Due : Sept. 12 (9/12), 13:00

Overview

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

General Notes

- *Read this homework guideline carefully.* If you do not follow the guidelines, you may receive a 0 regardless of whether your code works or not.

- Do not use any IDEs (Eclipse, IntelliJ IDEA, etc.)

    - We recommend Sublime Text (Linux/Mac/Windows), Atom (Linux /Mac/Windows), Notepad++ (Windows), or TextWrangler (Mac).

    - IDEs often create a "package" of your code, which breaks the auto-grader.

    - **If you know how to fix the package problem**, you can use any IDE you want. However, we will not answer any questions related to this problem since we have already recommended a solution.

- Do not change any method or class signatures. If your code changes any class or method names or signatures, you will receive an automatic 0.

- 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.

- To ensure that your code will be accepted by the autograder, you should submit your code on YSCEC, download it again, recompile it and check the provided test suite. This way, you know that the file you are submitting is the correct one.

- You can use any course materials. However, if you do not cite the source you referred to, it might be checked as copied code. Please write the material (and page) you referred in comments.

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 files in src/main directory.

- Implement all of the required methods.

- You should not import anything that is not already included in the file.

- Pay careful attention to the required return types and edge cases.

- All the codes we provided can be found in src/base directory. If you are unsure what a class/method exactly does, please refer to the code.

- You are free to implement any algorithm that you wish, but you must code it yourself. If you referred to any course materials, you must write the name of the material and page in comments. We will only be testing that your code produces a correct result and terminates in a reasonable amount of time.

Submission Procedure

Submit the files in src/main directory, excluding Main.java, as a zip file. You *must* make a zip file for submission using Gradle build tool (refer to Compiling section).

For this assignment, you should submit only the following two files.:

- Sorter.java
- your_student_id_number.txt

You must rename 2020xxxxxx.txt to your actual student ID number. Inside of that text file, you must include the following text and write your name at the bottom. Please be sure to write all the following text including the last period.

In completing this assignment, I pledge that I have not given nor received any unauthorized assistance.

If this file is missing, you will get 0 on the assignment. It should be named *exactly* your student id, with no other text. For example, *2020123456.txt* is correct while something like *2020123456_pa0.txt* will receive 0.

Compiling

You can test your Java code using the following command.:

```
% ./gradlew -q runTestRunner
```

You can also make a zip file for submission using the following command. The zip file named with your student id (the name of .txt file) will lie in "build" directory.:

```
% ./gradlew -q zipSubmission
```

We provide an empty Main class for testing using standard input/output:

```
% ./gradlew -q runMain
```

Since this file (src/main/Main.java) is not for submission, you can use any packages in the file.

On Windows, use `gradlew.bat` instead of `./gradlew`.

Testing

We have provided a small test suite (src/test) for you to check your code. You can test your code by compiling and running the tester program.

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.