

# Program Structures & Algorithms

Fall 2021

## Assignment No. 2

### ◎ Task

**Question:**

There are three parts of overall tasks:

1. Implement three methods of a class called Timer.
2. Implement InsertionSort by simply looking up the insertion code used by Arrays.sort.
3. Implement a main program to actually run the following benchmarks: measure the running times of this sort, using four different initial array ordering situations: random, ordered, partially-ordered and reverse-ordered. 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.

### ◎ Relationship Conclusion:

1. Running time: Reverse-Ordered > Random > Partially-Ordered > Ordered
2. For Ordered array, the relationship among N and Time is almost linear.

### 3. Increasing rate of Time with larger N: Reverse-Ordered > Random > Partially-Ordered > Ordered

#### ◎ Evidence to support the conclusion:

#### 1. Output

Here are times for four different types of arrays:

The screenshot shows the Eclipse IDE interface with the code for `InsertionSortResult.java`. The code implements a benchmark for insertion sort on arrays of integers. The `ddotit` method performs 10 runs of insertion sort on an array of size `N`. The `arr` method creates an array of `N` integers. The `main` method runs the benchmark. The `Failure Trace` view at the bottom shows the execution log:

```
2021-09-25 19:27:58 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:27:58 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:27:58 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:27:58 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:27:58 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:27:58 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:27:58 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:27:58 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:27:58 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:27:58 INFO Benchmark_Timer - Begin run: result with 10 runs
0.10635540000000002
0.3763622
0.71051
1.747063
7.152534
22.000000000000007
116.6566357
498.20325169999995
```

The screenshot shows the Eclipse IDE interface with the code for `InsertionSortResult.java`. The code is identical to the previous screenshot but uses a random array generator. The `Failure Trace` view at the bottom shows the execution log:

```
2021-09-25 19:28:01 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:28:01 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:28:01 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:28:01 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:28:01 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:28:01 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:28:01 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:28:01 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:28:01 INFO Benchmark_Timer - Begin run: result with 10 runs
2021-09-25 19:28:01 INFO Benchmark_Timer - Begin run: result with 10 runs
0.08869296
0.0176423
0.0197616
0.0193000000000005
0.0758622
0.1406195
0.2797576
```

eclipse-workspace - INFO6205/src/main/java/edu/neu/coe/info6205/sort/elementary/InsertionSortResult.java - Eclipse IDE

```

import java.util.ArrayList;
import java.util.function.Consumer;
import java.util.function.Supplier;

import edu.neu.coe.info6205.util.Benchmark_Timer;
public class InsertionSortResult {

    public static double doIt(int N, int type) {
        InsertionSort<Integer> insertionSort = new InsertionSort<Integer>();
        Supplier<Integer[]> supplier = () -> arr(N,type);
        Consumer<Integer[]> consumer = (f) -> {insertionSort.sort(f,0,f.length);};
        Benchmark_Timer<Integer[]> benchMark = new Benchmark_Timer<Integer[]>("result",consumer);
        //benchMark.run(arr(N,type), 10);
        double time = benchMark.runFromSupplier(supplier, 10);
        return time;
    }

    private static Integer[] arr(int n, int type) {
        Integer[] arr1 = new Integer[n];
        ...
    }
}

```

testWaitPeriods - edu.neu.coe.info6205.util.BenchmarkTest.java

getWarmupRuns - edu.neu.coe.info6205.util.BenchmarkTest.java

Failure Trace

Java Problems Declaration Search

InsertionSortResult.java (Java Application) /Library/Internet Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java (2021年9月25日下午7:27:49 - 下午7:28:33)

type: Partition-Ordered

2021-09-25 19:28:04 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:04 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:04 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:04 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:04 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:04 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:04 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:04 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:04 INFO Benchmark\_Timer - Begin run: result with 10 runs  
0.8357810999999999  
0.1382682  
0.224512  
1.7620565  
4.7706798  
7.93797  
26.31369  
114.9184528

eclipse-workspace - INFO6205/src/main/java/edu/neu/coe/info6205/sort/elementary/InsertionSortResult.java - Eclipse IDE

```

import java.util.ArrayList;
import java.util.function.Consumer;
import java.util.function.Supplier;

import edu.neu.coe.info6205.util.Benchmark_Timer;
public class InsertionSortResult {

    public static double doIt(int N, int type) {
        InsertionSort<Integer> insertionSort = new InsertionSort<Integer>();
        Supplier<Integer[]> supplier = () -> arr(N,type);
        Consumer<Integer[]> consumer = (f) -> {insertionSort.sort(f,0,f.length);};
        Benchmark_Timer<Integer[]> benchMark = new Benchmark_Timer<Integer[]>("result",consumer);
        //benchMark.run(arr(N,type), 10);
        double time = benchMark.runFromSupplier(supplier, 10);
        return time;
    }

    private static Integer[] arr(int n, int type) {
        Integer[] arr1 = new Integer[n];
        ...
    }
}

```

testWaitPeriods - edu.neu.coe.info6205.util.BenchmarkTest.java

getWarmupRuns - edu.neu.coe.info6205.util.BenchmarkTest.java

Failure Trace

Java Problems Declaration Search

InsertionSortResult.java (Java Application) /Library/Internet Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java (2021年9月25日下午7:27:49 - 下午7:28:33)

type: Reverse-Ordered

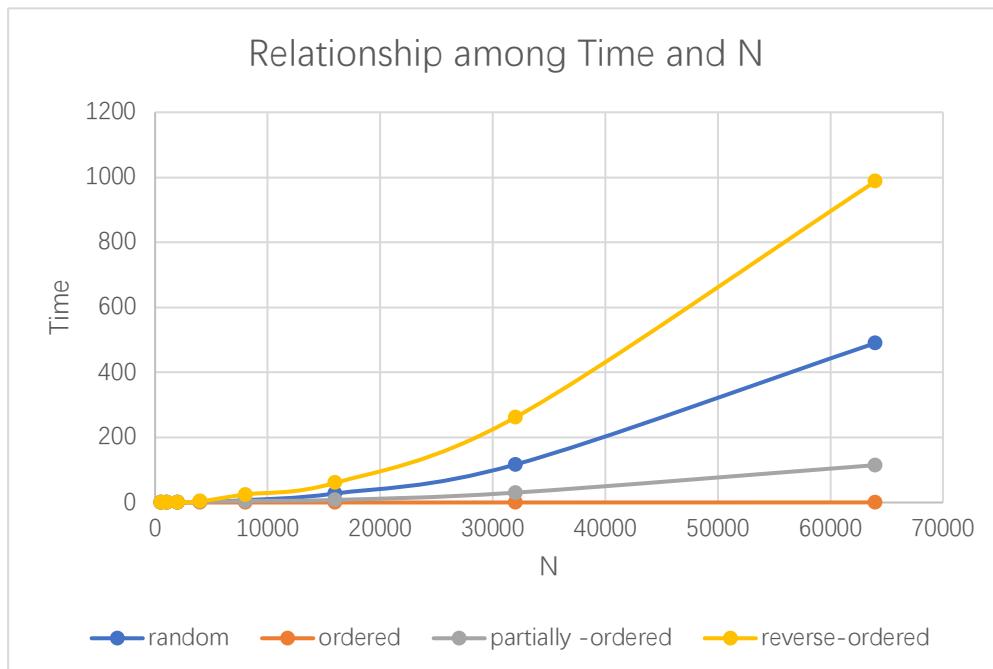
2021-09-25 19:28:07 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:07 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:07 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:07 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:07 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:07 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:07 INFO Benchmark\_Timer - Begin run: result with 10 runs  
2021-09-25 19:28:07 INFO Benchmark\_Timer - Begin run: result with 10 runs  
0.8574492  
0.224512  
0.224512  
3.5415711  
24.6398693  
60.932525  
261.6932525  
987.3596964000001

## 2. Graphical Representation

The graph observed:

N	random	ordered	partially - ordered	reverse-ordered
500	<b>0.106355</b>	<b>0.005419</b>	<b>0.035781</b>	<b>0.057449</b>
1000	<b>0.376302</b>	<b>0.00893</b>	<b>0.138268</b>	<b>0.224512</b>
2000	<b>0.857376</b>	<b>0.017642</b>	<b>0.48578</b>	<b>0.956718</b>

4000	<b>1.747063</b>	<b>0.019762</b>	<b>1.762057</b>	<b>3.541571</b>
8000	<b>7.152535</b>	<b>0.038781</b>	<b>4.77068</b>	<b>24.63061</b>
16000	<b>27.74747</b>	<b>0.075862</b>	<b>7.99329</b>	<b>60.42497</b>
32000	<b>116.6566</b>	<b>0.14062</b>	<b>30.31337</b>	<b>261.6932</b>
64000	<b>490.2033</b>	<b>0.279758</b>	<b>114.9185</b>	<b>987.3597</b>



## ◦ Unit tests result:

### 1. TimerTest unit tests are pass and none of code is modified:

eclipse-workspace - INFO6205/src/test/java/edu/neu/coe/info6205/util/TimerTest.java - Eclipse IDE

```

1 package edu.neu.coe.info6205.util;
2
3 import org.junit.Before;
4
5 public class TimerTest {
6
7     @Before
8     public void setup() {
9         run = 0;
10        post = 0;
11    }
12
13    @Test
14    public void testStop() {
15        final Timer timer = new Timer();
16        GoToSleep(TENTH, 0);
17        final double time = timer.stop();
18        assertEquals(TENTH_DOUBLE, time, 10);
19        timer.pauseAndLap();
20        GoToSleep(TENTH, 0);
21        final long ticks = (long) privateMethodTester.invokePrivate("getTicks");
22        assertEquals(TENTH_DOUBLE, ticks / 1e6, 12);
23        assertFalse((Boolean) privateMethodTester.invokePrivate("isRunning"));
24        assertEquals(0, privateMethodTester.invokePrivate("getLaps"));
25    }
26
27    @Test
28    public void testPauseAndLap() {
29        final Timer timer = new Timer();
30        final PrivateMethodTester privateMethodTester = new PrivateMethodTester(timer);
31        GoToSleep(TENTH, 0);
32        timer.pauseAndLap();
33        final long ticks = (long) privateMethodTester.invokePrivate("getTicks");
34        assertEquals(TENTH_DOUBLE, ticks / 1e6, 12);
35        assertFalse((Boolean) privateMethodTester.invokePrivate("isRunning"));
36        assertEquals(1, privateMethodTester.invokePrivate("getLaps"));
37    }
38
39    private void GoToSleep(double tenths, int millisecond) {
40        try {
41            Thread.sleep(tenths * 1000 + millisecond);
42        } catch (InterruptedException e) {
43            e.printStackTrace();
44        }
45    }
46}

```

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** eclipse-workspace - INFO6205/src/test/java/edu/neu/coe/info6205/util/TimerTest.java - Eclipse IDE
- Left Side:** Package Explorer (Shows 2.193 seconds, 10/10 runs, 0 errors, 0 failures), Test List (Shows 10 test cases: testPauseAndLapResume0, testPauseAndLapResume1, testLap, testPause, testStop, testRepeat1, testRepeat2, testRepeat3, testRepeat4, testPauseAndLap, all marked as passed).
- Middle Area:** Editor showing the `TimerTest.java` file with JUnit test cases for `testPauseAndLapResume0`, `testPauseAndLapResume1`, `testLap`, `testPause`, `testStop`, `testRepeat1`, `testRepeat2`, `testRepeat3`, `testRepeat4`, and `testPauseAndLap`.
- Bottom Bar:** Problems, JavaDoc, Declaration, Search, Console, and a message <terminated> TimerTest [JUnit] / Library / Internet Plug-ins / JavaAppletPlugin.plugin / Contents / Home / bin / java (2021年9月22日下午8:46:51 - 下午8:46:54)

```
62
63    @Test
64    public void testLap() {
65        final Timer timer = new Timer();
66        GoToSleep(TENTH, 0);
67        timer.lap();
68        GoToSleep(TENTH, 0);
69        final double time = timer.stop();
70        assertEquals(TENTH_DOUBLE, time, 10.0);
71        assertEquals(2, run);
72    }
73
74    @Test
75    public void testPause() {
76        final Timer timer = new Timer();
77        GoToSleep(TENTH, 0);
78        timer.pause();
79        GoToSleep(TENTH, 0);
80        timer.resume();
81        final double time = timer.stop();
82        assertEquals(TENTH_DOUBLE, time, 10.0);
83        assertEquals(2, run);
84    }
85
86    @Test
87    public void testMilliseconds() {
88        final Timer timer = new Timer();
89        GoToSleep(TENTH, 0);
90        timer.stop();
91        final double time = timer.milliseconds();
92        assertEquals(TENTH_DOUBLE, time, 10.0);
93        assertEquals(1, run);
94    }

```

```
95
96    @Test
97    public void testPrivate1() {
98        final Timer timer = new Timer();
99        final double mean = timer.repeat(10, () -> {
100            GoToSleep(HUNDREDTH, 0);
101            return null;
102        });
103        assertEquals(10, new PrivateMethodTester(timer).invokePrivate("getLaps"));
104        assertEquals(TENTH_DOUBLE / 10, mean, 6);
105        assertEquals(10, run());
106        assertEquals(0, pre);
107        assertEquals(0, post);
108    }
109
110    @Test
111    public void testRepeat2() {
112        final Timer timer = new Timer();
113        final int zzz = 20;
114        final double mean = timer.repeat(10, () -> zzz, t -> {
115            GoToSleep(t, 0);
116            return null;
117        });
118        assertEquals(10, new PrivateMethodTester(timer).invokePrivate("getLaps"));
119        assertEquals(zzz, mean, 8.5);
120        assertEquals(10, run());
121        assertEquals(0, pre);
122        assertEquals(0, post);
123    }
124
125    @Test // Slow
126    public void testRepeat3() {
127        final Timer timer = new Timer();
128        final int zzz = 20;
```

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - INFO6205/src/test/java/edu/neu/coe/info6205/util/TimerTest.java - Eclipse IDE
- Left Panel (Package Explorer):** Shows the project structure.
- Middle Panel (Editor):** Displays the `TimerTest.java` code. The code includes various test methods like `testPauseAndLapResume0`, `testPauseAndLapResume1`, etc., and a private helper method `GoToSleep`.
- Bottom Left (Status Bar):** Shows "Runs: 10/10 Errors: 0 Failures: 0".
- Bottom Middle (Status Bar):** Shows "Finished after 2.193 seconds".
- Bottom Right (Status Bar):** Shows the date and time: "2021年9月22日下午8:46:51 - 星期五 8:46:54".
- Right Side (Toolbars and Outline View):** Shows the JUnit outline for the `TimerTest` class.

## 2. BenchMarkTest unit tests are pass and none of code is modified:

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - INFO6205/src/test/java/edu/neu/coe/info6205/util/BenchmarkTest.java - Eclipse IDE
- Left Panel (Package Explorer):** Shows the project structure.
- Middle Panel (Editor):** Displays the `BenchmarkTest.java` code. It contains a single test method `testWaitPeriods` which creates a `Benchmark_Timer` object and performs assertions on its state.
- Bottom Left (Status Bar):** Shows "Runs: 2/2 Errors: 0 Failures: 0".
- Bottom Middle (Status Bar):** Shows "Finished after 1.502 seconds".
- Bottom Right (Status Bar):** Shows the date and time: "2021年9月23日下午12:40:03 - 星期六 12:40:06".
- Right Side (Toolbars and Outline View):** Shows the JUnit outline for the `BenchmarkTest` class.

## 3. InsertionSortTest unit tests are pass and none of code is modified:

eclipse-workspace - INFO6205/src/test/java/edu/neu/coe/info6205/sort/elementary/InsertionSortTest.java - Eclipse IDE

```
61    @Test
62    public void testMutatingInsertionSort() throws IOException {
63        final List<Integer> list = new ArrayList<>();
64        list.add(3);
65        list.add(4);
66        list.add(2);
67        list.add(1);
68        Integer[] xs = list.toArray(new Integer[0]);
69        BaseHelper<Integer> helper = new BaseHelper<>("InsertionSort", xs.length, Config.load(InsertionSortTest.class));
70        GenericSort<Integer> sorter = new InsertionSort<Integer>(helper);
71        sorter.mutatingSort(xs);
72        assertTrue(helper.sorted(xs));
73        assertEquals(true, helper.sorted(xs));
74    }
75
76    @Test
77    public void testStaticInsertionSort() throws IOException {
78        final List<Integer> list = new ArrayList<>();
79        list.add(3);
80        list.add(4);
81        list.add(2);
82        list.add(1);
83        Integer[] xs = list.toArray(new Integer[0]);
84        InsertionSort.sort(xs);
85        assertEquals(true, xs[0] < xs[1] && xs[1] < xs[2] && xs[2] < xs[3]);
86    }
87}
```

Failure Trace

```
terminated: InsertionSortTest (1) [JUnit] LibraryInternet Plug-InJavaAppletPlugin.plugin/Contents/Home/bin/java (2021年9月23日下午1:08:38 - 下午1:08:39)
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, instrument = true
2021-09-23 13:08:39 DEBUG Config - Config.getHelper, seed = 0
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, copies = true
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, swaps = true
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, compares = true
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, inversions = true
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, fixes = true
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, hits = true
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, cutoff = true
Helper for InsertionSort with 4 elements
StatPack (hits: 9,684; copies: 0; inversions: 2,421; swaps: 2,421; compares: 2,519)
StatPack (hits: 19,800; copies: 0; inversions: 4,950; swaps: 4,950; fixes: 4,950; compares: 4,950)
```

Writable Smart Insert 122:51:5104

eclipse-workspace - INFO6205/src/test/java/edu/neu/coe/info6205/sort/elementary/InsertionSortTest.java - Eclipse IDE

```
84    }
85    assertTrue(xs[0] < xs[1] && xs[1] < xs[2] && xs[2] < xs[3]);
86}
87
88    @Test
89    public void sort1() throws Exception {
90        final Config config = ConfigTest.setupConfig("true", "0", "1", "", "");
91        int n = 100;
92        Helper<Integer> helper = HelperFactory.create("InsertionSort", n, config);
93        sorter = new InsertionSort(helper);
94        final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);
95        final StatPack statPack = (StatPack) privateMethodTester.invokePrivate("getStatPack");
96        Integer[] xs = helper.randomUUID().class, r = r.nextInt(1000);
97        Sorter<Integer> sorter = new InsertionSort<Integer>(helper);
98        sorter.preProcess(xs);
99        Integer[] ys = sorter.sort(xs);
100       sorter.postProcess(ys);
101       final int compares = (int) statPack.getStatistics(InstrumentedHelper.COMPARES).mean();
102       // NOTE: these are supposed to match within about 12%.
103       // If we use a random seed and this test fails, just increase the delta a little.
104       // If we use true random seed and this test fails, just increase the delta a little.
105       assertEquals(1.0, 4.0 * compares / n / (n - 1), 0.12);
106       final int inversions = (int) statPack.getStatistics(InstrumentedHelper.INVERSIONS).mean();
107       final int fixes = (int) statPack.getStatistics(InstrumentedHelper.FIXES).mean();
108       final int swaps = (int) statPack.getStatistics(InstrumentedHelper.SWAPS).mean();
109       System.out.println(statPack);
110       assertEquals(inversions, fixes);
111   }
112
113    @Test
114    public void sort3() throws Exception {
115        final Config config = ConfigTest.setupConfig("true", "0", "1", "", "");
116    }
117}
```

Failure Trace

```
terminated: InsertionSortTest (1) [JUnit] LibraryInternet Plug-InJavaAppletPlugin.plugin/Contents/Home/bin/java (2021年9月23日下午1:08:38 - 下午1:08:39)
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, instrument = true
2021-09-23 13:08:39 DEBUG Config - Config.getHelper, seed = 0
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, copies = true
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, swaps = true
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, compares = true
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, inversions = 1
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, fixes = true
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, hits = true
2021-09-23 13:08:39 DEBUG Config - Config.getInstrumenting, cutoff = true
Helper for InsertionSort with 4 elements
StatPack (hits: 9,684; copies: 0; inversions: 2,421; swaps: 2,421; compares: 2,519)
```

Writable Smart Insert 122:51:5104

eclipse-workspace - INFO6205/src/test/java/edu/neu/coe/info6205/sort/elementary/InsertionSortTest.java - Eclipse IDE

```
21    @Test
22    public void sort0() throws Exception {
23        final List<Integer> list = new ArrayList<>();
24        list.add(1);
25        list.add(2);
26        list.add(3);
27        list.add(4);
28        Integer[] xs = list.toArray(new Integer[0]);
29        final Config config = ConfigTest.setupConfig("true", "0", "1", "", "");
30        int n = 100;
31        Helper<Integer> helper = HelperFactory.create("InsertionSort", list.size(), config);
32        sorter.initList(xs);
33        final StatPack statPack = (StatPack) privateMethodTester.invokePrivate("getStatPack");
34        Sorter<Integer> sorter = new InsertionSort<Integer>(helper);
35        sorter.preProcess(xs);
36        Integer[] ys = sorter.sort(xs);
37        sorter.postProcess(ys);
38        final int compares = (int) statPack.getStatistics(InstrumentedHelper.COMPARES).mean();
39        final int inversions = (int) statPack.getStatistics(InstrumentedHelper.INVERSIONS).mean();
40        final int fixes = (int) statPack.getStatistics(InstrumentedHelper.FIXES).mean();
41        final int swaps = (int) statPack.getStatistics(InstrumentedHelper.SWAPS).mean();
42        assertEquals(0L, inversions);
43        assertEquals(0L, fixes);
44        assertEquals(0L, swaps);
45    }
46
47    @Test
48    public void sort1() throws Exception {
49        final List<Integer> list = new ArrayList<>();
50        list.add(1);
51        list.add(4);
52        list.add(2);
53        list.add(3);
54        Integer[] xs = list.toArray(new Integer[0]);
55        BaseHelper<Integer> helper = new BaseHelper<>("InsertionSort", xs.length, Config.load(InsertionSortTest.class));
56        GenericSort<Integer> sorter = new InsertionSort<Integer>(helper);
57        sorter.mutatingSort(xs);
58        assertEquals(true, helper.sorted(ys));
59        System.out.println(sorter.toString());
60    }
61}
```

Failure Trace

```
terminated: InsertionSortTest (1) [JUnit] LibraryInternet Plug-InJavaAppletPlugin.plugin/Contents/Home/bin/java (2021年9月23日下午1:08:38 - 下午1:08:39)
```

Writable Smart Insert 122:51:5104

eclipse-workspace - INFO6205/src/test/java/edu/neu/coe/info6205/sort/elementary/InsertionSortTest.java - Eclipse IDE

```
111 }  
112 }  
113 @Test  
114 public void sort3() throws Exception {  
115     final Config config = ConfigTest.setupConfig("true", "0", "1", "", "");  
116     Helper<Integer> helper = HelperFactory.create("InsertionSort", n, config);  
117     helper.init(n);  
118     final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);  
119     final StatPack statPack = (StatPack) privateMethodTester.invokePrivate("getStatPack");  
120     Integer[] xs = new Integer[n];  
121     for (int i = 0; i < n; i++) xs[i] = n - i;  
122     Sorter<Integer, Integer> sorter = new InsertionSort<Integer>(helper);  
123     sorter.preProcess(xs);  
124     Integer[] ys = sorter.sort(xs);  
125     assertEquals(true, helper.sorted(ys));  
126     sorter.postProcess(xs);  
127     final int compares = (int) statPack.getStatistics(InstrumentedHelper.COMPARES).mean();  
128     // NOTE: these are supposed to match within about 12%  
129     // If we use true random seed and this fails, just increase the delta a little.  
130     assertEquals(4590, compares);  
131     final int inversions = (int) statPack.getStatistics(InstrumentedHelper.INVERSIONS).mean();  
132     final int fixes = (int) statPack.getStatistics(InstrumentedHelper.FIXES).mean();  
133     System.out.println(statPack);  
134     assertEquals(0, inversions);  
135     assertEquals(0, fixes);  
136 }  
137  
138 final static LazyLogger logger = new LazyLogger(InsertionSort.class);  
139  
140 }  
141 }
```

Failure Trace

```
terminated: InsertionSortTest [1] [exit] Library[Memory Plugin-Java API] (Contents/Home/bin/java) (2021/9月23日下午1:08:38 - 下午1:08:39)  
2021-09-23 13:08:39 DEBUG Config - Config.get(helper, instrument) = true  
2021-09-23 13:08:39 DEBUG Config - Config.get(helper, seed) = 0  
2021-09-23 13:08:39 DEBUG Config - Config.get(instrumenting, swaps) = true  
2021-09-23 13:08:39 DEBUG Config - Config.get(instrumenting, compares) = true  
2021-09-23 13:08:39 DEBUG Config - Config.get(instrumenting, inversions) = 1  
2021-09-23 13:08:39 DEBUG Config - Config.get(helper, cutoff) = 0  
2021-09-23 13:08:39 DEBUG Config - Config.get(helper, instrument) = true  
2021-09-23 13:08:39 DEBUG Config - Config.get(helper, seed) = 0  
2021-09-23 13:08:39 DEBUG Config - Config.get(instrumenting, swaps) = true  
2021-09-23 13:08:39 DEBUG Config - Config.get(instrumenting, compares) = true  
2021-09-23 13:08:39 DEBUG Config - Config.get(helper, cutoff) = 0  
Helper for InsertionSort with 4 elements  
StatPack (hits: 9,084; copies: 0; inversions: 2,421; swaps: 2,421; compares: 2,519);  
Writable SmartInsert 122.51:5104
```