Program Structures and Algorithms
Spring 2023(SEC –3)

NAME: Akshay Parab
NUID: 002766150

Task:
Run benchmark to measure the running time of Insertion Sort using four different initial array ordering situations: random, ordered, partially-ordered and reverse-ordered. By using the doubling method for choosing input size and test for at least five values of input size, draw any conclusions from your observations regarding the order of growth of the Insertion Sort algorithm.

Relationship Conclusion:
Please find detailed observation based on the benchmark test on Insertion Sort:

1. Best-Case Scenario: The best-case scenario for the Insertion Sort algorithm is an ordered array, meaning the elements in the array are already sorted. In this scenario, the algorithm has a linear time complexity of O(n). This is because each element only needs to be compared and inserted once, resulting in a running time proportional to the size of the input.
2. Average-Case Scenario: The average-case scenario for the Insertion Sort algorithm is a random array, meaning the elements in the array are not sorted. In this scenario, the algorithm has a quadratic time complexity of O(n^2). This is because each element needs to be compared and inserted several times, resulting in a running time proportional to the square of the size of the input.
3. Worst-Case Scenario: The worst-case scenario for the Insertion Sort algorithm is a reverse-ordered array, meaning the elements in the array are sorted in the opposite direction. In this scenario, the algorithm also has a quadratic time complexity of O(n^2). This is because each element needs to be compared and inserted several times, resulting in a running time proportional to the square of the size of the input.
4. Partially Ordered Scenario: The partially ordered scenario for the Insertion Sort algorithm refers to an array where some elements are sorted, but not all. In this scenario, the time complexity of the algorithm may be between O(n) and O(n^2), depending on the degree of partial order. If the degree of partial order is closer to the ordered scenario, the time complexity will be closer to O(n), and if the degree of partial order is closer to the reverse-ordered scenario, the time complexity will be closer to O(n^2).

Overall, the Insertion Sort algorithm has a quadratic time complexity in most cases and is generally not efficient for large datasets.

Evidence to support that conclusion:
Graphical Representation:
Input Type: Randomly Sorted

Sr. No.	Input Size	Iterations	Average Time (ms)
1	1000	100	1.2529600099999998
2	2000	100	4.8126466599999995
3	4000	100	18.962787919999997
4	8000	100	74.54104998
5	16000	100	444.38567997999996

Input Type: Partially Sorted

Sr. No.	Input Size	Iterations	Average Time (ms)
1	1000	100	0.9302303999999999
2	2000	100	3.6490608599999996
3	4000	100	13.7623567
4	8000	100	57.44689715
5	16000	100	240.82581793999998

Input Type: Ordered Input

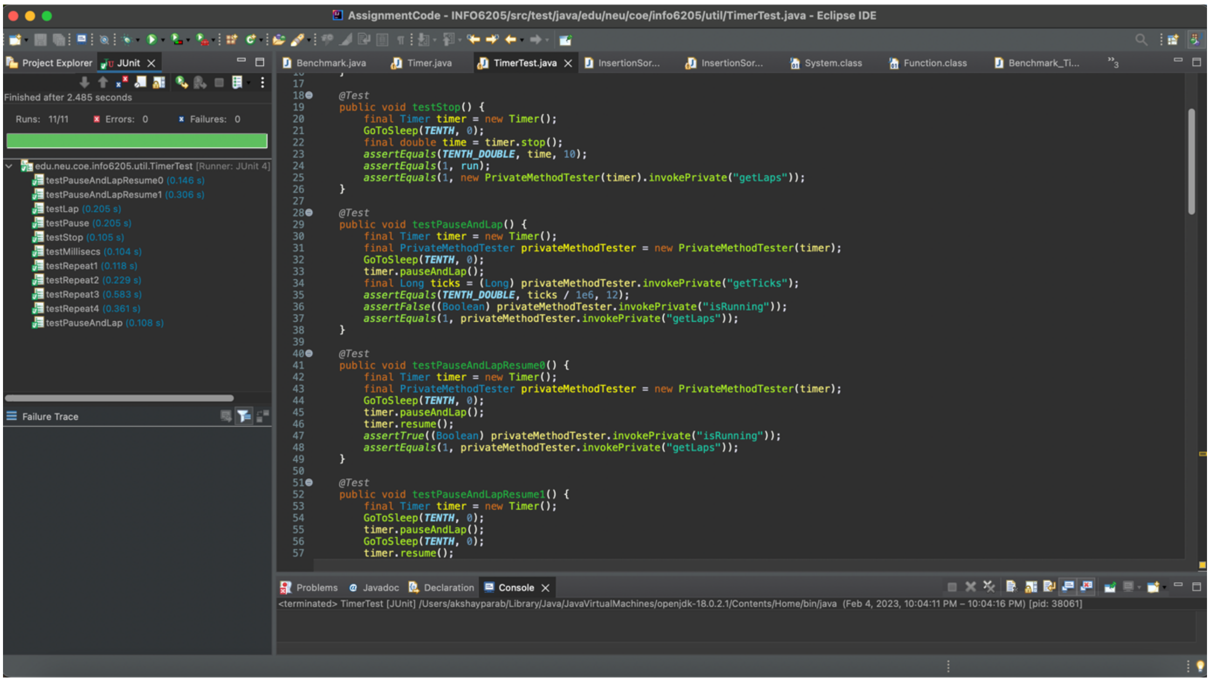
Sr. No.	Input Size	Iterations	Average Time (ms)
1	1000	100	0.00649289
2	2000	100	0.01214583
3	4000	100	0.025107499999999998
4	8000	100	0.051137489999999994
5	16000	100	0.10269331

Input Type: Reverse Sorted

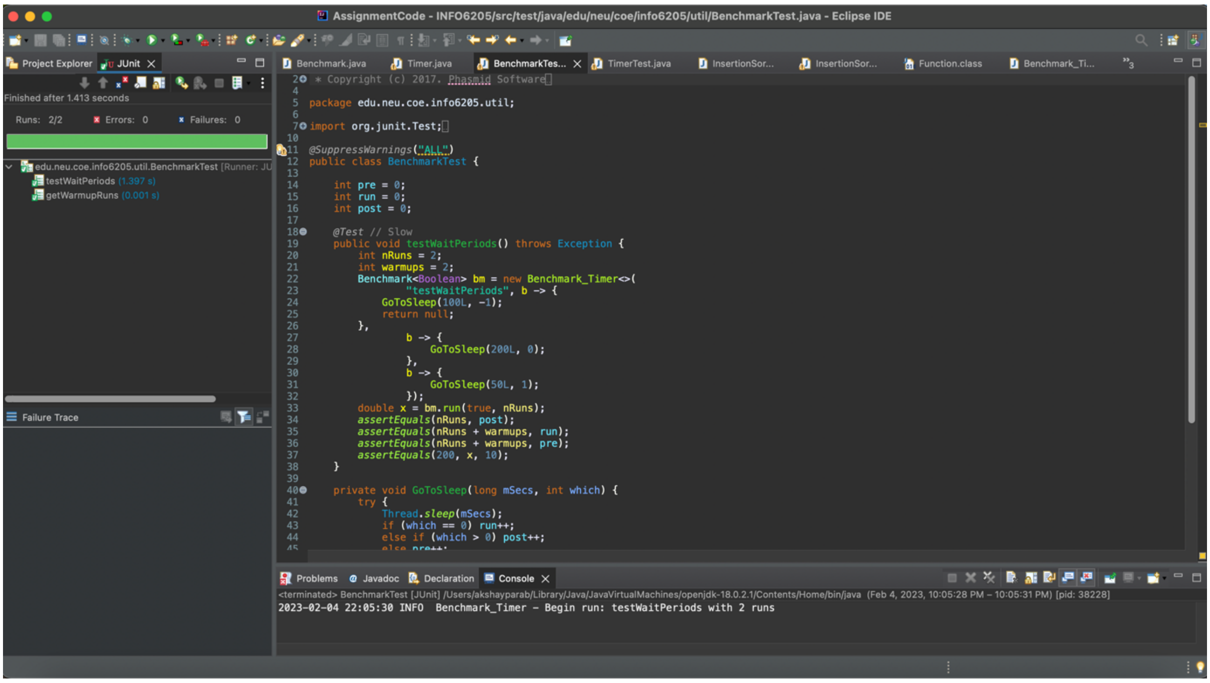
Sr. No.	Input Size	Iterations	Average Time (ms)
1	1000	100	2.39149791
2	2000	100	9.56261752
3	4000	100	36.42599541
4	8000	100	146.34479753
5	16000	100	662.29248377

Unit Test Screenshots:

1) TimerTest.java



2) BenchmarkTest.java



3) InsertSortTest.java