ITCS_6114_8114: Algorithms and Data Structures

Fall 2015

Programming Project 1: Insertion Sort

Due Date

The project is due on September 2nd, 2015 at 11 PM. No late submissions.

Project Description

Implement the insertion sort algorithm. To be specific, call your program "Isort". your code takes "test.txt" as an input file with the numbers to sort and outputs the result to "answer.txt" file.

Your program should compile and run from the command line. For example, if you implement your code in java, it should run using the following command assuming running in the default folder - the location of your source files:

javac Isort.java
java Isort test.txt

The input file "test.txt" should be a plain text file and it contains a sequence of numbers separated with semicolon. It can be generated manually or automatically with random numbers. Note that the name of this input file can be different so DO NOT hard code this in your code. It has to be read from the command line! When grading, different input files will be used.

The output of your program should be a plain text file called "answer.txt" recording the right sequence of all the numbers (the answer.txt file should be saved in the same folder as your default folder).

For example, the input.txt and output.txt may be shown as follows:
Input.txt: 12; -12; 10; 5; 100; -9; 1; 10
The output.txt: -12; -9; 1; 5; 10; 10; 12; 100

Programming Languages and IDE

The program MUST be written in Java or C++. Please contact the instructor or TA for other languages. As we are going to build upon this program, carefully choose the programming language and design your program.

We highly recommend that you use jGrasp environment for your coding. If you are using Mac or Unix system computer, you must test your code on windows environment and provide a detail step by step instruction of how it should be run. If you are using C/C++ language, use Cygwin or MinGW to compile and run your code. You can also use a plain text editor to write you code.

We will use *jGrasp* to grade all the submissions. As this first programming project is easy, the main purpose is to get you familiar with the requirements on programming, documentation, and submission. Please stick to the project description.

Grading Criteria

The total of 100 points for this project is broken up into:

• 20 points for proper construction of data structures required in the program.
• 20 points for correctly handing the input and output files.
• 40 points for efficiently implementing the insertion sort algorithm.
• 20 points for compilation, structure, and documentation (a readme file and comment in the code).

Within these criteria, your grade will be based on program structure, efficiency, and correct execution. The structure of your code will be judged for quality of the comments, quality of the data structure design, and especially the logic of the implementation. The comments need not be extremely long: just explain clearly the purpose of each class and each function within each class.

Submission Guidelines

Your submission should include all your source code files and a brief report as a README file. Do NOT include any IDE-specific project files, any compiled files, or any executable files. Every file should have your name in a comment line at the top. Your README file should have a brief description of your program design, the breakdown of the algorithm, the compiler you used, the platform you used, a summary of what you think works and fails in your program, and a short description of your data structure design.

You will submit your project on Moodle. You should submit a single zip file containing your source code files and README file. You can submit your project multiple times; only the most recent project submission will be graded. No late submission is allowed for this project and no submission through email will be considered. The submission link will automatically cut off at the end of the due date and time.

Final Warning

A project that does not follow the submission guidelines will receive a 10 point deduction. Proper submission is entirely your responsibility. Contact the TA if you have any doubts whatsoever about your submission. Do NOT submit your project via email. Please observe the academic integrity guidelines in the syllabus, and submit your own work.

Some useful links to setup your development environment:

- For Cygwin: https://cygwin.com/install.html

- For MinGW: https://users.cs.jmu.edu/bernstdh/web/common/help/cpp_mingw-setup.php
- For jGrasp: http://www.jgrasp.org/