ISE 315, Analysis of Algorithms, Fall 2018
Homework 2

Handed out: 14.11.2018
Due: 30.11.2018, until 23.00

PROBLEM: In this homework, you are expected to implement Quick Sort algorithm to analyze real time stock market data. You are also required to prepare a report including their analyses. The homework 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 homework 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 Quick Sort.

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

```
./yourExecutable -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.
An example execution command is given as follows:

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

This command executes the program using Quick 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)

a. (10 points) Give the asymptotic upper bound on the running time for Quicksort. If there is a recurrence for the algorithm, give and solve it.

b. (10 points) Run your program for N as {10, 100, 1000, 10000, 100000, 500000, 1000000}, and calculate the average time of execution (for example run 10 times for N = 10 and take the average, 10 times for N = 100 and take the average, and so on…) for each value of N. After calculating average execution times, you are required to prepare a two-lined plot in order to visualize the run time

complexity of Quicksort for different values of N. Comment on the results by considering the asymptotic bound that you have found in (a).
Note: You can use the clock() function under ctime library for calculating time of execution for the search functions. Refer to http://www.cplusplus.com/reference/clibrary/ctime/clock for more details.

c. (20 points) What is the worst case for Quicksort? Construct a data set in order to simulate this case, and execute your program for different values of N as {10, 100, 1000, 10000, 100000, 500000, 1000000}. Report the average execution times, and introduce a plot like you have done in part (b). Introduce a solution in order to overcome the worst case of Quicksort (You do not need to implement that solution).

d. (10 points) Is Quicksort stable? Explain your answer by illustrating your answer with a small fraction from the data set. You may modify the values if necessary.

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 homework 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 homework. 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).