CSCI 115 Lab Assignment 3
DUE: 2/15/15 (Sunday) at 11:59 P.M. on Mulan

Textbook Exercises

For the first part of the lab, you need to complete the following exercises from chapter 3 of the textbook:

3.3, 3.4, 3.5, 3.6, 3.11, 3.12

Exercises in red should be completed during the lab period. Make sure to work on these exercises first!

NOTE: For 3.11, you do NOT have to justify your answers using limits. You just need to determine which relationship is correct for each pair.

Textbook Project

For the second part of the lab, you need to complete the following project from chapter 3 of the textbook:

Project 3.2

NOTE 1: For each N = 10^i (i.e, N = 10^1, N = 10^2, …, N= 10^M), you need to perform 1,000 searches for a random number between 1 and N for both the sequential and binary search programs. As i gets larger and larger (and approaches M, the maximum power of ten that your machine can handle), you may do only 100 searches, or even only 10 searches if it takes too long to perform 1,000 searches.

NOTE 2: In addition to your program, you should also submit a graph. The X-axis of the graph contains the size of the input (N = 10^1, N = 10^2, …, N = 10^M), and the Y-axis contains the corresponding amount of time (in milliseconds) it takes for the program to run for the size of the input. You should have the time for the sequential search program and the binary search program on the same graph.

NOTE 3: Do NOT run your tests on Mulan! Mulan is set-up with timesharing, so if several students are simultaneously running tests on Mulan, the results will indicate that your programs are slower than they really are. Make sure to perform your tests on one of the machines in the lab, or on your own machine.

You will also need the following libraries:

stdlib.h (for rand and srand)
time.h (for time and difftime)

You should use srand to seed the random number generator (with a call to time(NULL) as argument), use rand to choose random numbers, and use time and difftime to compute the time it takes for the operations.

We are interested in the average time that it takes to do a single linear search vs. a single binary search, so you will be doing your many repetitions of these searches with random keys and dividing the total time by the number of searches you did to get the average time. You should produce a table with a row for each problem size (10, 100, 1000, ...), and a column for linear search and a column for binary search, where you give the average time for each search. Include a graph of the numbers in your table with the problem size on the x-axis, the time on the y-axis, and curves for both linear and binary search. At what value of x does binary search get faster than linear search?