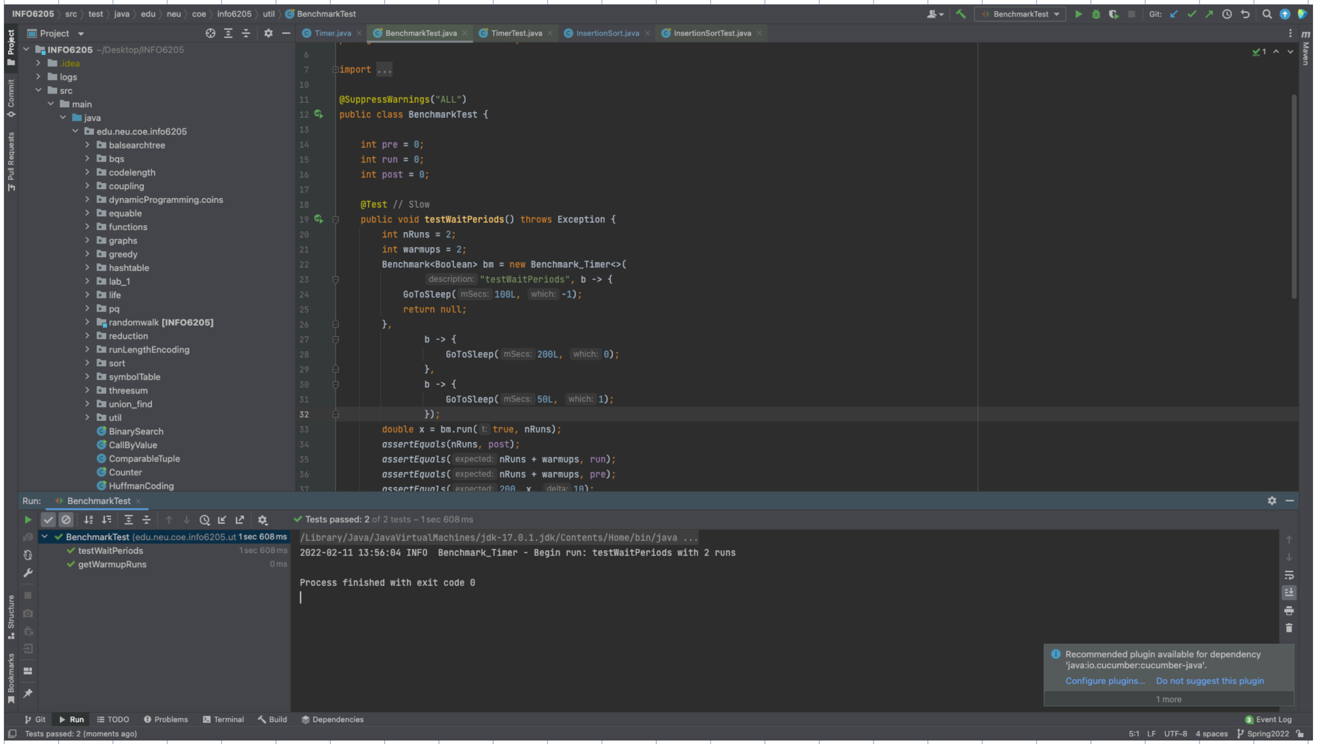
Part 1.



INFO6205 〉 src 〉 test 〉 java 〉 edu 〉 neu 〉 coe 〉 info6205 〉 util 〉 BenchmarkTest

Timer.java | BenchmarkTest.java | TimerTest.java | InsertionSort.java | InsertionSortTest.java

```java
import ...

@SuppressWarnings("ALL")
public class BenchmarkTest {

    int pre = 0;
    int run = 0;
    int post = 0;

    @Test // Slow
    public void testWaitPeriods() throws Exception {
        int nRuns = 2;
        int warmups = 2;
        Benchmark<Boolean> bm = new Benchmark_Timer<>(
                description: "testWaitPeriods", b -> {
            GoToSleep( mSecs: 100L,  which: -1);
            return null;
        },
                b -> {
                    GoToSleep( mSecs: 200L,  which: 0);
                },
                b -> {
                    GoToSleep( mSecs: 50L,  which: 1);
                });
        double x = bm.run( t: true, nRuns);
        assertEquals(nRuns, post);
        assertEquals( expected: nRuns + warmups, run);
        assertEquals( expected: nRuns + warmups, pre);
        assertEquals( expected: 200, x, delta: 10);
```

Run: BenchmarkTest

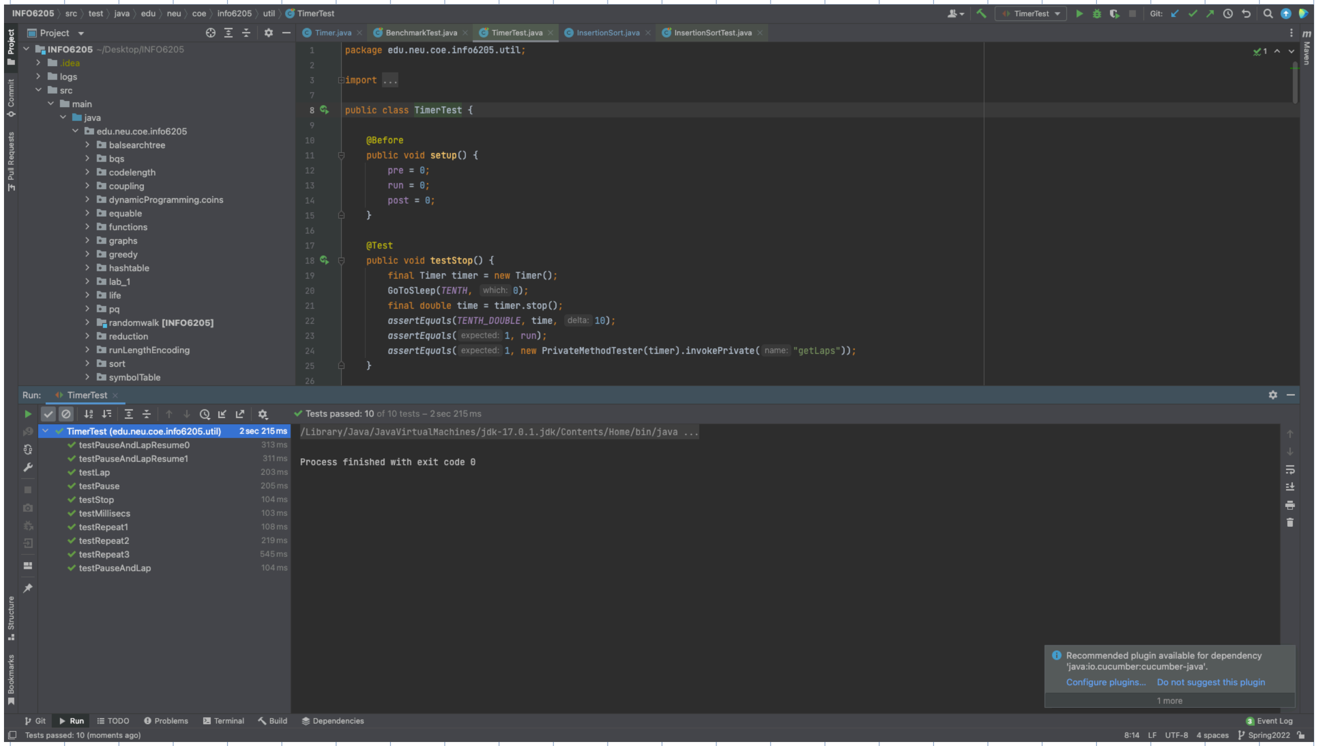
✓ Tests passed: 2 of 2 tests – 1 sec 608 ms

```
✓ BenchmarkTest (edu.neu.coe.info6205.ut 1 sec 608 ms
    ✓ testWaitPeriods                    1 sec 608 ms
    ✓ getWarmupRuns                             0 ms
```

```
/Library/Java/JavaVirtualMachines/jdk-17.0.1.jdk/Contents/Home/bin/java ...
2022-02-11 13:56:04 INFO  Benchmark_Timer - Begin run: testWaitPeriods with 2 runs

Process finished with exit code 0
```

Recommended plugin available for dependency 'java:io.cucumber:cucumber-java'.
Configure plugins... Do not suggest this plugin

Tests passed: 2 (moments ago)



INFO6205 〉 src 〉 test 〉 java 〉 edu 〉 neu 〉 coe 〉 info6205 〉 util 〉 TimerTest

Timer.java | BenchmarkTest.java | TimerTest.java | InsertionSort.java | InsertionSortTest.java

```java
package edu.neu.coe.info6205.util;

import ...

public class TimerTest {

    @Before
    public void setup() {
        pre = 0;
        run = 0;
        post = 0;
    }

    @Test
    public void testStop() {
        final Timer timer = new Timer();
        GoToSleep(TENTH,  which: 0);
        final double time = timer.stop();
        assertEquals(TENTH_DOUBLE, time,  delta: 10);
        assertEquals( expected: 1, run);
        assertEquals( expected: 1, new PrivateMethodTester(timer).invokePrivate( name: "getLaps"));
    }
```

Run: TimerTest

✓ Tests passed: 10 of 10 tests – 2 sec 215 ms

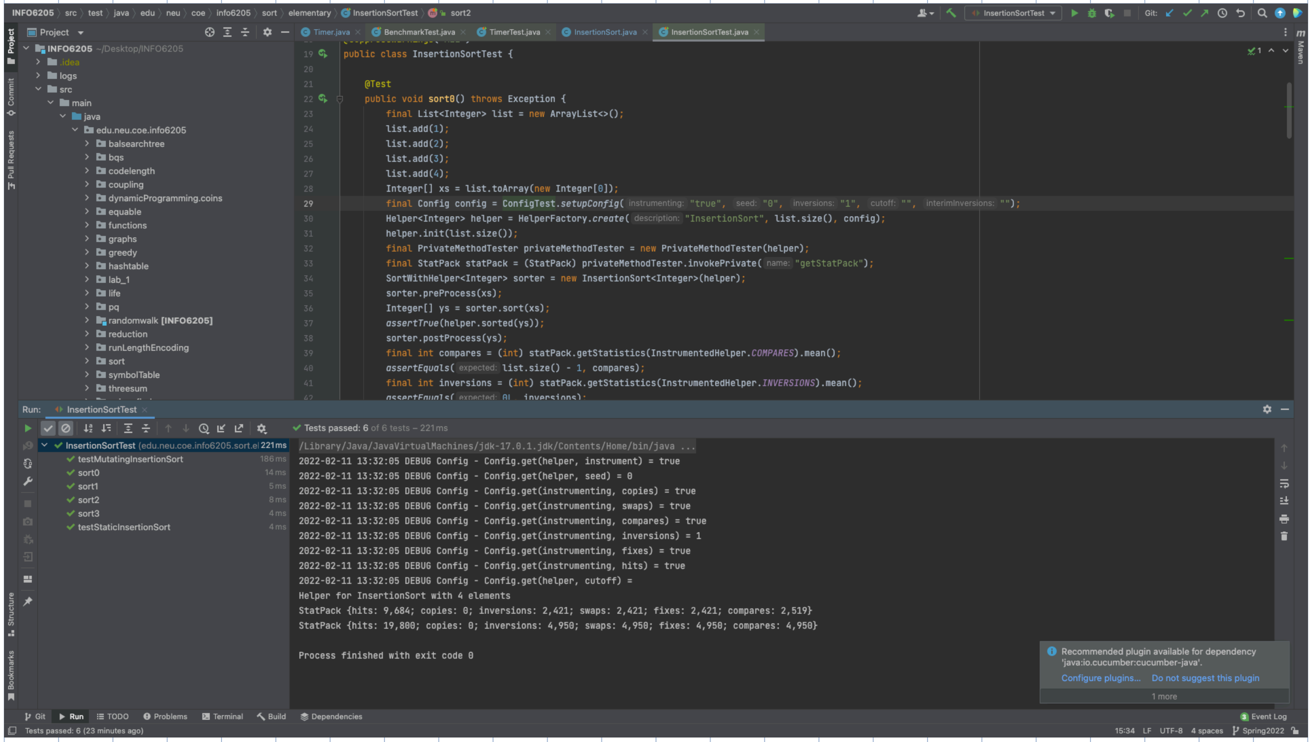
```
✓ TimerTest (edu.neu.coe.info6205.util)   2 sec 215 ms
    ✓ testPauseAndLapResume0                313 ms
    ✓ testPauseAndLapResume1                311 ms
    ✓ testLap                               203 ms
    ✓ testPause                             205 ms
    ✓ testStop                              104 ms
    ✓ testMillisecs                         103 ms
    ✓ testRepeat1                           108 ms
    ✓ testRepeat2                           219 ms
    ✓ testRepeat3                           545 ms
    ✓ testPauseAndLap                       104 ms
```

```
/Library/Java/JavaVirtualMachines/jdk-17.0.1.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```

Recommended plugin available for dependency 'java:io.cucumber:cucumber-java'.
Configure plugins... Do not suggest this plugin

Tests passed: 10 (moments ago)

Part 2.



```java
public class InsertionSortTest {

    @Test
    public void sort0() throws Exception {
        final List<Integer> list = new ArrayList<>();
        list.add(1);
        list.add(2);
        list.add(3);
        list.add(4);
        Integer[] xs = list.toArray(new Integer[0]);
        final Config config = ConfigTest.setupConfig("true", "0", "1", "", "");
        Helper<Integer> helper = HelperFactory.create("InsertionSort", list.size(), config);
        helper.init(list.size());
        final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);
        final StatPack statPack = (StatPack) privateMethodTester.invokePrivate("getStatPack");
        SortWithHelper<Integer> sorter = new InsertionSort<Integer>(helper);
        sorter.preProcess(xs);
        Integer[] ys = sorter.sort(xs);
        assertTrue(helper.sorted(ys));
        sorter.postProcess(ys);
        final int compares = (int) statPack.getStatistics(InstrumentedHelper.COMPARES).mean();
        assertEquals(list.size() - 1, compares);
        final int inversions = (int) statPack.getStatistics(InstrumentedHelper.INVERSIONS).mean();
        assertEquals(0L, inversions);
```

Tests passed: 6 of 6 tests – 221 ms

```
InsertionSortTest (edu.neu.coe.info6205.sort.el  221 ms
    testMutatingInsertionSort                    186 ms
    sort0                                         14 ms
    sort1                                          5 ms
    sort2                                          8 ms
    sort3                                          4 ms
    testStaticInsertionSort                        4 ms
```

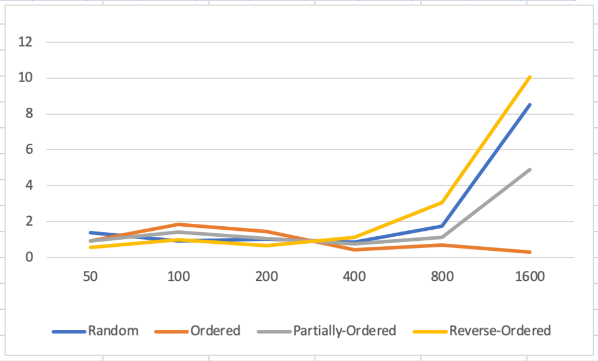
```
/Library/Java/JavaVirtualMachines/jdk-17.0.1.jdk/Contents/Home/bin/java ...
2022-02-11 13:32:05 DEBUG Config - Config.get(helper, instrument) = true
2022-02-11 13:32:05 DEBUG Config - Config.get(helper, seed) = 0
2022-02-11 13:32:05 DEBUG Config - Config.get(instrumenting, copies) = true
2022-02-11 13:32:05 DEBUG Config - Config.get(instrumenting, swaps) = true
2022-02-11 13:32:05 DEBUG Config - Config.get(instrumenting, compares) = true
2022-02-11 13:32:05 DEBUG Config - Config.get(instrumenting, inversions) = 1
2022-02-11 13:32:05 DEBUG Config - Config.get(instrumenting, fixes) = true
2022-02-11 13:32:05 DEBUG Config - Config.get(instrumenting, hits) = true
2022-02-11 13:32:05 DEBUG Config - Config.get(helper, cutoff) = 
Helper for InsertionSort with 4 elements
StatPack {hits: 9,684; copies: 0; inversions: 2,421; swaps: 2,421; fixes: 2,421; compares: 2,519}
StatPack {hits: 19,800; copies: 0; inversions: 4,950; swaps: 4,950; fixes: 4,950; compares: 4,950}

Process finished with exit code 0
```

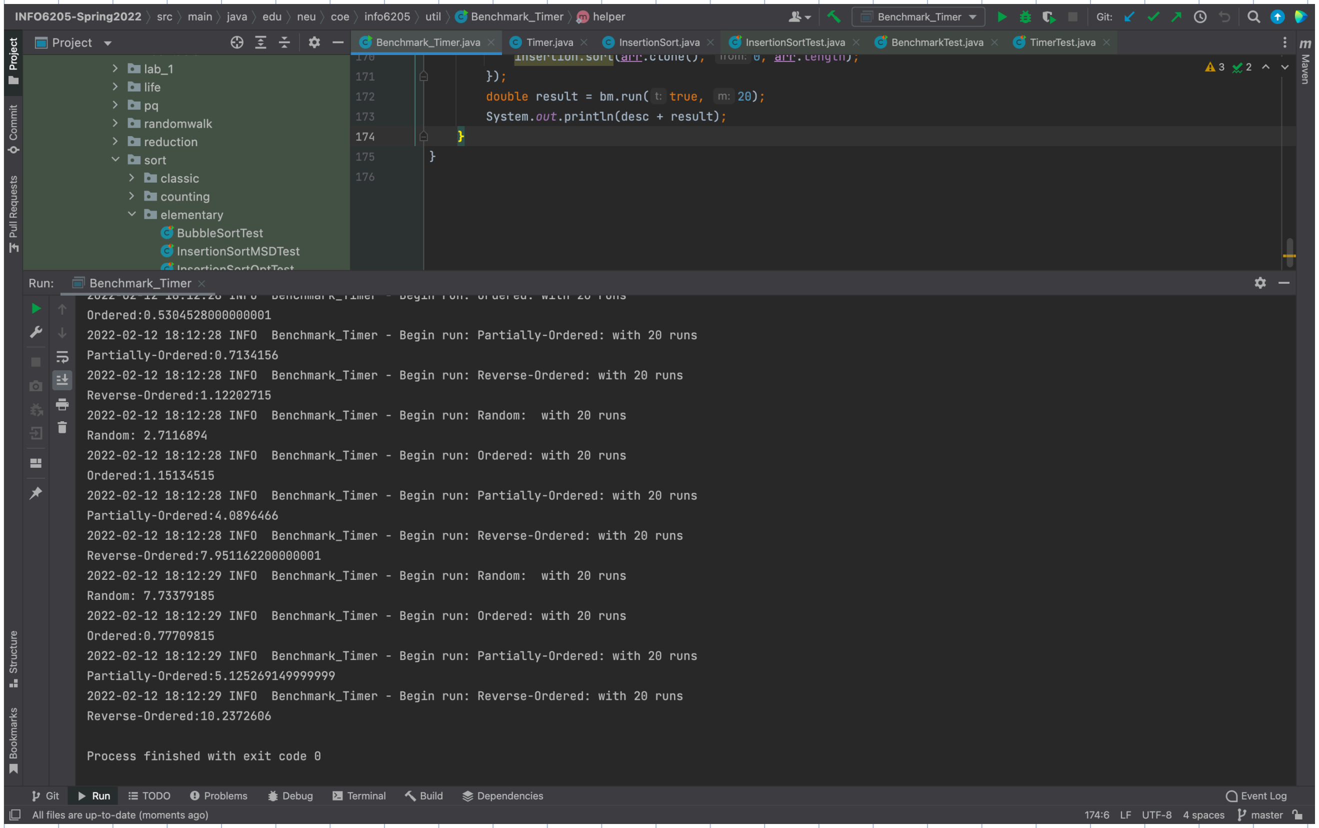
Tests passed: 6 (23 minutes ago)

Part 3.

The main method is implemented in Benchmark_Timer.java. The result and the screenshot are shown below. We can see that the reverse-ordered array has the highest running times with the worst-case time complexity scenario of $O(n^2)$, while the ordered array has the lowest running times, which corresponds to the best-case scenario of $O(n)$.

n	Random	Ordered	Partially-Ordered	Reverse-Ordered
50	1.3820293	0.91175955	0.9132063	0.5452313
100	0.9143646	1.8336113	1.401332	0.99345255
200	1.01796425	1.4506557	1.05725295	0.6673887
400	0.8601919	0.42171725	0.76073975	1.1125519
800	1.7448234	0.68113795	1.1101829	3.04168915
1600	8.5165534	0.28226975	4.89894345	10.05614275





```
            insertion.sort(arr.clone(), from: 0, arr.length);
        });
        double result = bm.run( t: true,  m: 20);
        System.out.println(desc + result);
    }
}
```

```
2022-02-12 18:12:28 INFO  Benchmark_Timer - Begin run: Ordered: with 20 runs
Ordered:0.5304528000000001
2022-02-12 18:12:28 INFO  Benchmark_Timer - Begin run: Partially-Ordered: with 20 runs
Partially-Ordered:0.7134156
2022-02-12 18:12:28 INFO  Benchmark_Timer - Begin run: Reverse-Ordered: with 20 runs
Reverse-Ordered:1.12202715
2022-02-12 18:12:28 INFO  Benchmark_Timer - Begin run: Random:  with 20 runs
Random: 2.7116894
2022-02-12 18:12:28 INFO  Benchmark_Timer - Begin run: Ordered: with 20 runs
Ordered:1.15134515
2022-02-12 18:12:28 INFO  Benchmark_Timer - Begin run: Partially-Ordered: with 20 runs
Partially-Ordered:4.0896466
2022-02-12 18:12:28 INFO  Benchmark_Timer - Begin run: Reverse-Ordered: with 20 runs
Reverse-Ordered:7.951162200000001
2022-02-12 18:12:29 INFO  Benchmark_Timer - Begin run: Random:  with 20 runs
Random: 7.73379185
2022-02-12 18:12:29 INFO  Benchmark_Timer - Begin run: Ordered: with 20 runs
Ordered:0.77709815
2022-02-12 18:12:29 INFO  Benchmark_Timer - Begin run: Partially-Ordered: with 20 runs
Partially-Ordered:5.125269149999999
2022-02-12 18:12:29 INFO  Benchmark_Timer - Begin run: Reverse-Ordered: with 20 runs
Reverse-Ordered:10.2372606

Process finished with exit code 0
```