Data Structures II - HW4

Instructions:

1. Download the hw4-files.zip file from the COL submission folder for HW4. Unzip the file into the src folder that is inside the workspace you created for this class (as you have done on previous homeworks). If you are unsure where that is, inside Eclipse, right click the src folder and select properties (at the bottom).

2. Start up Eclipse. You will probably need to refresh the project before you see the hw4 folder. (Right click on the src folder in the Project Explorer and select refresh.)

3. Open the hw4 package and you will see four files:
   - HW4Test – the JUnit test harness for the assignment
   - Price – stores a price (dollars and cents)
   - PriceQueue – implements a queue of prices using the book's code with a few modifications:
     - **boolean enqueue(Price p)** – prices in the queue must be unique so enqueue will insert p at the front of the queue if it is not already present in the queue and will do nothing if is already present in the queue. Returns true if the price was inserted and false if it was not.
     - **boolean delete(Price p)** – removes the price p from the queue if it is present. Returns true if the price was deleted (was present) and false if was not.
   - WrittenQuestions4.docx – a Word document containing some questions about your solution that you must answer. Don't forget to submit this file with your solution!

4. Currently, all tests in HW4Test pass, except for the timing test. This is because the enqueue and delete methods take linear time. Your task is to modify PriceQueue so that no method takes more than logarithmic time while not changing the input/output behavior of the PriceQueue. In other words, you need to make the PriceQueue faster without changing the answers it gives. In the process, you are allowed to make other methods slower (for example the running of functions that used to be constant could become logarithmic), as long as no method takes more than logarithmic time (except of course the iterator).

5. To do this, you will need to add a single new field to the PriceQueue class that is a TreeMap. (You can find the API for Java's TreeMap class online.) Use the TreeMap to keep track of where prices are in the queue so you can quickly find them. Because delete requires you to find the node **before** the one being deleted, you should map a price to the node behind the price in the queue. **Note:** You will likely need to make a change to the Price class as well.

6. Here is a summary of the requirements:
   - All methods should behave the same after your changes. In other words, all the tests should still pass after you make your changes.

- All methods must run in logarithmic time (or faster) in order to pass the timing test. (Note that all TreeMap operations run in logarithmic time.)
- The only new field you may add is **one** TreeMap variable. You may not add any other fields to the PriceQueue class nor may you make any changes to the Node class inside the PriceQueue class. (If you believe you need to add more fields, we should have a discussion because it probably points to some misunderstanding about the assignment.)
- You may modify the bodies of the methods in the PriceQueue class, but you may not modify the method headers of any of the methods already present.
- You may not change or remove the package declaration at the top of the files.
- You will need to make changes to the Price class as well, but you can only add new functionality. You may not make any changes to the methods that are already present. They must continue to behave as before.

7. Open the WrittenQuestions4.docx file and answer the questions in the file itself. You will need to submit this WrittenQuestions4.docx file on D2L along with your java source files.

8. Once you are satisfied with your solution, submit the Price.java and PriceQueue.java files (the solution requires both) as well as your WrittenQuestions4.docx file and your screenshot in the appropriate submission folder.

9. Double check your submission. Download it and make sure they are the files you intend to submit.

Submission:

Your submission should include the following files:

- Price.java
- PriceQueue.java file
- WrittenQuestions4.docx (after you have answered the questions)
- A screenshot of what happens when you run HW4Test.

Grading:

Each test in the HW4Test file has a number indicating how much that test is worth except for the timing test. Your code **must** pass the timing test (which has no number) to get any credit. (Failing the timing test will result in a grade of 0 regardless of your score on other tests.) If you code passes the timing test, then your score is the sum of the points for all the tests that pass minus any points deducted because of your answers to the written questions (at most 9 points).

Note that on the resubmission, the WrittenQuestions will not be required or graded. Instead the usual 15-point penalty will be assessed. However, the file is still included, because answering those questions can help you get your code correct.