Program Structures and Algorithms

Spring 2023(SEC –08)

NAME: Priyal Vimal Gudhka

NUID: 002747680

**Assignment-3 Benchmark**

1. ***Assignment Part 1***

**Task:** To implement repeat, getClock, and toMillisecs methods of a class called Timer.

**Unit Test Case Screenshots of BenchmarkTest Class**:

Graphical user interface, text

Description automatically generated

**Unit Test Case Screenshots of TimerTest Class**:

Graphical user interface, text, application

Description automatically generated

1. ***Assignment Part 2***

**Task:** To implement InsertionSort and run the unit tests in InsertionSortTest

**Unit Test Case Screenshots of InsertionSort Class:**

Graphical user interface, text

Description automatically generated

1. ***Assignment Part 3***

**Task:** Implement a main program \ to actually run the following benchmarks: measure the running times of this sort, using four different initial array ordering situations: random, ordered, partially ordered and reverse ordered. Draw any conclusions from your observations regarding the order of growth.

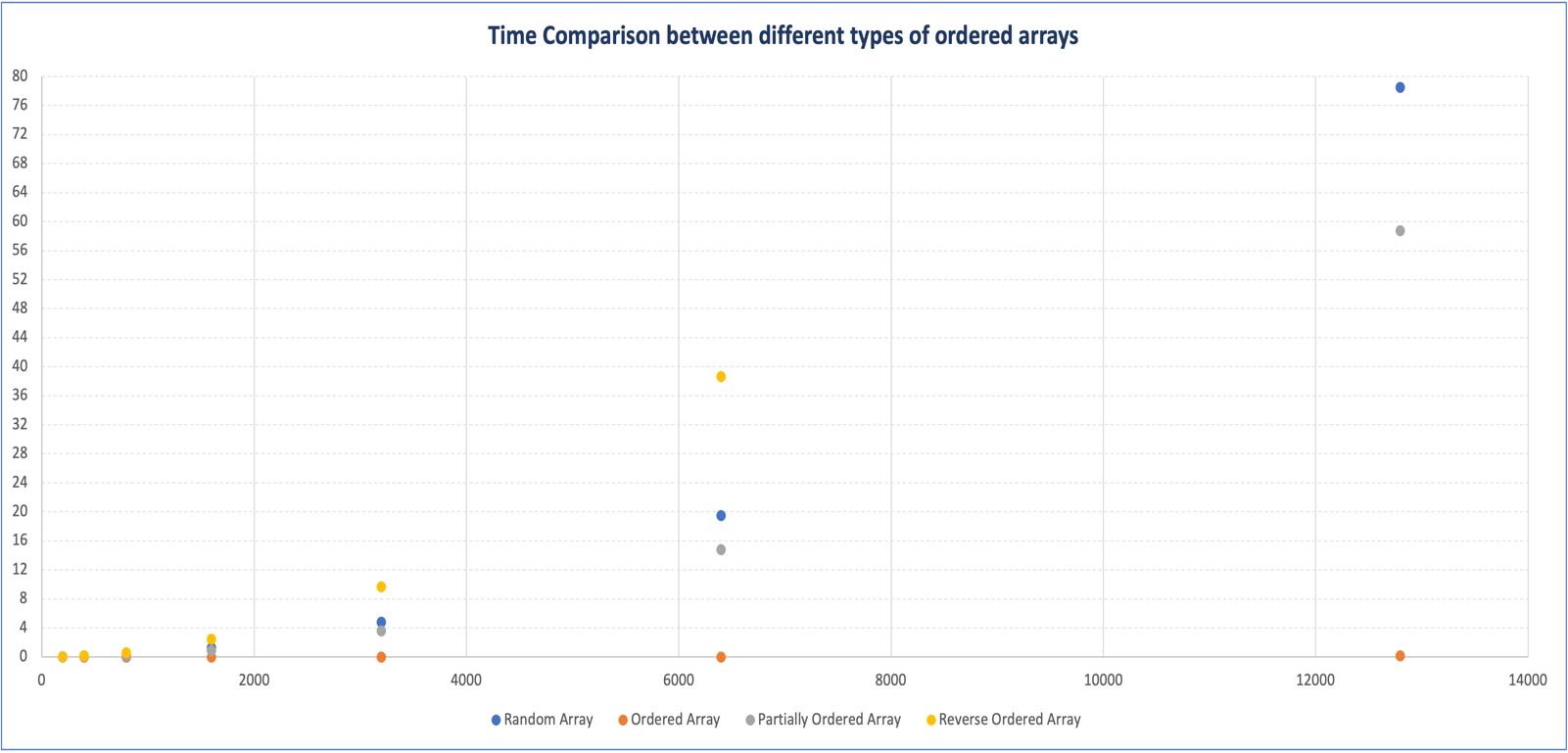
**Conclusion:** As per the results obtained after executing the main program, it is observed that reversely ordered array takes maximum time to perform the insertion sort in comparison to random array, ordered array and partially ordered array. Additionally, ordered array takes minimum time to perform the insertion sort in comparison to random array, ordered array and partially ordered array. Lastly, time taken by differently ordered array to perform sorting from minimum to maximum is ordered array, partially ordered array, random array and reverse ordered array

**Evidence to Support Conclusion:** Below table compares the time calculated in nanoseconds for the differently ordered arrays by executing the program with different array lengths. Seeing the results, it is quite evident that Ordered Array takes least time to sort and Reversely Ordered Array takes the maximum time to sort.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Time taken by Random Array | Time taken by Ordered Array | Time taken by Partially Ordered Array | Time taken by Reverse Ordered Array |
| 200 | 0.04875 | 0.0 | 0.0225 | 0.0775 |
| 400 | 0.165 | 0.00125 | 0.11125 | 0.175 |
| 800 | 0.3225 | 0.00125 | 0.22625 | 0.61 |
| 1600 | 1.245 | 0.0025 | 0.8825 | 2.455 |
| 3200 | 4.835 | 0.01375 | 3.6275 | 9.6775 |
| 6400 | 19.5075 | 0.02 | 14.8125 | 38.6325 |
| 12800 | 78.51125 | 0.17375 | 58.8 | 155.0325 |

**Graphical Representation:**

* **Graph of N and Time taken by Random Array**
* **Graph of N and Time taken by Ordered Array**
* **Graph of N and Time taken by Partially Ordered Array**
* **Graph of N and Time taken by Reverse Ordered Array**
* **Graph of N and Time taken by Differently Ordered Arrays**

****

**Output of the main code (InsertionSortMain class):**

**A screenshot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**