ECE 241 – HOMEWORK 2 Fall 2021

Due: Thursday, September 23, 2021 at 11:00 PM on Gradescope

Please answer the following questions. Submit your screenshots on Gradescope as one file (e.g., Word document or PDF). The starter code for three programming questions have been given. Please zip the folder as 'Submission/Code/question_.py' for the three questions and upload it to gradescope. (Replace the underscore with the actual problem number. E.g., for problem 3 it will be 'Submission/Code/questiom3.py, etc.)

1. **(20 points)** Order the following list of functions by the big-O notation.

$6n\log n$	2^{100}	$\log\log n$	$\log^2 n$	$2^{\log n}$
2^{2^n}	$\lceil\sqrt{n}\rceil$	$n^{0.01}$	$1/n$	$4n^{3/2}$
$3n^{0.5}$	$\log 5^n$	$3^n + \log 2021$	2^n	$\log_4 n^n$
4^n	$2n\log^2 n$	$2^n + n^2 + 3n$	$\sqrt{\log n}$	$(n+1)!$

2. **(20 points)** Find the Asymptotic time complexity for the following: (https://gist.github.com/chenzibin2019/c5865998b87bbababee2404ca3bca3a9)

3. **(20 points)** This problem will demonstrate how you can use data structures and algorithms for data science. In this specific case we look into the analysis of data that has been collected by weather stations. We have prepared a code skeleton (WeatherData.py) which will fetch temperature data from the Amherst weather station.

   a. It is your task to finalize that code such that the function `highest_temp()` determines the highest temperature for a certain period and the function `lowest_temp()` determines the lowest temperature for the same period.
   b. In addition, finalize the functions `average_high()` and `average_low()`, which have the goal to calculate the average of all maximum and minimum temperatures, respectively.

4. **(20 points)** A shelf contains some products sorted according to their width,
   e.g. |||10, 20|||
   we place more already sorted products with lesser width than the one that are already present.
   e.g. |||10, 20||1, 2, 4, 7|||
   Find the position of a given product (no sorting allowed and find it in O(log n) time).
   e.g. |||10, 20||1, 2, 4, 7|||
   position of product with width 10 is 0
   position of product with width 7 is 5

5. **(20 points)** Merge sort involves recursively splitting the array into 2 parts, sorting and finally merging them. A variant of merge sort is called **3-way merge sort**

where instead of splitting the array into 2 parts we split it into 3 parts. As shown in the MergeSort code, it recursively breaks down the arrays to subarrays of size half. Similarly, 3-way Merge sort breaks down the arrays to subarrays of size one third.

In this question, you need to implement the 3-way Merge sort algorithm (in function *mergeSort_3_way*) in a descending order. During the sorting, count the number of comparisons in each merge stage.