

Narendra Raj R K (001553969)
Program Structures & Algorithms
Spring 21
Assignment No. 2

Tasks:

Part 1: In this task I am to implement three methods of a class called Timer - **<T, U> double repeat, getClock() & toMillisecs(long ticks).**

⇒ Which I was successfully able to implement as all the test cases under **TimerTest** and **BenchmarkTest** have passed.

Part 2: Implement **Insertion Sort** under the sort() method available under the InsertionSort.java file only with the help of methods already defined under BaseHelper.java file.

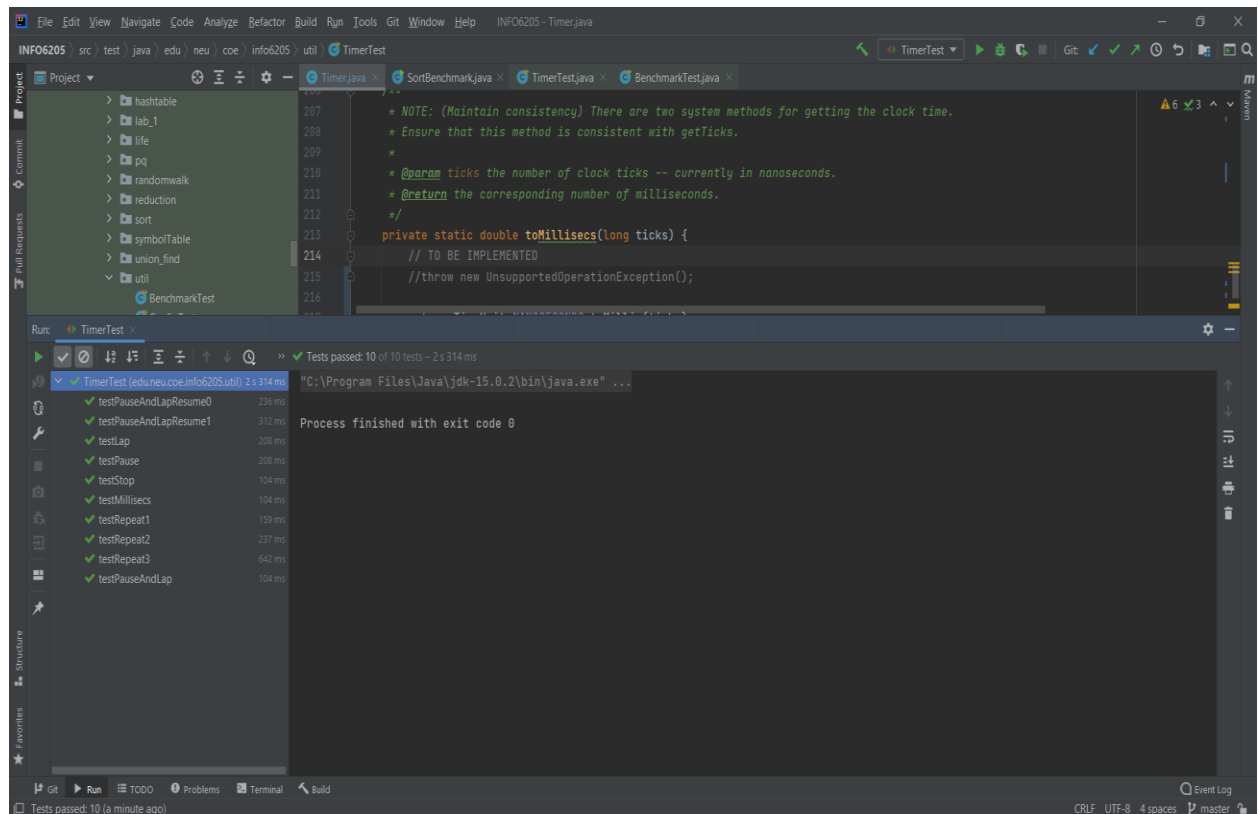
⇒ Successfully implemented the Insertion sort logic only using the methods available under BaseHelper file.

Part 3: Implement a **main method in Benchmark_Timer.java** to benchmark the Insertion sort algorithm against different array inputs.

- ⇒ Implemented a main method to successfully benchmark against four distinctly sorted arrays of type Integer (ordered, random, partial & reversed).

Unit Test Results:

- i) TimerTest.java (All 10 test cases passed)

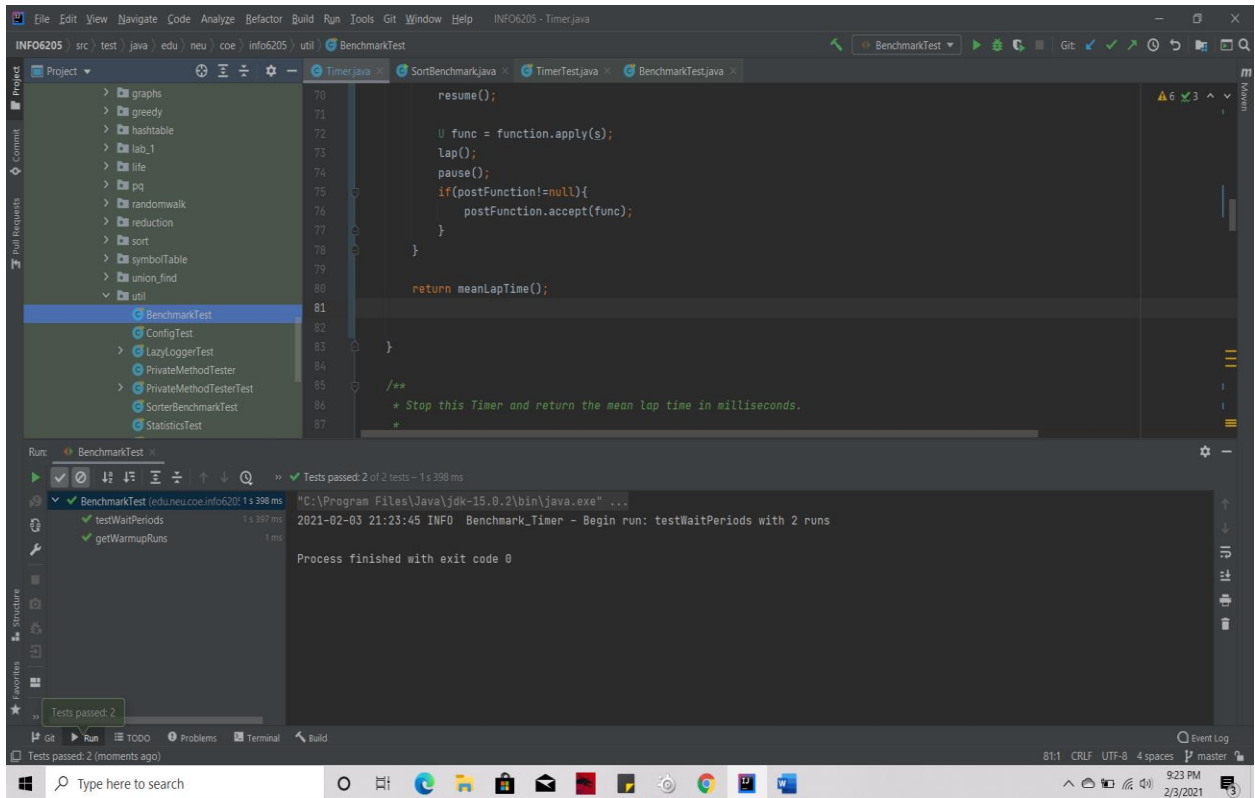


The screenshot displays an IDE window for a project named 'INFO6205 - Timer.java'. The 'TimerTest.java' file is open, showing a method `toMillisecs(long ticks)` that is currently a placeholder implementation. The code includes comments about maintaining consistency with `getTicks()` and converting clock ticks to milliseconds. Below the code editor, the 'Run' tab shows the test results for 'TimerTest'. The tests passed, with a total execution time of 26314 ms. The test results are as follows:

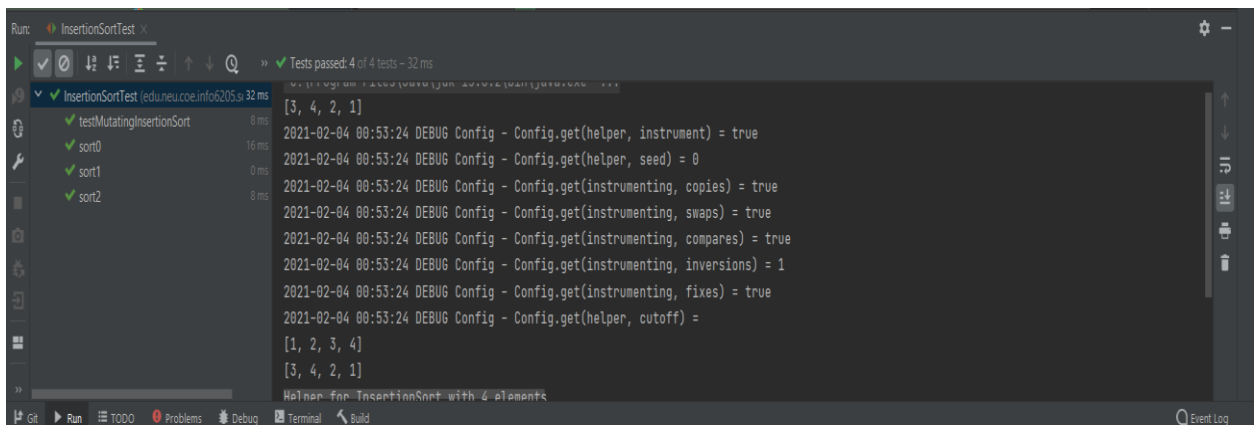
Test Case	Duration (ms)
testTest	236
testPauseAndLapResume0	312
testPauseAndLapResume1	208
testLap	208
testPause	104
testStop	104
testMillisecs	104
testRepeat1	104
testRepeat2	237
testRepeat3	642
testPauseAndLap	104

The IDE also shows the project structure on the left, including folders like 'hashtable', 'lab_1', 'ife', 'pq', 'randomwalk', 'reduction', 'sort', 'symbolTable', 'union_find', and 'util'. The 'util' folder contains the 'BenchmarkTest' folder.

ii) BenchmarkTest.java (All 2 test cases passed)



iii) InsertionSortTest.java (All 4 test cases passed)



Evidence to Support Conclusion:

I have created four different arrays as per the requirement of this assignment,

1. Sorted Array
2. Reversed Array
3. Random Array
4. Partial Array

Some useful Abbreviations

- n – Number of elements in the Array

I ran the experiment for various “ n ” values like 1000 & 10000.

- m – number of repetitions to arrive at the mean time.

Ran the experiment for different values of m like 10,50 & 100.

Please find observations, screenshots & graphs etc., supporting the conclusion

Run 1: m – 10 ; n – 1000 (10 Reps & 1000 elements in the array)

Type of Array	M – Repetitions	N – No. of Elements	Mean Time (Milliseconds)
Sorted	10	1000	0.0
Reversed	10	1000	2.8
Random	10	1000	1.2
Partial	10	1000	0.2

Screenshot of the Output

```

74         return array;
75     };
76
77     final Supplier<Integer[]> partialArray = () -> {
78         Integer[] array = (Integer[]) Array.newInstance(Integer.class, 1000);
79         for(int i=0; i<500; i++) array[i] = i;
80         for(int i=500; i<1000; i++) {
81             array[i] = getRandomValue( Min: 500, Max: 1000);
82         }
83         return array;
84     };
85
86     Benchmark<Integer []> b = new Benchmark_Timer<Integer[]>{
87         description: "Insertion Sort Benchmark",
88         run: run
89     };

```

```

Run: Benchmark_Timer
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.3.2\lib\idea_rt.jar=59091:C:\Program Files\JetBrains\I
2021-02-04 04:03:04 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 10 runs
The time taken by the Insertion Sort to sort an already Sorted array of Size 1000 is 0.0
2021-02-04 04:03:04 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 10 runs
The time taken by the Insertion Sort to sort an Reversed array of Size 1000 is 2.8
2021-02-04 04:03:04 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 10 runs
The time taken by the Insertion Sort to sort an Randomly ordered array of Size 1000 is 1.2
2021-02-04 04:03:04 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 10 runs
The time taken by the Insertion Sort to sort an Partially sorted array of Size 1000 is 0.2

Process finished with exit code 0

```

Run 2: $m = 50$; $n = 1000$ (50 Reps & 1000 elements in the array)

Type of Array	M – Repetitions	N – No. of Elements	Mean Time (Milliseconds)
Sorted	50	1000	0.02
Reversed	50	1000	2.44
Random	50	1000	1.04
Partial	50	1000	0.24

Screenshot of the Output

```

//Pre: null,
new InsertionSort().sort,
//Post: null
});

System.out.println("The time taken by the Insertion Sort to sort an already Sorted array of Size 1000 is " + b.run(sorted));
System.out.println("The time taken by the Insertion Sort to sort an Reversed array of Size 1000 is " + b.run(reversed));
System.out.println("The time taken by the Insertion Sort to sort an Randomly ordered array of Size 1000 is " + b.run(random));
System.out.println("The time taken by the Insertion Sort to sort an Partially sorted array of Size 1000 is " + b.run(partiallySorted));
}

/**
 * Calculate the appropriate number of warmup runs.
 *
 * @param m the number of runs.
 * @return at least 2 and at most m/10.
 */

```

```

C:\Program Files\Java\jdk-15.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.3.2\lib\idea_rt.jar=59869:C:\Program Files\JetBrains\I
2021-02-04 05:00:55 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 50 runs
The time taken by the Insertion Sort to sort an already Sorted array of Size 1000 is 0.02
2021-02-04 05:00:55 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 50 runs
The time taken by the Insertion Sort to sort an Reversed array of Size 1000 is 2.44
2021-02-04 05:00:55 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 50 runs
The time taken by the Insertion Sort to sort an Randomly ordered array of Size 1000 is 1.04
2021-02-04 05:00:55 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 50 runs
The time taken by the Insertion Sort to sort an Partially sorted array of Size 1000 is 0.24

Process finished with exit code 0

```

Build completed successfully in 1 sec, 73 ms (moments ago)

Run 3: $m = 100$; $n = 10000$ (100 Reps & 10000 elements in the array)

Type of Array	M – Repetitions	N – No. of Elements	Mean Time (Milliseconds)
Sorted	100	10000	0.08
Reversed	100	10000	237.57
Random	100	10000	111.9
Partial	100	10000	54.8

Screenshot of the Output

The screenshot shows an IDE with a Java project named 'INFO6205'. The code in 'Benchmark_Timer.java' defines a method 't()' that sorts an array of size 10000 using Insertion Sort and measures the time taken for 100 repetitions. The code includes comments for Sorted, Reversed, Random, and Partially sorted arrays.

```

89
90 t():sort,
91
92
93
94 ("The time taken by the Insertion Sort to sort an already Sorted array of Size 10000 is " + b.run(sortedArray.get(), m, 100));
95 ("The time taken by the Insertion Sort to sort an Reversed array of Size 10000 is " + b.run(reversedArray.get(), m, 100));
96 ("The time taken by the Insertion Sort to sort an Randomly ordered array of Size 10000 is " + b.run(randomArray.get(), m, 100));
97 ("The time taken by the Insertion Sort to sort an Partially sorted array of Size 10000 is " + b.run(partialArray.get(), m, 100));
98
99
100
101 te number of warmup runs.
102
103 runs.
104 at most m/10.

```

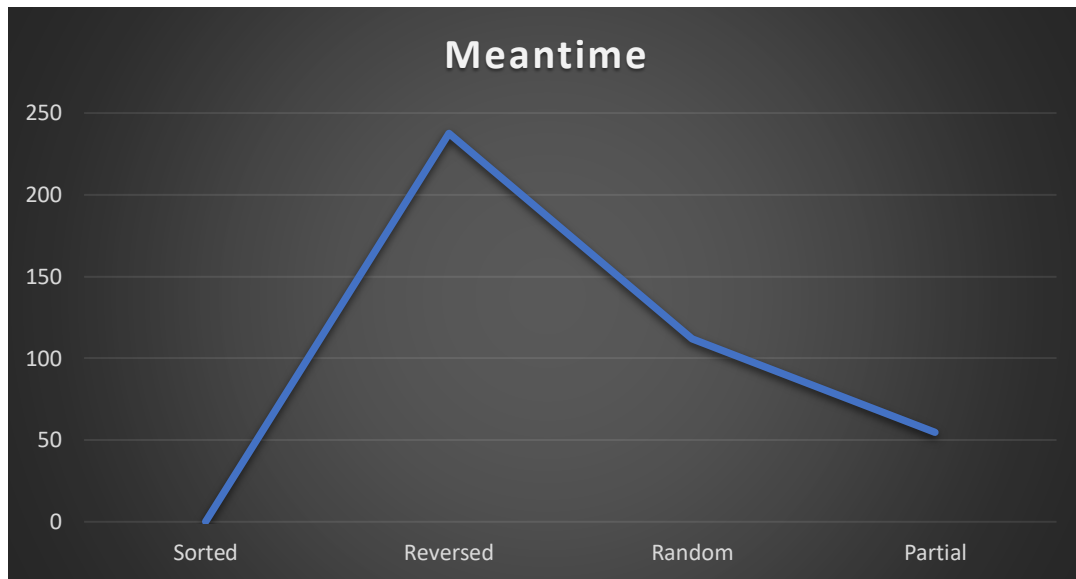
The Run console shows the output of the benchmarking process:

```

Run: Benchmark_Timer
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.3.2\lib\idea_rt.jar=58970:C:\Program Files\JetBrains\I
2021-02-04 03:57:07 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 100 runs
The time taken by the Insertion Sort to sort an already Sorted array of Size 10000 is 0.08
2021-02-04 03:57:07 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 100 runs
The time taken by the Insertion Sort to sort an Reversed array of Size 10000 is 237.57
2021-02-04 03:57:33 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 100 runs
The time taken by the Insertion Sort to sort an Randomly ordered array of Size 10000 is 111.9
2021-02-04 03:57:45 INFO Benchmark_Timer - Begin run: Insertion Sort Benchmark with 100 runs
The time taken by the Insertion Sort to sort an Partially sorted array of Size 10000 is 54.8
Process finished with exit code 0

```

Graphical Representation:



Final Conclusion:

- After benchmarking the Insertion Sort Algorithm against the different types of Arrays – Sorted, Reversed, Random & Partially ordered for multiple runs. I have arrived at the following conclusion, that time taken to sort an already sorted array is the least and to sort an reversed array consumes the most time.

Sorted Array < Partial Array < Random Array < Reversed Array