**Rongjin Dang (001565800)**

**Program Structures & Algorithms**

**Fall 2021**

**Assignment No. 2**

* **Task (List down the tasks performed in the Assignment)**

**Your task for this assignment is in three parts.**

**(Part 1) You are to implement three methods of a class called Timer. Please see the skeleton class that I created in the repository. Timer is invoked from a class called Benchmark\_Timer which implements the Benchmark interface. The APIs of these class are as follows:**

**(Part 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.**

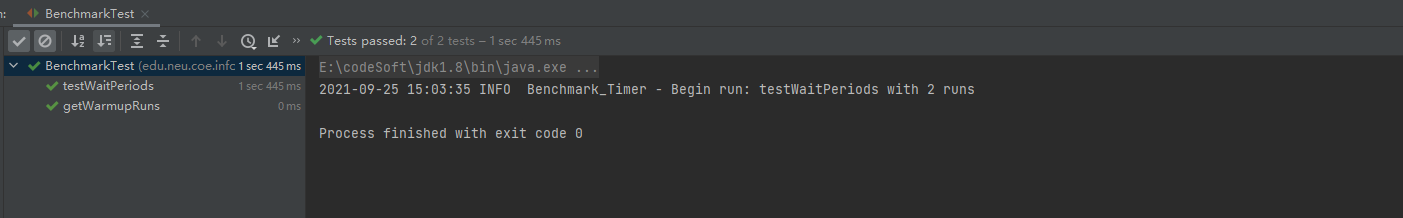
**(Part 3) Implement a main program (or you could do it via your own unit tests) to actually run the following benchmarks: measure the running times of this sort, using four different initial array ordering situations: random, ordered, partially-ordered and reverse-ordered. I suggest that your arrays 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.**

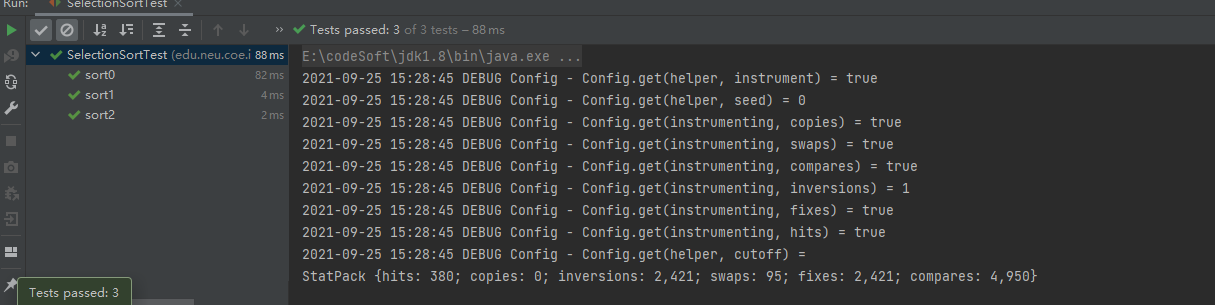
* **Relationship Conclusion: (For ex : z = a \* b)**

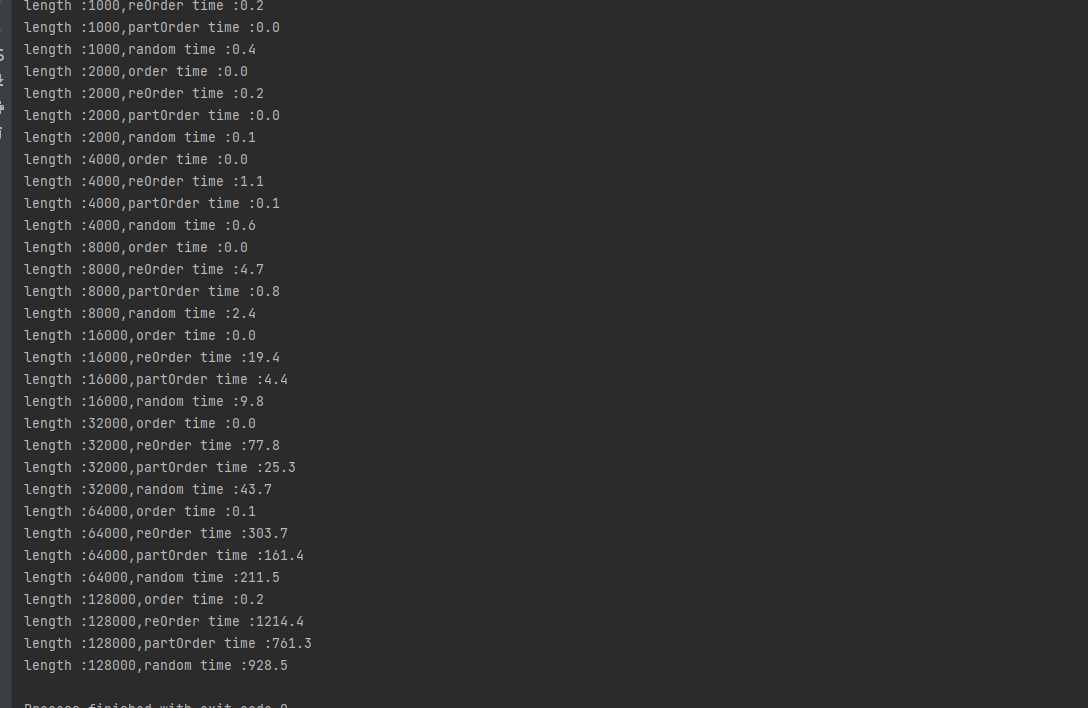
**When the length of the array to be sorted doubles, the time it takes to sort is Quadruple**

* **Evidence to support the conclusion: Graphical user interface, text, application

  Description automatically generated**

****

****

****

1. **Output (Snapshot of Code output in the terminal)**
2. **Graphical Representation(Observations from experiments should be tabulated and analyzed by plotting graphs(usually in excel) to arrive on the relationship conclusion)**

* **Unit tests result:(Snapshot of successful unit test run)**