Program Structures and Algorithms
Spring 2023(SEC – 08)

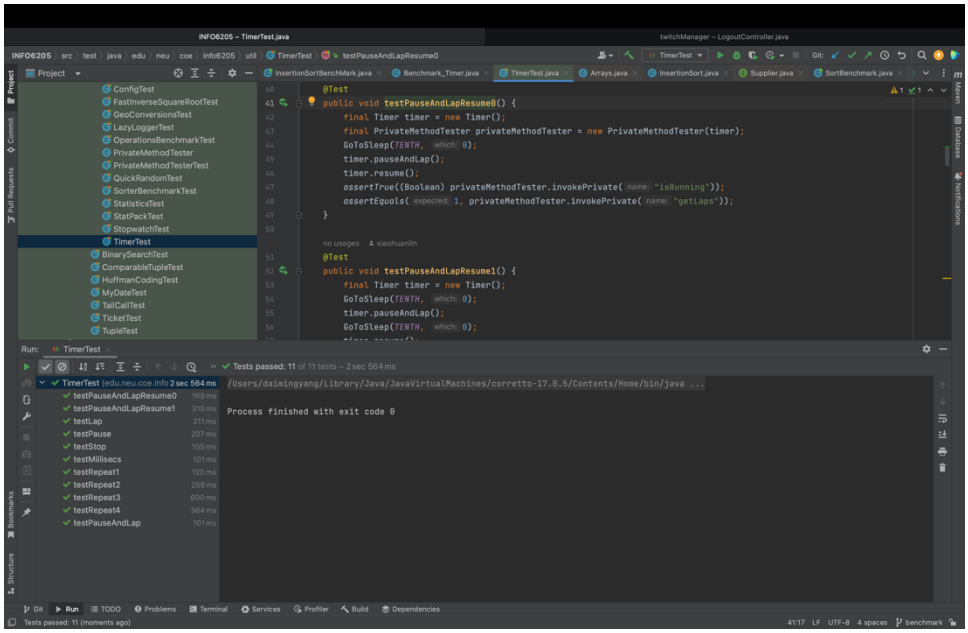
NAME: Daiming Yang
NUID: 002771605

**Task:**

1. **Task 1**

   (Part 1) You are to implement three (3) methods (repeat, getClock, and toMillisecs) of a class called Timer.

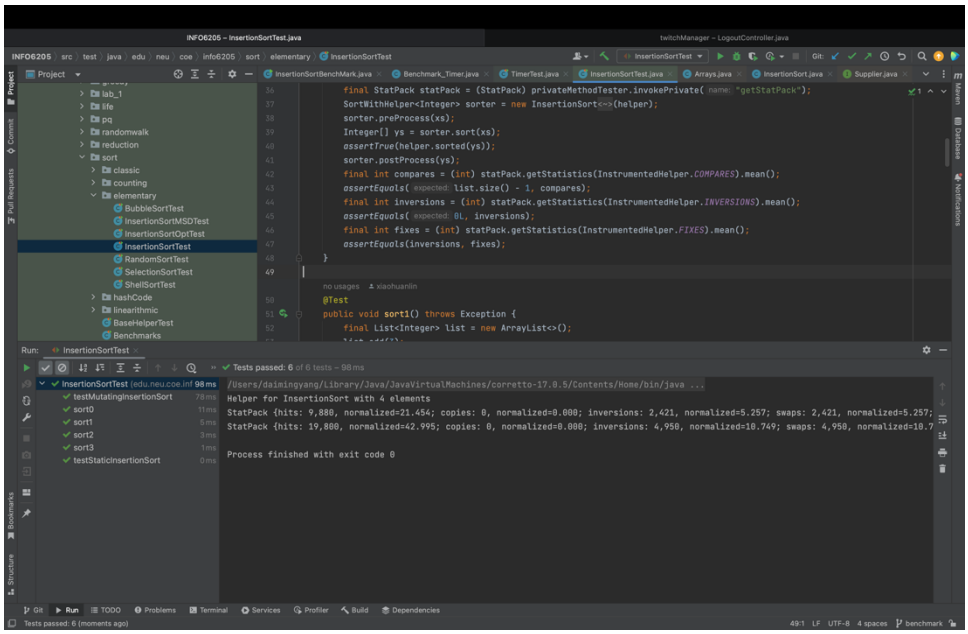
   Unit Test Screenshots:




2. **Task 2:**

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

   Unit Test Screenshots:



3. **Task 3:**

(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.

**Relationship Conclusion:**

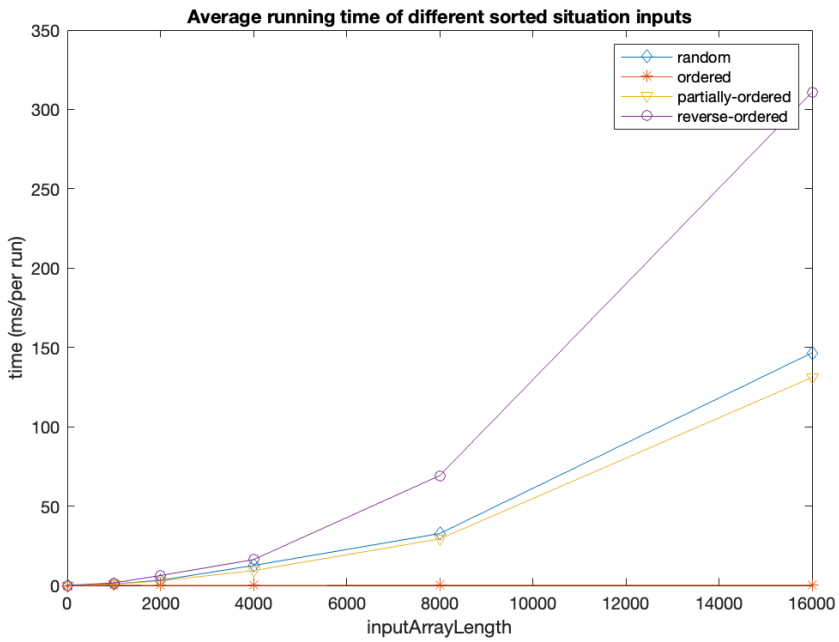
As shown in the average running time of different sorted situation input figuration, the reverse-ordered situation costs the most time, while the costed time of random input is more than the costed time of partially-ordered, and the ordered costs the smallest time.

$$t(reverse-ordered) > t(random) > t(partially-ordered) > t(ordered)$$

Besides, with the length of input array increasing, the differences between different situation are becoming larger, since the time complexity of insertion sort algorithm is $O(n^2)$.

**Evidence to support that conclusion:**

1. **Graphical Representation:**




2. **The run-time data:**

---

InsertionSort: N = 1000

2023-02-04 16:07:19 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs random array inputted, the average time per run is: 0.933ms

2023-02-04 16:07:20 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs ordered array inputted, the average time per run is: 0.088ms

2023-02-04 16:07:21 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs partially-ordered array inputted, the average time per run is: 0.762ms

2023-02-04 16:07:22 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs

reverse-ordered array inputted, the average time per run is: 1.603ms

---
InsertionSort: N = 2000
2023-02-04 16:07:23 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
random array inputted, the average time per run is: 3.214ms

2023-02-04 16:07:27 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
ordered array inputted, the average time per run is: 0.053ms

2023-02-04 16:07:27 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
partially-ordered array inputted, the average time per run is: 2.828ms

2023-02-04 16:07:30 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
reverse-ordered array inputted, the average time per run is: 6.237ms

---
InsertionSort: N = 4000
2023-02-04 16:07:36 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
random array inputted, the average time per run is: 12.575ms

2023-02-04 16:07:49 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
ordered array inputted, the average time per run is: 0.059ms

2023-02-04 16:07:50 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
partially-ordered array inputted, the average time per run is: 9.306ms

2023-02-04 16:07:59 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
reverse-ordered array inputted, the average time per run is: 16.242ms

---
InsertionSort: N = 8000
2023-02-04 16:08:16 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
random array inputted, the average time per run is: 32.681ms

2023-02-04 16:08:49 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
ordered array inputted, the average time per run is: 0.081ms

2023-02-04 16:08:50 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
partially-ordered array inputted, the average time per run is: 29.297ms

2023-02-04 16:09:20 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
reverse-ordered array inputted, the average time per run is: 69.168ms

---
InsertionSort: N = 16000
2023-02-04 16:10:31 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
random array inputted, the average time per run is: 146.592ms

2023-02-04 16:13:00 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs
ordered array inputted, the average time per run is: 0.094ms

2023-02-04 16:13:02 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs partially-ordered array inputted, the average time per run is: 131.085ms

2023-02-04 16:15:15 INFO  Benchmark_Timer - Begin run: InsertionSortBenchmark with 1,000 runs reverse-ordered array inputted, the average time per run is: 310.77ms

Process finished with exit code 0

Unit Test Screenshots: