Poojitha Konduparti (002103368)

PROGRAM STRUCTURES AND ALGORITHMS
FALL 2021
ASSIGNMENT NO. 2

Task

Implementation of Timer class and Insertion Sort and to time the insertion sort using the methods in Timer Class using the doubling method.

Task List:

- Implemented the Timer class and Insertion sort using the helper.swap() methods and passed the unit tests.
- A main program to time the insertion sort using the timer class and benchmark_timer methods.
- Plotted the values of the output onto a graph to deduce a conclusion.

Relationship Conclusion

Analysing the evidence provided below, it can be concluded that,

Best Case: Ordered Array O(n)

Average Case: Random Array, Partially Ordered Array $O(n^2)$

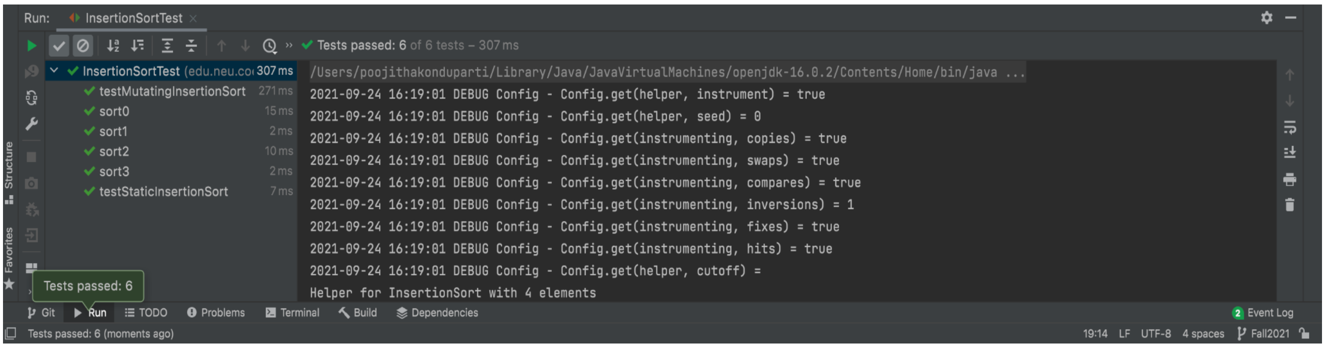
Worst Case: Reverse Ordered Array $O(n^2)$

Therefore, the run time for different array ordering cases is:

Reverse Ordered > Random Ordered > Partially Ordered > Ordered

---

Evidence



Run: InsertionSortTest

Tests passed: 6 of 6 tests – 307 ms

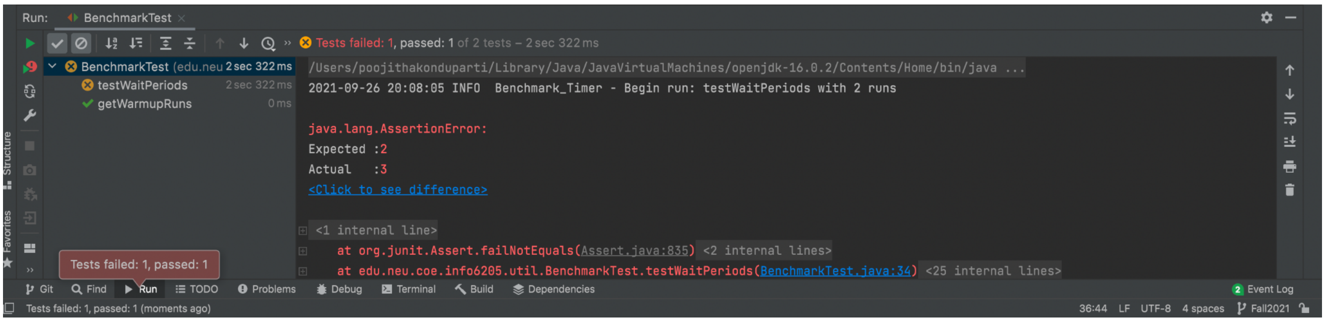
InsertionSortTest (edu.neu.co	307 ms
testMutatingInsertionSort	271 ms
sort0	15 ms
sort1	2 ms
sort2	10 ms
sort3	2 ms
testStaticInsertionSort	7 ms

```
/Users/poojithakonduparti/Library/Java/JavaVirtualMachines/openjdk-16.0.2/Contents/Home/bin/java ...
2021-09-24 16:19:01 DEBUG Config - Config.get(helper, instrument) = true
2021-09-24 16:19:01 DEBUG Config - Config.get(helper, seed) = 0
2021-09-24 16:19:01 DEBUG Config - Config.get(instrumenting, copies) = true
2021-09-24 16:19:01 DEBUG Config - Config.get(instrumenting, swaps) = true
2021-09-24 16:19:01 DEBUG Config - Config.get(instrumenting, compares) = true
2021-09-24 16:19:01 DEBUG Config - Config.get(instrumenting, inversions) = 1
2021-09-24 16:19:01 DEBUG Config - Config.get(instrumenting, fixes) = true
2021-09-24 16:19:01 DEBUG Config - Config.get(instrumenting, hits) = true
2021-09-24 16:19:01 DEBUG Config - Config.get(helper, cutoff) = 
Helper for InsertionSort with 4 elements
```

Tests passed: 6

Tests passed: 6 (moments ago)

Evidence of Insertion Sort Unit Tests running.



```
Tests failed: 1, passed: 1 of 2 tests – 2 sec 322 ms

BenchmarkTest (edu.neu 2 sec 322 ms
    testWaitPeriods     2 sec 322 ms
    getWarmupRuns               0 ms

Tests failed: 1, passed: 1
```

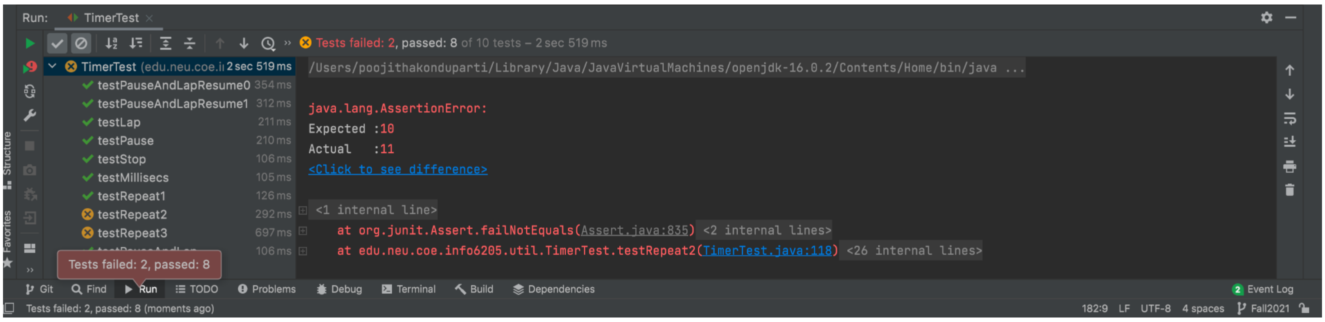
```
/Users/poojithakonduparti/Library/Java/JavaVirtualMachines/openjdk-16.0.2/Contents/Home/bin/java ...
2021-09-26 20:08:05 INFO  Benchmark_Timer - Begin run: testWaitPeriods with 2 runs

java.lang.AssertionError: 
Expected :2
Actual   :3
<Click to see difference>

<1 internal line>
    at org.junit.Assert.failNotEquals(Assert.java:835) <2 internal lines>
    at edu.neu.coe.info6205.util.BenchmarkTest.testWaitPeriods(BenchmarkTest.java:34) <25 internal lines>
```

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

Benchmark Unit Tests



```
Tests failed: 2, passed: 8 of 10 tests – 2 sec 519 ms

TimerTest (edu.neu.coe.i 2 sec 519 ms
    testPauseAndLapResume0  354 ms
    testPauseAndLapResume1  312 ms
    testLap                 211 ms
    testPause               210 ms
    testStop                106 ms
    testMillisecs           105 ms
    testRepeat1             126 ms
    testRepeat2             292 ms
    testRepeat3             697 ms
                            106 ms

Tests failed: 2, passed: 8
```

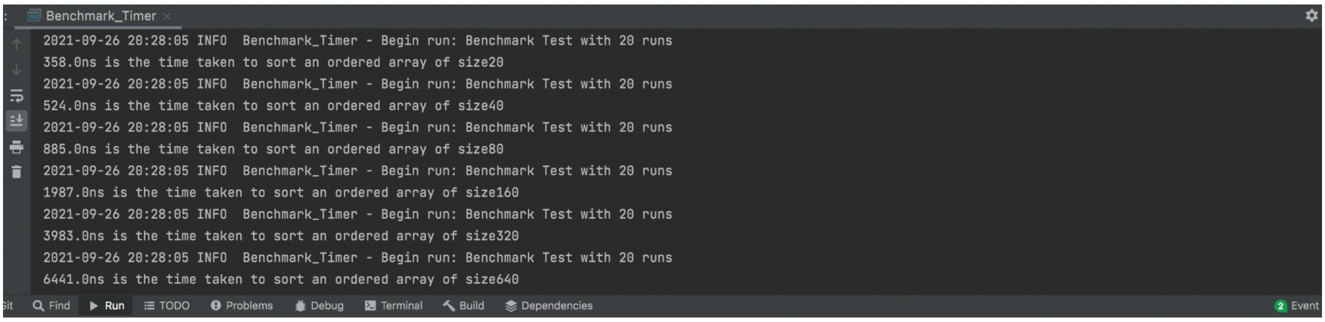
```
/Users/poojithakonduparti/Library/Java/JavaVirtualMachines/openjdk-16.0.2/Contents/Home/bin/java ...

java.lang.AssertionError: 
Expected :10
Actual   :11
<Click to see difference>

<1 internal line>
    at org.junit.Assert.failNotEquals(Assert.java:835) <2 internal lines>
    at edu.neu.coe.info6205.util.TimerTest.testRepeat2(TimerTest.java:118) <26 internal lines>
```

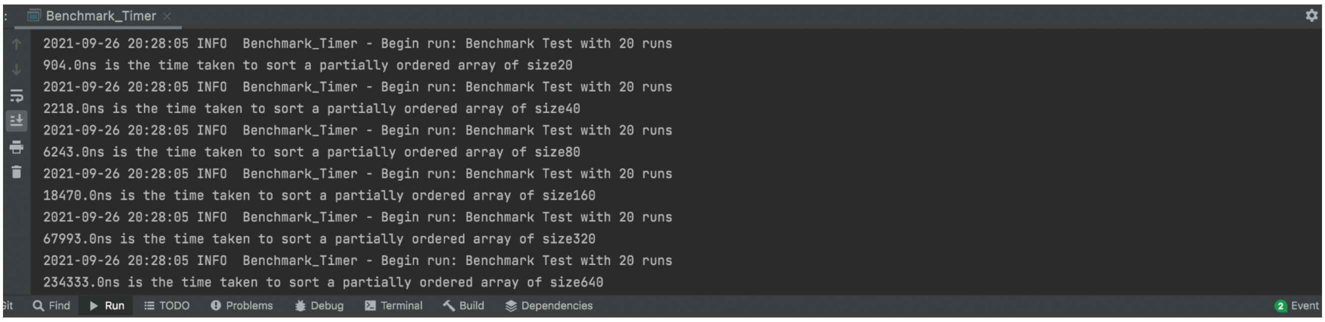
```
Tests failed: 2, passed: 8 (moments ago)
```

Timer Class Unit Tests



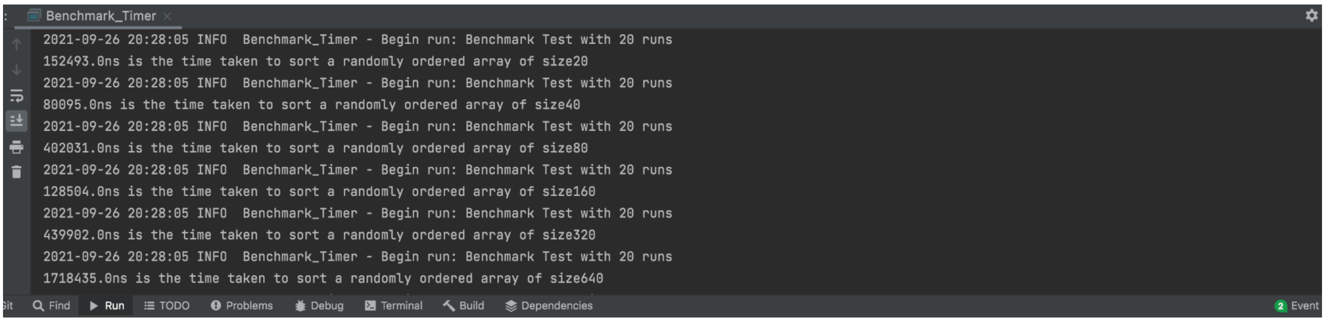
```
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
358.0ns is the time taken to sort an ordered array of size20
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
524.0ns is the time taken to sort an ordered array of size40
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
885.0ns is the time taken to sort an ordered array of size80
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
1987.0ns is the time taken to sort an ordered array of size160
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
3983.0ns is the time taken to sort an ordered array of size320
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
6441.0ns is the time taken to sort an ordered array of size640
```

Benchmark Timer for Ordered Array using the Doubling Method



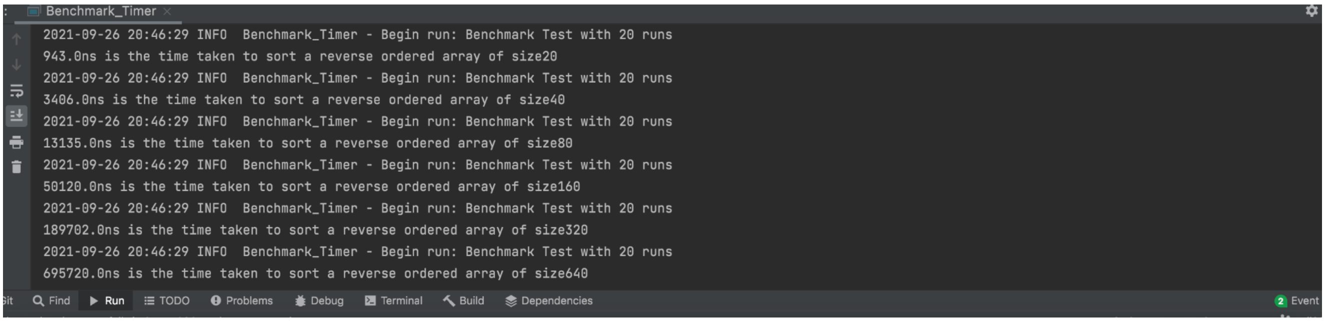
```
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
904.0ns is the time taken to sort a partially ordered array of size20
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
2218.0ns is the time taken to sort a partially ordered array of size40
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
6243.0ns is the time taken to sort a partially ordered array of size80
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
18470.0ns is the time taken to sort a partially ordered array of size160
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
67993.0ns is the time taken to sort a partially ordered array of size320
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
234333.0ns is the time taken to sort a partially ordered array of size640
```

Benchmark Timer for Partially Ordered Array using the Doubling Method



```
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
152493.0ns is the time taken to sort a randomly ordered array of size20
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
80095.0ns is the time taken to sort a randomly ordered array of size40
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
402031.0ns is the time taken to sort a randomly ordered array of size80
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
128504.0ns is the time taken to sort a randomly ordered array of size160
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
439902.0ns is the time taken to sort a randomly ordered array of size320
2021-09-26 20:28:05 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
1718435.0ns is the time taken to sort a randomly ordered array of size640
```

Benchmark Timer for Random Ordered Array using the Doubling Method



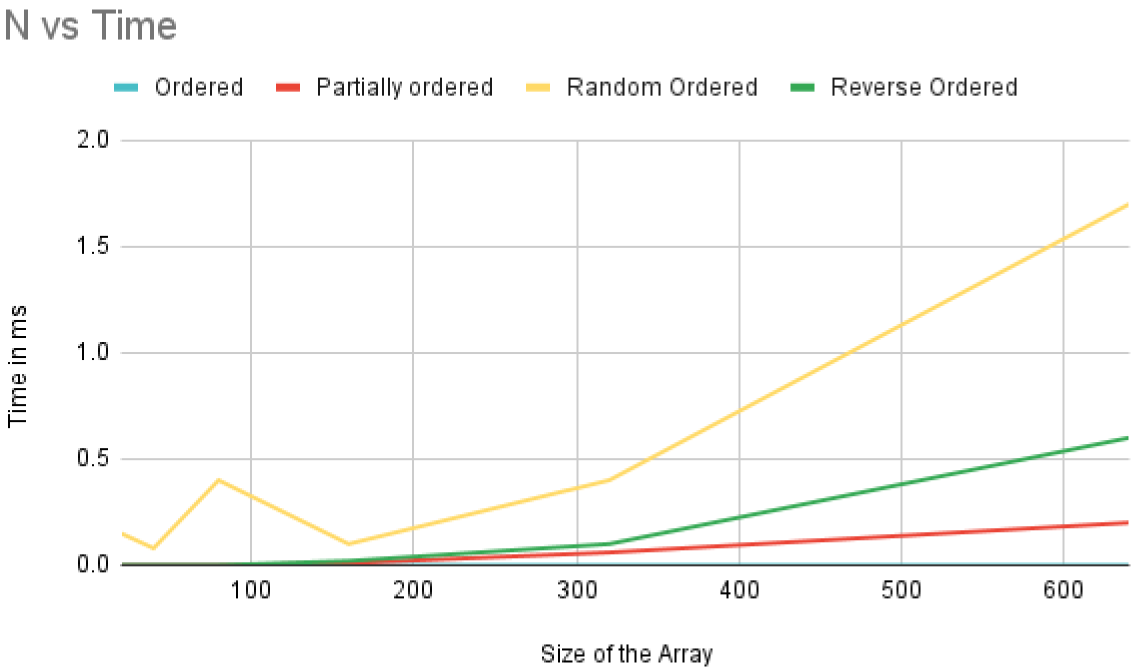
```
2021-09-26 20:46:29 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
943.0ns is the time taken to sort a reverse ordered array of size20
2021-09-26 20:46:29 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
3406.0ns is the time taken to sort a reverse ordered array of size40
2021-09-26 20:46:29 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
13135.0ns is the time taken to sort a reverse ordered array of size80
2021-09-26 20:46:29 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
50120.0ns is the time taken to sort a reverse ordered array of size160
2021-09-26 20:46:29 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
189702.0ns is the time taken to sort a reverse ordered array of size320
2021-09-26 20:46:29 INFO  Benchmark_Timer - Begin run: Benchmark Test with 20 runs
695720.0ns is the time taken to sort a reverse ordered array of size640
```

Benchmark Timer for Reverse Ordered Array using the Doubling Method

Tabular representation(Lesson 2.4)

N *(Size of the Array)*	Ordered	Partially Ordered	Random Ordered	Reverse Ordered
20	**0.0**	**0.0**	**0.15**	**0.0**
40	**0.0**	**0.0**	**0.08**	**0.0**
80	**0.0**	**0.0**	**0.4**	**0.0**
160	**0.0**	**0.01**	**0.1**	**0.02**
320	**0.0**	**0.06**	**0.4**	**0.1**
640	**0.0**	**0.2**	**1.7**	**0.6**

Graph


N vs Time
Ordered
Partially ordered
Random Ordered
Reverse Ordered
Time in ms
2.0
1.5
1.0
0.5
0.0
100
200
300
400
500
600
Size of the Array