Homework 6

Due March 4th, 11:00 AM

100 points

CS 2235

Data Structures and Algorithms

1. Scale up the size of the Scoreboard to 100, 1000, 10000, 50000, 100000 and 500000.

Note The print statements take time to print to the screen and are not reflective of computation time. Oftentimes, when experimentally timing algorithms, print statements are removed to time more purely the algorithm computation time. You may leave them in or remove them. (Make sure to display your scoreboard size and the time however.)

a) Find the computation time for your array implementation (from HW 4), for each of the 6 Scoreboard sizes.
b) Find the computation time required for the SinglyLinkedList implementation (from HW 5), for each of the 6 Scoreboard sizes.
c) Find the computation time required for the DoublyLinkedList implementation (from HW 5), for each of the 6 Scoreboard sizes.
d) In Excel, Word, etc. create a table for each of these times.
e) Graph Scoreboard size vs computation time for the different implementations, on the same plot (using Excel or other method).
f) Look at your code analytically and try to predict the Big-Oh [O(etc)] order of each of your simulations (look at each method in the simulation and the program as a whole).. Explain your reasoning. Which implementation was faster? Why do you think this was the case?

2. Consider the unique1 and unique2 codes that we coded together in class, (the code is given in slides 19 and 20), perform an experimental analysis to determine the largest value of n, such that the given algorithm runs in approximately 1 minute. (You may simply pass an array of random integers as the parameter for each method).

3. Consider the 5 code fragments in the handout from the Goodrich book. Analyze the fragments and determine the Big-Oh order of each. Justify your answer.

Submit the files with your graph and analysis in a word file or PDF. Submit your code for Part 1 and 2. Please make an effort to use complete sentences and good grammar.

Scoring

1. 10% -Completion times for array implementation, singly linked list implementation, doubly linked list implementation.
2. 10%- Graph of all 3 implementations, with appropriate labels.
3. 20% Analysis of all 3 implementations.
4. 20%- Part 2
5. 20%- Part 3
6. 20%- Clear and well written analysis throughout the submission.