CSCE 350: Project

Late submissions will not be accepted

Task - 1: (50 points) **Implement the Quick-Sort Algorithm using C++**

Requirements:

- Your code should be able to read an input ASCII file that contains unsorted floating-point numbers separated by a blank space.
- Choose a pivot using the *median-of-three method*
- Your code will produce an output ASCII file that contains the sorted floating-point numbers separated by a blank space.
- Both the input and output filenames should be passed as command line arguments.
  - Compile your program using the following command:

    yourLastname_yourFirstname_QuickSort input.txt output.txt
- Your program should also output the execution time in milliseconds

---

Task – 2: (50 points) **Empirical Analysis of Algorithm using C++**

1. (35 points) Study the time complexity of QuickSort using different input sizes: 10, 100, 1000, 10000, and 100000 (the number of unsorted floating-point numbers).

   Requirements:

   - Write C++ code to randomly generate 100 input files for each input size. You can use any uniform random number generator to create an input file.
   - For each input file, run QuickSort and record the execution time.
   - Create an ASCII file, named *yourLastname_yourFirstname_executionTime.txt* containing a table with all the execution times, as seen below:

Input Size	*Execution Time*
10	4e-09
10	3e-09
…	
100	#
100	#
…	

- Compute the average running time for each input size.
- Create an ASCII file, named *yourLastname_yourFirstname_averageExecutionTime.txt* containing a table with the average execution times for each input size, such as:

Input Size	*Average Execution Time*
10	2e-09
100	4e-09
1000	#
10000	#
100000	#

2. (15 points) Show the average execution times in a plot, where X-axis represents the input size and the Y-axis represents the time.

- You will have a curve for QuickSort, where a point on the curve represents the average execution time for an input size.
- Save the plot into a file named "*yourLastname_yourFirstname_plotAverageExecutionTime.jpg*"
- A *makefile* should be submitted together with your codes providing instructions on how to compile your codes.

Instructions:

- All code should be written in C++ for Linux.
- Program file submissions that do not compile automatically receive a grade of 0.

- Please test your code on the Departmental Linux machines prior to submission on BB
- Code must commented appropriately for major steps.

Submission:

- You must submit all your generated files and plots.
- Your zip folder must contain the following files:

  1. **QuickSort.cpp*
  2. **executionTime.txt*
  3. **averageExecutionTime.txt*
  4. **plotAverageExecutionTime.jpg*
  5. *makefile*
  6. *InputFileGenerator.cpp* → used to generate input ASCII files (Do NOT include in makefile)
  7. <u>Optional</u>: A ReadMe file that includes instructions on compiling your project

  **File Naming Convention: All your files(1-4 in the above list) should have the following prefix:*

  *yourLastName_yourFirstName_*

- Compress all your files into a single folder titled *"CSCE350Project_yourLastname_yourFirstname"* and submit it on Blackboard.
- Accepted Compression Formats: .tar.gz/ .zip only