BLG335E, Analysis of Algorithms I, Fall 2015
Project 1

Handed out: 08.10.2015
Due: 16.10.2015, until 23.30

PROBLEM: In this project, you are expected to implement Bubble Sort and Merge Sort. You are also required to prepare a report including their analyses.

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 Bubble Sort or Merge Sort.

You should refer to the lecture slides for Merge Sort algorithm. Additionally, pseudo code for Bubble Sort is given as follows:

```
for $i \leftarrow 1$ to $length[A]$
    do for $j \leftarrow length[A]$ downto $i + 1$
        do if $A[j] < A[j - 1]$
            then exchange $A[j] \leftrightarrow A[j - 1]$
```

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

```
./studentID_AoA1_P1 algorithmType N
```

N : Total number of integers to be sorted (1000, 10000, 100000, 1000000)
algorithmType : Method to be used to solve the problem (for Bubble Sort ‘b’ or for Merge Sort ‘m’)
An example execution command is given as follows:

```
./040050256 m 1000
```

This command executes the program using Merge 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)

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 Bubble Sort and Merge Sort (which you can find in the lecture slides) and show that your implementation of these algorithms fit these values.

b. (20 points) Run each search methods for each different value of **N** as {1000, 10000, 100000, 1000000}.

Calculate the average time of execution for each value of **N**.

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. (20 points) After calculating execution times you will prepare two line plots (in Excel or Matlab) in order to visualize the runtime complexity of Bubble 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***. Indicate in which cases you would choose which algorithm. Why?

DETAILED INSTRUCTIONS:

- All your code must be written in C++ using object oriented approach and able to compile and run on Linux using g++.
- Do not use external libraries such as STL.
- Submissions will be done through the Ninova server. You must submit all your program and header files. You must also submit a softcopy report.
- Each student must work individually for the project. Team work is not accepted!

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