

CS575: Programming Assignment 1

(Programming due at 11:59:59PM and Question 2 due at 10:05AM, Oct. 3, 2017)

1. [70%] Implement exchange sort, insertion sort, merge sort, and randomized quick sort algorithms. (In a randomized quick sort algorithm, the pivot is selected randomly.) More specifically, your program should do the following:
 - a) Let a user select the *sorting algorithm* to use and specify the number of input data n where $1 \leq n \leq 1,000$. Return an error message, if the selected sorting method is not supported by your program or $n < 1$ or $n > 1,000$.
Note 1: A textual command line interface is enough to do this. (Do not waste your precious time for a fancy GUI. This is not a graphics course!)
Note 2: To generate random numbers, use `srand(time(NULL))` and `rand()` functions. You can find a lot of information about these functions online. For example, see <http://www.cplusplus.com/reference/clibrary/cstdlib/rand/> and <http://www.cplusplus.com/reference/clibrary/cstdlib/srand/>
 - b) Randomly generate an array of n positive integers and print them.
Note 3: For the other sorting algorithms, data values are arbitrary integers returned by the pseudo random number generator.
 - c) Sort the randomly generated data in non-descending order using the selected sorting algorithm. Print the final result.
 - d) If $n \leq 20$, create random numbers that range between 0 and 15 and do simple visualization of the sorting process as follows. (Do not do this if $n > 20$.) For a number k that is randomly generated in Step 2, print "*" k times and move to a new line. For example, if the data to sort are (7, 3, 5), your program needs to display the following before starting to sort them:

After executing one step of the selected sorting algorithm, print the array to show the current state. For instance, if the array becomes (3, 7, 5) after executing one step of the selected sorting algorithm, your program needs to print the following:

2. [20%] Write down the time complexity for the four sorting algorithms: exchange sort, insertion sort, merge sort, and randomized quick sort. Justify your answers except the randomized quick sort. Please include your answers for the best-case, average-case and worst-case time complexity if that applies to the algorithm.
3. [10%] 10% of the grade will be based on good coding style and meaningful comments.

All programming must be done using **C or C++ in Linux** where your code will be tested. Create a tar file that includes (1) source code files, (2) a readme file that clearly describes how to compile and run your code, and (3) a plain text or pdf file for the time complexity justification. Submit only the tar file through the Blackboard. The name of the tar file should be `yourlastname_yourfirstname_prog1.tar` (Do not use special characters like #, @, or &, because they have caused Blackboard problems in the

past.) Suppose that your assignment files are under the directory of /your_userid/yourlastname_prog1/ and you are under that directory right now. To create a tar file under /your_userid directory, do the following in Linux command line:

```
>cd ..  
>tar cvf yourlastname_yourfirstname_prog1.tar yourlastname_prog1
```

To view the content of the created tar file, do the following in Linux command line:

```
>tar tvf yourlastname_yourfirstname_prog1.tar
```

Finally, read the following policies carefully:

- *All work must represent each individual student's own effort. If you show your code or any other part of your work to somebody else or copy or adapt somebody else's work found online or offline, you will get zero.*
- *To detect software plagiarism, your programs will be checked using an automated tool, called Moss. If the programs of two or more students have 90% or higher similarity, they will automatically get zero for the programming assignment. Even if the similarity score is lower than 90, the instructor can require students to prove that they did an assignment on their own if there is any even slight indication or hint of plagiarism.*
- *Your code will be compiled and executed. If your code does not compile or produce any runtime errors such as segmentation faults or bus errors, you will get zero.*
- *The instructor and TA will not read or debug your code. The instructor and TA will not take a look at an emailed code. If you need general directions, show your code to the TA during his office hours. The TA will not do programming or debugging for you though. He will only help you understand algorithms to be implemented and answer basic questions related to implementation.*