

Lab 2
CSCI232
Assignment due 9/18, at 11:30pm

Purpose: To analyze algorithms

1. Programming: Starting with the ThreeSum program that we used in class, create a four sum program. In the comments – tell me the algorithm complexity (Big O) of this program. Create a main program that generates 4000 random numbers (1-8000) and produces the number of 4-number sets that equal 0. Use the Stopwatch class to time execution of the count method. Print out the running time.
2. Do Not Program: Using the ThreeSumFast program as a reference, tell me how you would change the code (in words, not code) to become FourSumFast.
3. Questions:
 - a. Give me 5 different classes of time complexities and give me an example of each.
 - b. Give me a code snippet showing a loop of constant time complexity.
 - c. If an algorithm executes $O(\log n)$ -time computation for each entry of an array storing n elements, what is the Big O for storing the entire array?
 - d. Using an n of 5, show that time complexity of the recursive Fibonacci algorithm is $O(2^n)$ – write it out.
 - e. Using the running time and Big O from your programming assignment, predict what the running time of 8000 items would be.
 - f. Give me the Big O of the algorithms with the following worst case runtimes ($T(N)$):
 - i. $T(N)=3N^2 + 10N + 17$
 - ii. $T(N)=N + 9999$
 - iii. $T(N)=734N$

Due: Wednesday, 9/18 at 11:30pm

Note: work with a lab partner – only 1 of you will submit your answers to brightspace in the lab2 folder. In each of your files that you submit, put lab partners: <your name and partner's name> at the top of the file (along with your name of course).

Submission You will submit the source files and screen shot of output for 1. For 2 and 3, write up your answers and submit to brightspace – either a picture or scan of your handwritten answer or a file from the computer. (Do not turn anything in to the TAs). Zip the files together and name them: <partner1first/last name>_<partner2first/lastname>.zip.

Rubric:

1. – 4 pts (2 pts for changed source, 2 pts for showing proper output)
2. – 2 pts
3. – 4 pts