ISE 315 Analysis of Algorithms
Fall 2016 - Homework 2

Handed out: November 2, 2016
Due: November 16, 2016, 23.30

Submission type: An archive file including all source code files and report (pdf).

Randomized Quick Sort

In this homework, you are asked to implement Quick Sort. You are also required to prepare a report including its analysis.

Part A. Implementation (50 points)

An input file (data.txt) that includes integers to be sorted is provided. Read N numbers from the file and sort them using Randomized Quick Sort. You should refer to the lecture slides for Randomized Quick Sort algorithm.

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

```
./studentID N
```

N : Total number of integers to be sorted (1000, 10000, 100000, 1000000)

An example execution command is given as follows:

```
./040050256 1000
```

This command executes the program using Randomized Quick sort with the first 1000 elements of the input file.

After the execution of your program, an output file should be created (sorted.txt) with the sorted integers in ascending order.

Part B. Report (50 points)

a. (10 points) Give the worst case, best case and average case on the running time for Quick Sort and Randomized Quick Sort.

b. (10 points) Run Randomized Quick Sort for each different value of **N** as {1000, 10000, 100000, 1000000} for the given input file. Calculate the average time of execution for each value of **N**.

c. ***(15 points)*** Why do we need Randomized version of Quicksort? Discuss in detail by presenting examples. Give an example on a simple scenario (give an example on at least 5 element-array) to illustrate the advantage of randomized version over Quick Sort.

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.

d. (15 points) After calculating execution times you will prepare two line plots (in Excel or Matlab) in order to visualize the runtime complexity of Randomized Quick Sort for different values of **N** (Both for

part b and c). Then you are expected to interpret the results with respect to the asymptotic upper bounds you have given in ***a***. You may also include previous results from Assignment 1 (Merge Sort and Insertion Sort) in your plot.

Detailed Instructions:

- All your code must be written in C++ using object oriented approach and able to compile and run on Linux using g++.
- Submissions will be done through the Ninova server. You must submit all your program and header files. You must also submit a softcopy report.

This is an individual homework. You should write your own code. Academic dishonesty including but not limited to cheating, plagiarism, collaboration is unacceptable and subject to disciplinary actions. Any student found guilty will get grade F.