Program Structures and Algorithms
Spring 2023(SEC –08)

NAME: **ALTAF SALIM SHAIKH**
NUID: **002774748**

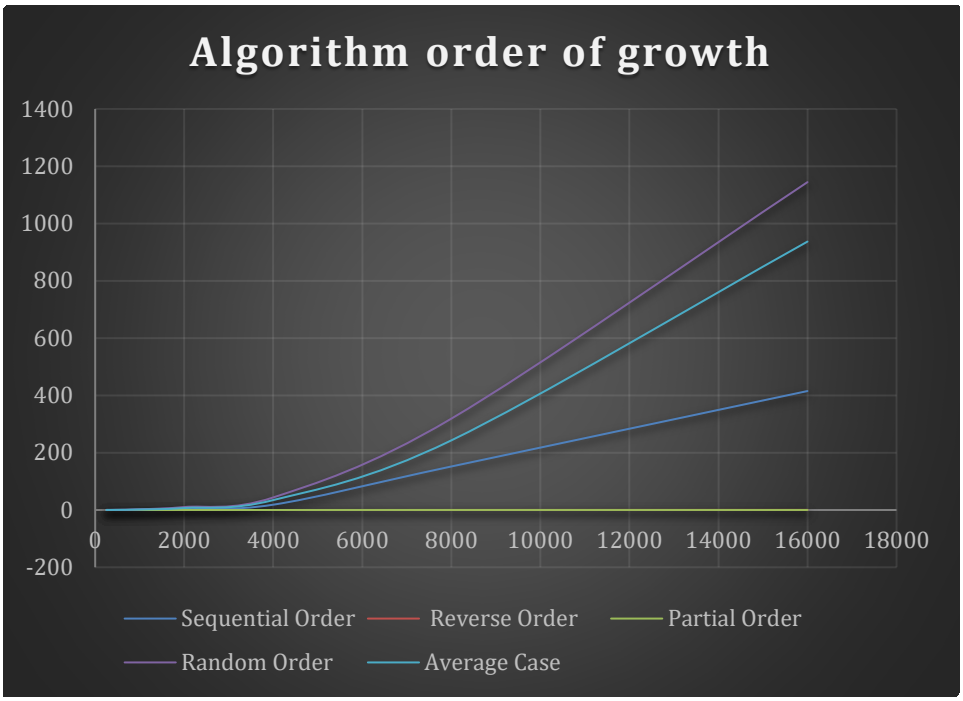
Task: To fix the Timer Class, you need to run the test cases in BenchmarkTest.java, TimerTest.java, and InsertionSortTest.java. Additionally, you should use the helper function to implement Insertion Sort. To evaluate the performance of the sorting algorithm, create a main method that generates arrays of varying sizes using the doubling method. Finally, draw conclusions from your observations and evidence to analyze the performance of the Timer Class.

Relationship Conclusion: The order of speed for Insertion Sort on arrays of the same size would be sorted arrays being the fastest, followed by partially ordered arrays, then random arrays, and finally reverse ordered arrays being the slowest. The execution time of reverse ordered, random, and partially ordered arrays grows quadratically. Insertion Sort is well-suited for partially ordered arrays, but its worst-case scenario is when the array is in reverse order

Graphical Representation:

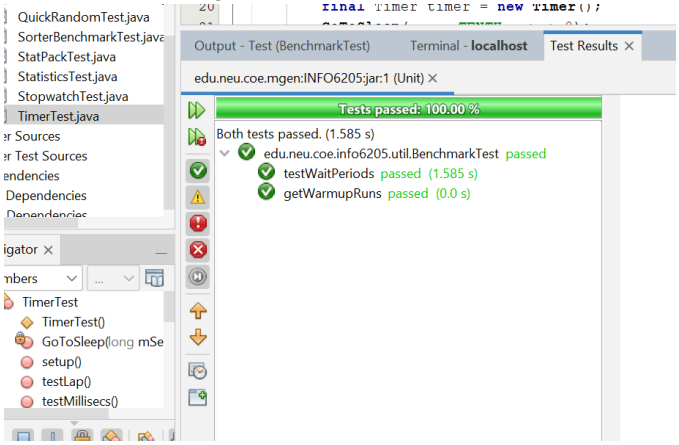
Run Time (ms)

Array Length	Sequential Order	Reverse Order	Partial Order	Random Order	Average Case
250	0.11	0	0	0.3	0.245
500	0.32	0	0	0.64	0.48
1000	1.36	0.01	0.01	2.45	1.775
2000	6.76	0.03	0.01	10.39	7.015
4000	18.56	0.03	0.04	43.95	34.69
8000	151.96	0.15	0.19	318.99	243.105
16000	415.24	0.37	0.16	1144.51	936.97

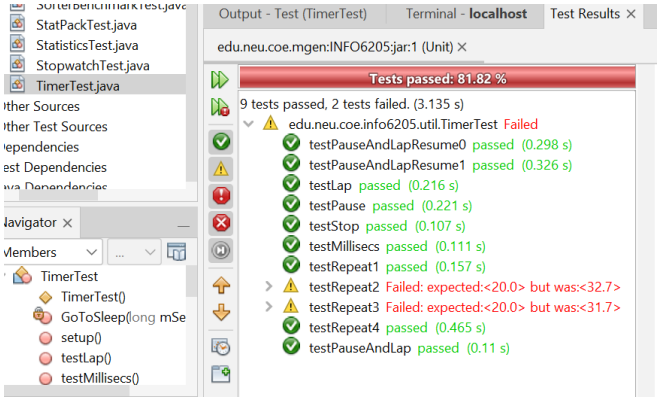


Unit Test Screenshots:

- BenchmarkTest.java



- TimerTest.java (Issue as discussed is due to different machine architectures)



- InsertionSortTest .java