Name (NUID)

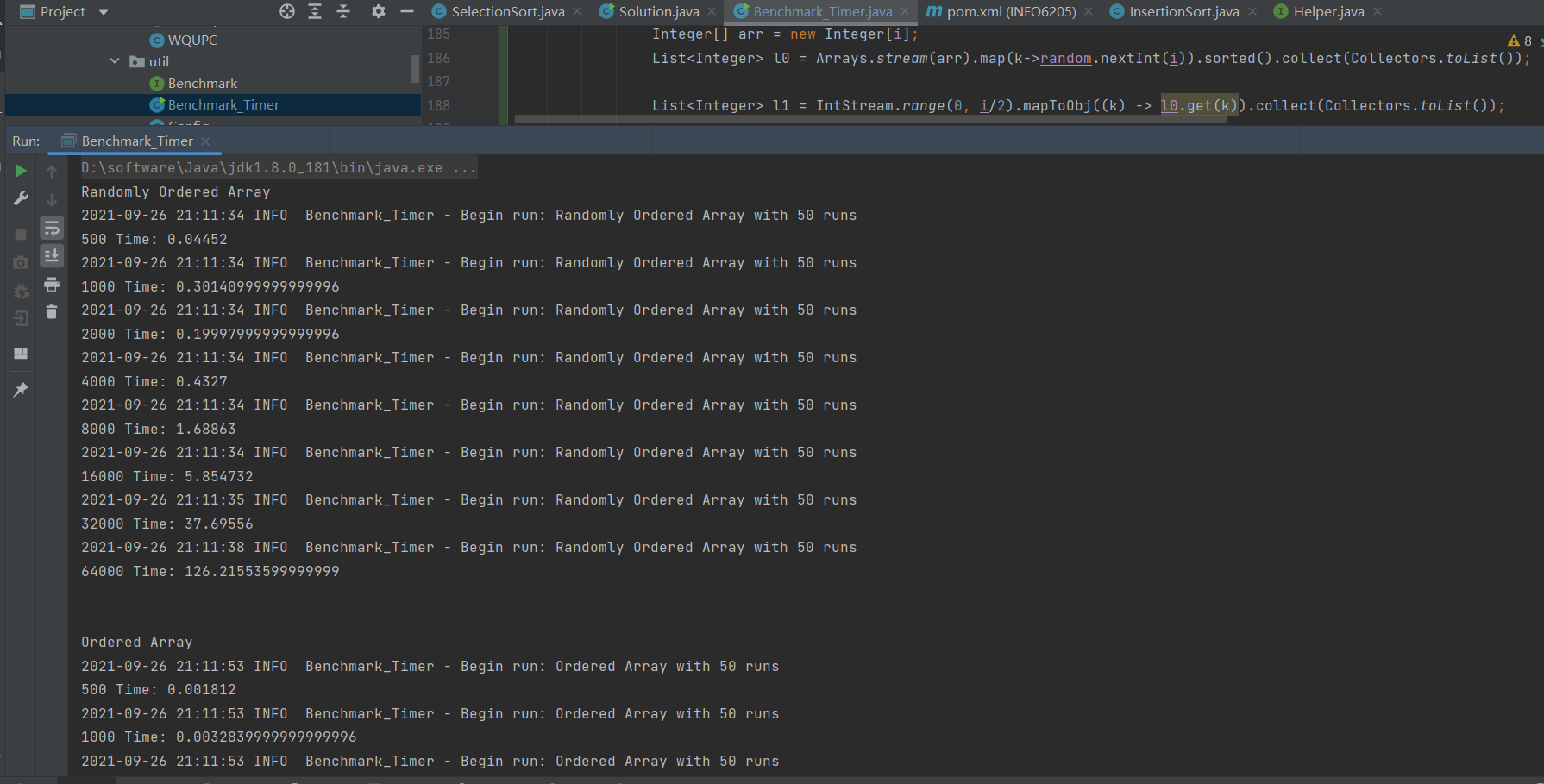
**Program Structures & Algorithms**

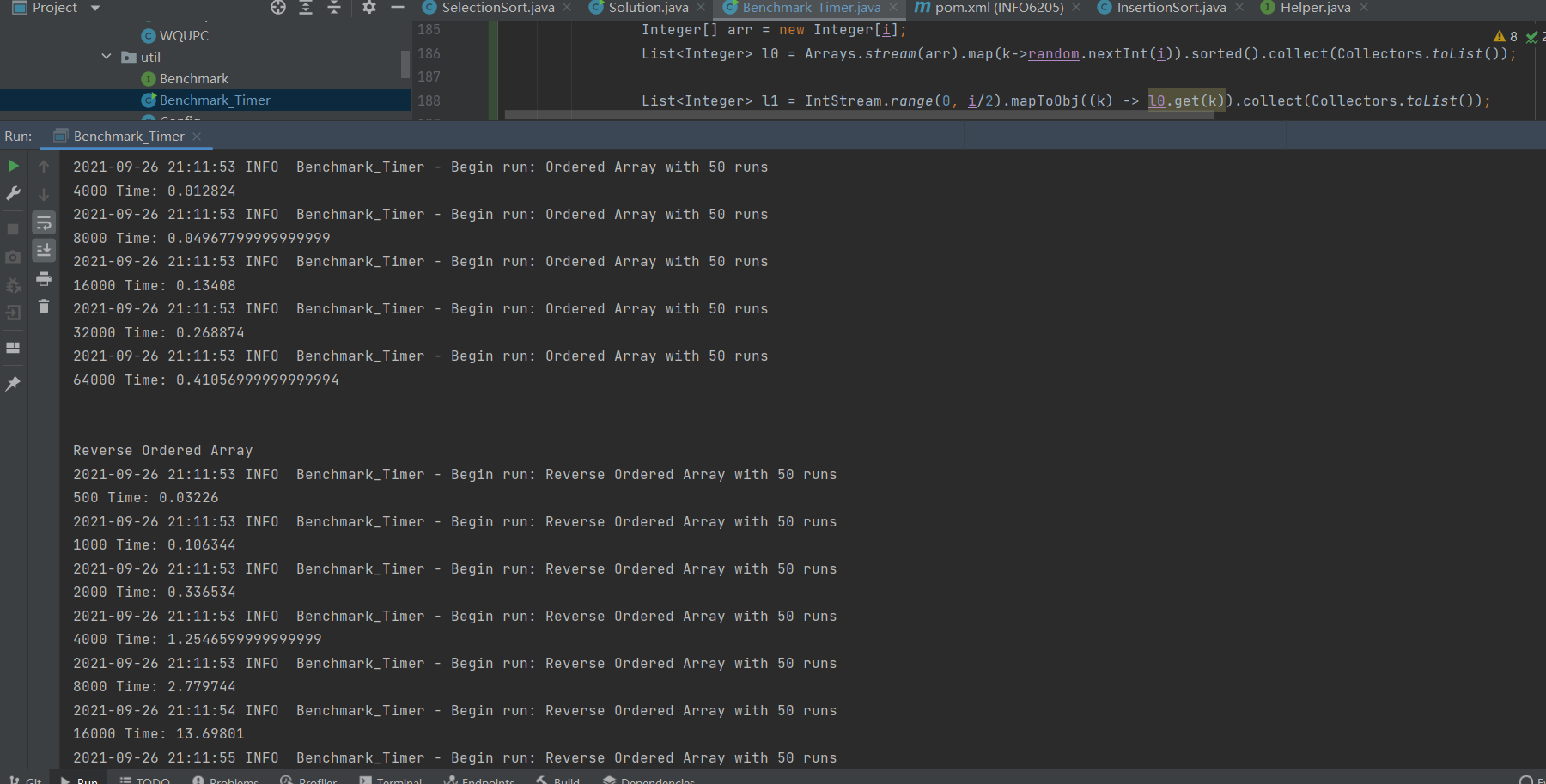
**Fall 2021**

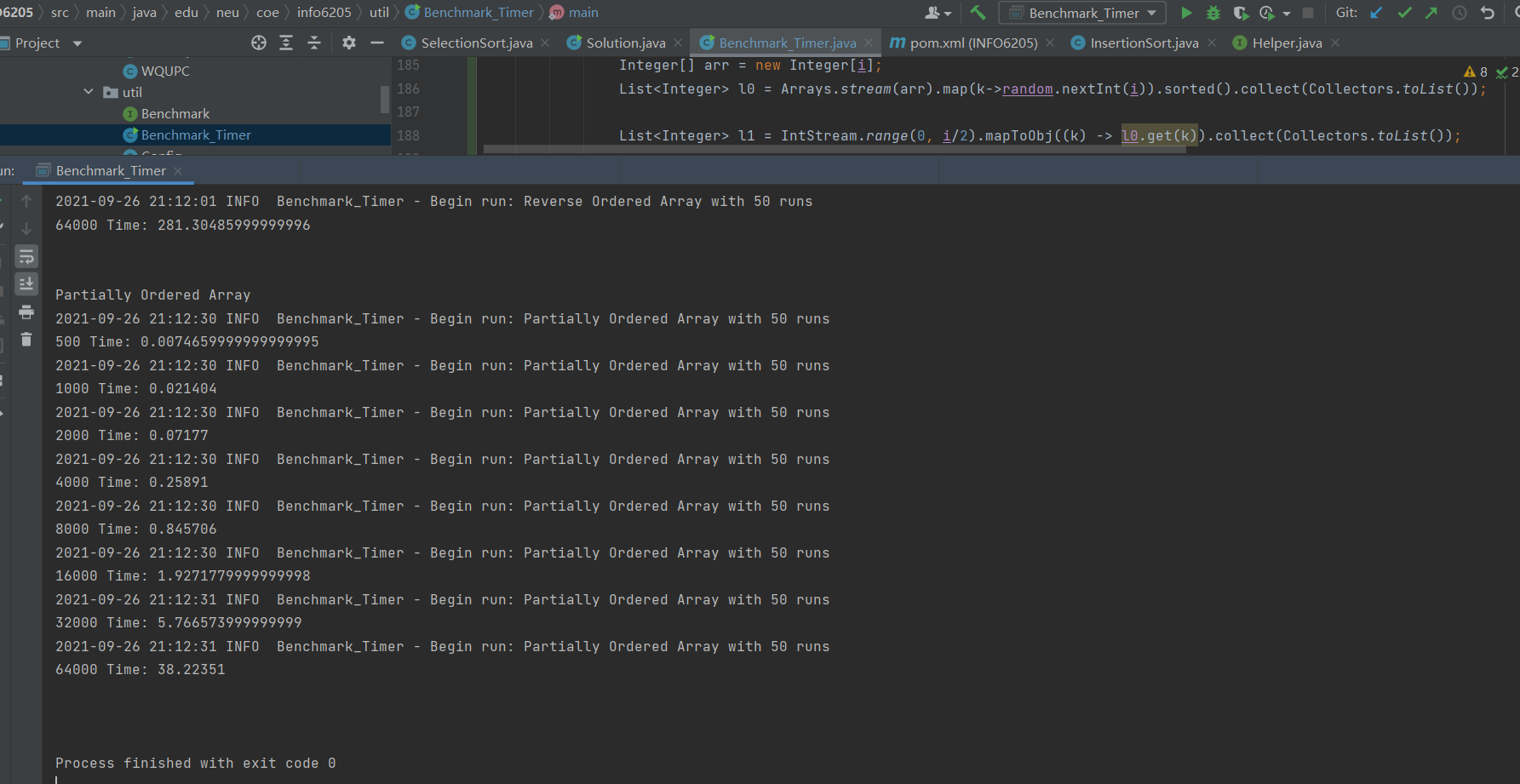
**Assignment No. N**

* **I. Task**
* **1. To implement three methods of a class called Timer. Don't forget to check your implementation by running the unit tests in BenchmarkTest and TimerTest.**
* **2. Implement InsertionSort (in the InsertionSort class) by simply looking up the insertion code used by Arrays.sort. 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.**
* **II. Relationship Conclusion: (For ex : z = a \* b)**
* While N becomes larger, the increasing rate of time is:
  + **Reversed-ordered > Random > Partially-ordered > Ordered**
* **III. Evidence to support the conclusion:**

1. **Output (Snapshot of Code output in the terminal)**

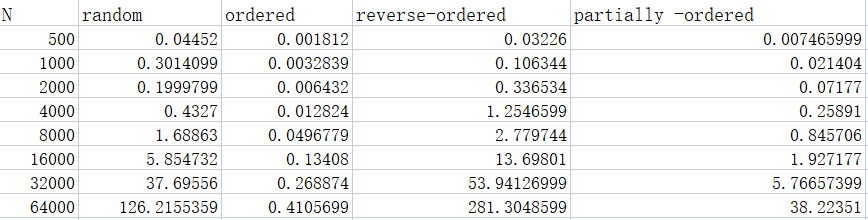




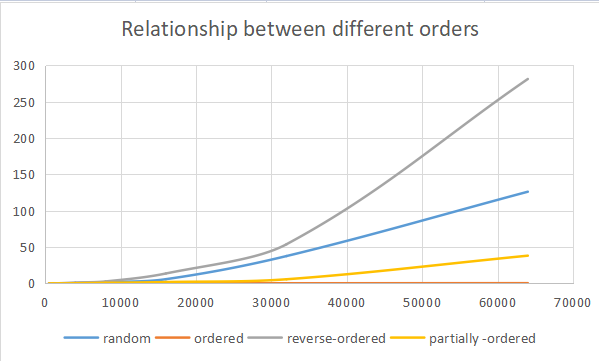


1. **Graphical Representation(Observations from experiments should be tabulated and analyzed by plotting graphs(usually in excel) to arrive on the relationship conclusion)**

**Data:**

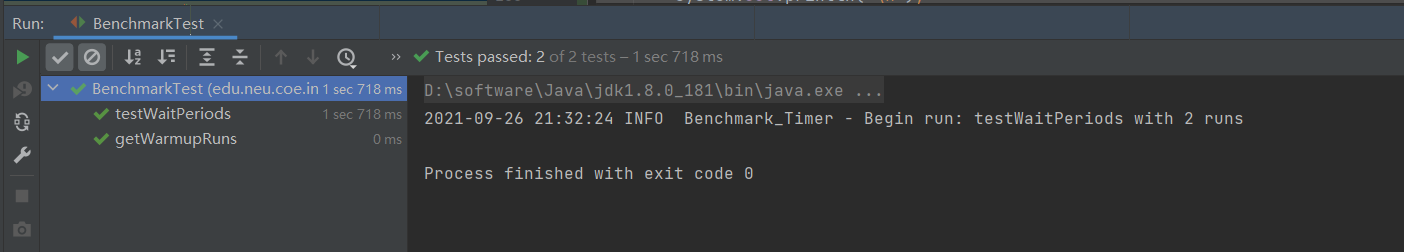


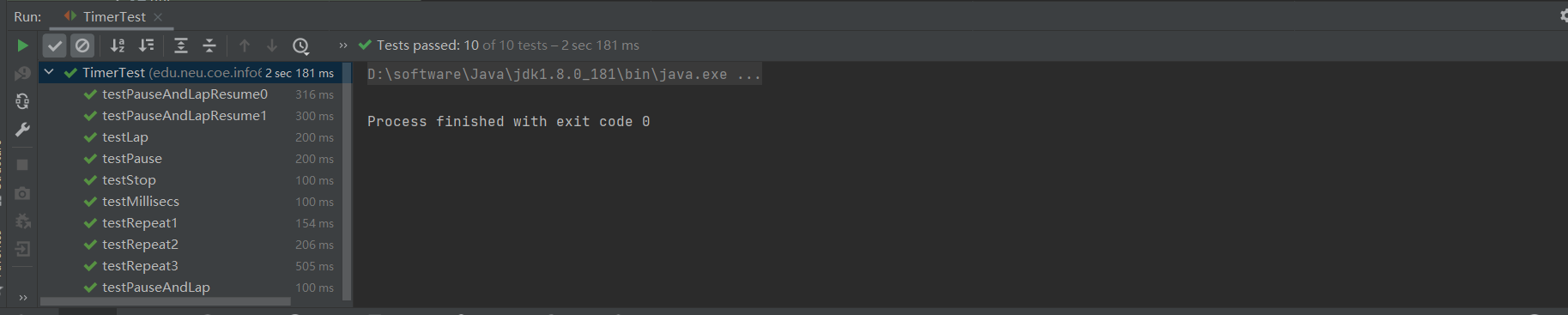
**Graph:**



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

1. BenchmarkTest



1. TimerTest
2. InsertionSortTest