Lab 5 - Implementation of Max Heaps and Heapsort

Due Date: Lab Sessions March 15 – March 28, 2020
Assessment: 4% of the total course mark.

Description:

In this lab you will write a `Java` class `MaxHeap` implementing MAX binary heaps, the heapsort algorithm, and a test class `TestMaxHeap`. The heaps will store objects of type `Integer` and must be implemented using an array. Each heap may contain items with equal integer value. Then, the value in any tree node must be larger than or equal to the values in any of its descendants.

Specifications:

- Class `MaxHeap` must contain a field of type `Integer[]`, which is a reference to the array which stores the items (references to objects of type `Integer`). There must be other fields to store the size of the array (the amount of memory allocated) and the size of the heap (number of items stored). All fields must be `private`. Pay attention to update accordingly these fields when performing the heap operations (when necessary).
- Class `MaxHeap` must contain **at least** the following constructors:
  - A `public` constructor which creates an empty heap. The size of the array to be allocated must be passed as a parameter.
  - `public MaxHeap(Integer[] someArray)` - creates a `MaxHeap` which stores the items from the input array.
- Class `MaxHeap` must contain at least the following `public` methods:
  1) `public void insert(int n)`: Inserts the value `n` in `this` max heap (inserts an object of type `Integer` representing `n`). Duplicates are allowed. If there is no room for the insertion in the current array storing the items, then an array of double size has to be allocated and all items are copied into the new array, after which the insertion is performed.
  2) `private int deleteMax()`: Removes the item with the largest value and returns its value.
  3) `public String toString()`: Returns a string representing the sequence of integer values stored in the heap, in the order they are stored in the array, i.e., in level order.
  4) `public static void heapsort(Integer[] arrayToSort)`: Applies heapsort algorithm to sort the input array. You may construct a `MaxHeap` object storing the items (using the second constructor), apply the delete max repeatedly on this heap, then copy back the items in the array in sorted order.
  5) Accessor public methods (`get` methods) to allow the user to see the values of the fields.

- The test class has to be designed such that to carefully check that all specifications are met. During testing instructive messages should be printed to clarify what is tested (i.e., your method performing the tests should print such messages). You may write additional methods in your `MaxHeap` class to aid in testing, if necessary.

NOTES:

To get credit for the assignment you have to demonstrate your code in front of a TA during your lab session. A 50% penalty will be applied for late demo. A 25% penalty will be applied if the demo is on time, but the electronic submission is late.

You are additionally required to compute the asymptotic run time and space complexity of all methods you develop in this lab. The TA will ask you to explain your implementation and run time for your methods as part of your demo.

SUBMISSION INSTRUCTIONS:

NO REPORT IS NEEDED. Submit the source code for each of the classes `MaxHeap` and `TestMaxHeap` in a separte text file. Include your student number in the name of each file. For instance, if your student number is 12345 then the files should be named as follows: MaxHeap12345.txt, TestMaxHeap12345.txt, etc. Submit the files in the Assignments Box on Avenue by the end of your designated lab session.