CSCI 230 HW#8

Collaboration policy: **Individual Assignment**
Total Points: **100**

Source Code

The only Java classes provided in the zip file attached to this Dropbox assignment are:

- `Utils.java`
- `HeapTest.java`

Under no circumstances are you allowed to modify or create new `Utils` class. For this assignment, you will reuse the `SinglyLinkedList.java`, `ArrayList.java`, `Node.java`, and `List.java` files. You must use these files **as is**.

You **may only** modify the `Utils` class. In particular, in this class you **may only** modify the methods listed in Part 1, and under no circumstances are you allowed to remove, add, or modify any other line of code in these classes (this include instance variables, class variables, constants, etc.).

Lastly, you **may not** change the package structure! Specifically, `edu.cofc.cs.csci230` cannot be removed or modified. If a solution is submitted with a different package structure, it will not be graded, no exceptions.

Part 1

In the `Utils` class please fully implement the methods listed below:

- `public static <AnyType extends Comparable> void maxHeapify( List<AnyType> list ) throws IndexOutOfBoundsException`
- `public static <AnyType extends Comparable> void minHeapify( List<AnyType> list ) throws IndexOutOfBoundsException`
- `public static <AnyType extends Comparable> void heapSort( List<AnyType> sorted_list, List<AnyType> list, Boolean increasing ) throws IndexOutOfBoundsException`

In each method, you will see a TODO comment, this is where you add your code. In the provided source code, comments are provided; please ensure you read them carefully. Additionally, the course notes (Heap_and_Heapsort.pdf) and Chapter 6 in the supplemental course textbook (Chap6_Levintin.pdf) have been placed on OAKs in the content section.

Part 2

The provided `HeapTest` class has a main method with one example test case for each list. In the main please add additional test cases that demonstrate you have fully evaluated the

operational correctness of the methods you implemented in Part 1. To receive full credit, these test cases **must** be included.

Submission

Create a zip file that **only** includes the completed `Utils.java` and `HeapTest.java` file. The name of the zip file must be your last name. For example, *ritchie.zip* would be correct if the original co-developer of UNIX (Dennis Ritchie) submitted the assignment. Only assignments submitted in the correct format will be accepted (no exceptions). Please submit the zip file (via OAKS) to the Dropbox setup for this assignment by the due date. You may resubmit the zip file as many times as you like, Dropbox will only keep the newest submission.

Per the syllabus, late assignments will not be accepted – no exceptions. Please do not email Paul or I your assignment after the due date, we will not accept it.

Grading Rubric

Utils Compiles	10 points
Thoroughness of your test cases in the HeapTest class	10 points
Instructor test cases. Several random test cases that sort random values (may include duplicates) in both the ArrayList and SinglyLinkedList data structures. In total these will be 80 points.	80 points
	100 points

In particular, each data structure will be graded as follows. If the submitted solution

- Does not compile: 0 of 100 points
- Compiles but does not run: 10 of 100 points
- Thoroughness of your test cases: 20 of 100 points
- Passes test cases developed by instructor: 100 of 100 points.