Due : Sept. 8 (9/8), 13:00

Overview

This assignment consists of one part, implementing a basic sorting algorithm.

General Notes

- Do not use Eclipse. We recommend Sublime Text (Linux/Mac/Windows), Atom (Linux/Mac/Windows), Notepad++ (Windows), or TextWrangler (Mac).
  - Code that contains the line "package ..." at the beginning of the file breaks our autograder and Eclipse automatically adds that line.
- Do not change any method or class signatures. You should only edit inside of the functions. If your code changes any class or method names or signatures, you will receive an automatic 0. You should not implement any other functions or instance variables besides the ones that are provided, unless explicitly allowed.
- Make sure your code compiles. Non-compiling code will automatically receive a 0. If you have a problem that is causing you to not be able to compile, it may be better to just comment out the incorrect code and return a dummy value (something like null or -1) so the rest can compile.
- Make sure that your code does not print out anything (there should be no System.out.println in your code). You will receive an automatic 0 if your code outputs something to STDOUT during the tests.
- To ensure that your code will be accepted by the autograder, you should submit your code on YSCEC, download it again, unzip it, recompile it and check the provided test suite. This way, you know that the file you are submitting is the correct one.

Sorting

Sorting is a fundamental algorithm in computer science. Here you will implement a basic sorting algorithm of your choice.

In this assignment, you will implement both ascending and descending order sorting on arrays of ints. You are free to implement any sorting algorithm that you wish, but you must code it yourself (for example, you cannot just call Java's sort on your array or copy a solution from the internet). We will only be testing that your code produces a correct result and terminates in a reasonable amount of time.

General Directions

- Write your name and student ID number in the comment at the top of the Sorter.java file.
- Implement all of the required functions in the Sorter class.
- You should not import anything that is not already included in the file.
- Pay careful attention to the required return types and edge cases.

Deliverables

Submit a zip file of the submission folder and all of its contents. You should zip the actual folder so that when you unzip it, you get a folder called “hw0” containing all of the files. The name of the zip file should be “your_student_id_number.zip”.

Testing

We have provided a small test suite for you to check your code. You can test your code by compiling and running the tester program. If any of the test cases fail, you will be notified. If you pass all of the tests, the output will be blank.

Note that the test suite we will use to grade your code will be much more rigorous than the one provided here (and not necessarily a superset of the provided tests). You should consider making your own test cases to check your code more thoroughly.