UNIVERSITY OF VIRGINIA, DEPARTMENT OF COMPUTER SCIENCE

Priority Queues - Heap and Heap Sort

Nada Basit and Mark Floryan

March 8, 2022

1 SUMMARY

For this homework, you will be building a custom MinHeap class as discussed in class. You will use this heap to implement an efficient HeapSort algorithm.

1. Download the provided starter code.
2. Implement the MinHeap.java class.
3. Implement the heapSort() method inside the file *HeapSort.java*.
4. Test the correctness of your Min-Heap using the provided tester.
5. **FILES TO DOWNLOAD:** heaps.zip
6. **FILES TO SUBMIT:** MinHeap.java, HeapSort.java

1.1 MINHEAP.JAVA

First, you will be implementing the *MinHeap.java* class, which implements the following PriorityQueue interface:

```java
public interface PriorityQueue<T extends Comparable<T>> {

    /* places the value T onto the heap */
    public void push(T data);

    /* removes and returns the item with next priority
     * (i.e., lowest value)
     */
    public T poll();

    /* returns the next item to be polled, without removing */
    public T peek();

    /* returns number of elements on the heap */
    public int size();

}
```

Your MinHeap class will also have a few extra supporting methods necessary to make it work. You heap MUST be a heap implementation, as discussed in class, using an array (no linked list / tree like implementations). The other methods necessary are:

```java
// Constructs a heap from the given array
// pre-filled with the data in the heap
// data may need to be restructured
public MinHeap(ArrayList<T> data);

// Turns the internal array without
// heap ordering property into the
// equivalent heap with the ordering property
private void heapify();

// Percolate the item at index up until
// the ordering property is restored
private void percolateUp(int index);

// Percolate the item at index down until
// the ordering property is restored.
private void percolateDown(int index);
```

Once you are done, you can test your implementation using the provided tester. Remember that the tester is NOT meant to be a thorough check of your implementation. You should still write your own tests to make sure that it works. Also note that not all the tests will pass at this point, because you haven't implemented HeapSort yet (see next section).

1.2 HEAP SORT

Heap Sort is an algorithm that sorts a list using the methods from a MinHeap or MaxHeap. In the provided *HeapSort.java* file, you'll find a method called heapSort(). Implement this method so that it uses the MinHeap you built to sort the provided list of numbers. Once you are done, you can re-run the provided tester to confirm that it seems to work. *Note: This method returns void and is meant to alter the provided list of numbers so that it changes to be sorted. You will not be returning anything here.*

1.3 GRADESCOPE

You should submit your code to *Gradescope*. If you are having trouble with your submission, you should double check the following common problems:

1. Make sure you are only submitting two files, and they are called *MinHeap.java* and *HeapSort.java* exactly.

2. Make sure you keep the package statement at the top of the files.

3. Your code should **NOT contain any additional import statements**.

4. Make sure your file is **NOT printing out anything extra to the console**. This will confuse the Gradescope autograder.