

INFO 6205

Program Structures & Algorithms

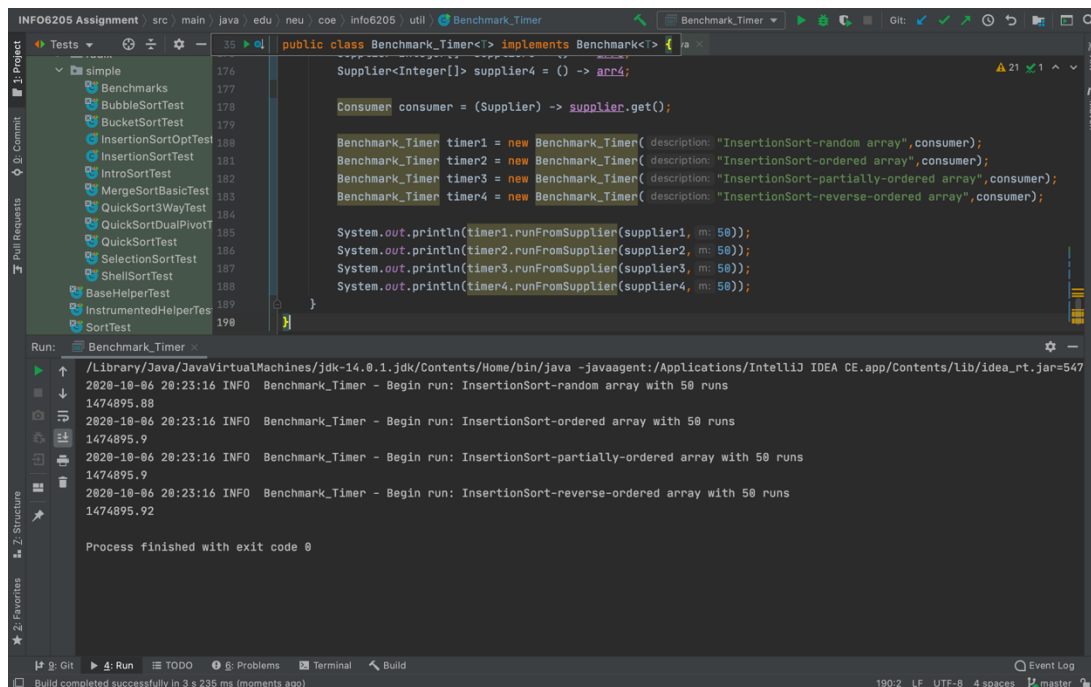
Fall 2020

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:



The screenshot shows an IDE with the following components:

- Project Explorer:** A tree view on the left showing a project named 'INFO6205 Assignment'. Under 'src/main/java/edu/neu/coe/info6205/util', there is a 'Benchmarks' folder containing several test classes like 'BubbleSortTest', 'BucketSortTest', 'InsertionSortOptTest', etc.
- Code Editor:** The main window displays the 'Benchmark_Timer.java' file. It implements the 'Benchmark' interface. It defines four suppliers: 'supplier1' (random), 'supplier2' (ordered), 'supplier3' (partially-ordered), and 'supplier4' (reverse-ordered). Each supplier provides an array of 10 integers. The 'run' method uses 'Benchmark_Timer' objects to measure the time taken to sort these arrays using insertion sort. The output is printed to the console.
- Run Console:** At the bottom, the output of the program is shown. It indicates that the program was run successfully and provides the execution time for each of the four array types. The times are approximately 1474895.88, 1474895.9, 1474895.9, and 1474895.92 milliseconds.

2. 4 different arrays of 20 elements respectively:

The screenshot shows the IntelliJ IDEA IDE with the `Benchmark_Timer.java` file open. The code defines a `main` method that generates four different arrays of 20 elements each: random, ordered, partially-ordered, and reverse-ordered. The `Run` console shows the execution of these benchmarks, each taking approximately 147.6843 seconds to complete 50 runs.

```
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];
    for (int i = 0; i < length; i++) {
        arr2[i] = i;
    }

    //partially-ordered array
    Integer[] arr3 = new Integer[length];
    for (int i = 0; i < length; i++) {
        arr3[i] = random.nextInt(i);
    }

    //reverse-ordered array
    Integer[] arr4 = new Integer[length];
    for (int i = 0; i < length; i++) {
        arr4[i] = length - i - 1;
    }

    //run benchmarks
    BenchmarkTest.run(arr1, arr2, arr3, arr4);
}
```

Run: Benchmark_Timer

```
/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
147.6843.02
2020-10-06 20:24:53 INFO Benchmark_Timer - Begin run: InsertionSort-ordered array with 50 runs
147.6843.04
2020-10-06 20:24:53 INFO Benchmark_Timer - Begin run: InsertionSort-partially-ordered array with 50 runs
147.6843.04
2020-10-06 20:24:53 INFO Benchmark_Timer - Begin run: InsertionSort-reverse-ordered array with 50 runs
147.6843.04

Process finished with exit code 0
```

3. 4 different arrays of 100 elements respectively:

The screenshot shows the IntelliJ IDEA IDE with the `Benchmark_Timer.java` file open. The code defines a `main` method that generates four different arrays of 100 elements each: random, ordered, partially-ordered, and reverse-ordered. The `Run` console shows the execution of these benchmarks, each taking approximately 147.7748 seconds to complete 50 runs.

```
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
    Integer[] arr2 = new Integer[length];
    for (int i = 0; i < length; i++) {
        arr2[i] = i;
    }

    //partially-ordered array
    Integer[] arr3 = new Integer[length];
    for (int i = 0; i < length; i++) {
        arr3[i] = random.nextInt(i);
    }

    //reverse-ordered array
    Integer[] arr4 = new Integer[length];
    for (int i = 0; i < length; i++) {
        arr4[i] = length - i - 1;
    }

    //run benchmarks
    BenchmarkTest.run(arr1, arr2, arr3, arr4);
}
```

Run: Benchmark_Timer

```
/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
147.7748.44
2020-10-06 20:25:38 INFO Benchmark_Timer - Begin run: InsertionSort-ordered array with 50 runs
147.7748.46
2020-10-06 20:25:38 INFO Benchmark_Timer - Begin run: InsertionSort-partially-ordered array with 50 runs
147.7748.46
2020-10-06 20:25:38 INFO Benchmark_Timer - Begin run: InsertionSort-reverse-ordered array with 50 runs
147.7748.46

Process finished with exit code 0
```

4. 4 different arrays of 500 elements respectively:

```
INFO6205 Assignment > src > main > java > edu > neu > coe > info6205 > util > Benchmark_Timer > main
Tests
  simple
    Benchmarks
    BubbleSortTest
    BucketSortTest
    InsertionSortOptTest
    InsertionSortTest
    IntroSortTest
    MergeSortBasicTest
    QuickSort3WayTest
    QuickSortDualPivotTest
    QuickSortTest
    SelectionSortTest
    ShellSortTest
    BaseHelperTest
    InstrumentedHelperTest
    SortTest
  Run: Benchmark_Timer
    /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)
```

5. 4 different arrays of 1000 elements respectively:

```
INFO6205 Assignment > src > main > java > edu > neu > coe > info6205 > util > Benchmark_Timer > main
Tests
  simple
    Benchmarks
    BubbleSortTest
    BucketSortTest
    InsertionSortOptTest
    InsertionSortTest
    IntroSortTest
    MergeSortBasicTest
    QuickSort3WayTest
    QuickSortDualPivotTest
    QuickSortTest
    SelectionSortTest
    ShellSortTest
    BaseHelperTest
    InstrumentedHelperTest
    SortTest
  Run: Benchmark_Timer
    /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)
```

6. 4 different arrays of 5000 elements respectively:

```
INFO6205 Assignment > src > main > java > edu > neu > coe > info6205 > util > Benchmark_Timer > main
Tests
Benchmark_Timer.java x Timer.java x InsertionSort.java x BenchmarkTest.java x
simple 127
Benchmarks 128
BubbleSortTest 129
BucketSortTest 130
InsertionSortOptTest 131
InsertionSortTest 132
IntroSortTest 133
MergeSortBasicTest 134
QuickSort3WayTest 135
QuickSortDualPivotT 136
QuickSortTest 137
SelectionSortTest 138
ShellSortTest 139
BaseHelperTest 140
InstrumentedHelperTes 141
SortTest 142

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();
    }

    //ordered array
    Integer[] arr2 = new Integer[length];
    for (int i = 0; i < length; i++) {
        arr2[i]=i;
    }

    //partially ordered array
    Integer[] arr3 = new Integer[length];
    for (int i = 0; i < length; i++) {
        arr3[i]=i%2;
    }

    //reverse ordered array
    Integer[] arr4 = new Integer[length];
    for (int i = 0; i < length; i++) {
        arr4[i]=length-i-1;
    }

    fRun.accept(arr1);
    fRun.accept(arr2);
    fRun.accept(arr3);
    fRun.accept(arr4);
    fPost.accept(arr1);
    fPost.accept(arr2);
    fPost.accept(arr3);
    fPost.accept(arr4);
}

Run: Benchmark_Timer x
/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

Build completed successfully in 2 s 157 ms (moments ago)
133:27 LF UTF-8 4 spaces master
```

7. 4 different arrays of 10000 elements respectively:

```
INFO6205 Assignment > src > main > java > edu > neu > coe > info6205 > util > Benchmark_Timer > main
Tests
Benchmark_Timer.java x Timer.java x InsertionSort.java x BenchmarkTest.java x
simple 127
Benchmarks 128
BubbleSortTest 129
BucketSortTest 130
InsertionSortOptTest 131
InsertionSortTest 132
IntroSortTest 133
MergeSortBasicTest 134
QuickSort3WayTest 135
QuickSortDualPivotT 136
QuickSortTest 137
SelectionSortTest 138
ShellSortTest 139
BaseHelperTest 140
InstrumentedHelperTes 141
SortTest 142

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();
    }

    //ordered array
    Integer[] arr2 = new Integer[length];
    for (int i = 0; i < length; i++) {
        arr2[i]=i;
    }

    //partially ordered array
    Integer[] arr3 = new Integer[length];
    for (int i = 0; i < length; i++) {
        arr3[i]=i%2;
    }

    //reverse ordered array
    Integer[] arr4 = new Integer[length];
    for (int i = 0; i < length; i++) {
        arr4[i]=length-i-1;
    }

    fRun.accept(arr1);
    fRun.accept(arr2);
    fRun.accept(arr3);
    fRun.accept(arr4);
    fPost.accept(arr1);
    fPost.accept(arr2);
    fPost.accept(arr3);
    fPost.accept(arr4);
}

Run: Benchmark_Timer x
/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

Build completed successfully in 2 s 803 ms (moments ago)
133:28 LF UTF-8 4 spaces master
```

- **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**

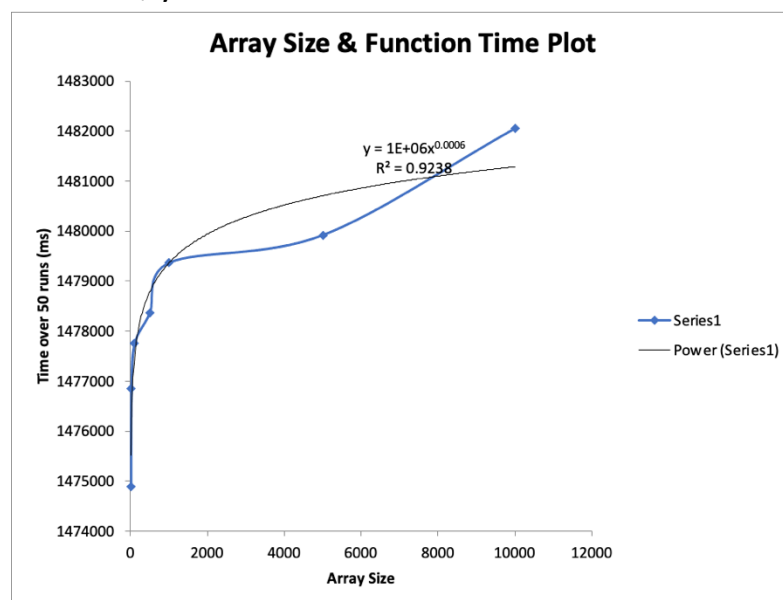
By using Excel to analyze and deduce the relationship of the two numbers, scatter plot with different trend lines are generated to prove the relationship of the two numbers. At the same time, the correlation test which is used to evaluate the association between two or more variables is automatically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

- 1 indicates a strong negative correlation: this means that every time x increases, y decreases;
- 0 means that there is no association between the two variables (x and y);
- 1 indicates a strong positive correlation: this means that y increases with x.

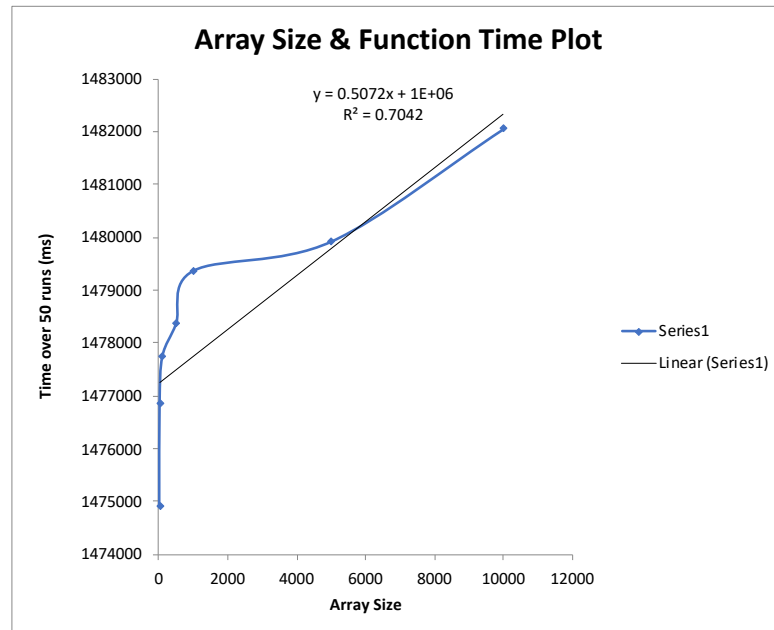
Usually, R^2 is used to demonstrate the correlation degree. R^2 is bigger and the correlation is stronger.

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

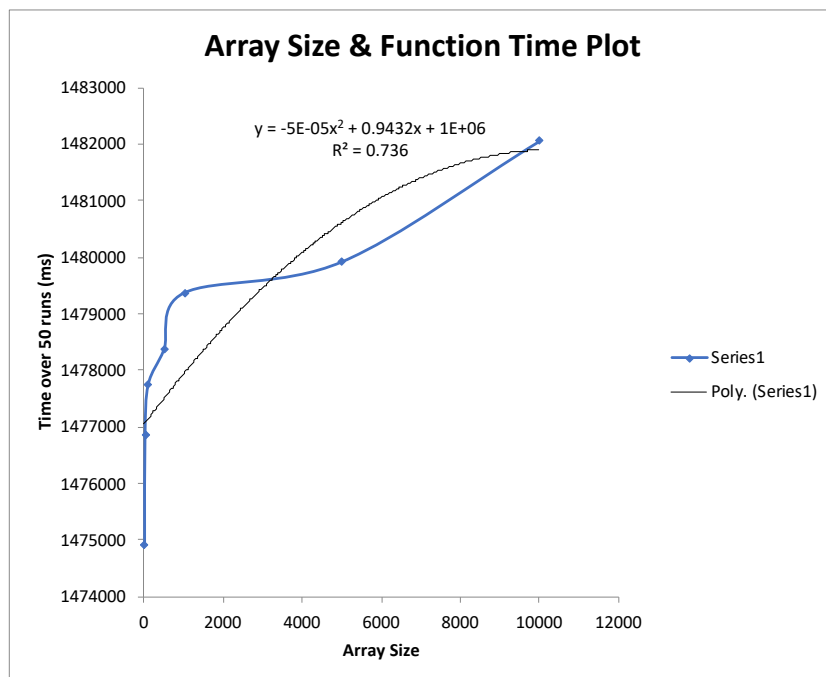
1. Power, $y = 1E+06x^{0.0006}$. $R^2 = 0.9238$



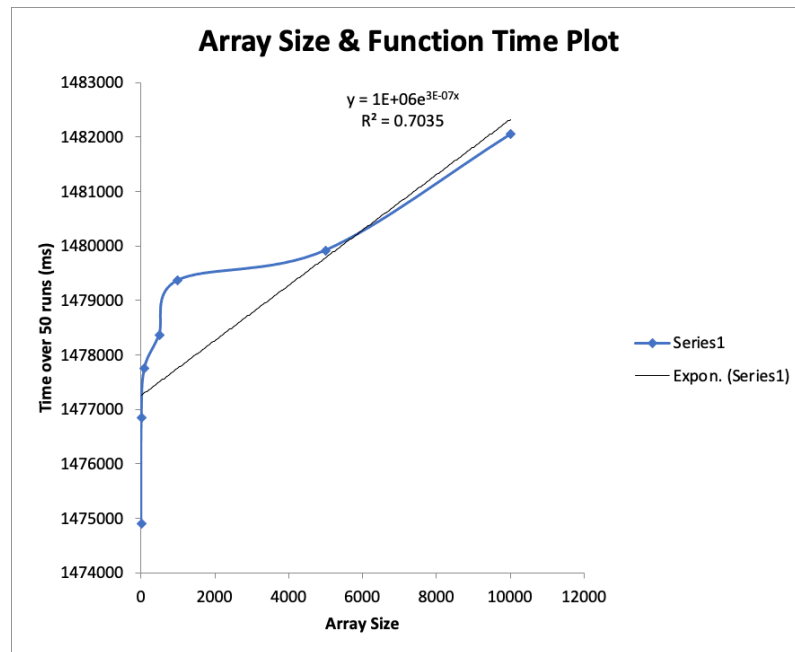
2. logarithmic, $y = 835.53\ln(x) + 1E+06$. $R^2 = 0.9237$



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:

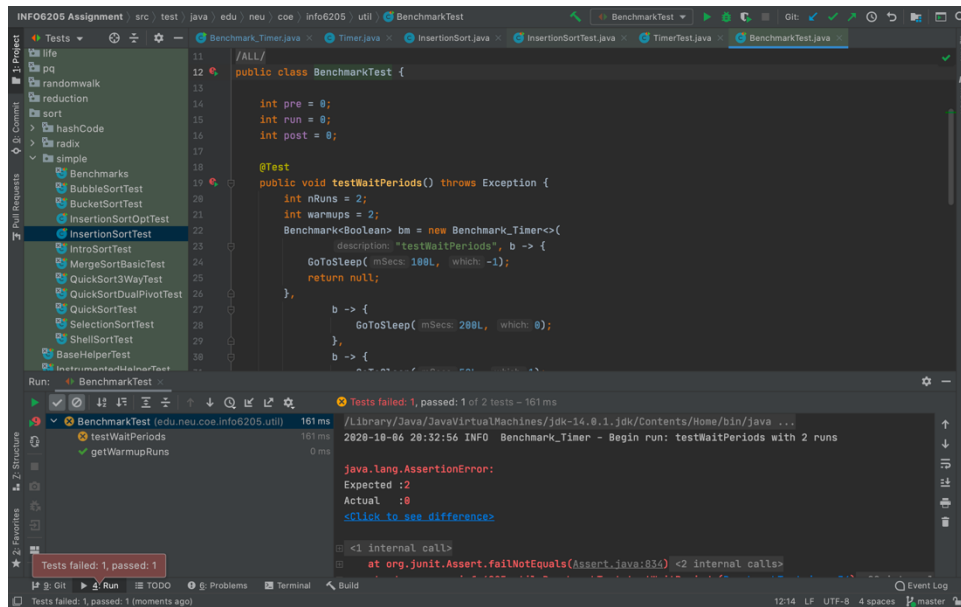
```
public class InsertionSortTest {  
  
    @Test  
    public void sort0() throws Exception {  
        final List<Integer> list = new ArrayList<>();  
        list.add(1);  
        list.add(2);  
        list.add(3);  
        list.add(4);  
        Integer[] xs = list.toArray(new Integer[0]);  
        final Config config = ConfigTest.setupConfig( instrumenting: "true", seed: "0", inversions: "1", cutoff: "", interminver  
        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));  
    }  
}
```

Run: InsertionSortTest
Tests failed: 1, passed: 3 of 4 tests - 37 ms

Test	Time
sort0	3 ms
sort1	23 ms
sort2	11 ms

Xi Zhao (NUID: 001054333)

Benchmark test:



```
11 //ALL/
12 public class BenchmarkTest {
13
14     int pre = 0;
15     int run = 0;
16     int post = 0;
17
18     @Test
19     public void testWaitPeriods() throws Exception {
20         int nRuns = 2;
21         int warmups = 2;
22         Benchmark<Boolean> bm = new Benchmark_Timer<>() {
23             description: "testWaitPeriods", b -> {
24                 GoToSleep( mSecs: 100L, which: -1);
25                 return null;
26             },
27             b -> {
28                 GoToSleep( mSecs: 200L, which: 0);
29             },
30             b -> {
31                 // ...
32             }
33         };
34         bm.run();
35     }
36 }
37
38 Run: BenchmarkTest
39 Tests failed: 1, passed: 1 of 2 tests - 161 ms
40 testWaitPeriods 161 ms
41 getWarmupRuns 0 ms
42 java.lang.AssertionError:
43     Expected :2
44     Actual   :0
45     <Click to see difference>
46     <1 internal call>
47     at org.junit.Assert.failNotEquals(Assert.java:836) <2 internal calls>
```

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