This assignment will be to create an instrumentation test to determine operation efficiency for different data structures.

Tested operations should include

Collection equality

Addition of a new element (and at location if possible)

Removing an element (and at location if possible)

Size

Contains an element

Change an element

Sorting

Sorting based on a custom comparator

Each of these operations should be performed 1 million times to average out the timing of the operation.

You should use the following structures.

List<Integer> al = **new** ArrayList<>();  
List<Integer> ll = **new** LinkedList<>();  
  
Set<Integer> ts = **new** TreeSet<>();  
Set<Integer> hs = **new** HashSet<>();  
  
Map<Integer, Integer> hm = **new** HashMap<>();  
Map<Integer, Integer> tm = **new** TreeMap<>();

You may rename the variables to your liking.

You should be efficient in your code design for implementation and use polymorphism to minimize coding and maximize code reuse.

Note: Not all operations will work for each data structure, but you should examine the API to determine to which structures and which operations this applies and be able to explain to the grader.

**Grading:**   
Correctness: You can lose up to 20% if your solution is not correct   
Quality: You can lose up to 20% if your solution is poorly designed   
Testing: You can lose up to 20% if your solution is not well tested   
Explanation: You can lose up to 40% if you cannot explain your solution during the grading session