Programming Assignment #3

125 points

Due Date/Time:

Your project is due on Thursday April 5th at 8:00 A.M. Hardcopy of your Java source code file and report are also due at that same time.

The Program:

For this assignment, you will write one more implementations of a **`Priority Queue`**. Using the same interface as program #1, you will write a binary heap implementation. Additionally, you will write a report (details follow).

Your heap implementation must have identical behavior, and must implement the `PriorityQueue` interface used in program #1. The implementation must have two constructors as in the first assignment--use the `DEFAULT_MAX_CAPACITY` in the interface for one constructor and take a size parameter for the second.

Thus, your project will consist of the following files. You must use exactly these filenames.

- `PriorityQueue.java` The ADT interface (provided in prog1)
- `BinaryHeapPriorityQueue.java` The binary heap implementation.

Additional Details:

- Each method must be as efficient as possible. That is, a O(n) is unacceptable if the method could be written with O(log n) complexity. Both `insert` and `remove` must be O(log n)
- Your BinaryHeap must be stable--able to determine the oldest entry among those with identical priority.
- Your iterator must be fail-fast.
- You may not make any modifications to the `PriorityQueue` interface provided. I will grade your project with my copy of this file. This interface is UNCHANGED from projects #1 and #2
- All relevant requirements fro m the first assignment apply here.

The Report

You will provide complexity analysis for the `insert(E obj)` and `remove()` methods in all five implementations (10 methods total). You will also run empirical timing tests on all five of your implementations to provide evidence that your two methods perform at the expected complexity. You should graph the runtime results from the 10 methods and provide written justification for you results. Attach your report to your Binary Heap source code to make a single submission package; do not give me two items.

Turning in your project:

To submit your project, you must copy your Java source code files into your `.handin/prog3` subdirectory. You will submit a printout of this file in class on the due date.*[**IMPORTANT NOTE:** Do not recreate the data_structures subdirectory in the handin subdirectory--just copy your file into the* `handin/prog3` *directory itself.]* Be sure to check the Program Submission Guidelines below. **Attach your report to your source code printout; do not give me two items.** Late programs may be turned in up to one class meeting after the due date with a late penalty of 20%.

Cheating Policy

There is a zero tolerance policy on cheating in this course. You are expected to complete all programming assignments on your own. Collaboration with other students in the course is not permitted. You may discuss ideas or solutions in general terms with other students, but you must not exchange code. During the grading process I will examine your code carefully. Anyone caught cheating on a programming assignment (or on an exam) will receive an "F" in the course, and a referral to Judicial Procedures.

General Submission Guidelines

Note that your program **will NOT compile in your handin subdirectory**. If it does, then you have done something wrong and you will receive a zero for the assignment. This directory is only a place to turn in your files, not a suitable location for project development. Do not create any package folders within the `handin/` directory tree.

To be collected for grading, your program file must have the correct name and be in the correct location. If the grading script fails to find your program file, you will not get any credit for the assignment.

Programs must compile with no errors. Programs that fail to compile will receive very little or no credit. Programs will be compiled on edoras using the Sun JDK 1.8 (or JDK 8) compiler.

The timestamp on your file will be used to determine the date and time when your project was submitted. The act of copying your program files into your `handin/` subdirectory constitutes submission of your program for grading. You must not place incomplete programs in your `handin/` subdirectory. At any time after the due date, programs may be collected from this `handin/prog3/` subdirectory. Once collection has occurred, you may not resubmit.

IMPORTANT:
Resubmission of projects is not allowed. Once your program has been collected for grading, no further changes or resubmissions will be permitted. On rare exceptions (as determined by me) where resubmission is allowed due to packaging or other trivial errors on your part, the minimum grade penalty will be 15%.

Also, you must not open or edit your program file after the due date, as this will change the timestamp on your file, making it late. You may print your files or view them without modifying the timestamp. Use the UNIX command '`view FILENAME`' to run the read-only version of the text editor `vi`.

A hardcopy printout of your source code files will be due in class on the due date. Although the timestamp on your file is used to determine that your program was completed on time, there may be a grade penalty for hard copy submissions not turned in at the beginning of class on the due date. In all cases, the printed copy of your program must exactly match the file in your `handin/`subdirectory. If you submit your program late, the hardcopy is due at the next class meeting after submission. Be sure to staple all of your code into a single package. Do not submit loose pages or multiple bundles.