

Assignment 6

Due: April 4th, 2013 before 8:30pm

Objectives

- More practice implementing an interface in Java
- Exposure to the Priority Queue ADT
- Practice implementing the Heap ADT
- Exposure to Comparable interface
- Exposure to inheritance

Introduction

This assignment asks you to implement the Priority Queue interface using a Heap data structure. In this assignment, the heap will store objects that implement the Comparable interface.

You've been provided with a LinkedList implementation of the Priority Queue interface so that you can run the tester against both implementations and see that the Heap implementation provides a significant increase in performance.

Quick Start:

1. Download `a6tester.java`, `PriorityQueue.java`, `HeapPriorityQueue.java`, `LinkedPriorityQueue.java`, `ComparableNode.java`, `HeapEmptyException.java` and `HeapFullException.java`
2. Read the comments in `HeapPriorityQueue.java` carefully
3. Compile and run the test program `a6tester.java`
4. If no errors reported by test program, see the grading section of this document
5. Implement one of the methods in `HeapPriorityQueue.java`
6. goto 3

Understanding the test program: a6tester.java

You've been given a program that will test your implementation of the PriorityQueue interface.

One of the first things you should do after downloading the source code files is to run the test program.

Compile the test program by typing:

```
javac a6tester.java
```

Run the test program by typing:

```
java a6tester
```

You should see the following output:

```
Testing Heap implementation.  
Basic testing of size, isEmpty  
Failed test: 0 at line 53
```

The tester is reporting that your implementation is failing the very first test. This is hardly surprising, since you haven't written any code yet!

The tester is written to be able to test both a linked list implementation (provided to you in the file `LinkedPriorityQueue.java`) and the heap implementation that you are required to implement in `HeapPriorityQueue.java`

To see the result of testing your instructor's linked list based solution, type:

```
java a6tester linked
```

There are 51 test cases for the linked list implementation and 52 test cases for the heap implementation.

There is one fewer test case for the linked list implementation because one of the test cases is using the priority queue to sort random integer values and the resulting algorithm is too slow for 1 million elements.

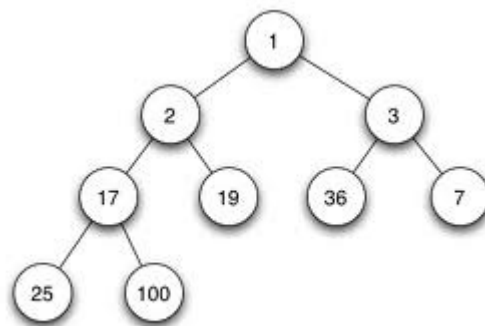
Heaps

Your implementation of the Priority Queue interface must use the Heap data structure.

The Heap data structure has been discussed at length in class, and is also covered in detail in the text book. If you'd like a third source, the Wikipedia page might be helpful:

[http://en.wikipedia.org/wiki/Heap_\(data_structure\)](http://en.wikipedia.org/wiki/Heap_(data_structure))

A reminder that when implementing heaps, the complete tree is stored in an array. For example, the following minimum heap:



Would be stored in an array:

0	1	2	3	4	5	6	7	8	9
	1	2	3	17	19	36	7	25	100

Notice that position 0 isn't used. Using this representation, given a node at position i in the array:

i 's parent is at $i / 2$

i 's left child is at $i * 2$

i 's right child is at $i * 2 + 1$

The sample code in `HeapPriorityQueue.java` includes comments on how to complete this assignment. Be sure to read them!

Submission

Submit your `HeapPriorityQueue.java` using Connex. **Please be sure you submit your assignment, not just save a draft.**

A reminder that it is OK to talk about your assignment with your classmates, and you are encouraged to design solutions together, but each student must implement their own solution.

We will be using plagiarism detection software on your assignment submissions.

Grading

If you submit something that does not compile, you will receive a grade of 0 for the assignment. It is your responsibility to make sure you submit the correct files.

Requirement	Marks
You submit something that compiles	1
Your code passes all the test cases in <code>a6tester.java</code>	9

Total 10

To be clear about the above guidelines – you must pass all test cases to score higher than 10% on this assignment. Please start early!