BLG335E, Analysis of Algorithms I, Fall 2018
Project 1

Handed out: 11.10.2018
Due: 29.10.2018, until 23.30

PROBLEM: In this project, you are expected to implement Insertion Sort and Merge Sort algorithms to analyze real time stock market data. You are also required to prepare a report including their analyses. The project includes 2 phases, implementation and report, respectively. Please note that you may not receive any grade if you only submit a report. Please read this document carefully and prepare a well-structured report that includes your analyses.

Part A. Implementation (50 points)

You are provided a data set taken from Kaggle (*https://www.kaggle.com/deeiip/1m-real-time-stock-market-data-nse*) that includes real time stock data for 3 weeks. The file (*log_inf.csv*) includes several lines and each line contains data of a distinct stock information for a specific time. Note that the first line is header which represents the names of the features. Please refer to the given link above for more information about the data set.

Download the input file from the link provided in the definition of the project on Ninova. Later on, you are required to sort the data considering a given feature (timestamp or last_price) as a criterion. Read **N** numbers from the file and sort them using Insertion Sort or Merge Sort.

Your program should be run from the command line with the following format.

```
./yourExecutable -algo algorithmType -feature criterion -size N
```

-feature : Sorting criterion (for timestamp ‘t’ or for last_price ‘p’). Sort the data in ascending order considering the criterion.
-size : Total number of items to be sorted (1000, 10000, 100000, 1000000). You can simply read first N line from the file and sort them.
-algo : Algorithm to be used to solve the problem (for Insertion Sort ‘i’ or for Merge Sort ‘m’)
An example execution command is given as follows:

```
./myExecutable -algo m -feature p -size 100000
```

This command executes the program using Merge Sort with the first 100000 elements of the input file considering the price feature. After the execution of your program, an output file should be created (sorted.csv) with the same format of the input file.

Part B. Report (50 points)

In your report, you are expected to analyze and compare the running times of algorithms with respect to their computational complexity. Therefore, you need to complete the following tasks and include the results in your report.

a. (10 points) Give the asymptotic upper bound on the running time for Insertion Sort and Merge Sort (which you can find in the lecture slides) and show that your implementation of these algorithms fit these values.

b. (10 points) Run 10 times each sorting algorithm for each different value of **N** as {1000, 10000, 100000, 1000000} considering a sorting criterion that you select. You may also choose additional N values if necessary.
Calculate the average time of execution for each value of **N** and construct a table that summaries these results.
NOTE: You can use the *clock()* function under *ctime* library to calculate time of execution for the sort functions. Refer to http://www.cplusplus.com/reference/clibrary/ctime/clock for more details.
c. (15 points) After calculating execution times you will prepare two line plots (in Excel or Matlab) in order to visualize the runtime complexity of Insertion Sort and Merge Sort for different values of **N.** Then you are expected to interpret the results with respect to the asymptotic upper bounds you have given in (***a)***.
d. (15 points) Assume that after you have your data sorted (can be size of 1000000 considering the last price feature), you are required to insert a new entry to your data. Run these two algorithms on this data in order to do so and report the results in terms of elapsed time. Comment on the results by indicating in which cases you would choose which algorithm. Why?

DETAILED INSTRUCTIONS:

- You should be aware that the Ninova system clock may not be synchronized with your computer, watch, or cell phone. Do not e-mail the teaching assistant or the instructors your submission after the Ninova submission has closed. If you have submitted to Ninova once and want to make any changes to your report, you should do it before the Ninova submission system closes. Your changes will not be accepted by e-mail. Connectivity problems to the Internet or to Ninova in the last few minutes are not valid excuses for being unable to submit. You should not risk leaving your submission to the last few minutes. After uploading to Ninova, check to make sure that your submission appears there. You must submit all your program and header files. You must also submit a softcopy report.
- Policy: You may discuss the problem addressed by the project at an abstract level with your classmates, but you should not share or copy code from your classmates or from the Internet. You should submit your own, individual project. Plagiarism and any other forms of cheating will have serious consequences, including failing the course or disciplinary investigations.
- Submission Instructions: Please submit your assignment through Ninova. Include both your report (as a PDF file) and your code (including header files) in the archive file you submit.
- All your code must be written in C++, and must compile and run on the common ITU Linux Server (accessible through SSH) using g++. If your code requires non-standard compiling, please state compilation instructions in your report.
- When you write your code, try to follow an object-oriented methodology with well-chosen variable, method, and class names and comments where necessary. Your code must compile without any errors; otherwise, you may get a grade of zero on the coding part of the assignment.

If you have any questions, please feel free to contact Res. Asst. Doğan Altan via e-mail (daltan@itu.edu.tr).