Program Structures and Algorithms
Spring 2023(SEC –1)
Assignment 3 : BenchMark Insertion Sort

NAME: Shubham Sharad Bagal
NUID: 002708621

Task:

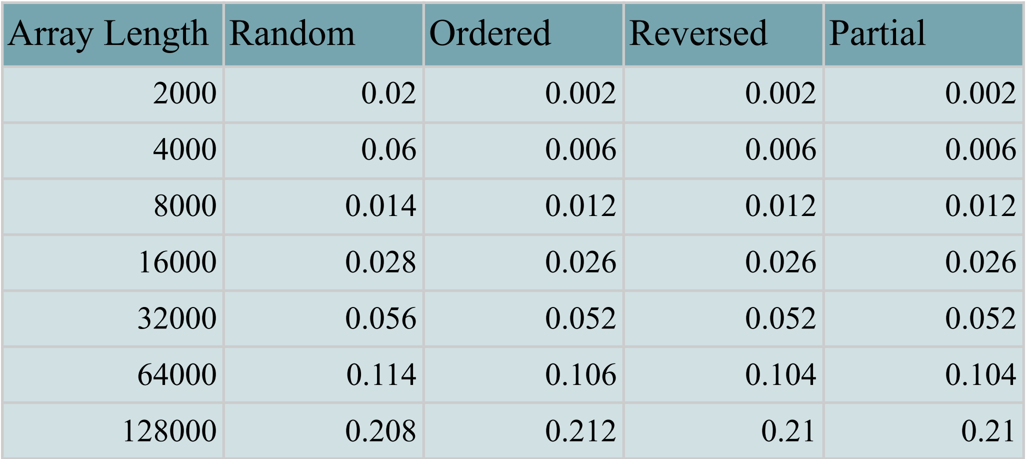
Implement three methods of the Timer class to measure the average running time of target functions. The repeat method takes repetitions, an input supplier, a function to be tested, and pre- and post-functions. GetClock returns clock time in ticks, and toMillisecs converts ticks to milliseconds. Part 2 is to implement InsertionSort, and Part 3 runs a benchmark on four different array orders and seven values of n, reporting results and observations.

Relationship Conclusion:

The relationship between the different initial array orderings (random, ordered, reversed, and partially ordered) and the running time of the Insertion Sort algorithm is being studied. The benchmark measures the average running time for seven different values of n, with n representing the length of the array to be sorted. The results show that there is little difference in running time between the different initial array orderings, with all values being relatively close to each other. This suggests that the initial ordering of the array has little impact on the performance of the Insertion Sort algorithm, at least for the specific data set used in this study.

Evidence to support that conclusion:

According to the doubling hypothesis(Random Array) :
Time $T(n) = aN^b$
Order of growth with N = log ratio = b ≈ 1.3



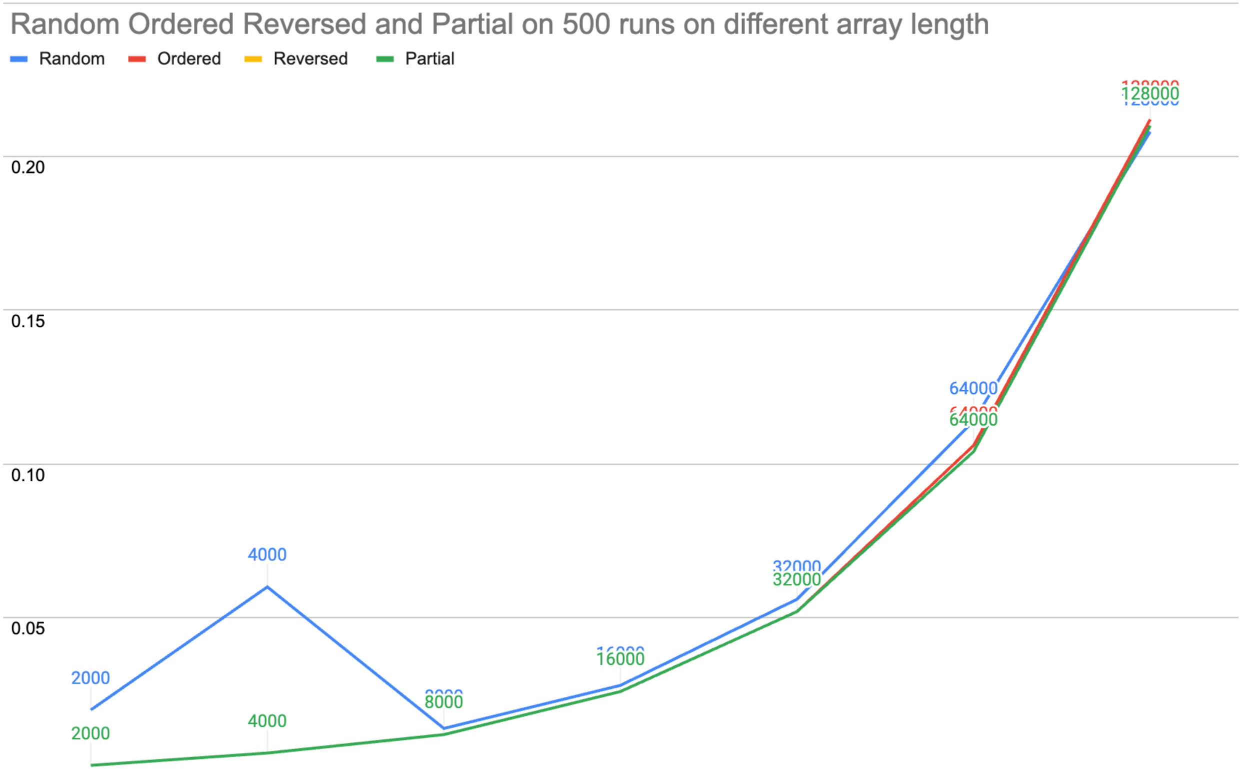
Array Length	Random	Ordered	Reversed	Partial
2000	0.02	0.002	0.002	0.002
4000	0.06	0.006	0.006	0.006
8000	0.014	0.012	0.012	0.012
16000	0.028	0.026	0.026	0.026
32000	0.056	0.052	0.052	0.052
64000	0.114	0.106	0.104	0.104
128000	0.208	0.212	0.21	0.21

Considering , b ≈ 1.3
Time(T) for 16000,
0.028 = a* (16000)1.3
a = 0.028 / (16000)^1.3

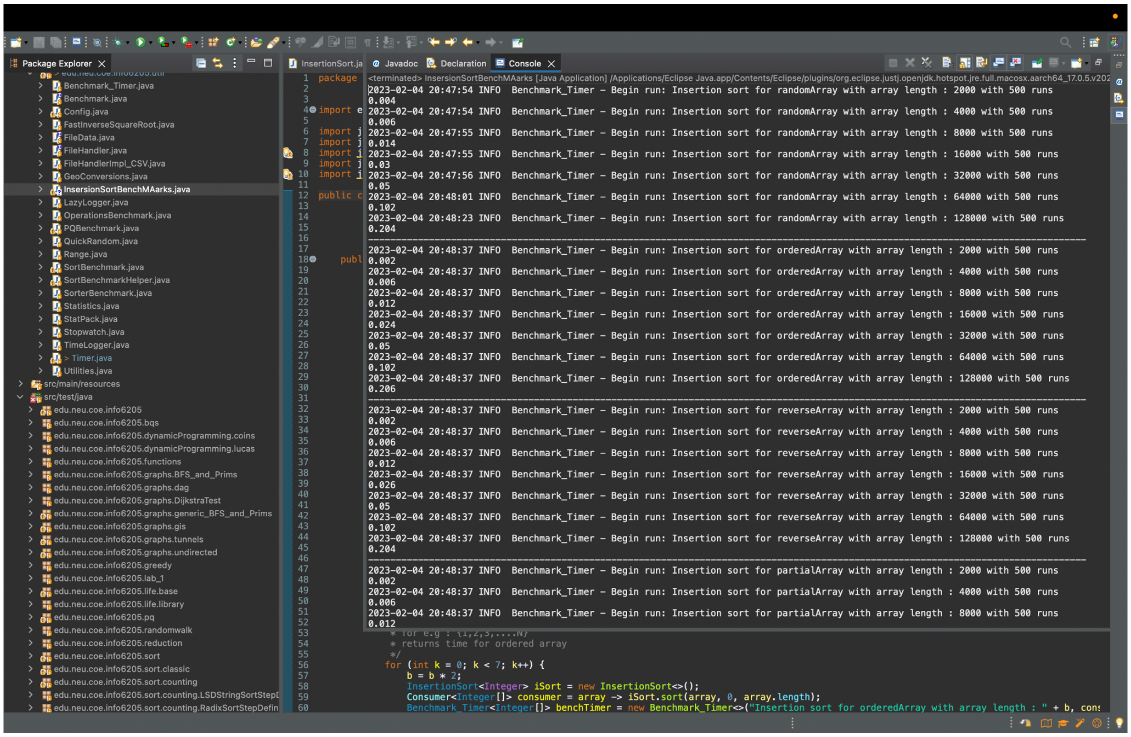
Now, for a=2.808*10-7 , we will calculate T where n=32000

T =2.81*10-7 * (32000)1.3
T = 0.1857

Graphical Representation:

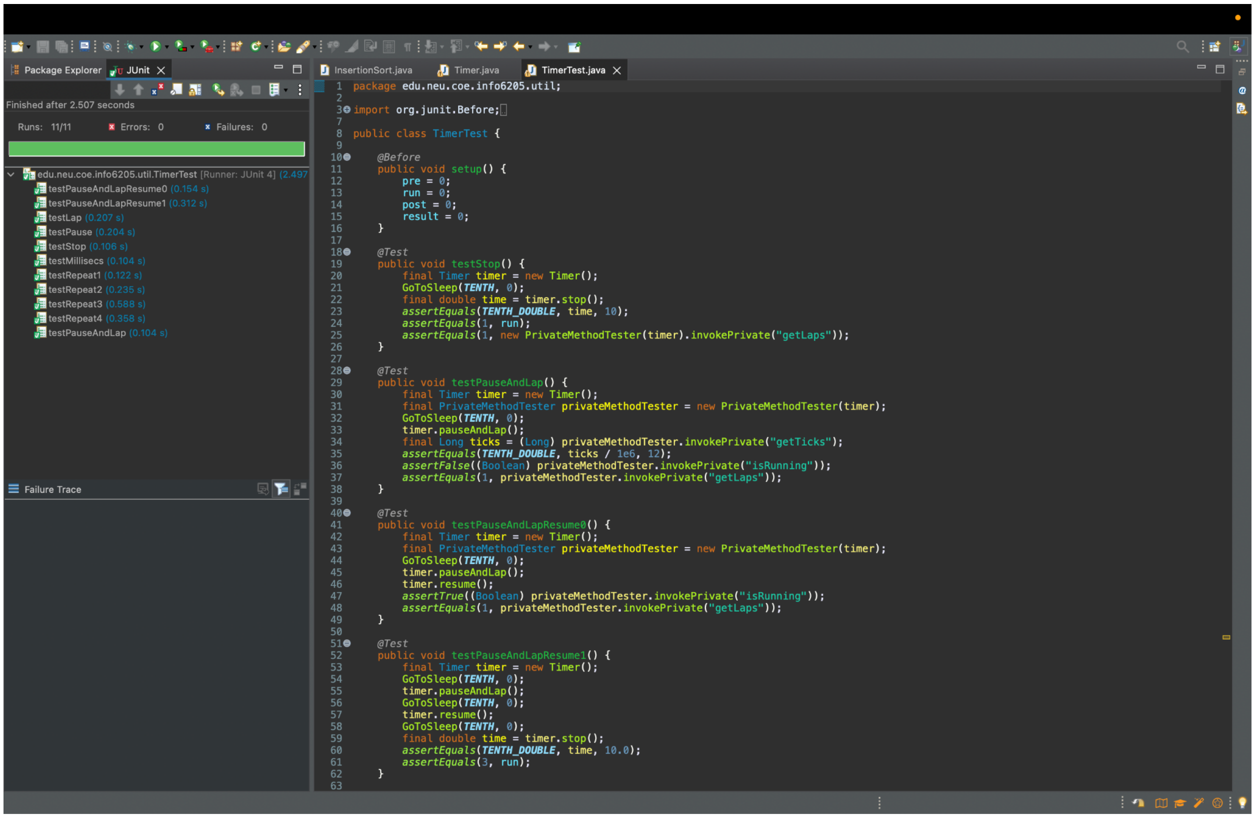


Unit Test Screenshots:


Package Explorer
Benchmark_Timer.java
Benchmark.java
Config.java
FastInverseSquareRoot.java
FileData.java
FileHandler.java
FileHandlerImpl_CSV.java
GeoConversions.java
InsersionSortBenchMAarks.java
LazyLogger.java
OperationsBenchmark.java
PQBenchmark.java
QuickRandom.java
Range.java
SortBenchmark.java
SortBenchmarkHelper.java
SorterBenchmark.java
Statistics.java
StatPack.java
Stopwatch.java
TimeLogger.java
> Timer.java
Utilities.java
src/main/resources
src/test/java
edu.neu.coe.info6205
edu.neu.coe.info6205.bqs
edu.neu.coe.info6205.dynamicProgramming.coins
edu.neu.coe.info6205.dynamicProgramming.lucas
edu.neu.coe.info6205.functions
edu.neu.coe.info6205.graphs.BFS_and_Prims
edu.neu.coe.info6205.graphs.dag
edu.neu.coe.info6205.graphs.DijkstraTest
edu.neu.coe.info6205.graphs.generic_BFS_and_Prims
edu.neu.coe.info6205.graphs.gis
edu.neu.coe.info6205.graphs.tunnels
edu.neu.coe.info6205.graphs.undirected
edu.neu.coe.info6205.greedy
edu.neu.coe.info6205.lab_1
edu.neu.coe.info6205.life.base
edu.neu.coe.info6205.life.library
edu.neu.coe.info6205.pq
edu.neu.coe.info6205.randomwalk
edu.neu.coe.info6205.reduction
edu.neu.coe.info6205.sort
edu.neu.coe.info6205.sort.classic
edu.neu.coe.info6205.sort.counting
edu.neu.coe.info6205.sort.counting.LSDStringSortStepD
edu.neu.coe.info6205.sort.counting.RadixSortStepDefin
InsertionSort.ja
Javadoc
Declaration
Console

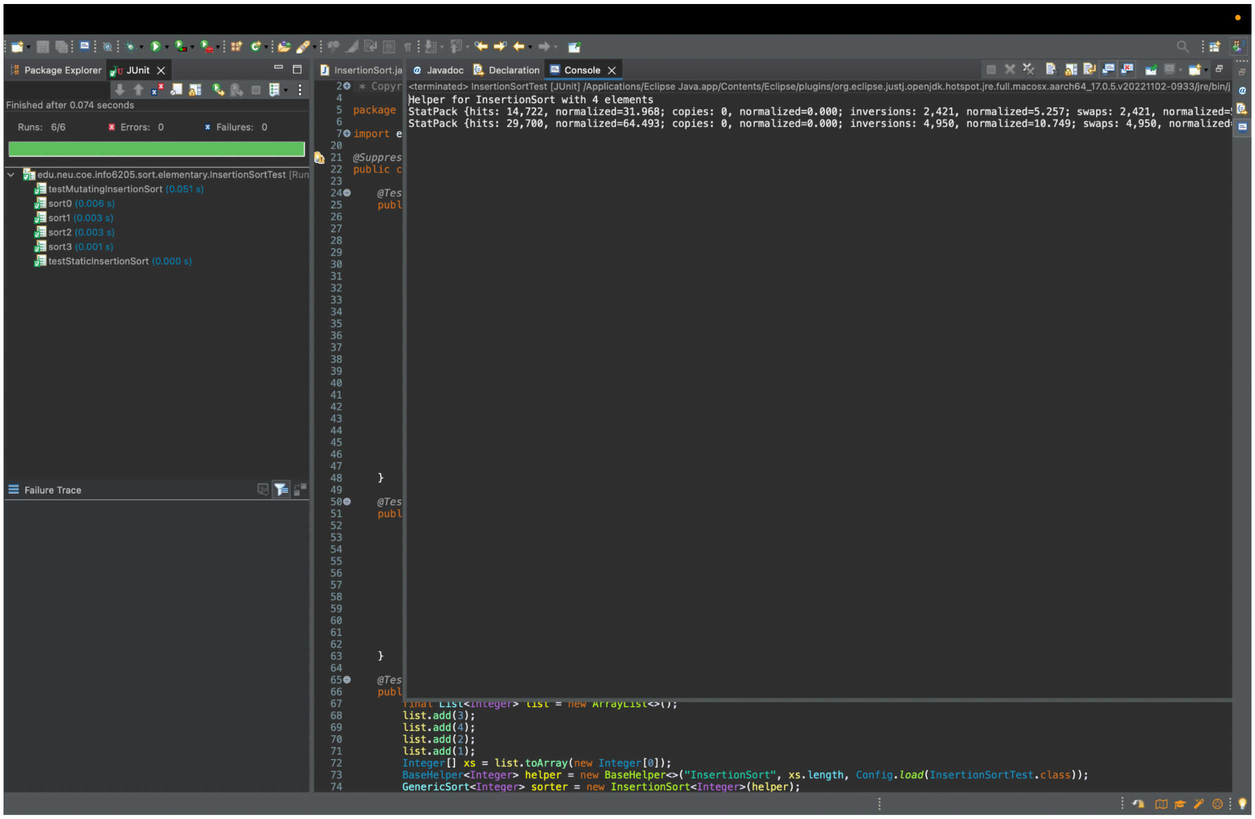
```
<terminated> InsersionSortBenchMAarks [Java Application] /Applications/Eclipse Java.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.aarch64_17.0.5.v202
2023-02-04 20:47:54 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 2000 with 500 runs
0.004
2023-02-04 20:47:54 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 4000 with 500 runs
0.006
2023-02-04 20:47:55 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 8000 with 500 runs
0.014
2023-02-04 20:47:55 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 16000 with 500 runs
0.03
2023-02-04 20:47:56 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 32000 with 500 runs
0.05
2023-02-04 20:48:01 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 64000 with 500 runs
0.102
2023-02-04 20:48:23 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 128000 with 500 runs
0.204
------------------------------------------------------------------------------------------------------------------------------
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 2000 with 500 runs
0.002
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 4000 with 500 runs
0.006
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 8000 with 500 runs
0.012
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 16000 with 500 runs
0.024
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 32000 with 500 runs
0.05
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 64000 with 500 runs
0.102
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 128000 with 500 runs
0.206
------------------------------------------------------------------------------------------------------------------------------
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 2000 with 500 runs
0.002
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 4000 with 500 runs
0.006
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 8000 with 500 runs
0.012
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 16000 with 500 runs
0.026
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 32000 with 500 runs
0.05
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 64000 with 500 runs
0.102
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 128000 with 500 runs
0.204
------------------------------------------------------------------------------------------------------------------------------
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for partialArray with array length : 2000 with 500 runs
0.002
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for partialArray with array length : 4000 with 500 runs
0.006
2023-02-04 20:48:37 INFO  Benchmark_Timer - Begin run: Insertion sort for partialArray with array length : 8000 with 500 runs
0.012
```

```
         * for e.g : {1,2,3,....N}
         * returns time for ordered array
         */
        for (int k = 0; k < 7; k++) {
            b = b * 2;
            InsertionSort<Integer> iSort = new InsertionSort<>();
            Consumer<Integer[]> consumer = array -> iSort.sort(array, 0, array.length);
            Benchmark_Timer<Integer[]> benchTimer = new Benchmark_Timer<>("Insertion sort for orderedArray with array length : " + b, cons
```



```
Finished after 2.507 seconds

Runs:  11/11      Errors:  0      Failures:  0

edu.neu.coe.info6205.util.TimerTest [Runner: JUnit 4] (2.497
    testPauseAndLapResume0 (0.154 s)
    testPauseAndLapResume1 (0.312 s)
    testLap (0.207 s)
    testPause (0.204 s)
    testStop (0.106 s)
    testMillisecs (0.104 s)
    testRepeat1 (0.122 s)
    testRepeat2 (0.235 s)
    testRepeat3 (0.588 s)
    testRepeat4 (0.358 s)
    testPauseAndLap (0.104 s)
```

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

import org.junit.Before;

public class TimerTest {

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

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

    @Test
    public void testPauseAndLap() {
        final Timer timer = new Timer();
        final PrivateMethodTester privateMethodTester = new PrivateMethodTester(timer);
        GoToSleep(TENTH, 0);
        timer.pauseAndLap();
        final Long ticks = (Long) privateMethodTester.invokePrivate("getTicks");
        assertEquals(TENTH_DOUBLE, ticks / 1e6, 12);
        assertFalse((Boolean) privateMethodTester.invokePrivate("isRunning"));
        assertEquals(1, privateMethodTester.invokePrivate("getLaps"));
    }

    @Test
    public void testPauseAndLapResume0() {
        final Timer timer = new Timer();
        final PrivateMethodTester privateMethodTester = new PrivateMethodTester(timer);
        GoToSleep(TENTH, 0);
        timer.pauseAndLap();
        timer.resume();
        assertTrue((Boolean) privateMethodTester.invokePrivate("isRunning"));
        assertEquals(1, privateMethodTester.invokePrivate("getLaps"));
    }

    @Test
    public void testPauseAndLapResume1() {
        final Timer timer = new Timer();
        GoToSleep(TENTH, 0);
        timer.pauseAndLap();
        GoToSleep(TENTH, 0);
        timer.resume();
        GoToSleep(TENTH, 0);
        final double time = timer.stop();
        assertEquals(TENTH_DOUBLE, time, 10.0);
        assertEquals(3, run);
    }

```



```
Finished after 0.074 seconds

Runs:  6/6      Errors:  0      Failures:  0

edu.neu.coe.info6205.sort.elementary.InsertionSortTest [Run
    testMutatingInsertionSort (0.051 s)
    sort0 (0.006 s)
    sort1 (0.003 s)
    sort2 (0.003 s)
    sort3 (0.001 s)
    testStaticInsertionSort (0.000 s)
```

```
<terminated> InsertionSortTest [JUnit] /Applications/Eclipse Java.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.aarch64_17.0.5.v20221102-0933/jre/bin/j
Helper for InsertionSort with 4 elements
StatPack {hits: 14,722, normalized=31.968; copies: 0, normalized=0.000; inversions: 2,421, normalized=5.257; swaps: 2,421, normalized=
StatPack {hits: 29,700, normalized=64.493; copies: 0, normalized=0.000; inversions: 4,950, normalized=10.749; swaps: 4,950, normalized=
```

```java
 * Copyr

package

import e

@Suppres
public c

    @Tes
    publ

    }

    @Tes
    publ

    }

    @Tes
    publ
        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<>("InsertionSort", xs.length, Config.load(InsertionSortTest.class));
        GenericSort<Integer> sorter = new InsertionSort<Integer>(helper);
```