Programming Assignment 4

Due Date: Tuesday 21 March 2023 by 12:30 PM

General Guidelines.

In general, you are not allowed to import any additional classes in your code without explicit permission from your instructor!

Note on academic dishonesty: Please note that it is considered academic dishonesty to read anyone else's solution code, whether it is another student's code, code from a textbook, or something you found online. You MUST do your own work! It is also considered academic dishonesty to share your code with another student. Anyone who is found to have violated this policy will be subject to consequences according to the syllabus and university policy.

Note on grading and provided tests: The provided tests are to help you as you implement the project and (in the case of auto-graded sections) to give you an idea of the number of points you are likely to receive. Please note that the points indicated when you run these tests locally are not your final grade. Your solution will be graded by the TAs after you submit. Please also note that these test cases are not likely to be exhaustive and find every possible error. Part of programming is learning to test and debug your own code, so if something goes wrong, we can help guide you in the debugging process but we will not debug your code for you.

Project Overview.

In this project, you will implement a Max Priority Queue for patient triage using an array-based binary heap.

Specific Requirements

- Make sure your implementation includes all the required methods below (which should be public).
- Make sure your method signatures are correct according to the API given below.

- Your implementation should be in a file called *MaxPQ.java* and the structure backing the PQ should be an array of *Patient* objects.
- Note that the following files are provided for you: *EmptyQueueException.java, MaxPQTest.java, patient_file.txt, Patient.java*
- A note about coding style: It is always a good idea to write separate helper methods for things that are reusable inside your code–I recommend having separate *private* methods for (at the very least): *swapping two elements, resizing the array, sink,* and *swim.* Abstracting these out makes it much easier to read your code, and much easier to debug your code!

Required Methods in MaxPQ.java

Method Signature	Description	Runtime
`MaxPQ()`	`constructor: set the starting capacity of the array to 10`	
`MaxPQ(int cap)`	`constructor: set the starting capacity of the array to <cap>`	
`void insert(Patient item)`	`insert the new Patient into the PQ and re-heapify; before inserting, if the array is full, resize to be twice as large`	`best: O(logN) worst: O(N)--for resizing amortized: O(logN)`
`Patient delMax() throws EmptyQueueException`	`if the PQ is empty, throw the exception; if not, remove and return the max and re-heapify; if the size of the PQ falls below ¼ of the array capacity AND halving it would not be below the initial capacity of the starting array (from constructor), then resize to be half the current capacity`	`best: O(logN) worst: O(N)--for resizing amortized: O(logN)`

`Patient getMax() throws EmptyQueueException`	`if the PQ is empty, throw the exception; if not, return the max`	`O(1)`
`int size()`	`return the number of elements in the PQ`	`O(1)`
`boolean isEmpty()`	`return true if the PQ is empty and false otherwise`	`O(1)`

Grading.

The test cases provided will test the required functions. However, they may not catch every possible error, so additional testing on your end may be necessary.

Your code will also be checked manually, and additional deductions may be made based on that, including, but not necessarily limited to:

- not following instructions (up to 100% deduction)
- wasting space (up to 25% deduction)
- wasting time (up to 25% deduction)
- bad coding style/hard-to-read code (up to 10 points)

In order to be graded & receive any credit, your project must be submitted properly, and it must compile and run on lectura, so make sure you test that before you submit.

Submission.

After transferring and testing your code on lectura, use the following turnin command.

```
turnin cs345p4 MaxPQ.java
```

Useful terminal commands:

javac	Use this for compiling java files. You can follow it with specific .java file names OR you can follow it with *.java to compile all the .java files in the current directory.
java	Use this to run a compiled java file. I recommend following it with the .java file name, which will let you know if there are any accessibility errors. e.g. java MaxPQTest.java
ls	Use this to list the files in your current directory.
cat	Use this to see the contents of a file.
emacs	If you use a mac, you can open and edit a file using this command if you follow it with the name of the file you want to edit.
vi	Same as emacs, but it is a different text editor.
cp	Use this to copy a file from one place to another. e.g cp src/MaxPQ.java ./ The above command will copy the file in the src directory called MaxPQ.java to the current directory.
mv	Use this to move a file from one place to another. The usage is basically the same as cp except it will actually move the file instead of just copying it.
rm	Use this one very cautiously! It can be used to delete files, but it is very hard (if not impossible) to recover files deleted in this way, so be very careful. e.g. rm MaxPQ.java The above command will remove MaxPQ.java permanently. e.g. rm *.java The above command will remove all the .java files in the current directory

	permanently.
cd	Use this to change the current directory. e.g. cd /local/cs345/ The above command would navigate you to the above directory (if it exists).

Regrade Requests/Resubmissions

See the syllabus and read the announcement on D2L when the grades are released.