CS4349 Advanced Algorithm Design and Analysis

Assignment 1:

Due back on: Friday, February 4, 2022 at 11:00pm.

The following is from syllabus:

No e-mail submissions are accepted. No late submissions are accepted. So, please plan accordingly, do not leave your submissions to the last minute. If you encounter a problem during elearning submission, please contact 24/7 elearning Help IMMEDIATELY. This help is available 24/7 at:

eLearning Help URL: http://www.utdallas.edu/elearning/eLearningHelpdesk.html
eLearning Help Phone: 1 866 588 3192

Any submission that is missed will be graded with a zero. Please do not insist for exceptions.

Purpose: Demonstrate the ability to use asymptotic notations, solve recurrences, and perform algorithm analysis.

IMPORTANT NOTE: Ideally, please use automated tools wherever applicable, such as an equation writer, graph drawing tool, etc. so as to make sure that your equations involving exponents, logarithm representations, etc. are typed correctly. Manual (handwritten) submissions are ok too. If you are submitting your answers as handwriting, please make sure you provide a good quality scanning with no ambiguity in your answers (avoid poor handwriting where it is hard to discover whether your text is a a subscript or a normal text, etc.).

Question 1 (40 POINTS): Given the input string "WELOVEALGORITHMS", answer the following questions:
1.A.) (15 POINTS) Using the MergeSort algorithm provided in the course slides, illustrate the operation of MergeSort to **manually** sort this string. Please show each step of your recursive solution clearly.

1.B.) (15 POINTS) Using the Analysis of Merge Sort algorithm approach we studied in Ch1 starting with slide 31, perform a running time analysis (Θ only) of executing the MergeSort algorithm. Please use your own words in the way you understood the algorithmic analysis. Use a generic "n" value for the input size.
Hint: As the given input is a string with an arbitrary listing of characters, i.e. not necessarily in increasing or decreasing order, you should be considering average running time.

1.C) (10 POINTS) Using the size of the given input string, what will be the numeric value of the running time? Please show your work.

Question 2. (40 POINTS): Implement the InsertionSort and MergeSort algorithms in any programming language of your choice. Then measure the runtime of each algorithm for integer arrays of sizes given in 2A through 2D.

2.A.) (10 POINTS) size n=100
- **case 1:** for a sorted array of ascending order,
- **case 2:** for a sorted array of descending order,
- **case 3:** for an unsorted (random) array

2.B.) (10 POINTS) size n=1000
- **case 1:** for a sorted array of ascending order,
- **case 2:** for a sorted array of descending order,
- **case 3:** for an unsorted (random) array

2.C.) (10 POINTS) size n=10000
- **case 1:** for a sorted array of ascending order,
- **case 2:** for a sorted array of descending order,
- **case 3:** for an unsorted (random) array

2.D.) (10 POINTS) size n=100000 elements
- **case 1:** for a sorted array of ascending order,
- **case 2:** for a sorted array of descending order,
- **case 3:** for an unsorted (random) array

Plot the input size vs. runtime graph for each algorithm for each size and for each case given in 2A through 2D. Therefore, you should draw one graph for 2A, another for 2B, another for 2C, and one for 2D.
A sample graph is provided for one value of input array size n below to model after. Still, you are free to use your own graph styling. Remember, you will need to draw the graph for four different sizes of n as provided in 2A through 2D. The sort algorithm names (colored columns), as well as time values (y coordinate) of the graph below are deliberately blocked so as not to reveal the answer that should be provided by students.

You can plot the graph either manually, or by using an automated tool. Compare and comment on the results. Please attach your program source code to your submission.

Question 3. (20 POINTS) Given the following recurrence:

$$T(n) = T\left(\frac{n}{2}\right) + \sqrt{n}$$

Prove that solution to this recurrence is $O(\sqrt{n})$ by using the **substitution method**. Please show your work.

Naming Convention:

If you are submitting multiple files, please create a ZIP file of all your files and use the following naming convention for your ZIP file:
CS4349-Assignment<number>-<FirstName><LastName>.zip.

So, student John Smith will name his 1st assignment zip file as:

CS4349-Assignment1-JohnSmith.zip

If you are submitting a single file, please name your file as:

C4349-Assignment1-JohnSmith.doc or .pdf, etc.

Good luck.