

PSA ASSIGNMENT 2

- SUBMITTED BY DAISY QUADROS
- NU ID – 001568823

Part 1 of Assignment

Test cases for TimerTest

The screenshot displays an IDE window for the project 'INFO6205'. The left sidebar shows a project structure with a 'sort' package containing various tests, including 'TimerTest'. The main editor shows the source code of 'TimerTest.java', which includes a 'warmup' phase and a 'timed' phase. The 'warmup' phase uses 'new Timer().repeat' to run a function multiple times. The 'timed' phase uses 'return new Timer().repeat' to run a function multiple times. The 'TimerTest' class has a constructor that takes a description, a function, and a pre/post function. The test results at the bottom show that all 10 tests passed, with a total execution time of 2 seconds and 388 milliseconds. The tests include 'testPauseAndLapResume0', 'testPauseAndLapResume1', 'testLap', 'testPause', 'testStop', 'testMilliSecs', 'testRepeat1', 'testRepeat2', 'testRepeat3', and 'testPauseAndLap'. The IDE also shows a terminal window with the command 'C:\Users\daisy\.jdk\corretto-1.8.0_302\bin\java.exe ...' and the message 'Process finished with exit code 0'.

```
53 logger.info("Begin run: " + description + " with " + formatWhole(n) + " runs");
54 // Warmup phase
55 final Function<T, T> function = t -> {
56     fRun.accept(t);
57     return t;
58 };
59 new Timer().repeat(getWarmupRuns(n), supplier, function, fPre, postfunction: null);
60
61 // Timed phase
62 return new Timer().repeat(n, supplier, function, fPre, fPost);
63 }
64
65 /**
66  * Constructor for a Benchmark_Timer with option of specifying all three functions.
67  *
68  * @param description the description of the benchmark.
69  * @param fPre a function of T => T.
70  * @param fPost Function fPre is run before each invocation of fRun (but with the clock stopped).
71  * The result of fPre (if any) is passed to fRun.
```

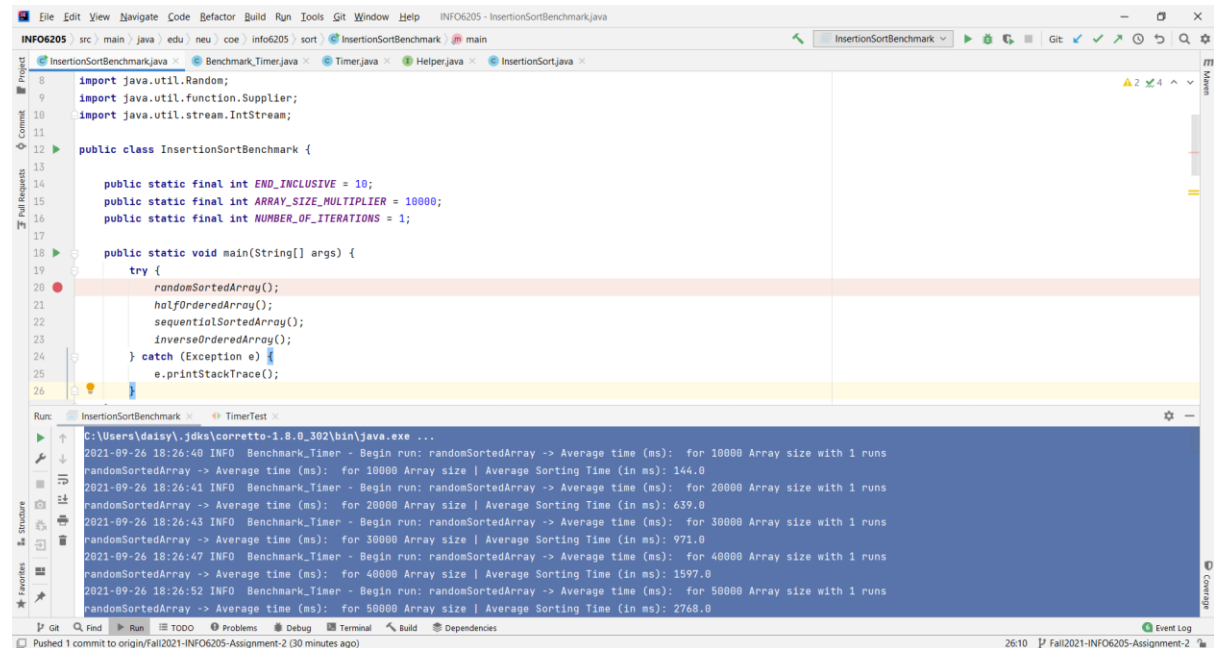
Run: InsertionSortTest x TimerTest x
Tests passed: 10 of 10 tests - 2 sec 388 ms
C:\Users\daisy\.jdk\corretto-1.8.0_302\bin\java.exe ...
Process finished with exit code 0

Test	Time
testPauseAndLapResume0	174 ms
testPauseAndLapResume1	325 ms
testLap	218 ms
testPause	220 ms
testStop	109 ms
testMilliSecs	109 ms
testRepeat1	186 ms
testRepeat2	313 ms
testRepeat3	624 ms
testPauseAndLap	430 ms

Tests passed: 10 (moments ago)

Part 2 Of Assignment

Output of Main Insertion Sort Program



```
import java.util.Random;
import java.util.function.Supplier;
import java.util.stream.IntStream;

public class InsertionSortBenchmark {

    public static final int END_INCLUSIVE = 10;
    public static final int ARRAY_SIZE_MULTIPLIER = 10000;
    public static final int NUMBER_OF_ITERATIONS = 1;

    public static void main(String[] args) {
        try {
            randomSortedArray();
            halfOrderedArray();
            sequentialSortedArray();
            inverseOrderedArray();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Run: InsertionSortBenchmark x TimerTest x

```
C:\Users\daisy\jdk\corretto-1.8.0_302\bin\java.exe ...
2021-09-26 18:26:40 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 10000 Array size with 1 runs
randomSortedArray -> Average time (ms): for 10000 Array size | Average Sorting Time (in ms): 144.0
2021-09-26 18:26:41 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 20000 Array size with 1 runs
randomSortedArray -> Average time (ms): for 20000 Array size | Average Sorting Time (in ms): 639.0
2021-09-26 18:26:43 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 30000 Array size with 1 runs
randomSortedArray -> Average time (ms): for 30000 Array size | Average Sorting Time (in ms): 971.0
2021-09-26 18:26:47 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 40000 Array size with 1 runs
randomSortedArray -> Average time (ms): for 40000 Array size | Average Sorting Time (in ms): 1597.0
2021-09-26 18:26:52 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 50000 Array size with 1 runs
randomSortedArray -> Average time (ms): for 50000 Array size | Average Sorting Time (in ms): 2768.0
```

2021-09-26 18:26:40 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 10000 Array size with 1 runs

randomSortedArray -> Average time (ms): for 10000 Array size | Average Sorting Time (in ms): 144.0

2021-09-26 18:26:41 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 20000 Array size with 1 runs

randomSortedArray -> Average time (ms): for 20000 Array size | Average Sorting Time (in ms): 639.0

2021-09-26 18:26:43 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 30000 Array size with 1 runs

randomSortedArray -> Average time (ms): for 30000 Array size | Average Sorting Time (in ms): 971.0

2021-09-26 18:26:47 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 40000 Array size with 1 runs

randomSortedArray -> Average time (ms): for 40000 Array size | Average Sorting Time (in ms):
1597.0

2021-09-26 18:26:52 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 50000 Array size with 1 runs

randomSortedArray -> Average time (ms): for 50000 Array size | Average Sorting Time (in ms):
2768.0

2021-09-26 18:27:00 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 60000 Array size with 1 runs

randomSortedArray -> Average time (ms): for 60000 Array size | Average Sorting Time (in ms):
4606.0

2021-09-26 18:27:13 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 70000 Array size with 1 runs

randomSortedArray -> Average time (ms): for 70000 Array size | Average Sorting Time (in ms):
6452.0

2021-09-26 18:27:31 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 80000 Array size with 1 runs

randomSortedArray -> Average time (ms): for 80000 Array size | Average Sorting Time (in ms):
8343.0

2021-09-26 18:27:59 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 90000 Array size with 1 runs

randomSortedArray -> Average time (ms): for 90000 Array size | Average Sorting Time (in ms):
10962.0

2021-09-26 18:28:33 INFO Benchmark_Timer - Begin run: randomSortedArray -> Average time (ms):
for 100000 Array size with 1 runs

randomSortedArray -> Average time (ms): for 100000 Array size | Average Sorting Time (in ms):
15043.0

2021-09-26 18:29:13 INFO Benchmark_Timer - Begin run: partiallySortedArray -> Average time (ms):
for 10000 Array size with 1 runs

partiallySortedArray -> Average time (ms): for 10000 Array size | Average Sorting Time (in ms): 88.0

2021-09-26 18:29:13 INFO Benchmark_Timer - Begin run: partiallySortedArray -> Average time (ms):
for 20000 Array size with 1 runs

partiallySortedArray -> Average time (ms): for 20000 Array size | Average Sorting Time (in ms): 267.0

2021-09-26 18:29:14 INFO Benchmark_Timer - Begin run: partiallySortedArray -> Average time (ms):
for 30000 Array size with 1 runs

partiallySortedArray -> Average time (ms): for 30000 Array size | Average Sorting Time (in ms): 684.0

2021-09-26 18:29:16 INFO Benchmark_Timer - Begin run: partiallySortedArray -> Average time (ms):
for 40000 Array size with 1 runs

partiallySortedArray -> Average time (ms): for 40000 Array size | Average Sorting Time (in ms):
1557.0

2021-09-26 18:29:20 INFO Benchmark_Timer - Begin run: partiallySortedArray -> Average time (ms):
for 50000 Array size with 1 runs

partiallySortedArray -> Average time (ms): for 50000 Array size | Average Sorting Time (in ms):
2026.0

2021-09-26 18:29:28 INFO Benchmark_Timer - Begin run: partiallySortedArray -> Average time (ms):
for 60000 Array size with 1 runs

partiallySortedArray -> Average time (ms): for 60000 Array size | Average Sorting Time (in ms):
2749.0

2021-09-26 18:29:36 INFO Benchmark_Timer - Begin run: partiallySortedArray -> Average time (ms):
for 70000 Array size with 1 runs

partiallySortedArray -> Average time (ms): for 70000 Array size | Average Sorting Time (in ms):
5223.0

2021-09-26 18:29:52 INFO Benchmark_Timer - Begin run: partiallySortedArray -> Average time (ms):
for 80000 Array size with 1 runs

partiallySortedArray -> Average time (ms): for 80000 Array size | Average Sorting Time (in ms):
5607.0

2021-09-26 18:30:09 INFO Benchmark_Timer - Begin run: partiallySortedArray -> Average time (ms):
for 90000 Array size with 1 runs

partiallySortedArray -> Average time (ms): for 90000 Array size | Average Sorting Time (in ms):
8713.0

2021-09-26 18:30:34 INFO Benchmark_Timer - Begin run: partiallySortedArray -> Average time (ms):
for 100000 Array size with 1 runs

partiallySortedArray -> Average time (ms): for 100000 Array size | Average Sorting Time (in ms):
8965.0

2021-09-26 18:31:05 INFO Benchmark_Timer - Begin run: sequentialSortedArray -> Average time
(ms): for 10000 Array size with 1 runs

sequentialSortedArray -> Average time (ms): for 10000 Array size | Average Sorting Time (in ms): 1.0

2021-09-26 18:31:05 INFO Benchmark_Timer - Begin run: sequentialSortedArray -> Average time
(ms): for 20000 Array size with 1 runs

sequentialSortedArray -> Average time (ms): for 20000 Array size | Average Sorting Time (in ms): 1.0

2021-09-26 18:31:05 INFO Benchmark_Timer - Begin run: sequentialSortedArray -> Average time
(ms): for 30000 Array size with 1 runs

sequentialSortedArray -> Average time (ms): for 30000 Array size | Average Sorting Time (in ms): 1.0

2021-09-26 18:31:05 INFO Benchmark_Timer - Begin run: sequentialSortedArray -> Average time
(ms): for 40000 Array size with 1 runs

sequentialSortedArray -> Average time (ms): for 40000 Array size | Average Sorting Time (in ms): 1.0

2021-09-26 18:31:05 INFO Benchmark_Timer - Begin run: sequentialSortedArray -> Average time
(ms): for 50000 Array size with 1 runs

sequentialSortedArray -> Average time (ms): for 50000 Array size | Average Sorting Time (in ms): 1.0

2021-09-26 18:31:05 INFO Benchmark_Timer - Begin run: sequentialSortedArray -> Average time
(ms): for 60000 Array size with 1 runs

sequentialSortedArray -> Average time (ms): for 60000 Array size | Average Sorting Time (in ms): 1.0

2021-09-26 18:31:05 INFO Benchmark_Timer - Begin run: sequentialSortedArray -> Average time (ms): for 70000 Array size with 1 runs

sequentialSortedArray -> Average time (ms): for 70000 Array size | Average Sorting Time (in ms): 1.0

2021-09-26 18:31:05 INFO Benchmark_Timer - Begin run: sequentialSortedArray -> Average time (ms): for 80000 Array size with 1 runs

sequentialSortedArray -> Average time (ms): for 80000 Array size | Average Sorting Time (in ms): 1.0

2021-09-26 18:31:05 INFO Benchmark_Timer - Begin run: sequentialSortedArray -> Average time (ms): for 90000 Array size with 1 runs

sequentialSortedArray -> Average time (ms): for 90000 Array size | Average Sorting Time (in ms): 2.0

2021-09-26 18:31:05 INFO Benchmark_Timer - Begin run: sequentialSortedArray -> Average time (ms): for 100000 Array size with 1 runs

sequentialSortedArray -> Average time (ms): for 100000 Array size | Average Sorting Time (in ms): 1.0

2021-09-26 18:31:05 INFO Benchmark_Timer - Begin run: reversedSortedArray -> Average time (ms): for 10000 Array size with 1 runs

reversedSortedArray -> Average time (ms): for 10000 Array size | Average Sorting Time (in ms): 159.0

2021-09-26 18:31:06 INFO Benchmark_Timer - Begin run: reversedSortedArray -> Average time (ms): for 20000 Array size with 1 runs

reversedSortedArray -> Average time (ms): for 20000 Array size | Average Sorting Time (in ms): 646.0

2021-09-26 18:31:08 INFO Benchmark_Timer - Begin run: reversedSortedArray -> Average time (ms): for 30000 Array size with 1 runs

reversedSortedArray -> Average time (ms): for 30000 Array size | Average Sorting Time (in ms): 1431.0

2021-09-26 18:31:12 INFO Benchmark_Timer - Begin run: reversedSortedArray -> Average time (ms): for 40000 Array size with 1 runs

reversedSortedArray -> Average time (ms): for 40000 Array size | Average Sorting Time (in ms): 2658.0

2021-09-26 18:31:20 INFO Benchmark_Timer - Begin run: reversedSortedArray -> Average time (ms): for 50000 Array size with 1 runs

reversedSortedArray -> Average time (ms): for 50000 Array size | Average Sorting Time (in ms): 4008.0

2021-09-26 18:31:32 INFO Benchmark_Timer - Begin run: reversedSortedArray -> Average time (ms): for 60000 Array size with 1 runs

reversedSortedArray -> Average time (ms): for 60000 Array size | Average Sorting Time (in ms): 5876.0

2021-09-26 18:31:50 INFO Benchmark_Timer - Begin run: reversedSortedArray -> Average time (ms): for 70000 Array size with 1 runs

reversedSortedArray -> Average time (ms): for 70000 Array size | Average Sorting Time (in ms): 7957.0

2021-09-26 18:32:14 INFO Benchmark_Timer - Begin run: reversedSortedArray -> Average time (ms): for 80000 Array size with 1 runs

reversedSortedArray -> Average time (ms): for 80000 Array size | Average Sorting Time (in ms): 10420.0

2021-09-26 18:32:46 INFO Benchmark_Timer - Begin run: reversedSortedArray -> Average time (ms): for 90000 Array size with 1 runs

reversedSortedArray -> Average time (ms): for 90000 Array size | Average Sorting Time (in ms): 13870.0

2021-09-26 18:33:27 INFO Benchmark_Timer - Begin run: reversedSortedArray -> Average time (ms): for 100000 Array size with 1 runs

reversedSortedArray -> Average time (ms): for 100000 Array size | Average Sorting Time (in ms): 16859.0

Test Cases for Insertion Sort

The screenshot shows an IDE window titled 'INFO6205 - InsertionSortBenchmark.java'. The main editor displays the source code for 'InsertionSortBenchmark.java'. The code includes imports for 'java.util.Random', 'java.util.function.Supplier', and 'java.util.stream.IntStream'. It defines a class 'InsertionSortBenchmark' with static final variables: 'END_INCLUSIVE = 10', 'ARRAY_SIZE_MULTIPLIER = 10000', and 'NUMBER_OF_ITERATIONS = 1'. The 'main' method calls 'randomSortedArray()', 'halfOrderedArray()', 'sequentialSortedArray()', and 'inverseOrderedArray()', each followed by a call to 'e.printStackTrace()'. The IDE's 'Run' tab at the bottom shows the test results for 'InsertionSortTest'. The tests passed: 6 of 6 tests - 122 ms. The test results table is as follows:

Test Case	Time (ms)
testMutatingInsertionSort	79 ms
sort0	37 ms
sort1	1 ms
sort2	3 ms
sort3	1 ms
testStaticInsertionSort	1 ms

The bottom status bar indicates 'Tests passed: 6 (moments ago)' and '25:13 P Fail2021-INFO6205-Assignment-2'.

Part 3 Of Assignment

Insertion Sort Output Analysis

Output Obtained

Quantity	Random Sorted Array	Partially Sorted Array	Sequential Sorted Array	Reversed Sorted Array
10000	144	88	1	159
20000	639	267	1	646
30000	971	684	1	1431
40000	1597	1557	1	2658
50000	2768	2026	1	4008
60000	4606	2749	1	5876
70000	6452	5223	1	7957
80000	8343	5607	1	10420
90000	10962	8713	2	13870
100000	15043	8965	1	16859

