CS 418-301: Algorithms Design and Analysis **Due: 7/07/2022 11:59pm**

Homework #2

[redacted]

100 points

Note You must provide one compressed file that includes your all your work. Make sure you test your implementations on different inputs. I may ask you for a demo of your work.

Problem 1(10 pts). What is the difference between Big-O, Big-Ω, Big-Θ and best case, average case, worse case scenarios?

Problem 2(10 pts). Complete the following table and sort all big-O notations from smallest (fastest) to largest (slowest).

Expression.	Dominating term	Dominating term in Big-O
$5 + 0.001n^3 + 0.025n$	$0.001n^3$	$O(n^3)$
$n! + n^n$		
$2^{3^2} + 3^{2^n}$		
$n^2 \log n + n(\log n)^2$		
$n \log n + 9^{999999999999}$		

Problem 3(20 pts). Use Master method to solve the below recurrences:

(a) $T(n) = 2T(n/4) + 1$
(b)$T(n) = 2T(n/4) + \sqrt{n}$
(c) $T(n) = 2T(n/4) + n$
(d) $T(n) = 2T(n/4) + n^2$

Problem 4(30 pts) Solve the following recurrences (using recursive tree or substitution method) and prove your result by induction.

(a) $T(n) = T(n-1) + 1$ where $T(1) = 1$.
(b) $T(n) = T(n-1) + n$ where $T(1) = 10$.
(c) $T(n) = T(n-1) + 2n^2$ where $T(1) = 10$.
(d) $T(n) = 2T(n-1)$ where $T(1) = 1$.
(e) $T(3^n) = T(3^n/3) + 1$ where $T(1) = 1$.

Problem 5(30 pts) For each of the following pairs of functions $f(n)$ and $g(n)$, determine whether $f(n) = O(g(n)), g(n) = O(f(n))$, or both (Hint: use mathematical limits.)

(a) $f(n) = n^5$–$100n^{17} + 2^5$, $g(n) = n^{17}$
(b) $f(n) = n^{-2} + \log(n^{10})$, $g(n) = n \log n$
(c) $f(n) = n(\log n)^2$, $g(n) = 2^n + n^2 \log n$
(d) $f(n) = n^2 \log n$, $g(n) = n^2 \log(n^2)$
(e) $f(n) = 2 \log(2^n)$, $g(n) = 2n + 1$
(f) $f(n) = n^n$, $g(n) = n!$
(g) $f(n) = 2^n$, $g(n) = n^2$