# Assignment2

**Yutong Wang 001530602**

1. **Task**

(part 1)  Implement three methods of a class called *Timer, then* check implementation by running the unit tests in *BenchmarkTest*and*TimerTest*.

(part 2) Implement *InsertionSort* (in the *InsertionSort* class) by simply looking up the insertion code used by *Arrays.sort.*

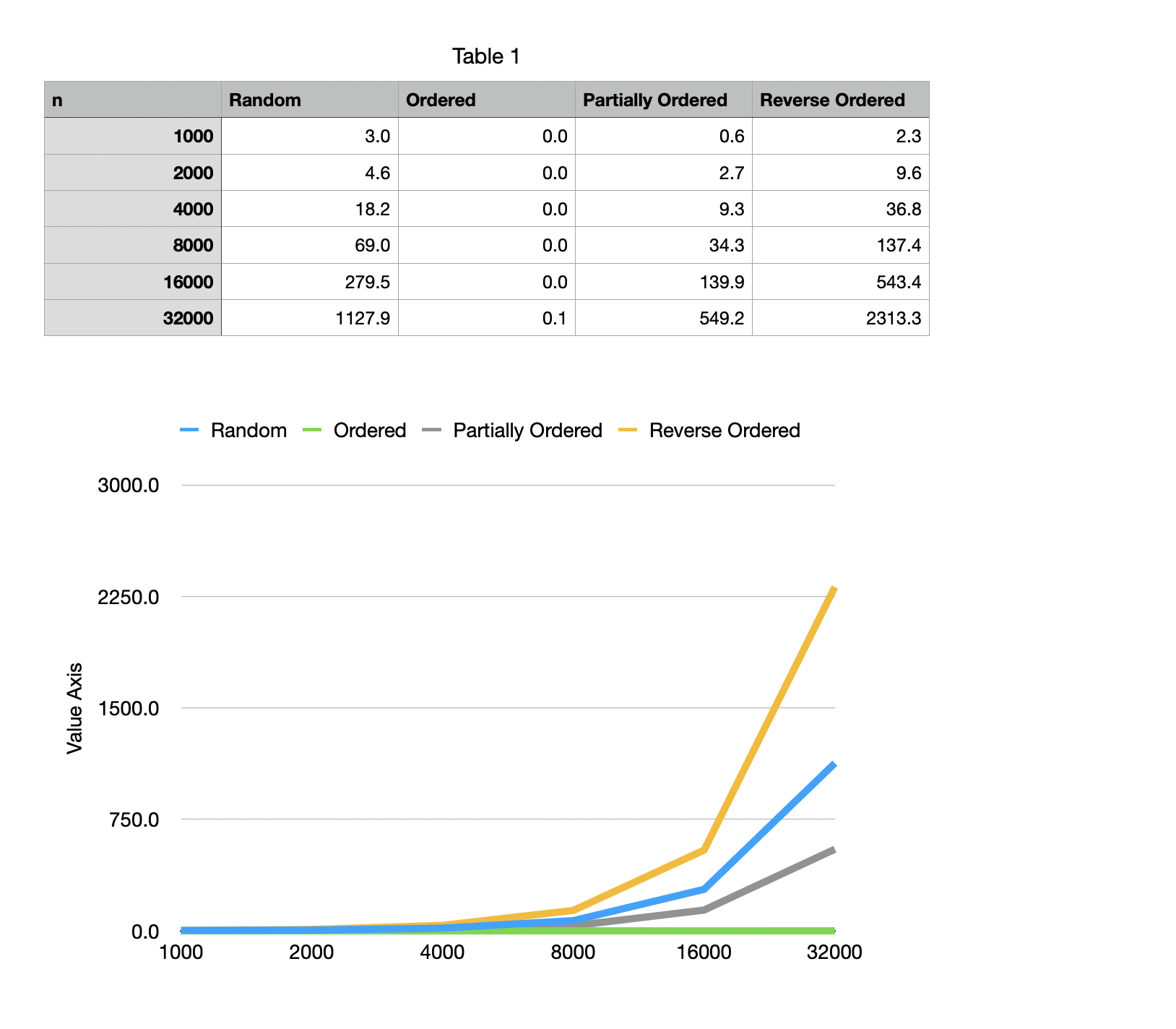
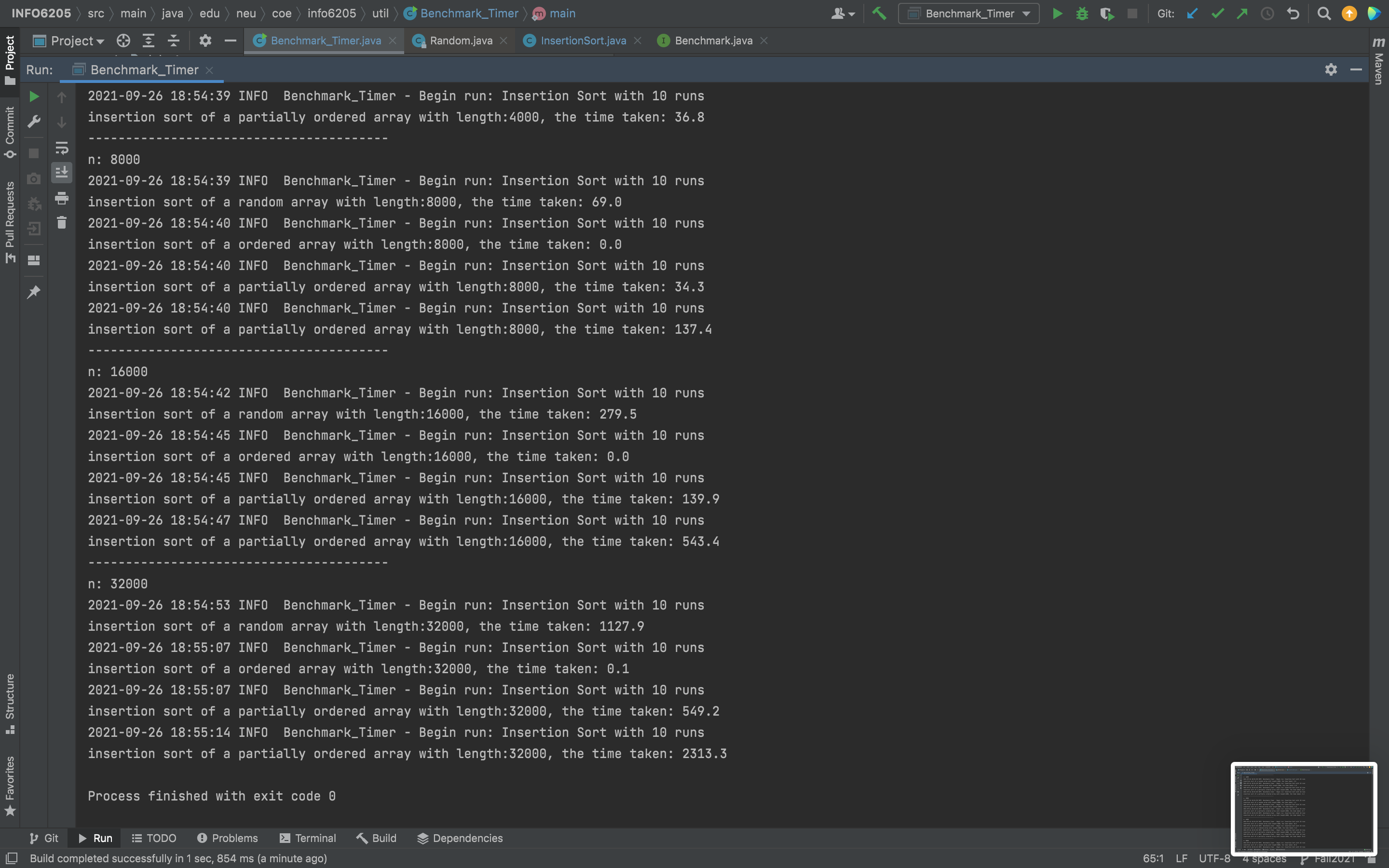
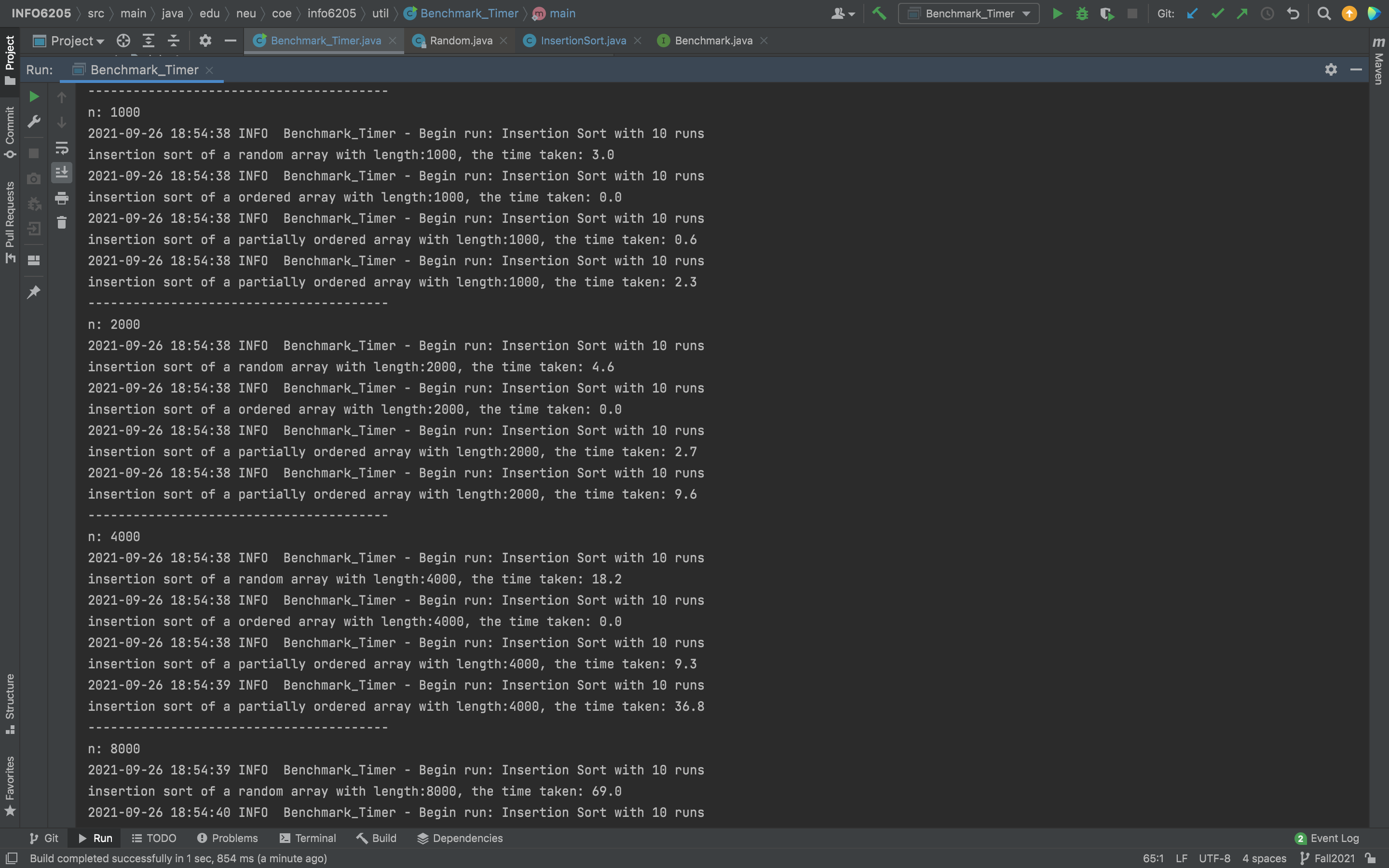
(part 3) Implement a main program to actually run the following benchmarks: measure the running times of this sort, using four different initial array ordering situations: random, ordered, partially-ordered and reverse-ordered. Use the doubling method for choosing *n*and test for at least five values of *n.*Draw any conclusions from observations regarding the order of growth.

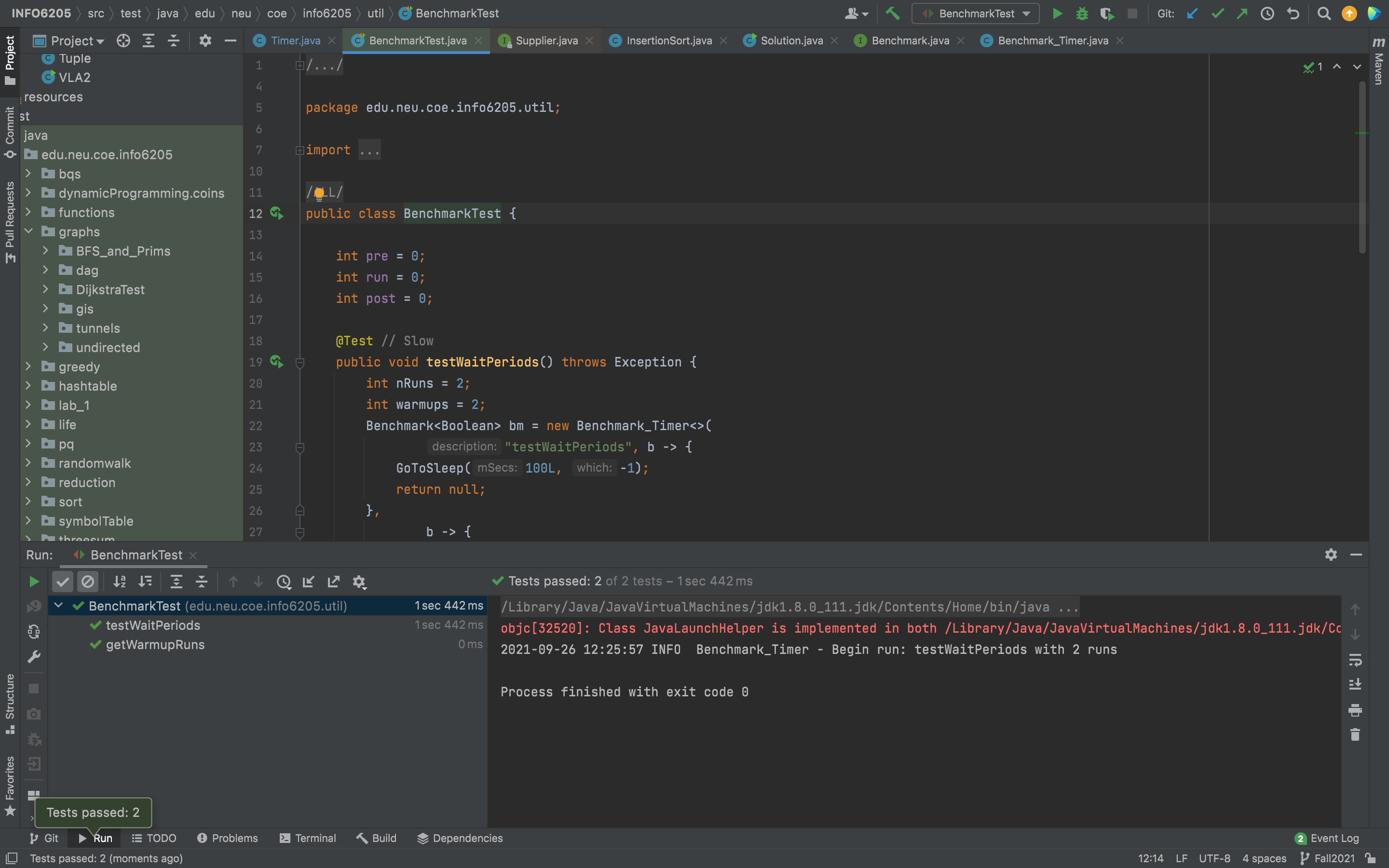
1. **Conclusion**

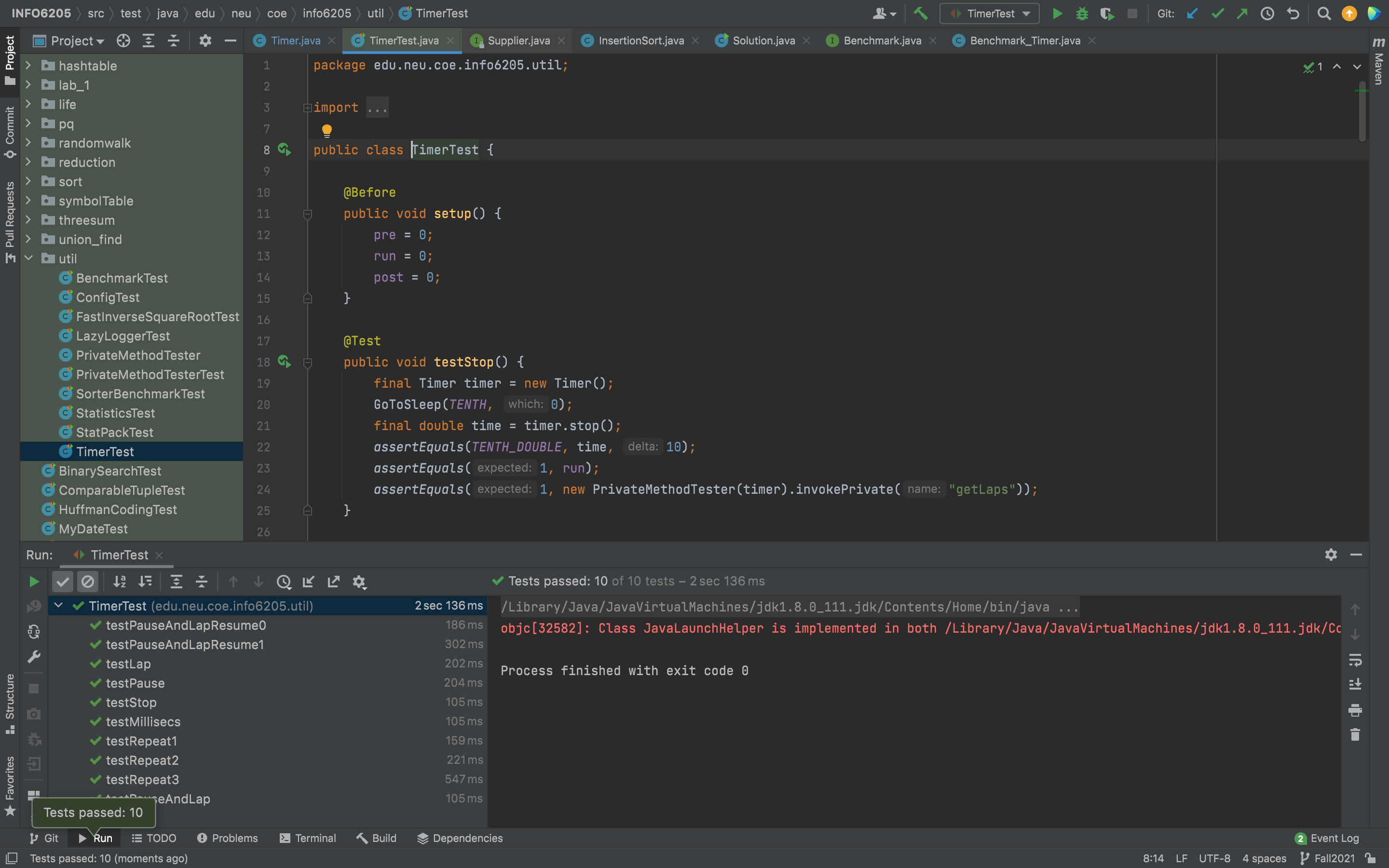
By comparing the running time of insertion sort with four arrays(random, ordered, partially-ordered and reverse-ordered), and six different size n(1000, 2000, 4000, 8000, 16000, 32000), a obvious difference in the processing time growth rate can be seen, which is

*ordered array < partially ordered array < random array < reversed array*

The time growth of ordered array performs almost like constant, and the rest are like exponential function.

1. **Evidence**
2. **Scree****n Shot**

benchmarkTest

timerTest

insertionSortTest