

ISE 315 Analysis of Algorithms

Fall 2016 - Homework 1

Handed out: October 5, 2016

Due: October 21, 2016, 23.00

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

Merge Sort and Insertion Sort

In this homework, you are asked to implement Insertion Sort and Merge Sort. You are also required to prepare a report including their analyses.

Part A. Implementation (60 points)

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

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

`./studentID N AlgorithmType`

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

Algorithm Type: Algorithm type to be used for sorting

An example execution command is given as follows:

`./040050256 1000 MergeSort`

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 (40 points)

a. (10 points) Give the asymptotic upper bound on the running time for both Merge Sort and Insertion Sort (which you can find in the lecture slides).

b. (15 points) Run Merge Sort and Insertion Sort for each different value of N as {1000, 10000, 100000, 1000000}. Calculate the average time of execution for each value of N for each algorithm.

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/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 Merge Sort and Insertion Sort for different values of N .

ISE 315 Analysis of Algorithms

Fall 2016 - Homework 1

Then you are expected to interpret the results with respect to the asymptotic upper bounds you have given in a .

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.