CSCI 220 -- PA 4

Recursion, Analysis Tools, and Experimental Analysis

Feel free to discuss and help each other out but does not imply that you can give away your code or your answers! Make sure to read all instructions before attempting this lab. You cannot work with a lab partner for this assignment.

You must use an appropriate provided template from Canvas or my website (zeus.mtsac.edu/~tvo) and output "Author: Your Name(s)" for all your programs. If you are modifying an existing program, use "Modified by: Your Name(s)".

Review recursion and how to analyze run times with big-Oh notation.

Exercise 1: Implement the linear Fibonacci version based on pseudocode on page 146 of the C++ textbook.

Exercise 2: Perform max sum problem below or P-4.1 from C++ textbook as an alternate problem. Run your program with the 10 values below as one test case to confirm it is working correctly and output results to the screen. In addition, run your program and output only running times in ms for 100, 1000, 10000, and 10000 elements (generate random values between -10000 and 10000). What is the running time for your solution using Big-Oh notation? Do the collected running times match the expected running times in Big-Oh notation? Explain.

Max Sum Problem: Given the integer array x[n], compute the maximum sum found in any contiguous sub-array. If the input array is 31 -41 59 26 -53 58 97 -93 -23 84, then the program returns the sum of x[2..6] which is 187.

Question 1: What are limitations with experimental studies of running times?

Question 2: Perform R-4.22 in C++ book; it is best to set up a table showing approximate values for requested run times.

Extra Credit: Implement the remaining problem above (max sum problem or P-4.1).

Fill out and turn in the PA submission file for this assignment (save as PDF format).