Bowen Jiang (NUID: 001582174)

INFO 6205

Program Structures & Algorithms

Fall 2020

Assignment 2

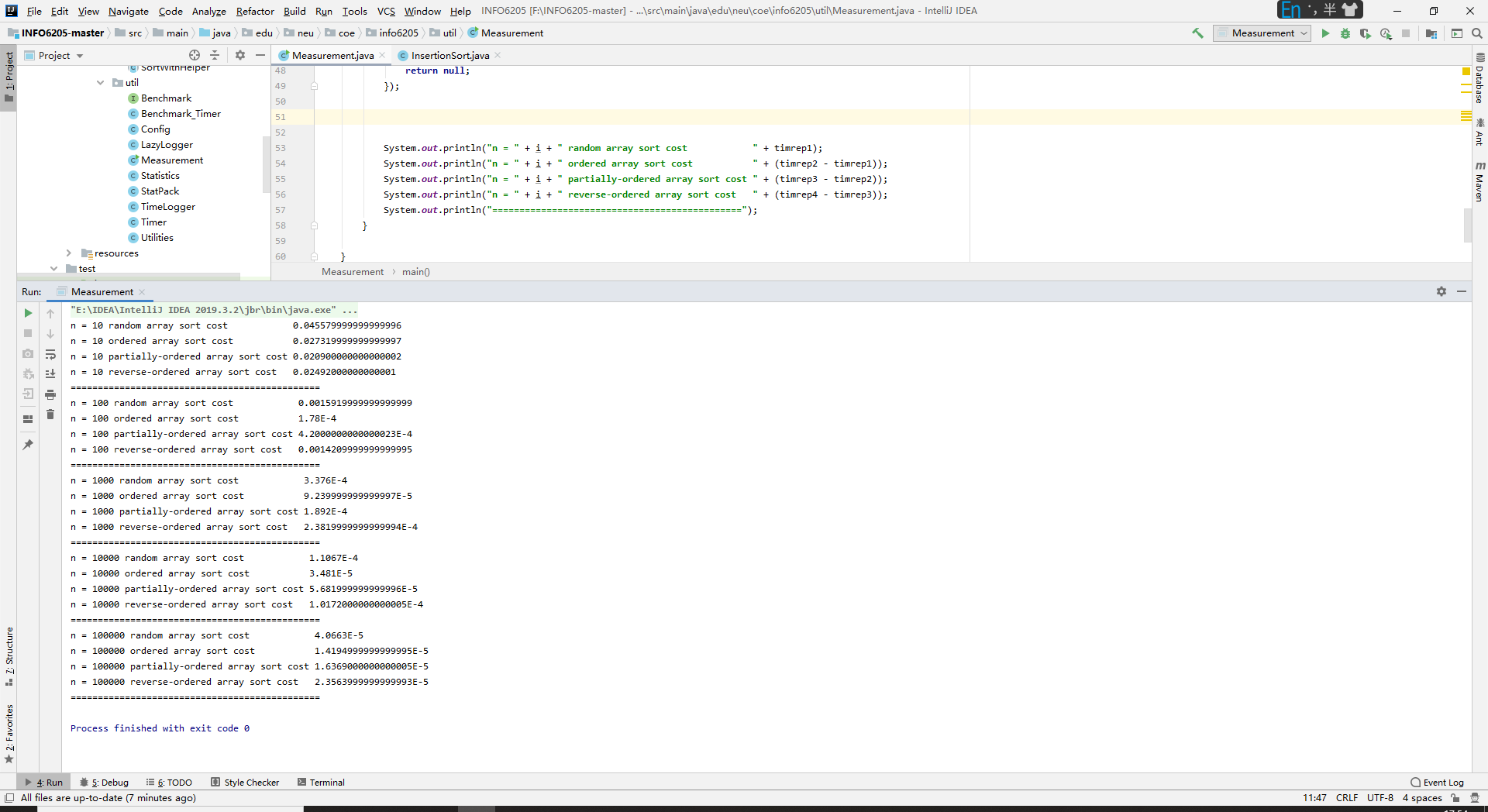
Task: Your task for this assignment is in three parts.

(Part1) You are to implement four 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.

(Part2) 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*.

(Part3) 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.

**Output**



**Relationship Conclusion**

When n = 10, Sorting time of four arrays :

Partially-ordered < Reverse-ordered < Ordered < Random

When n = 100, Sorting time of four arrays :

Ordered < Partially-ordered < Reverse-ordered < Random

When n = 1000, Sorting time of four arrays :

Ordered < Partially-ordered < Reverse-ordered < Random

When n = 10000, Sorting time of four arrays :

Ordered < Partially-ordered < Reverse-ordered < Random

When n = 100000, Sorting time of four arrays :

Ordered < Partially-ordered < Reverse-ordered < Random

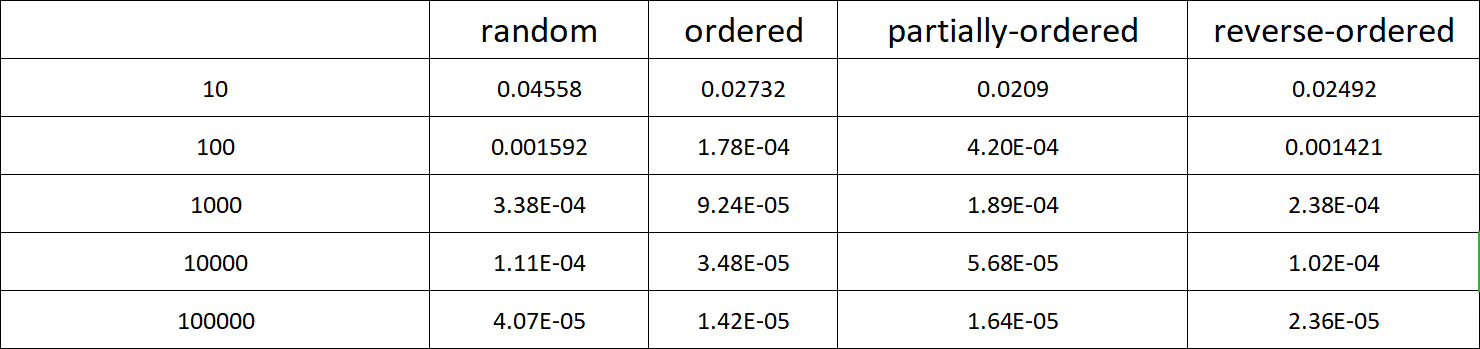
After the observations we can get the conclusion that if n is bigger enough, the sorting time of four arrays is

Ordered < Partially-ordered < Reverse-ordered < Random

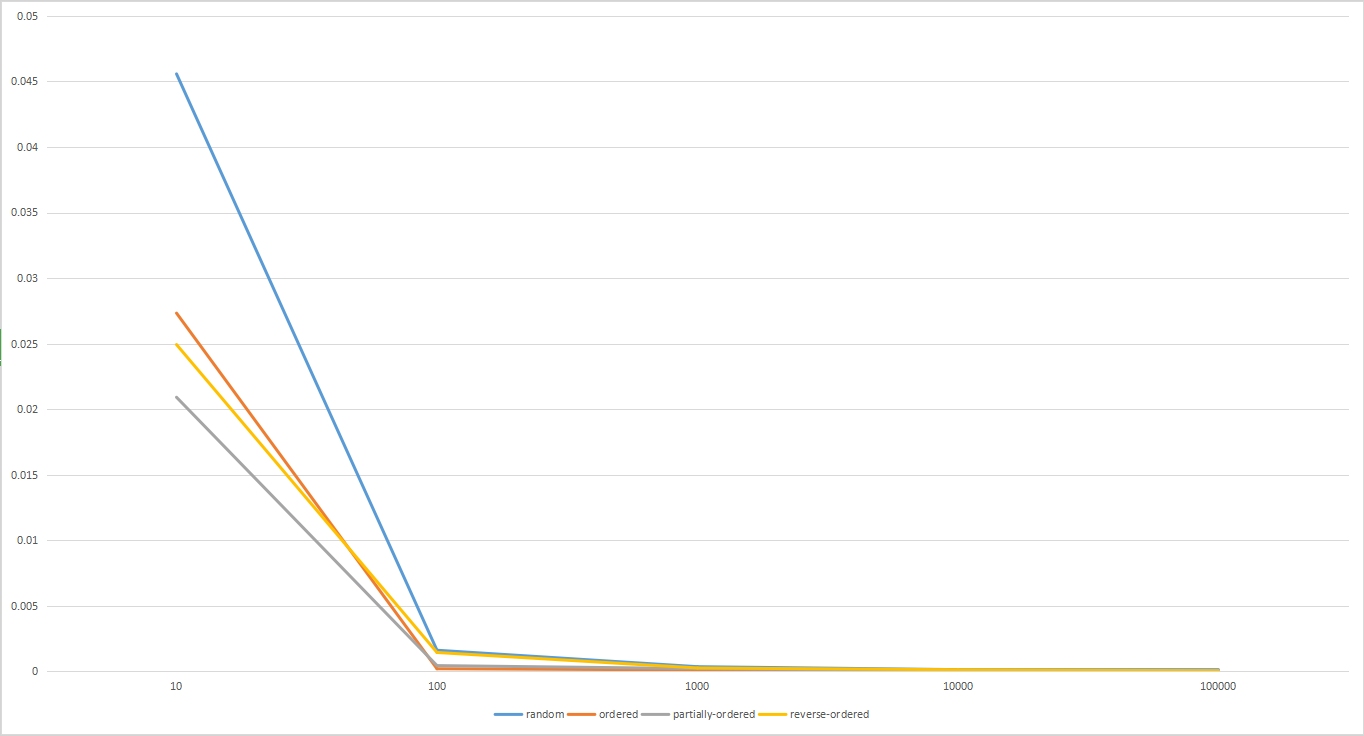
**Evidence to support relationship**

Use 10, 100, 1000, 10000, 100000 as different values of n.

The table shows my experimental data.

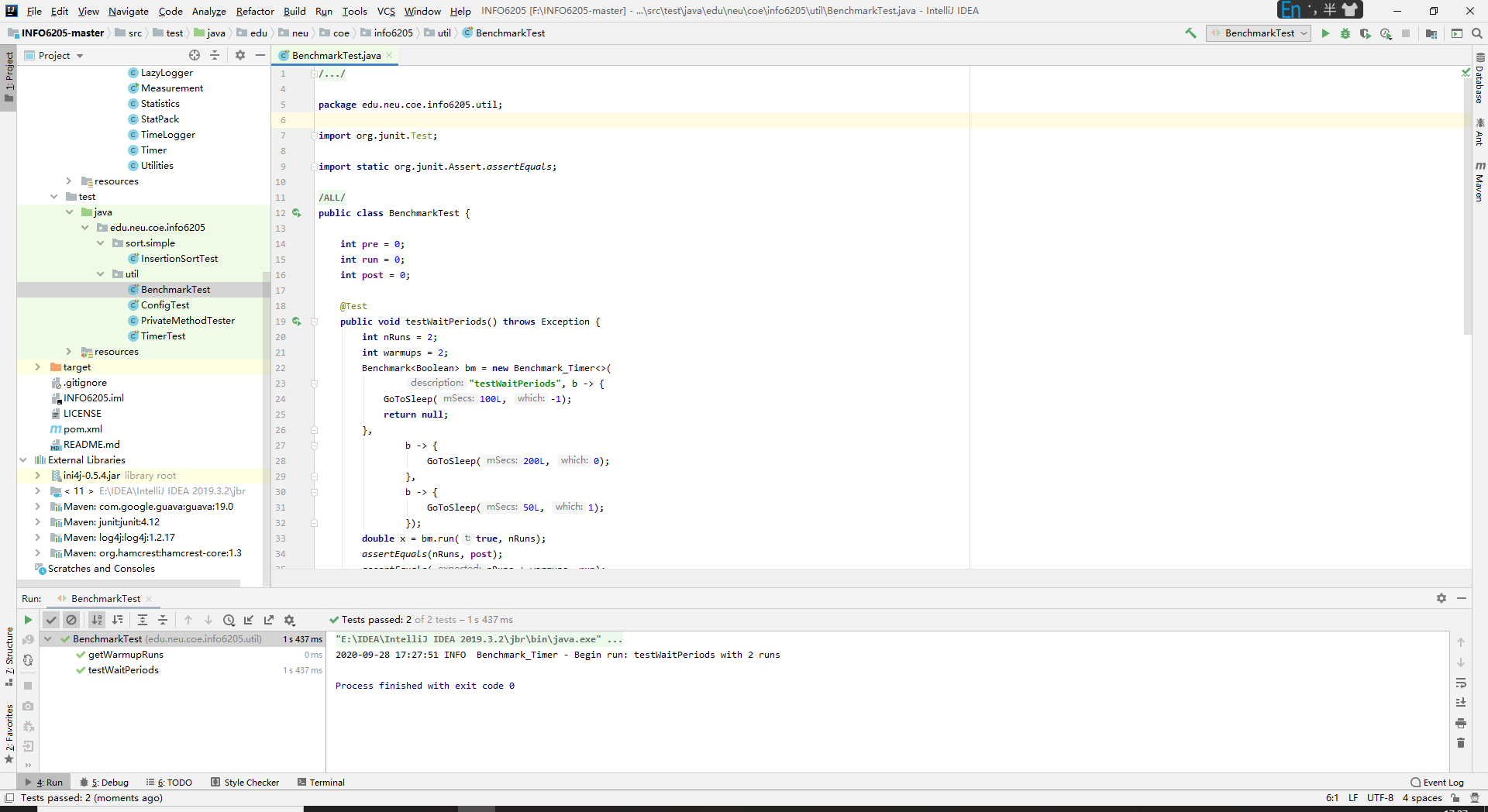


And I use a line chart to analyze the relationship between steps and distance.



**Screenshot of Unit test passing**

BenchmarkTest.java



TimerTest.java

