Xi Zhao (NUID: 001054333)

INFO 6205

Program Structures & Algorithms

Fall 2020

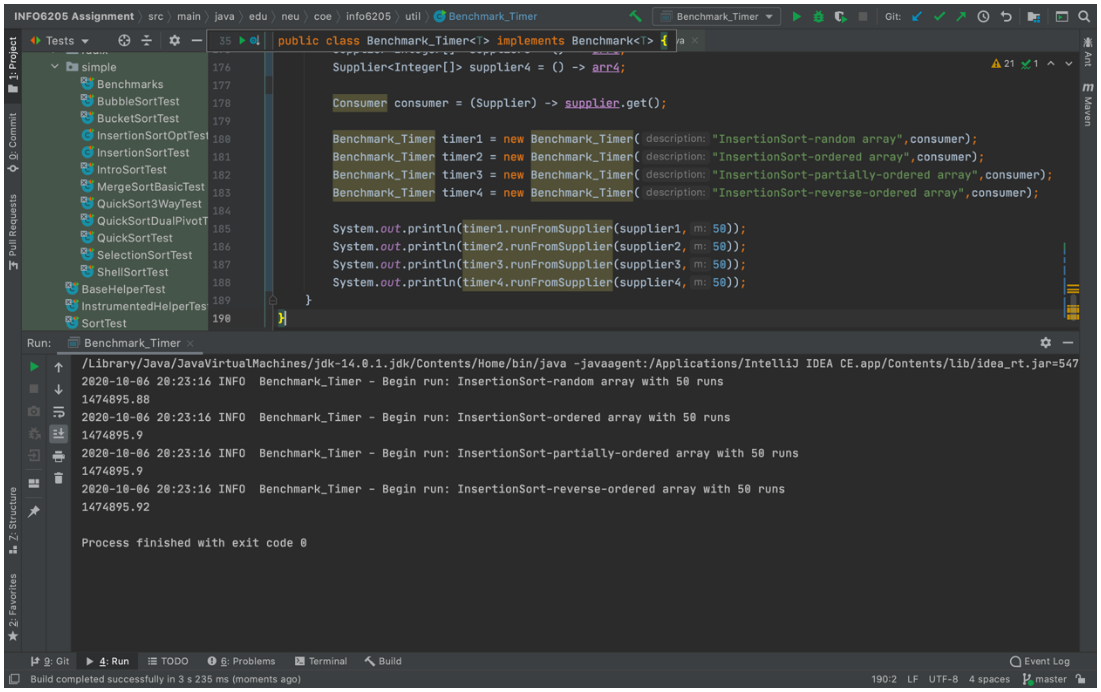
Section 6

Assignment 2

- **Task:**
  Measure the running times of insertion 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.

- **Output (Over 50 times run of each N value, 7 N values in total)**

1. 4 different arrays of 10 elements respectively:

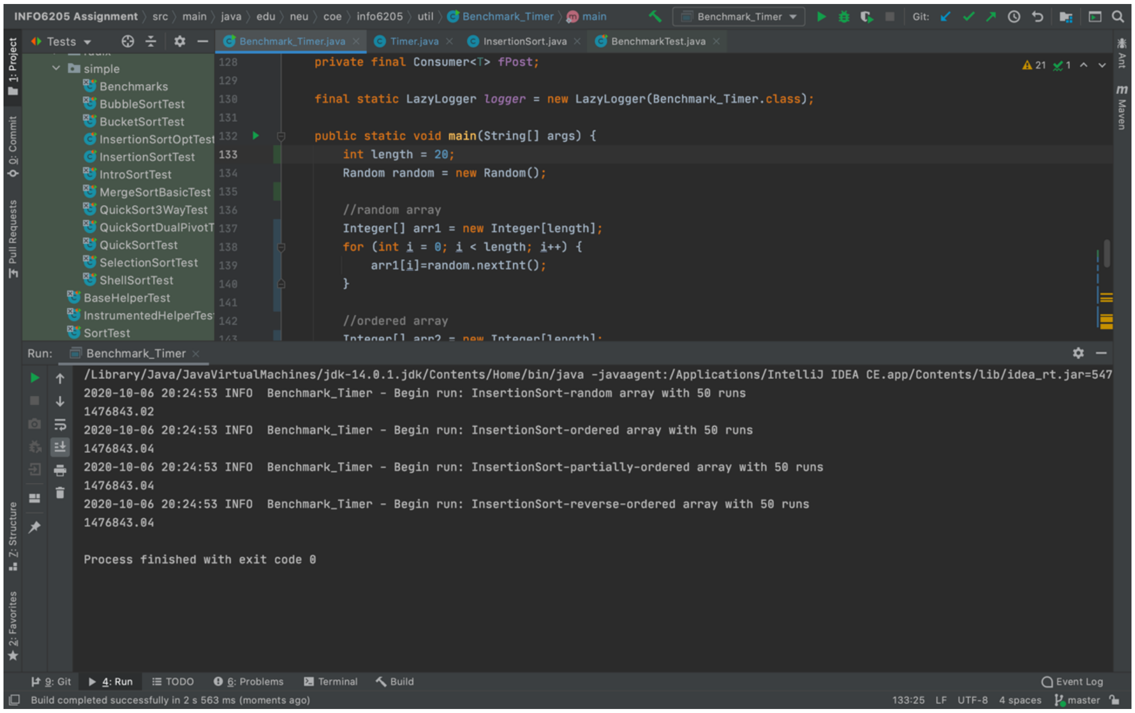


```java
public class Benchmark_Timer<T> implements Benchmark<T> {
        Supplier<Integer[]> supplier4 = () -> arr4;

        Consumer consumer = (Supplier) -> supplier.get();

        Benchmark_Timer timer1 = new Benchmark_Timer( description: "InsertionSort-random array",consumer);
        Benchmark_Timer timer2 = new Benchmark_Timer( description: "InsertionSort-ordered array",consumer);
        Benchmark_Timer timer3 = new Benchmark_Timer( description: "InsertionSort-partially-ordered array",consumer);
        Benchmark_Timer timer4 = new Benchmark_Timer( description: "InsertionSort-reverse-ordered array",consumer);

        System.out.println(timer1.runFromSupplier(supplier1, m: 50));
        System.out.println(timer2.runFromSupplier(supplier2, m: 50));
        System.out.println(timer3.runFromSupplier(supplier3, m: 50));
        System.out.println(timer4.runFromSupplier(supplier4, m: 50));
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=547
2020-10-06 20:23:16 INFO  Benchmark_Timer - Begin run: InsertionSort-random array with 50 runs
1474895.88
2020-10-06 20:23:16 INFO  Benchmark_Timer - Begin run: InsertionSort-ordered array with 50 runs
1474895.9
2020-10-06 20:23:16 INFO  Benchmark_Timer - Begin run: InsertionSort-partially-ordered array with 50 runs
1474895.9
2020-10-06 20:23:16 INFO  Benchmark_Timer - Begin run: InsertionSort-reverse-ordered array with 50 runs
1474895.92

Process finished with exit code 0
```

Xi Zhao (NUID: 001054333)

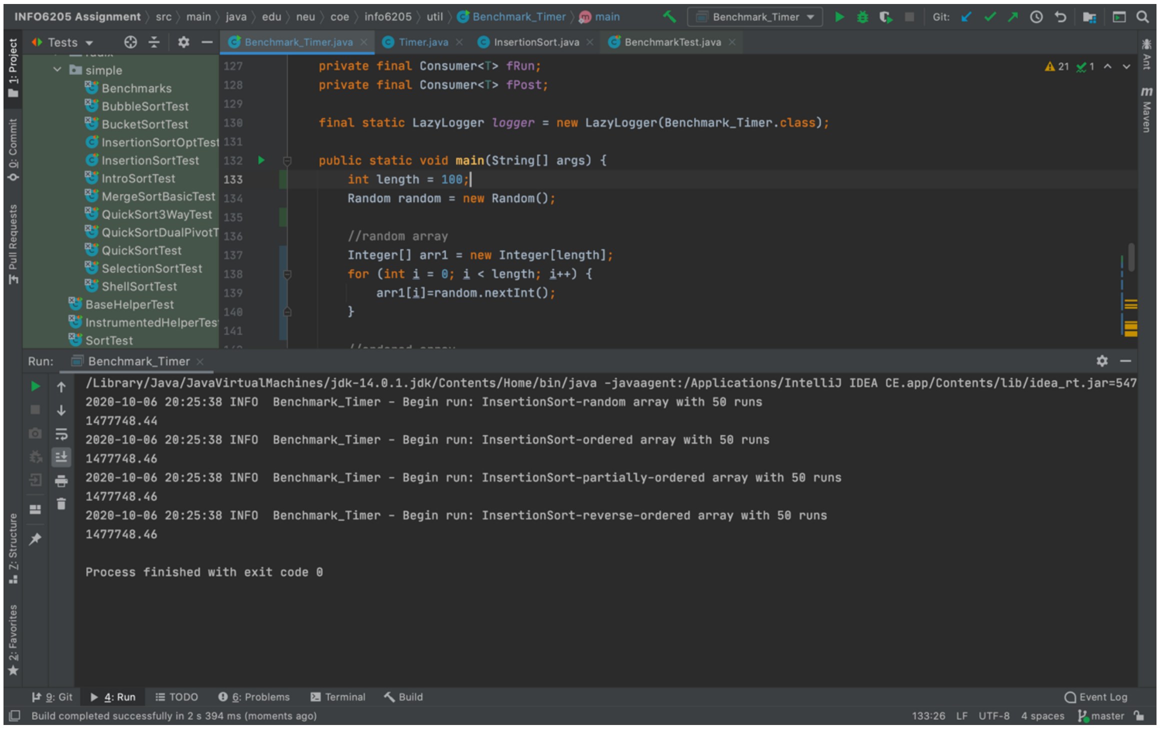
2. 4 different arrays of 20 elements respectively:



```java
    private final Consumer<T> fPost;

    final static LazyLogger logger = new LazyLogger(Benchmark_Timer.class);

    public static void main(String[] args) {
        int length = 20;
        Random random = new Random();

        //random array
        Integer[] arr1 = new Integer[length];
        for (int i = 0; i < length; i++) {
            arr1[i]=random.nextInt();
        }

        //ordered array
        Integer[] arr2 = new Integer[length];
```

```
/Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=547
2020-10-06 20:24:53 INFO  Benchmark_Timer - Begin run: InsertionSort-random array with 50 runs
1476843.02
2020-10-06 20:24:53 INFO  Benchmark_Timer - Begin run: InsertionSort-ordered array with 50 runs
1476843.04
2020-10-06 20:24:53 INFO  Benchmark_Timer - Begin run: InsertionSort-partially-ordered array with 50 runs
1476843.04
2020-10-06 20:24:53 INFO  Benchmark_Timer - Begin run: InsertionSort-reverse-ordered array with 50 runs
1476843.04

Process finished with exit code 0
```

3. 4 different arrays of 100 elements respectively:

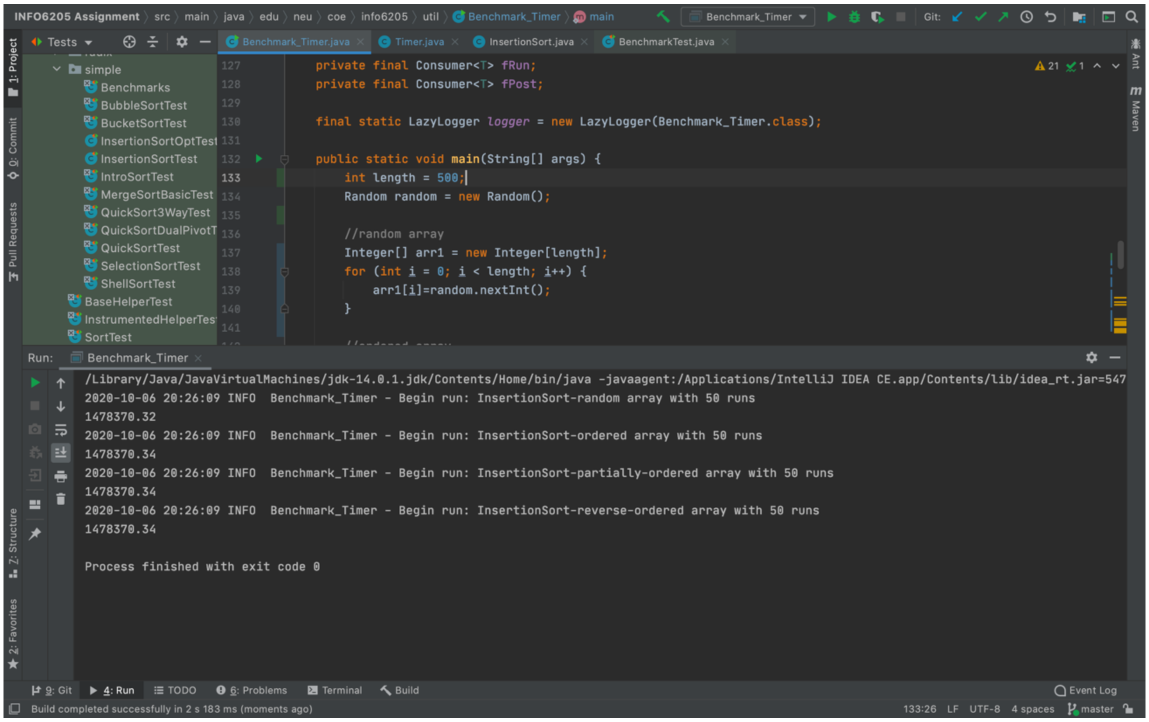


```java
    private final Consumer<T> fRun;
    private final Consumer<T> fPost;

    final static LazyLogger logger = new LazyLogger(Benchmark_Timer.class);

    public static void main(String[] args) {
        int length = 100;
        Random random = new Random();

        //random array
        Integer[] arr1 = new Integer[length];
        for (int i = 0; i < length; i++) {
            arr1[i]=random.nextInt();
        }

        //ordered array
```

```
/Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=547
2020-10-06 20:25:38 INFO  Benchmark_Timer - Begin run: InsertionSort-random array with 50 runs
1477748.44
2020-10-06 20:25:38 INFO  Benchmark_Timer - Begin run: InsertionSort-ordered array with 50 runs
1477748.46
2020-10-06 20:25:38 INFO  Benchmark_Timer - Begin run: InsertionSort-partially-ordered array with 50 runs
1477748.46
2020-10-06 20:25:38 INFO  Benchmark_Timer - Begin run: InsertionSort-reverse-ordered array with 50 runs
1477748.46

Process finished with exit code 0
```

Xi Zhao (NUID: 001054333)

4. 4 different arrays of 500 elements respectively:

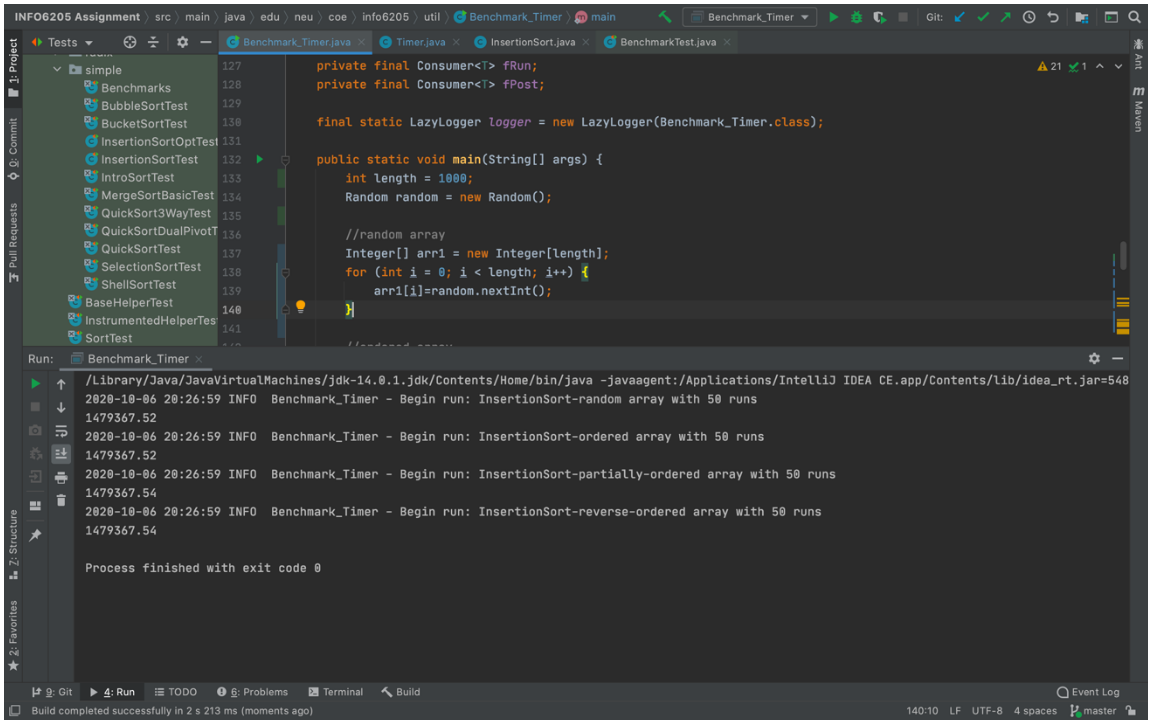


```java
    private final Consumer<T> fRun;
    private final Consumer<T> fPost;

    final static LazyLogger logger = new LazyLogger(Benchmark_Timer.class);

    public static void main(String[] args) {
        int length = 500;
        Random random = new Random();

        //random array
        Integer[] arr1 = new Integer[length];
        for (int i = 0; i < length; i++) {
            arr1[i]=random.nextInt();
        }
```

```
/Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=547
2020-10-06 20:26:09 INFO  Benchmark_Timer - Begin run: InsertionSort-random array with 50 runs
1478370.32
2020-10-06 20:26:09 INFO  Benchmark_Timer - Begin run: InsertionSort-ordered array with 50 runs
1478370.34
2020-10-06 20:26:09 INFO  Benchmark_Timer - Begin run: InsertionSort-partially-ordered array with 50 runs
1478370.34
2020-10-06 20:26:09 INFO  Benchmark_Timer - Begin run: InsertionSort-reverse-ordered array with 50 runs
1478370.34

Process finished with exit code 0
```

Build completed successfully in 2 s 183 ms (moments ago)   133:26 LF UTF-8 4 spaces master

5. 4 different arrays of 1000 elements respectively:



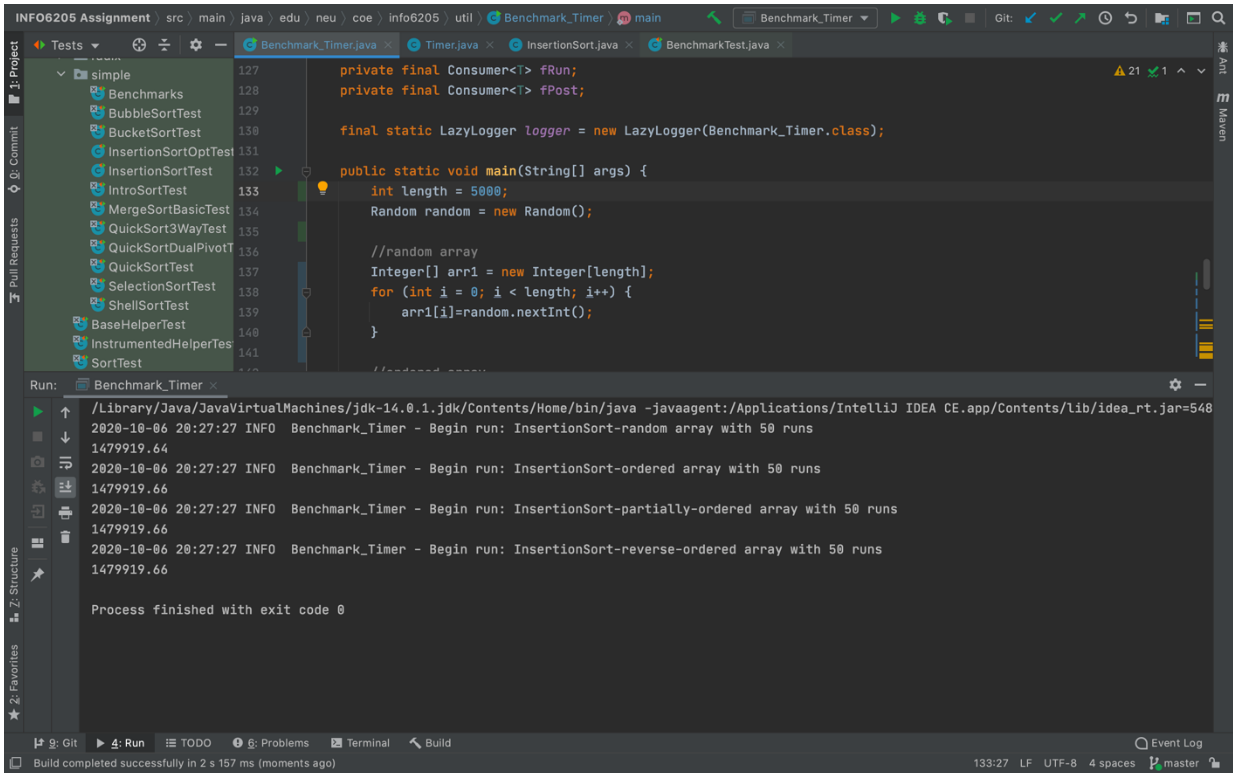
```java
    private final Consumer<T> fRun;
    private final Consumer<T> fPost;

    final static LazyLogger logger = new LazyLogger(Benchmark_Timer.class);

    public static void main(String[] args) {
        int length = 1000;
        Random random = new Random();

        //random array
        Integer[] arr1 = new Integer[length];
        for (int i = 0; i < length; i++) {
            arr1[i]=random.nextInt();
        }
```

```
/Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=548
2020-10-06 20:26:59 INFO  Benchmark_Timer - Begin run: InsertionSort-random array with 50 runs
1479367.52
2020-10-06 20:26:59 INFO  Benchmark_Timer - Begin run: InsertionSort-ordered array with 50 runs
1479367.52
2020-10-06 20:26:59 INFO  Benchmark_Timer - Begin run: InsertionSort-partially-ordered array with 50 runs
1479367.54
2020-10-06 20:26:59 INFO  Benchmark_Timer - Begin run: InsertionSort-reverse-ordered array with 50 runs
1479367.54

Process finished with exit code 0
```

Build completed successfully in 2 s 213 ms (moments ago)   140:10 LF UTF-8 4 spaces master

Xi Zhao (NUID: 001054333)

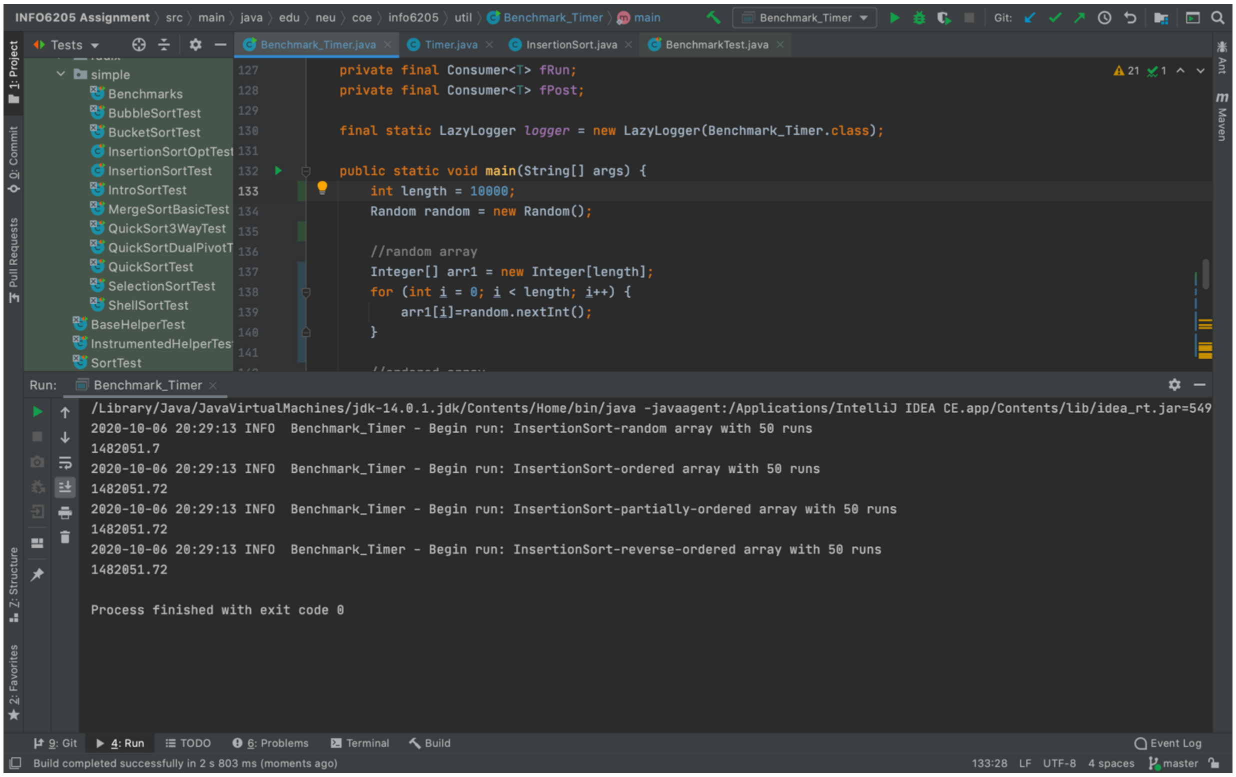
6. 4 different arrays of 5000 elements respectively:



```java
    private final Consumer<T> fRun;
    private final Consumer<T> fPost;

    final static LazyLogger logger = new LazyLogger(Benchmark_Timer.class);

    public static void main(String[] args) {
        int length = 5000;
        Random random = new Random();

        //random array
        Integer[] arr1 = new Integer[length];
        for (int i = 0; i < length; i++) {
            arr1[i]=random.nextInt();
        }
```

```
/Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=548
2020-10-06 20:27:27 INFO  Benchmark_Timer - Begin run: InsertionSort-random array with 50 runs
1479919.64
2020-10-06 20:27:27 INFO  Benchmark_Timer - Begin run: InsertionSort-ordered array with 50 runs
1479919.66
2020-10-06 20:27:27 INFO  Benchmark_Timer - Begin run: InsertionSort-partially-ordered array with 50 runs
1479919.66
2020-10-06 20:27:27 INFO  Benchmark_Timer - Begin run: InsertionSort-reverse-ordered array with 50 runs
1479919.66

Process finished with exit code 0
```

7. 4 different arrays of 10000 elements respectively:



```java
    private final Consumer<T> fRun;
    private final Consumer<T> fPost;

    final static LazyLogger logger = new LazyLogger(Benchmark_Timer.class);

    public static void main(String[] args) {
        int length = 10000;
        Random random = new Random();

        //random array
        Integer[] arr1 = new Integer[length];
        for (int i = 0; i < length; i++) {
            arr1[i]=random.nextInt();
        }
```

```
/Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=549
2020-10-06 20:29:13 INFO  Benchmark_Timer - Begin run: InsertionSort-random array with 50 runs
1482051.7
2020-10-06 20:29:13 INFO  Benchmark_Timer - Begin run: InsertionSort-ordered array with 50 runs
1482051.72
2020-10-06 20:29:13 INFO  Benchmark_Timer - Begin run: InsertionSort-partially-ordered array with 50 runs
1482051.72
2020-10-06 20:29:13 INFO  Benchmark_Timer - Begin run: InsertionSort-reverse-ordered array with 50 runs
1482051.72

Process finished with exit code 0
```

- **Relationship conclusion**
  The relationship between the array length N value and function time is power. Roughly, the formula of the array size n(x) and the function time (y) is
  y = 1E+06x^0.0006.

- **Evidence to support relationship**
  By using Excel to analyze and deduce the relationship of the two numbers, scatter plot with different trend lines are generated to prove the relationship of the two numbers. At the same time, the correlation test which is used to evaluate the association between two or more variables is automatically conducted along with the trend line. Correlation coefficient R is comprised between -1 and 1:

  i. -1 indicates a strong negative correlation: this means that every time x increases, y decreases;

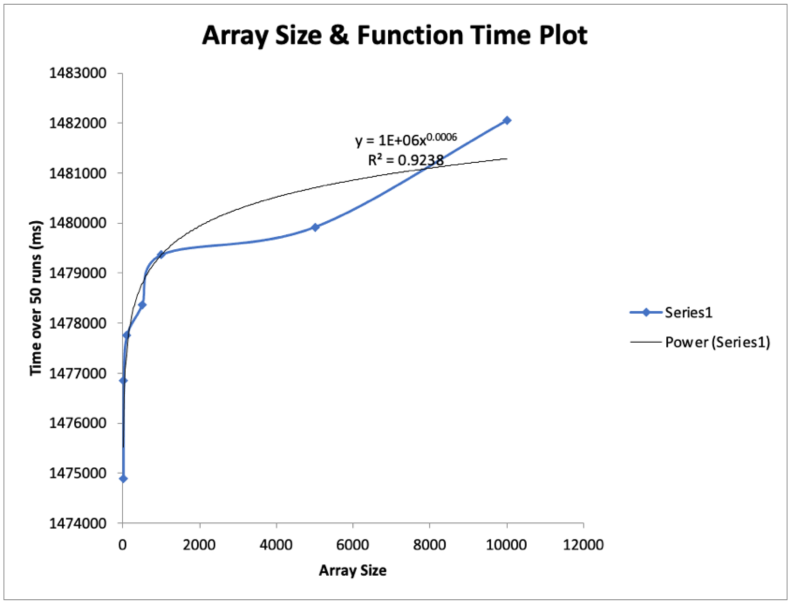
  ii. 0 means that there is no association between the two variables (x and y);

  iii. 1 indicates a strong positive correlation: this means that y increases with x.

  Usually, R^2 is used to demonstrate the correlation degree. R^2 is bigger and the correlation is stronger.

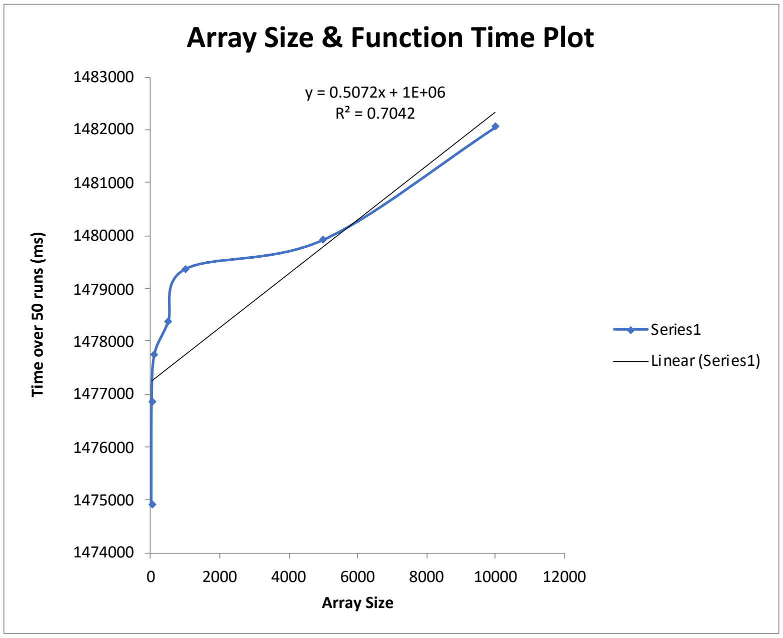
  In this case, after comparing 4 possible trend lines which illustrate different possible correlation between the two numbers:

  1. Power, y = 1E+06x^0.0006. $R^2 = 0.9238$

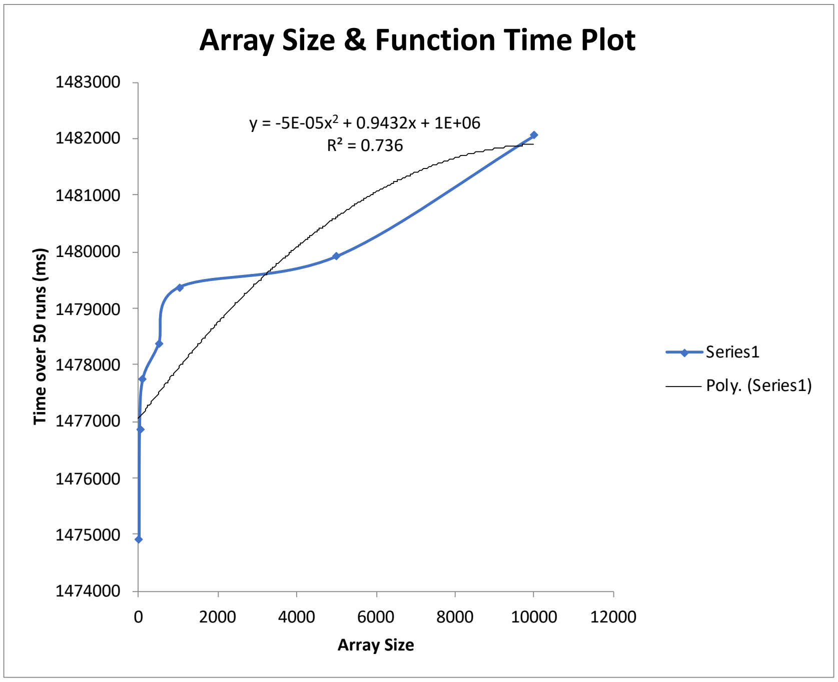


Xi Zhao (NUID: 001054333)

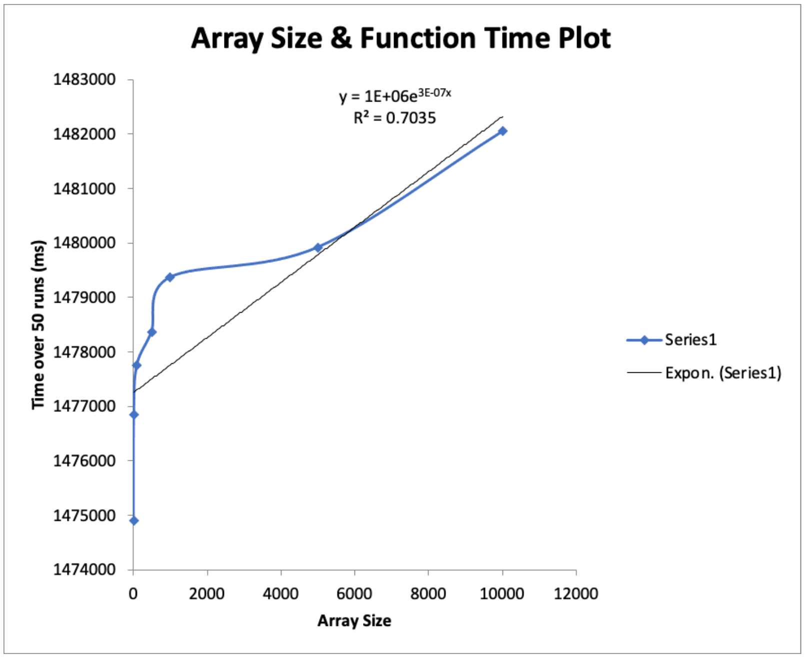
2. logarithmic, y = 835.53ln(x) + 1E+06. R^2 = 0.9237



3. Polynomial, y = -5E-05x^2 + 0.9432x + 1E+06. R^2 = 0.736



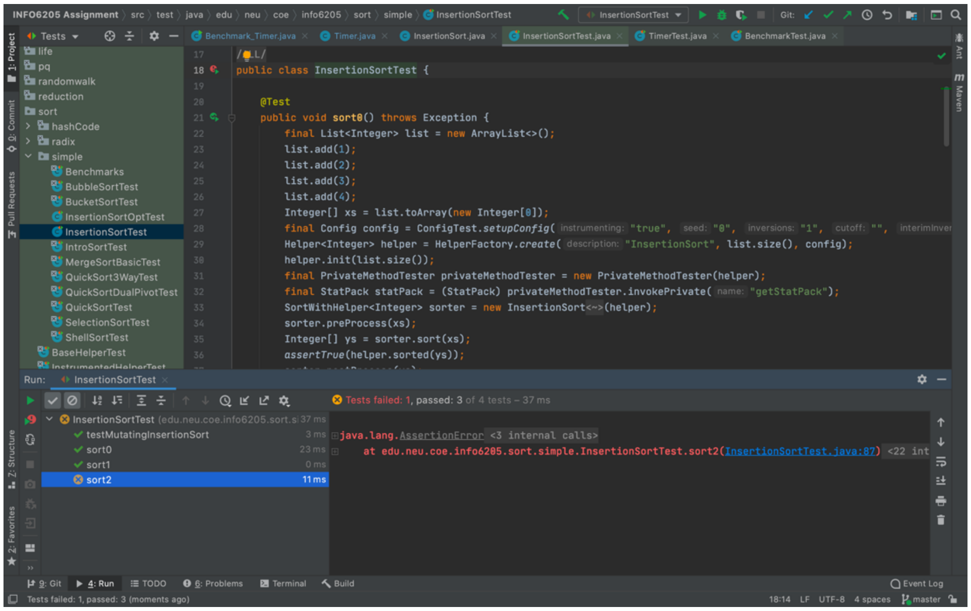
4. Exponential, $y = 1E+06e^{3E-07x}$. $R^2 = 0.7035$



The highest R^2 is the power trend line. Therefore, the highest possible correlation of the array size n and the function time is power correlation.

- **Screenshot of Unit test**

  Insertion sort test:

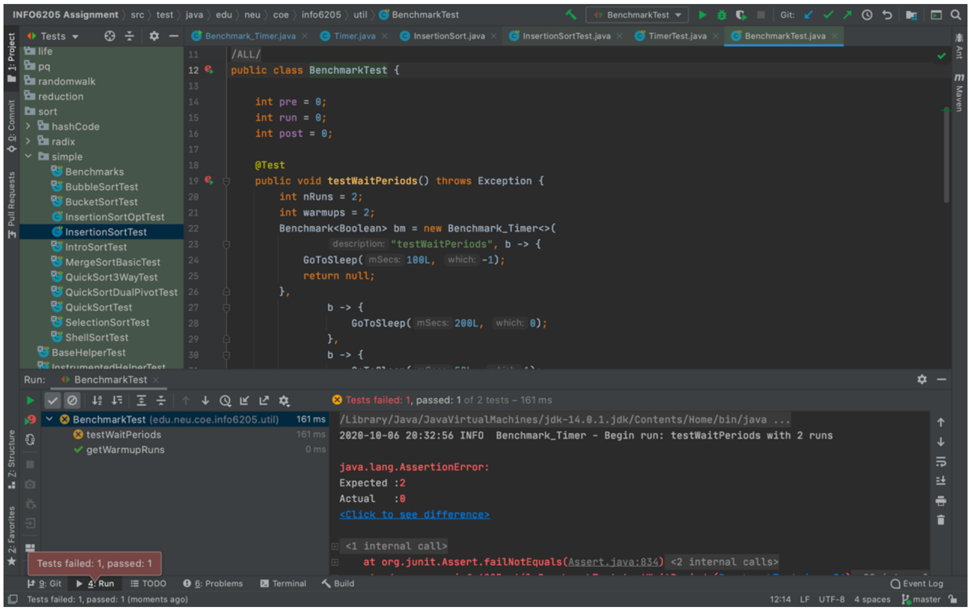


```
/*..L/
public class InsertionSortTest {

    @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( instrumenting: "true",  seed: "0",  inversions: "1",  cutoff: "",  interimInver
        Helper<Integer> helper = HelperFactory.create( description: "InsertionSort", list.size(), config);
        helper.init(list.size());
        final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);
        final StatPack statPack = (StatPack) privateMethodTester.invokePrivate( name: "getStatPack");
        SortWithHelper<Integer> sorter = new InsertionSort<~>(helper);
        sorter.preProcess(xs);
        Integer[] ys = sorter.sort(xs);
        assertTrue(helper.sorted(ys));
```

```
Tests failed: 1, passed: 3 of 4 tests – 37 ms

InsertionSortTest (edu.neu.coe.info6205.sort.si 37 ms
    testMutatingInsertionSort                    3 ms
    sort0                                       23 ms
    sort1                                        0 ms
    sort2                                       11 ms

java.lang.AssertionError <3 internal calls>
    at edu.neu.coe.info6205.sort.simple.InsertionSortTest.sort2(InsertionSortTest.java:87) <22 int
```

```
Tests failed: 1, passed: 3 (moments ago)                18:14  LF  UTF-8  4 spaces  master
```

Benchmark test:



```java
/ALL/
public class BenchmarkTest {

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

    @Test
    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 -> {
```

Tests failed: 1, passed: 1 of 2 tests – 161 ms

BenchmarkTest (edu.neu.coe.info6205.util) 161 ms
testWaitPeriods 161 ms
getWarmupRuns 0 ms

```
/Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Contents/Home/bin/java ...
2020-10-06 20:32:56 INFO  Benchmark_Timer - Begin run: testWaitPeriods with 2 runs

java.lang.AssertionError:
Expected :2
Actual   :0
<Click to see difference>

<1 internal call>
    at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal calls>
```

Tests failed: 1, passed: 1

Tests failed: 1, passed: 1 (moments ago)

Reference:

Taylor, R. (1990). Interpretation of the Correlation Coefficient: A Basic Review. *Journal of Diagnostic Medical Sonography, 6*(1), 35-39. doi:10.1177/875647939000600106