CS 2420  
Search Project

## Purpose

The purpose of this lab is to demonstrate the speed difference between recursive and iterative binary search and sequential search. This difference is not shown if the array size is too small. Therefore you should make your arrays as large as possible, and still have 2 significant digits of output.  
Binary search is much faster than iterative search. Therefore you may need to put binary search into a loop and divide the result by the number of iterations. You should also search for a different number each time, to give fair results to each search.

## Tasks

* Create a Search Class
  + Member variables
    - Dynamic array of ints
    - Size of the array
    - Whatever else you need
  + Constructor(array size)
  + Destructor
  + Overloaded << operator
    - friend ostream& operator<< (ostream& out, const Search& s)
    - this should display the elements in the array.
      * useful for debugging.
    - Plus I just want you to have practice programming this operator.
  + Sequential Search
  + Recursive Binary Search
  + Iterative Binary Search
  + A function to set the seed: just call srand(num)
  + Function to initialize the array with random numbers roughly the same size as the array
    - Init\_array();
    - So if the array was size 20 the random numbers should be between 1-60
  + Function to initialize the array with sorted random numbers
    - Init\_sorted\_array
    - Don’t use a sort() function here. Rather, when initializing, insert a number a bit larger than the previous number.
      * Array [0] = rand () %5
      * Array[i+1] = Array[i] + rand () %5 etc.
* Create a **non-interactive** driver that runs each of these searches and times and displays the outputs.
  + You will need a large array size: say 1000 - 10,000,000
  + You may need to time a loop of searching many times say 100-100,000 or so
    - Your results should be at least 2 significant digits
    - Your loop size should be different for your binary search vs. your sequential search
    - Search for a new random item each time through the loop.
    - **Do not do any printing inside the loop**.
      * This will give you incorrect results.
  + Display the output to do **one search of array size: ‘size’ in micro-seconds** for each of the different searches
    - Please make the array size the same for all searches and use larger or smaller loops to give you the significant digits required.
    - Do not make the driver interactive, rather just display the results.
    - You may need to do an appropriate division or multiplication (or both).
    - No other output is required.

#### Random Numbers

* Random Number generation in C/C++ is defined in the <random> header file.
* Use void “srand(long)” to set the random number seed. The seed can be any value, but should be the same each time (so you can duplicate your results).
* Use “int rand()” to generate a random number from 0 to MAXINT.
* Use the % (mod) operator to shrink the random numbers to something more reasonable – like 0-10000 (depending on your array size).

#### Timing

* Timing in C/C++ is defined in the <ctime> header file
* Elapsed time is measured in the clock\_t type.
* Elapsed time to this point in clock ticks is returned by ctime()
* The number of clock ticks per second is defined by CLOCKS\_PER\_SEC
* One clock tick is way too slow to measure one search, even in a large array, therefore you will need to put these searches inside of a large loop in order to get any data other than zero.
* Sample Code:

clock\_t start = clock();

// some code to time here

clock\_t end = clock();

clock\_t diff = end - start;

#### Turn in

* Solution or Project.
* Screen Shot.
  + Your computer might be faster or slower than the graders.