Sorting Runtime Analysis

You’ve been given a project by your manager to give a visual representation the “Big-Oh runtime” of four different sorting algorithms by graphing each runtime in a Spreadsheet:

1. Bubble Sort
2. Insertion Sort
3. Merge Sort
4. Quick Sort

# The Coding

The project does not require you to write a summary or recommendation in terms of which sort algorithm to use. What *is* required is that you write programs and/or functions that do the following:

1. Write a program that not only creates a CSV file with random numbers in it but write a program that generates a CSV file with the following characteristics in it:
   1. Each row will have random numbers between 1 and 10000
   2. The first row in the file will have 250 random numbers
   3. Each subsequent line will have 100 more numbers
   4. Your CSV file will have 200 rows in it.
   5. The program will also ask for a filename to save the data to. If you give it an existing file, make sure you confirm with the user if they wish to overwrite the data.

This(These) file(s) is the data set to run your timer to collect your data set.

1. Write a program that has:
   1. each of the four different sorting algorithms implemented as a function. You pass in an arbitrary list of integers and it returns the sorted list.
   2. A menu which lets the user to select one of the four sorting algorithms to test/time, after which it will do the following:
      1. Ask for the CSV data file to load.
      2. Ask for the name of to save the timing data in. If you give it an existing file name, make sure you confirm with the user that they want to overwrite the existing file.
      3. How many runs to do (up to 200)
      4. Open the CSV data file for reading.
      5. Loop the number of runs:
         1. Read the current line from the data file into a list
         2. Start the timer
         3. Apply the appropriate sort function
         4. Stop the timer.
         5. Write into the timing data file these two pieces of information as a CSV record
            1. Number of items sorted
            2. How long it took
      6. Close the input and output files.

Now to help you “easily” write CSV files here are two URLs that you may find useful:

1. <https://appdividend.com/2020/12/10/how-to-convert-python-list-to-csv-file/>
2. <https://stackoverflow.com/questions/3191528/csv-in-python-adding-an-extra-carriage-return-on-windows> (for people writing on Windows machines)

For the stopwatch/timer object, you can use the following class from GitHub. Be sure to give credit where credit is due!

<https://gist.github.com/igniteflow/1253276>

# The “Analysis”

Once the data of the various sorting algorithms is done and you have your various CSV files for the different sorting algorithms you are to do the following:

1. import each of the CSV data as a separate sheet into an Excel Workbook.
2. Label each sheet with the related sorting algorithm name.
3. Do a “scatter plot” graph of the X-Y data.

# What to Submit onto Teams for this project:

* All of your source code and Python scripts you wrote for this project
* All of the generated .csv files you made (at least 5 different files – your “randomnumbers.csv”, and one csv file for each of the four different sorting algorithm timings.
* Your Excel spreadsheet which contains the imported sort-algorithm timings. It will also have the scatter plot graph

# Assessment

* Basic Programming Rubric (out-of 25)
  + Docstrings (1-5)
  + Comments (1-5)
  + Variable Names (1-5)
  + Functions (1-5)
  + Overall Program (1-5)
* CSV Files (10 points total)
  + Up-to 2 points for each of the required CSV files.
* Excel Workbook (5 points)
  + Data imported as separate sheets into workbook
  + Appropriate graphs representing the runtime is generated.

This assignment will not be scaled to be out of 25. It is out of 40 points.

This assignment must be handed in by no later than **JUNE 22, 2022.**