Due : Oct. 10 (10/31), 13:00
Late submission due: Nov. 11 (11/1), 13:00

Overview

This assignment contains one part.

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 However, if you do not cite the source you referred to, it might be checked as copied code.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.

1   Heap

A min (max) heap is a specialized tree-based data structure. A heap is a complete tree that satisfies the following property: for any given node N of the tree and its parent node P, the value of P is less than or equal (greater than or equal) to the value of N. Also, it is an efficient implementation of an abstract data type called a priority queue.

1.1   Max-Heap

You have learned that there are two ways of constructing a heap. In this assignment, you will build a Max-Heap bottom-up consisting of integers. Also, you must implement the Max-Heap data structure in Heap.java file based on a tree using Node.java. The details of each methods are in the source files.

   You must implement the following methods as well as its constructor.

   `insert`: Insert the given item to the heap.

   `removeMax`: Remove and return the item that has maximum value in the heap. If the heap is empty, throw an exception.

   `max`: Return the maximum value in the heap. If the heap is empty, throw an exception.

   `getRoot`: Return the root node's reference of the heap. If the heap is empty, throw an exception.

   `size`: Return the size of heap.

The running time of the construction must be in $O(n)$ and the insert and removeMax methods must be in $O(\log n)$.

1.2   The Oldest Cars

There are n cars, all of which have different travel distances. We want to find the top $k < n$ cars that have the longest travel distances. Use the data structure (Max-Heap) you built in 1.1. You will implement a program to find the top k (given at the construction) cars in the Cars.java file.

You must implement the following methods as well as its constructor:

carDistance: Given a travel distance d of a car, determine whether or not to keep the distance d. This method will be called n times. (Note that you should consider the total time complexity of the program)

getCandidates: Return an array of the top k cars that have the longest travel distances within the travel distances up to this point.

The running time of the whole program must be in $O(n \log k)$ time using $O(k)$ space. (You do not consider the space for inputs.)

General Directions

- Write your name and student ID number in the comment at the top of the files you submit.

- 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 type and edge cases.

- All the codes we provide 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

You *must* make a zip file for submission using Gradle build tool (refer to Compiling section). For this assignment, the zip file will contain only the following three files:

- Heap.java
- Cars.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_pa4.txt* will receive 0.

Compiling

This assignment uses Gradle build tool to automate compiling and testing procedure. The following command will test your Java code against the provided testcases:

        % ./gradlew -q runTestRunner

The following command will zip the files for your submission. The zip file will be named with your student id (the name of .txt file) and will lie in "build" directory. Be aware that the command will be interrupted if your pledge does not comply the guideline.

        % ./gradlew -q zipSubmission

Since the testrunner blocks the standard output from printing, it is hard to test your code fragment while writing the code. For this purpose, we also provide an empty Main class. As this file is not for submission, you may use any features Java provides. The following command will run the Main class instead of the testcases.

        % ./gradlew -q runMain

On Windows, try `gradlew.bat` instead of `./gradlew` if you met an error. Moreover, you may omit the '`-q`' option to review the compile log.

Testing

We have provided small testcases (src/test) for you to check your code. You can test your code by the means mentioned above.

Note that the testcases we will use to grade your code is `very 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.