**INFO 6205**

**Program Structures & Algorithms**

**Fall 2020**

**Assignment No 2**

* **Task**

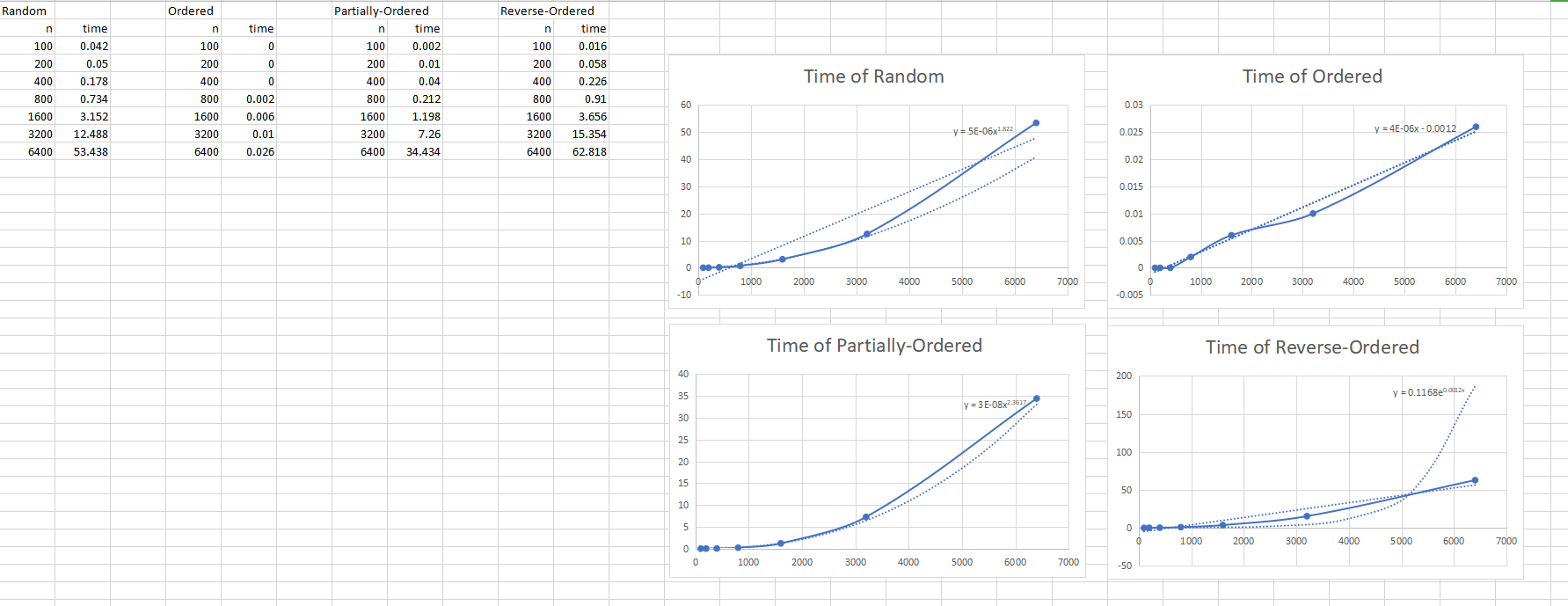
(Part 1) You are to implement four methods of a class called *Timer*. Please see the skeleton class that I created in the repository.

(Part 2) Implement *InsertionSort*(in the *InsertionSort* class) by simply looking up the insertion code used by*Arrays.sort.* You should use the *helper.swap* method although you could also just copy that from the same source code. In the *main* method of *Benchmark*, remove the reference to *SelectionSort*.

(Part 3) 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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Random | | Ordered | | Partially-Ordered | | Reverse-Ordered | |
| n | time | n | time | n | time | n | time |
| 100 | 0.042 | 100 | 0 | 100 | 0.002 | 100 | 0.016 |
| 200 | 0.05 | 200 | 0 | 200 | 0.01 | 200 | 0.058 |
| 400 | 0.178 | 400 | 0 | 400 | 0.04 | 400 | 0.226 |
| 800 | 0.734 | 800 | 0.002 | 800 | 0.212 | 800 | 0.91 |
| 1600 | 3.152 | 1600 | 0.006 | 1600 | 1.198 | 1600 | 3.656 |
| 3200 | 12.488 | 3200 | 0.01 | 3200 | 7.26 | 3200 | 15.354 |
| 6400 | 53.438 | 6400 | 0.026 | 6400 | 34.434 | 6400 | 62.818 |

* **Output** (few outputs to prove relationship)
* **Relationship conclusion and Evidence** (screen shot and/or graph and/or spreadsheet)

****

* **Screenshot of Unit test passing**

