CS317: Intro to Algorithm Analysis — Programming Project

Due: See Canvas for Due Dates — (50 points)

Problem Statement

You will write two programs to sort arrays of strings. One version will implement the QuickSort, the other will implement MergeSort.

You must use basic sequential arrays in your programming language (no library data structures such as ArrayList or Vector). You must also your own sort code. Do not use any built-in libraries for sorting. (`Example: in C++, the declaration may look like` **`string lines[N];`**)

You may write the solution to this assignment in either Java or C++.

Requirements

- Prompt the user for the name of the input and output text files.
- Read the lines of the file into an array of strings. Your program should work for any number of lines in a file, from 1 through 200000 (two hundred thousand lines, at least).
- Sort the lines using Quicksort. The string comparisons should be case-insensitive.
- Sort the original set of data using Mergesort.
- Include metric gathering instructions to measure the elapsed time for each sort. Are they the same? Is it significantly different? What if the data set was already sorted? Would there be a significant difference?
- Print the sorted data to two text files, for verification purposes.

Sample Input File

An input data file for this program is simply a file with a collection of text lines in it.

```
hello goodbye Once Upon a time there
was a monster named ogre This is a word
list that we can store into string type
there is no puncts in this file and
there may be up to 200000 lines in this
file should be Sorted case less Ann bob
CAROL
```

TURN IN:

Submit the electronic version of your project to Canvas by the due date. Along with a copy of your test input file. Please zip up the project and include a note on the canvas submission indicating which environment it should be tested in (Netbeans 11/12, IntelliJ 11, Visual Studio, etc).

Note if you are zipping a visual studio project, DO NOT INCLUDE the DEBUG subfolder or .vs hidden directory. They are too large for canvas to handle.

ALSO – turn in an executive summary explaining the timing results (1 to 2 pages)

- Include, for your chosen programming language what is involved in comparing two strings? I.E. what does it cost? Is it a fixed cost per pair of strings?
- At what data set size / characteristics do you see a significant divergence of timing (if any).

Grading Requirements

- Your program must be well-commented. Comment all variables, functions and remember to have a section of comments at the top of each of your program files that includes your name, date, course section and a solid description of what your software does.
- Use good variable names.
- Use proper code indentation to make sure your program is easy to read and understand.