libra   ○ Aquarius
◉ Cancer   ○ Scorpio   ○ Pisces
You were born between
June 21 And July 22
Y N Were you born after July 6

Guess? What?

❖ What information is needed by the guess procedure?
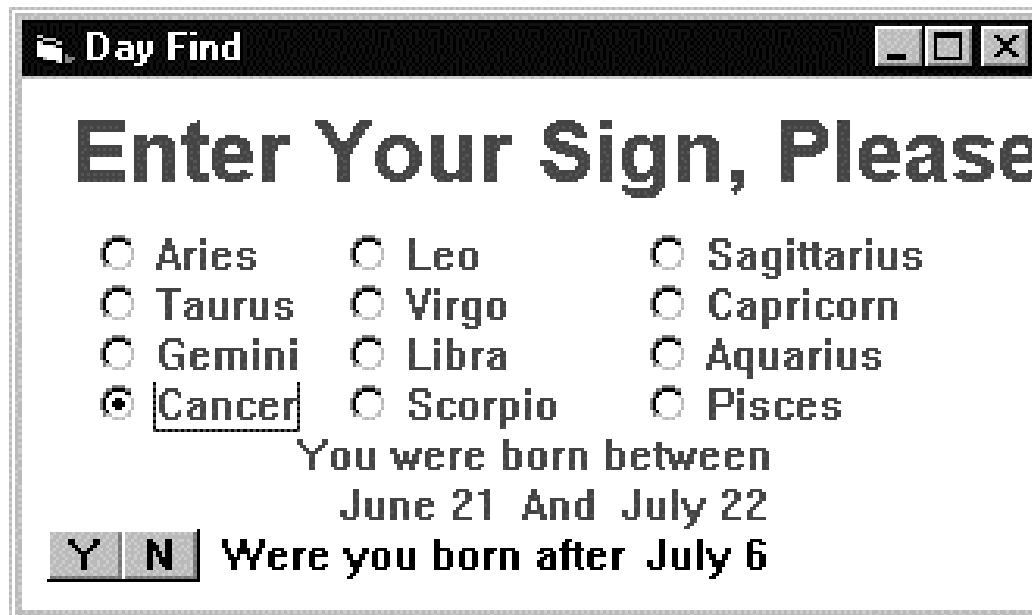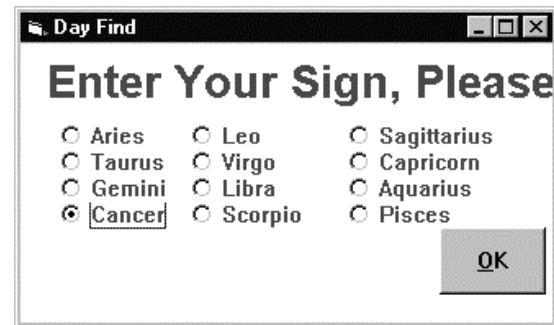

Day Find

Enter Your Sign, Please

○ Aries   ○ Leo   ○ Sagittarius
○ Taurus   ○ Virgo   ○ Capricorn
○ Gemini   ○ Libra   ○ Aquarius
◉ Cancer   ○ Scorpio   ○ Pisces

You were born between
June 21 And July 22

Y N Were you born after July 6

Using Binary Search In Day Finder

❖ Inherit the initial configuration from Zodiac

❖ The guess Procedure asks one probe at a time ...

❖ When is Guess called?

Overall Data Flow ...

❖ Where do the initial values come from?

 ❏ After the Zodiac computation, loEnd and hiEnd can be set

❖ When are these values used?

 ❏ In the guess procedure to compute the midPt for the guess

❖ How are these values updated?

 ❏ In the yes and no button event handlers

 ❏ In the case of "yes," which end moves?

   ✦ loEnd = midPt + 1

 ❏ In the case of "no", which end moves?

   ✦ hiEnd = midPt

 ❏ Why are the two setting not "opposite" one another?

❖ When the does the questioning terminate?

 ❏ When the end points are equal

Structure Of Solution

Declarations   *-- additional variable declarations*
Private Sub optAri
Private Sub optTau
Private Sub optGem
Private Sub optCan
Private Sub optLeo
Private Sub optVir   *Inherit from Zodiac*
Private Sub optLib
Private Sub optSco
Private Sub optSag
Private Sub optCap
Private Sub optAqu
Private Sub optPis
Private Sub cmdOK   *-- initialize, make first guess*
Private Sub cmd Yes   *-- revise interval, make guess*
Private Sub cmd No   *-- revise interval, make guess*
Private Sub guess   *-- formulate guess*