Zeen Wang (001082883)

**Program Structures & Algorithms**

**Fall 2021**

**Assignment No. 2**

* **Task**

**Benchmark**

1. Implement three methods of a class called Timer.
2. Implement InsertionSort (in the InsertionSort class) by simply looking up the insertion code used by Arrays.sort. If you have the instrument = true setting in test/resources/config.ini, then you will need to use the helper methods for comparing and swapping (so that they properly count the number of swaps/compares). The easiest is to use the helper.swapStableConditional method, continuing if it returns true, otherwise breaking the loop. Alternatively, if you are not using instrumenting, then you can write (or copy) your own compare/swap code. Either way, you must run the unit tests in InsertionSortTest.
3. Implement a main program (or you could do it via your own unit tests) to actually run the following benchmarks: measure the running times of this sort, using four different initial array ordering situations: random, ordered, partially-ordered and reverse-ordered. I suggest that your arrays to be sorted are of type Integer. Use the doubling method for choosing n and test for at least five values of n. Draw any conclusions from your observations regarding the order of growth.

* **Relationship Conclusion:**

As the N increase doubling, the running time gradually increase. But for the ordered array, it is almost not increase like linear.

The total running time “Reversed” > “Random” > “Partially – Ordered” > “Ordered”.

* **Evidence to support the conclusion:**

1. **Output**

Ordered  
**Text

Description automatically generated**  
Partially – Ordered  
**Text

Description automatically generated**  
Random  
**Text

Description automatically generated**  
Reversed  
**Text

Description automatically generated**

1. **Graphical Representation**

* **Unit tests result:**Graphical user interface, text, application

  Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text

Description automatically generated