Skill Problem B: Heap with Deletion

The primary (public)operations on an ordinary (min) heap are insertItem and removeMin. For this problem, you must create an enhanced heap that also supports deletion of elements (by key) in the heap. The enhanced heap should support insertItem and removeMin, with the same functionality as in an ordinary heap but should now support deleteItem as well. (The operation deleteItem(Object k) will delete the first node in the heap that stores the key k – if there are several nodes that have k as a key, only one will be deleted.) All operations on the enhanced heap must have O(log n) running time. For your enhanced heap, it will be useful to make use of a hashtable to keep track of keys and the nodes they are stored in. In the code shell provided, a hashtable is provided as an instance variable. This problem is worth 9 points. For full credit you must do both parts of the problem, described below – each part is worth 4.5 points.

1. [4.5 points] In the Skill Problems folder, you will see a Java file Heap.java. This file is a shell for the heap implementation you need to do. The expected implementation of the insertItem and removeMin methods is the implementation of these operations discussed in class (and also in the text for the course). You will need to do a bit more work in each of these operations to keep the hashtable that you are using up to date.

   The file Heap.java contains a nested class BtreePrinter which is designed to give a visual representation (in the console) of your heap. To use it, call the printAsDiagram method that is provided. This will help you test your code.

   Include in a separate file unit tests that test the operations insertItem, removeMin, and deleleItem. Your tests should provide convincing evidence that your heap is working properly.
2. [4.5 points] In a separate document, explain in English how your implementation of deleteItem works. Also, you must explain clearly why:
   a. deleteItem runs in O(log n)
   b. insertItem and removeMin still run in O(log n) even though they will now be maintaining the hashtable

What to turn in:

- Your implementation of Heap.java, along with unit test code
- A document that addresses the points raised in part 2.