CS 106
Introduction to Computer Science I

03 / 04 / 2020

Instructor: Michael Eckmann

Today's Topics

- Questions? Comments?
- Binary search
- Analysis of search methods and bubblesort
- Start Object Oriented programming
- Exam 1 corrections can get up to 50% of points off. Submit by Friday class. Can use notes and/or meet with me.
- Please write all corrections on new sheets of paper. Please do not write anything on the exam itself. Please submit both the exam and your sheets of paper with corrections

Binary Search

- We could compare to the middle element of the array.
- If it is equal to the middle element, we're done.
- If it is less than the middle element, where would we now concentrate our search?
- If it is greater than the middle element, where would we now concentrate our search?

Binary Search

- Any idea how we might write code to implement this algorithm?

Binary Search

- Any idea how we might write code to implement this algorithm?

- Let's discuss some ideas before we get right to the code.
  - What parameters might our method have?
  - What element to compare to first?
    - How do we calculate that index?
  - How do we determine what part of the array to now do a search?
- Let's take a look at an implementation and do an example call.

Binary Search

```
// method to perform binary search of an array
public static int binarySearch( int array2[], int key )
{
   int low = 0;                 // low element subscript
   int high = array2.length - 1;  // high element subscript
   int middle;                  // middle element subscript

   // loop until low subscript is greater than high subscript
   while ( low <= high )
   {
      // determine middle element subscript
      middle = ( low + high ) / 2;

      // if key matches middle element, return middle location
      if ( key == array2[ middle ] )
         return middle;

      // if key less than middle element, set new high element
      else if ( key < array2[ middle ] )
         high = middle - 1;

      // key greater than middle element, set new low element
      else
         low = middle + 1;

   } // end while loop

   return -1;   // key not found

}  // end method binarySearch
```

Searching arrays (Binary search)

- Let's analyze the binary search.
- To simplify the discussion, we can count the 2 comparisons in the if/else/if/else together to be 1 comparison.
- How long (that is, how many comparisons) does it take to find the value?
  - What's the minimum number of comparisons it would take?
  - What's the maximum number of comparisons it would take?

Searching arrays

- n is the size of the array (the number of elements)
- Linear search
  - Worst case when not found or found at last slot
    - makes n compares
  - Best case when found at first slot
    - makes 1 compare
- Binary search
  - Worst case when not found
    - makes lg n compares (lg n is log base 2)
  - Best case when found in middle slot
    - makes 1 compare

Analyze bubblesort

- n is the size of the array (the number of elements)
- Outer loop iterates exactly n-1 times
  - Each time the inner loop starts it iterates a different number of times
    - n-1, n-2, n-3, …, 1
    - Add these up

Object orientation

- Object orientation is a programming paradigm that views the pieces of large programs as
  - objects with “attributes” and “behaviors”.
- Java is an object-oriented language. C++ is another. Other languages with which you may be familiar, like C, Pascal, and Fortran, are not object-oriented. These are what are called procedural languages.
- So far in this course, we have used Java in a very procedural way and did not use it to its potential as an object-oriented language.

Object orientation

- An example of an object.
- If we wanted to write a program that worked with rectangles:
  - The *attributes* of a rectangle include its length and width.
  - Other *attributes* might be its color, ...
  - *Behaviors* of the rectangle could include: changing its size, computing its area, etc...

Object orientation

- So, in our example, a rectangle would have as its data:
  - length
  - width
  - color
- As its methods we would have things like:
  - setLength
  - setWidth
  - setColor
  - calculateArea