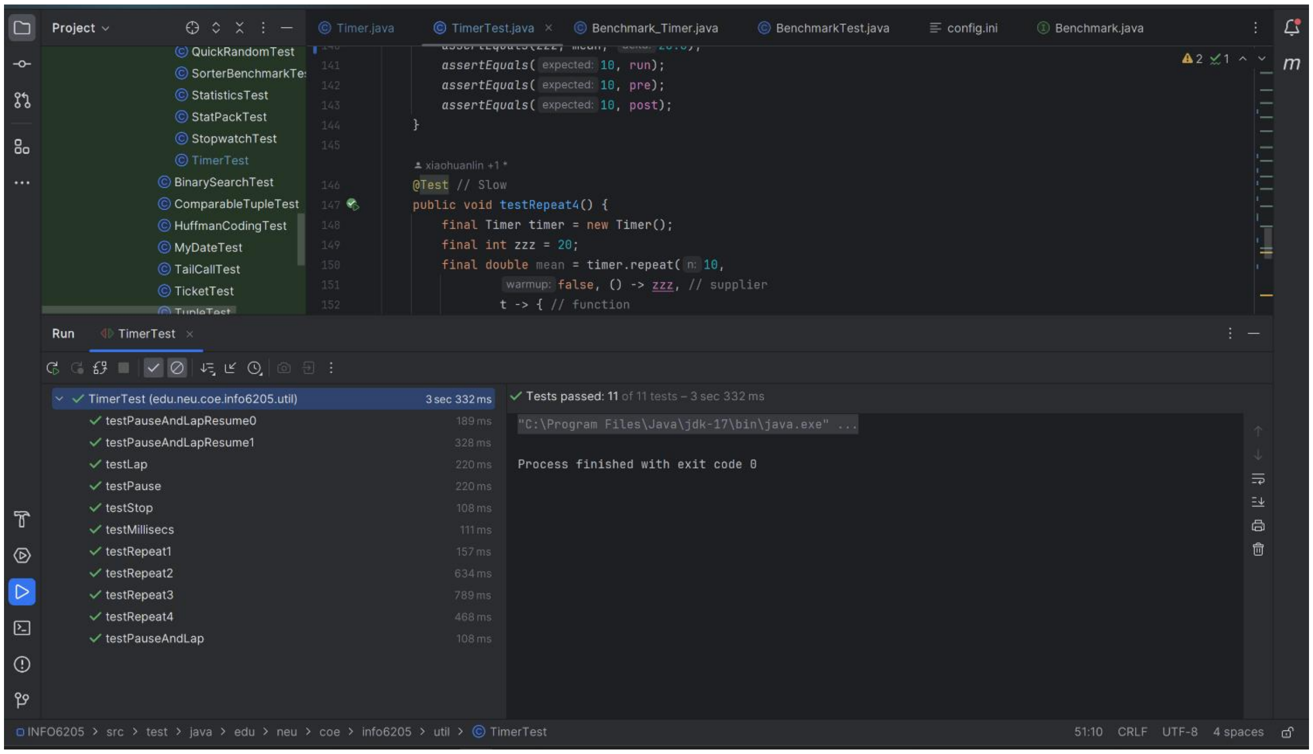
TimerTest Unit test cases passed(11/11) :

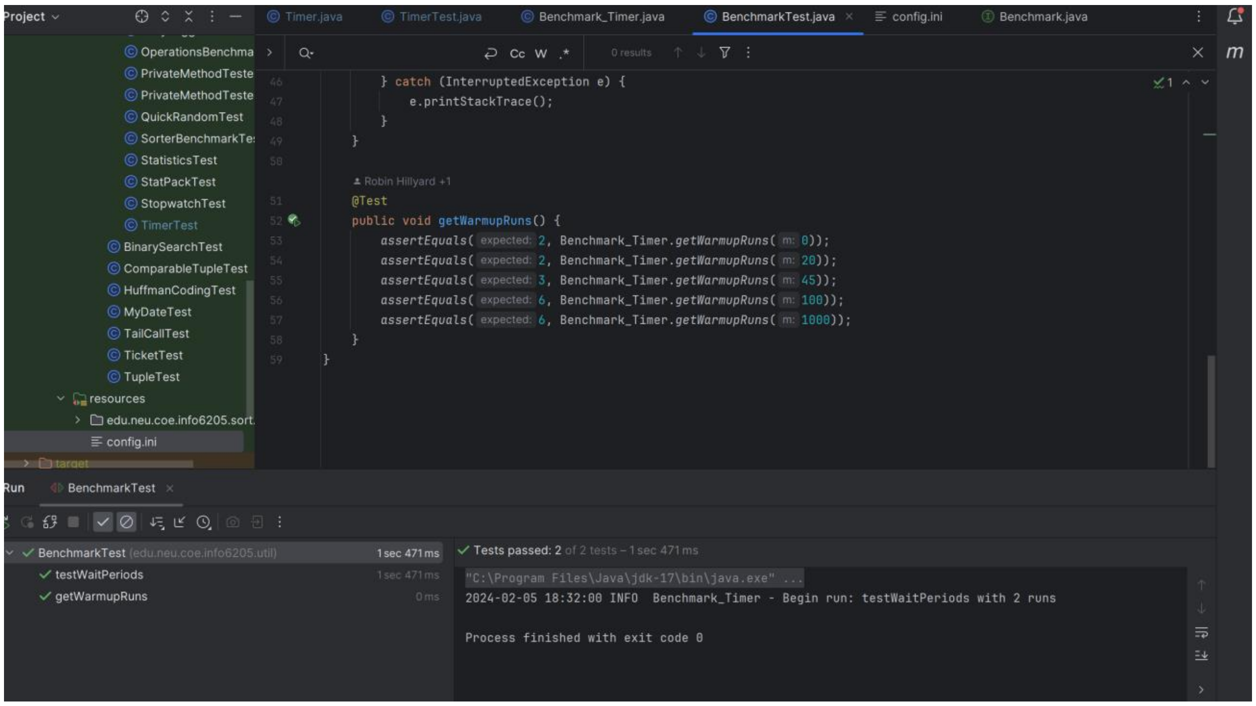


```
        assertEquals( expected: 10, run);
        assertEquals( expected: 10, pre);
        assertEquals( expected: 10, post);
    }

    @Test // Slow
    public void testRepeat4() {
        final Timer timer = new Timer();
        final int zzz = 20;
        final double mean = timer.repeat( n: 10,
                warmup: false, () -> zzz, // supplier
                t -> { // function
```

```
TimerTest (edu.neu.coe.info6205.util)   3 sec 332 ms
  testPauseAndLapResume0                 189 ms
  testPauseAndLapResume1                 328 ms
  testLap                                220 ms
  testPause                              220 ms
  testStop                               108 ms
  testMillisecs                          111 ms
  testRepeat1                            157 ms
  testRepeat2                            634 ms
  testRepeat3                            789 ms
  testRepeat4                            468 ms
  testPauseAndLap                        108 ms
```

```
Tests passed: 11 of 11 tests – 3 sec 332 ms

"C:\Program Files\Java\jdk-17\bin\java.exe" ...

Process finished with exit code 0
```

Benchmark testcases passed(2/2):

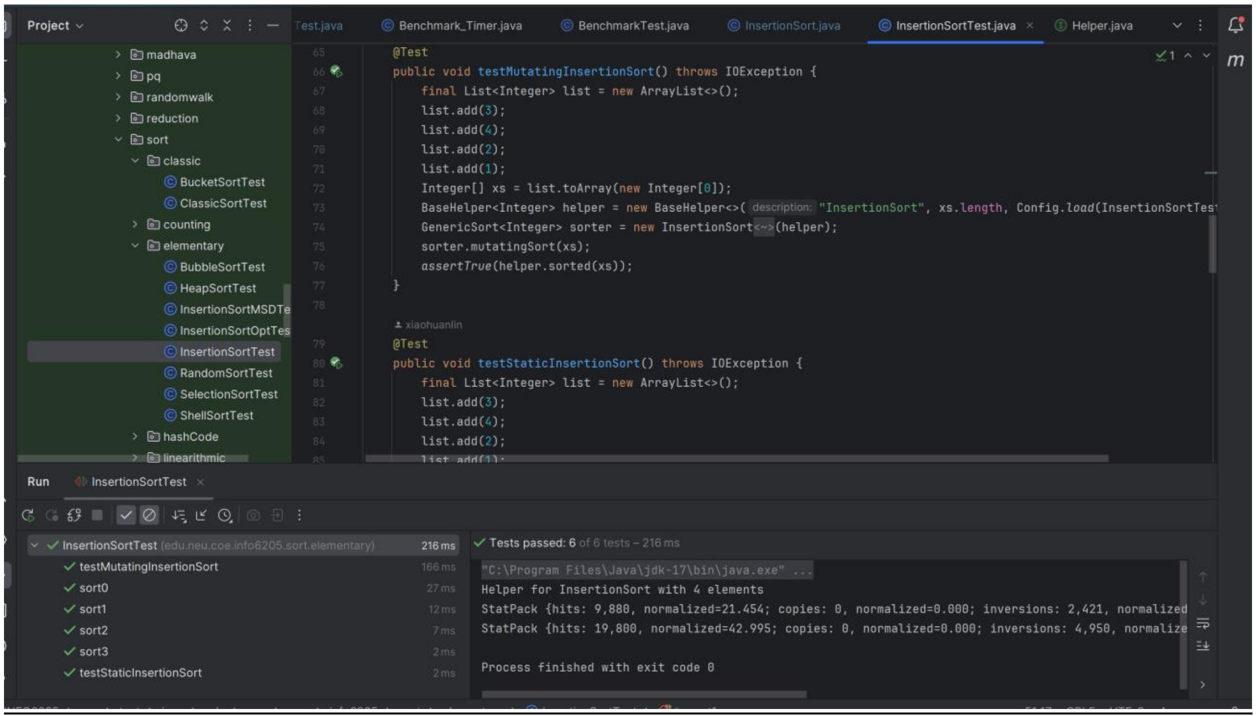


```
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

    @Test
    public void getWarmupRuns() {
        assertEquals( expected: 2, Benchmark_Timer.getWarmupRuns( m: 0));
        assertEquals( expected: 2, Benchmark_Timer.getWarmupRuns( m: 20));
        assertEquals( expected: 3, Benchmark_Timer.getWarmupRuns( m: 45));
        assertEquals( expected: 6, Benchmark_Timer.getWarmupRuns( m: 100));
        assertEquals( expected: 6, Benchmark_Timer.getWarmupRuns( m: 1000));
    }
}
```

```
BenchmarkTest (edu.neu.coe.info6205.util)   1 sec 471 ms
  testWaitPeriods                            1 sec 471 ms
  getWarmupRuns                              0 ms
```

```
Tests passed: 2 of 2 tests – 1 sec 471 ms

"C:\Program Files\Java\jdk-17\bin\java.exe" ...
2024-02-05 18:32:00 INFO  Benchmark_Timer - Begin run: testWaitPeriods with 2 runs

Process finished with exit code 0
```

<u>Insertion Sort test cases passed(6/6):</u>

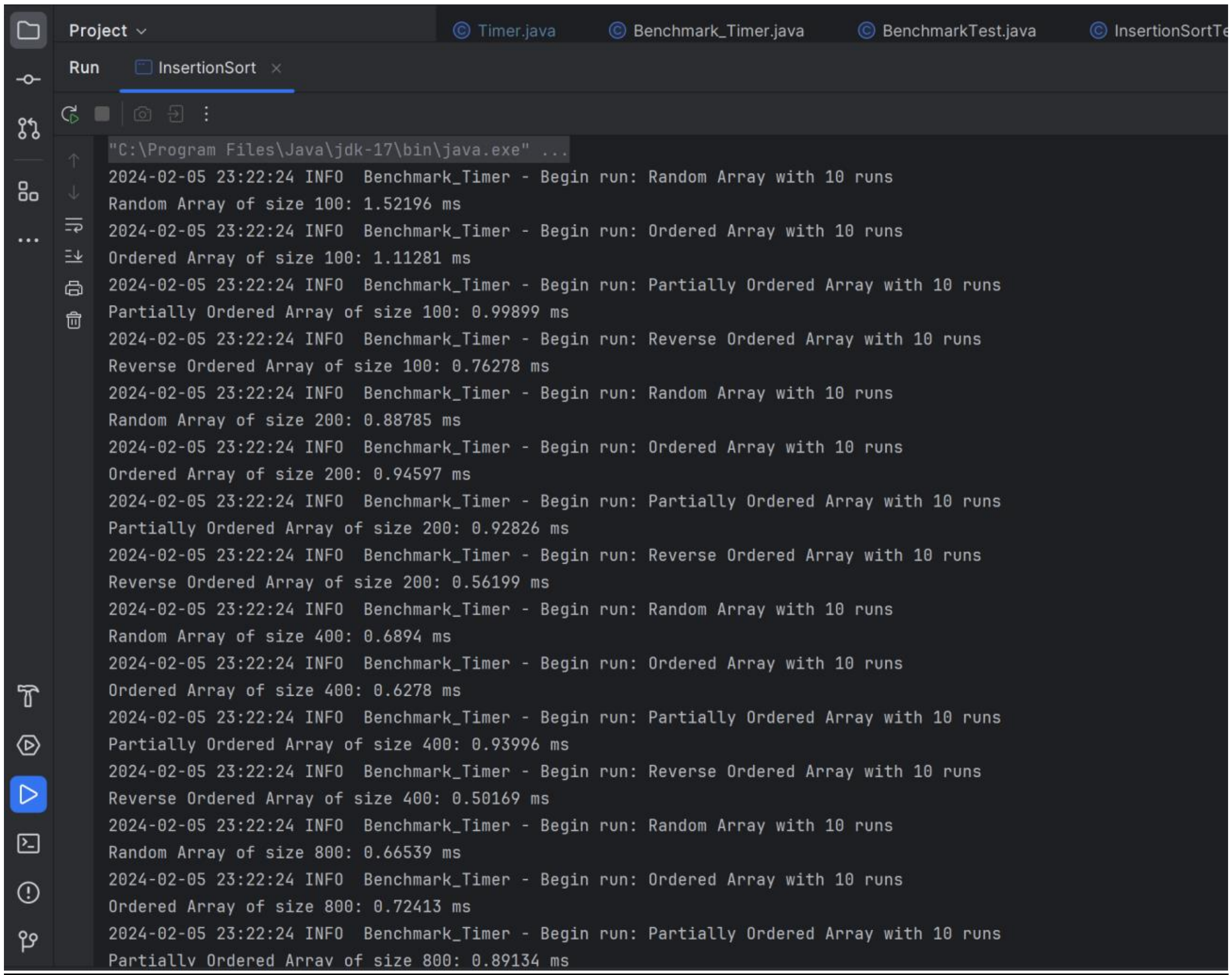


```java
    @Test
    public void testMutatingInsertionSort() throws IOException {
        final List<Integer> list = new ArrayList<>();
        list.add(3);
        list.add(4);
        list.add(2);
        list.add(1);
        Integer[] xs = list.toArray(new Integer[0]);
        BaseHelper<Integer> helper = new BaseHelper<>( description: "InsertionSort", xs.length, Config.load(InsertionSortTes
        GenericSort<Integer> sorter = new InsertionSort<~>(helper);
        sorter.mutatingSort(xs);
        assertTrue(helper.sorted(xs));
    }

    xiaohuanlin
    @Test
    public void testStaticInsertionSort() throws IOException {
        final List<Integer> list = new ArrayList<>();
        list.add(3);
        list.add(4);
        list.add(2);
        list.add(1);
```

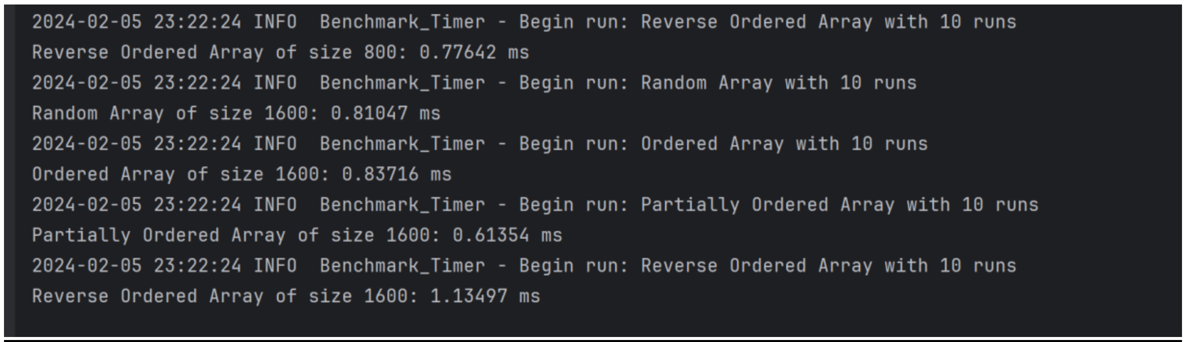
```
✓ InsertionSortTest (edu.neu.coe.info6205.sort.elementary)   216 ms
    ✓ testMutatingInsertionSort                              166 ms
    ✓ sort0                                                   27 ms
    ✓ sort1                                                   12 ms
    ✓ sort2                                                    7 ms
    ✓ sort3                                                    2 ms
    ✓ testStaticInsertionSort                                  2 ms
```

```
✓ Tests passed: 6 of 6 tests – 216 ms

"C:\Program Files\Java\jdk-17\bin\java.exe" ...
Helper for InsertionSort with 4 elements
StatPack {hits: 9,880, normalized=21.454; copies: 0, normalized=0.000; inversions: 2,421, normalized
StatPack {hits: 19,800, normalized=42.995; copies: 0, normalized=0.000; inversions: 4,950, normalize

Process finished with exit code 0
```

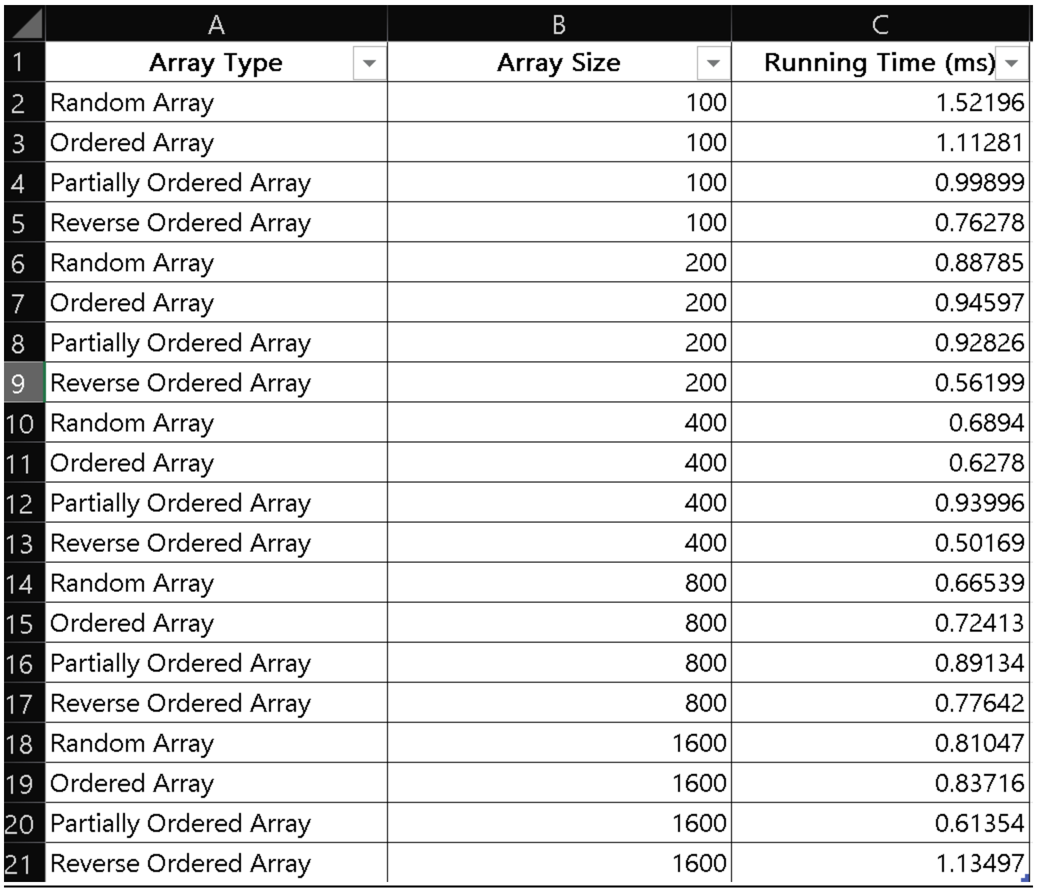
<u>Insertion Sort Runs(Observations):-</u>



```
"C:\Program Files\Java\jdk-17\bin\java.exe" ...
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Random Array with 10 runs
Random Array of size 100: 1.52196 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Ordered Array with 10 runs
Ordered Array of size 100: 1.11281 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Partially Ordered Array with 10 runs
Partially Ordered Array of size 100: 0.99899 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Reverse Ordered Array with 10 runs
Reverse Ordered Array of size 100: 0.76278 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Random Array with 10 runs
Random Array of size 200: 0.88785 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Ordered Array with 10 runs
Ordered Array of size 200: 0.94597 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Partially Ordered Array with 10 runs
Partially Ordered Array of size 200: 0.92826 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Reverse Ordered Array with 10 runs
Reverse Ordered Array of size 200: 0.56199 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Random Array with 10 runs
Random Array of size 400: 0.6894 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Ordered Array with 10 runs
Ordered Array of size 400: 0.6278 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Partially Ordered Array with 10 runs
Partially Ordered Array of size 400: 0.93996 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Reverse Ordered Array with 10 runs
Reverse Ordered Array of size 400: 0.50169 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Random Array with 10 runs
Random Array of size 800: 0.66539 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Ordered Array with 10 runs
Ordered Array of size 800: 0.72413 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Partially Ordered Array with 10 runs
Partially Ordered Array of size 800: 0.89134 ms
```



```
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Reverse Ordered Array with 10 runs
Reverse Ordered Array of size 800: 0.77642 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Random Array with 10 runs
Random Array of size 1600: 0.81047 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Ordered Array with 10 runs
Ordered Array of size 1600: 0.83716 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Partially Ordered Array with 10 runs
Partially Ordered Array of size 1600: 0.61354 ms
2024-02-05 23:22:24 INFO  Benchmark_Timer - Begin run: Reverse Ordered Array with 10 runs
Reverse Ordered Array of size 1600: 1.13497 ms
```



Array Type	Array Size	Running Time (ms)
Random Array	100	1.52196
Ordered Array	100	1.11281
Partially Ordered Array	100	0.99899
Reverse Ordered Array	100	0.76278
Random Array	200	0.88785
Ordered Array	200	0.94597
Partially Ordered Array	200	0.92826
Reverse Ordered Array	200	0.56199
Random Array	400	0.6894
Ordered Array	400	0.6278
Partially Ordered Array	400	0.93996
Reverse Ordered Array	400	0.50169
Random Array	800	0.66539
Ordered Array	800	0.72413
Partially Ordered Array	800	0.89134
Reverse Ordered Array	800	0.77642
Random Array	1600	0.81047
Ordered Array	1600	0.83716
Partially Ordered Array	1600	0.61354
Reverse Ordered Array	1600	1.13497

<u>Conclusions from the above observations:</u>

- Insertion sort shows good performance on ordered arrays, approaching linear time complexity, which aligns with theoretical expectations
- Partially Ordered Arrays sorting time tend to be lower than for random arrays, which suggests that insertion sort can benefit from pre-existing order in the array, as expected.
- Some anomalies in the data may be due to external factors like system load or JVM.