Lab 14

CSE 274

I. THE `DELETE` METHOD

A draft of `delete` method is provided in the `Application.java` file. In this draft the body of some `if` statements are missing. Complete development of `delete` method by filling in the body of those `if` statements. To test the developed method, copy the following lines of code into the body of the `main` method and run the application:

```
Heap myHeap = new Heap();
myHeap.add(3);
myHeap.add(1);
myHeap.add(19);
myHeap.add(25);
myHeap.add(4);
myHeap.add(9);
myHeap.add(18);
myHeap.add(7);
myHeap.add(6);
myHeap.add(5);
while(!myHeap.isEmpty())
      System.out.println(myHeap.delete());
```

The expected output is printed below:

```
25
19
18
9
7
6
5
4
3
1
```

II. HEAP SORTING

From above, calling the `delete` method of the heap results in the values of the heap being `return`-ed in a decreasing order. This result suggests that a heap can be used to sort a set of values. More precisely, the values are added to the heap one by one by calling the `add` method of the `Heap` class. Once all the values are added to the heap, the `delete` method is called over and over until the heap becomes empty. The values `return`-ed from the `delete` method are sorted.
The time complexity of `add` and `delete` methods are both O(log(n)). The time complexity of adding n values to the heap is O(nlog(n)). Also, the time complexity of deleting n values from the heap is O(nlog(n)). Accordingly, the time complexity of sorting n values with a heap data structure is O(nlog(n)). Sorting numbers with a heap is referred to as heap sorting.

III. SUBMITTING THE ASSIGNMENT

When submitting your response to this assignment keep the above lines of code into the body of the `main` method.