CSCD320 Homework1, Eastern Washington University, Spokane, Washington.

Name: **EWU ID:**

Please follow these rules strictly:

1. Verbal discussions with classmates are encouraged, but each student must independently write his/her own solutions, without referring to anybody else's solution.
2. The deadline is sharp. Late submissions will **NOT** be accepted (it is set on the Canvas system). Send in whatever you have by the deadline.
3. Submission must be computer typeset in the **PDF** format and sent to the Canvas system. I encourage you all to use the LaTeX system for the typesetting, as what I am doing for this homework as well as the lecture slides. LaTeX is a free software used by publishers for professional typesetting and are used by almost all computer science and math professionals for paper writing. But of course you are also allowed to use other software for editing (for ex., Microsoft Office Word) but save and submit the file as a PDF.
4. Your submission PDF file must be named as: **firstname_lastname_EWUID_cscd320_hw1.pdf**
(1) We use the underline '_' not the dash '-'.
(2) All letters are in the lower case including your name and the filename's extend.
(3) If you have middle name(s), you don't have to put them into the submission's filename.
5. Sharing any content of this homework and its keys in any way with anyone who is not in this class of this quarter is NOT permitted.

Note: In this homework, as well as in those of the rest of the quarter, n assumes natural numbers.

Problem 1 (5 points)**.** *Based on your learning from the CSCD300 Data Structures course, describe your understanding of the connection and difference between the "data structures" and "algorithms". Say your opinions in your own language. Any reasonable opinion is welcome.*

Problem 2 (10 points)**.** *Show:* $5000n^2 + n \log n = O(n^2)$.

Problem 3 (10 points)**.** *Show:* $5000n^2 + n \log n = o(n^3)$.

Problem 4 (10 points)**.** *Show:* $5000n^2 + n \log n = \Omega(n^2)$.

Problem 5 (10 points)**.** *Show:* $5000n^2 + n \log n = \omega(n)$.

Problem 6 (15 points)**.** *Let $f(n)$ be a nonnegative increasing function. Is $f(n) = \Theta(f(2n))$ always true ? If yes, prove it; otherwise, give a counter example. Hint: try many different such possible functions for f.*

Problem 7 (25 points)**.** *The idea of* Merge Sort *that we have discussed in the class is to split the input sequence of size n into two subsequences of each sized $n/2$, recursively sort each subsequence, and merge the two sorted subsequences into a sorted version of the original sequence. Now, someone proposed the following new idea: why don't we split the input sequence into more subsequences of smaller size, so that the algorithm can reach the exit condition (which is when the sequence size becomes 1) more quickly and thus make the algorithm faster? Your job:*

Name: **EWU ID:**

1. *Change the algorithm and give the pseudocode for this proposed new* Merge Sort*, where the input sequence is divided into* 4 *subsequences of each sized* $n/4$*.*

2. *Analyze the time complexity of this new algorithm and present the result using the* Θ *notation.*

3. *Is this new algorithm asymptotically faster than the one in the textbook ? Justify your answer.*

4. *Can this algorithm be faster in practice than the one in the textbook ? Justify your answer.*

5. *Let's think of the extreme case. We split the input sequence into* n *subsequences of each sized just* 1*. Can this algorithm be asymptotically faster than the one in the textbook ? Justify your answer.*

6. *Do you get any insight why the textbook* Merge Sort *only splits the input sequence into two halves ?*

Problem 8 (15 points total; 5 points for each algorithm.)**.** *Search and learn three existing algorithms that use the divide-conquer strategy. For each algorithm, in your own language, concisely and clearly describe:*

1. *the problem statement*

2. *the algorithmic idea in the solution (don't just copy the code or the text on the webpage to me)*

3. *the time complexity*

4. *the condition, on which the worst-case running time appears.*

5. *the source of your finding. For example, the url of the webpages, the title and page of a book, the title/author/year of an article, etc.*

Note: If you just copy and paste or with small modification without your own understanding, you will get zero for this problem.