STAT 304: HW1

Due Date: January 16th at 11:59pm

You may talk to your fellow STAT304-ers about the problems. However:

- Write your answers yourself in your own words after collaborating. You should never share your typed-up solutions with your collaborators.
- If you collaborated, list the names of the students with whom you collaborated at the beginning of each problem.

Read chapter 3 of the textbook and answer the following questions. Questions 2, 3, and 4 are pulled from the end-of-chapter exercises and are reproduced exactly in this document.

Please submit a single pdf file. I'd suggest writing answers to all questions in a single jupyter notebook then downloading as a pdf or downloading your jupyter notebook as a pdf for Question 1 and combining it with your answers to questions 2, 3, and 4.

Question 1 (50 points) Read the code in `hw1_code.ipynb` and complete the following tasks for all seven functions and all five examples.

(a) Give the actual running time for several (at least 4) values of n and plot the execution time versus the input size trend line. Hint: write runtime functions for each function with a different type of input. For example, `fun1` takes an integer as an input, `example1` takes a sequence, and `example5` takes two sequences.

(b) Give an asymptotic analysis of the running time using big-O. Hint: break your analysis into Assignments, Array/List Accesses, Arithmetic Operations, and Comparison Operations (**show this work!**). Add them together to get your total operations and conclude some big-O.

Note: for `fun6`, you do not need to count primitive operations. Just provide a big-O if you can.

Question 2: R-3.28 on Page 142 (15 points) For each function $f(n)$ and time t in the following table, determine the largest size n of a problem P that can be solved in time t if the algorithm for solving P takes $f(n)$ microseconds (one entry is already completed).

	1 Second	1 Hour	1 Month	1 Century
$\log n$	$2^{10^6} \approx 10^{300000}$			
n				
$n \log n$				
n^2				
2^n				

Question 3: R-3.17-R-3.19 (10 points)

(a) Show that $(n+1)^5$ is $O(n^5)$.

(b) Show that 2^{n+1} is $O(2^n)$.

(c) Show that n is $O(n \log n)$.

Question 4: R-3.2, R-3.3 (10 points)

(a) The number of operations executed by algorithms A and B is $8n \log n$ and $2n^2$, respectively. Determine n_0 such that A is better than B for $n \geq n_0$.

(b) The number of operations executed by algorithms A and B is $40n^2$ and $2n^3$, respectively. Determine n_0 such that A is better than B for $n \geq n_0$.

Question 5 (10 points) Order the following 14 functions by growth rate:

$$n, \sqrt{n}, n^{1.5}, n^2, n \log n, n \log \log n, n \log^2 n, n \log(n^2), 2/n, 2^n, 2^{n/2}, 37, n^2 \log n, n^3$$

Indicate which functions grow at the same rate.

Question 6 (5 points) Suppose $T_1(n) = O(f(n))$ and $T_2(n) = O(f(n))$. Which of the following are always true (for all T_1, T_2, and f)?

(a) $T_1(n) + T_2(n) = O(f(n))$

(b) $T_1(n) - T_2(n) = \Theta(f(n))$

(c) $\frac{T_1(n)}{T_2(n)} = O(1)$

(d) $T_1(n) = O(T_2(n))$

For this question you do not need to prove an item is true (saying true is enough for full credit). If an item is false then you need to give a counterexample by providing values for $T_1(n), T_2(n)$ and $f(n)$ for which the statement is false. Hint: think about the definitions of big-O and big-Θ.