Keyur Ashokbhai Barot (NU ID: 001568664)

**Program Structures and Algorithms**

**Fall 2021**

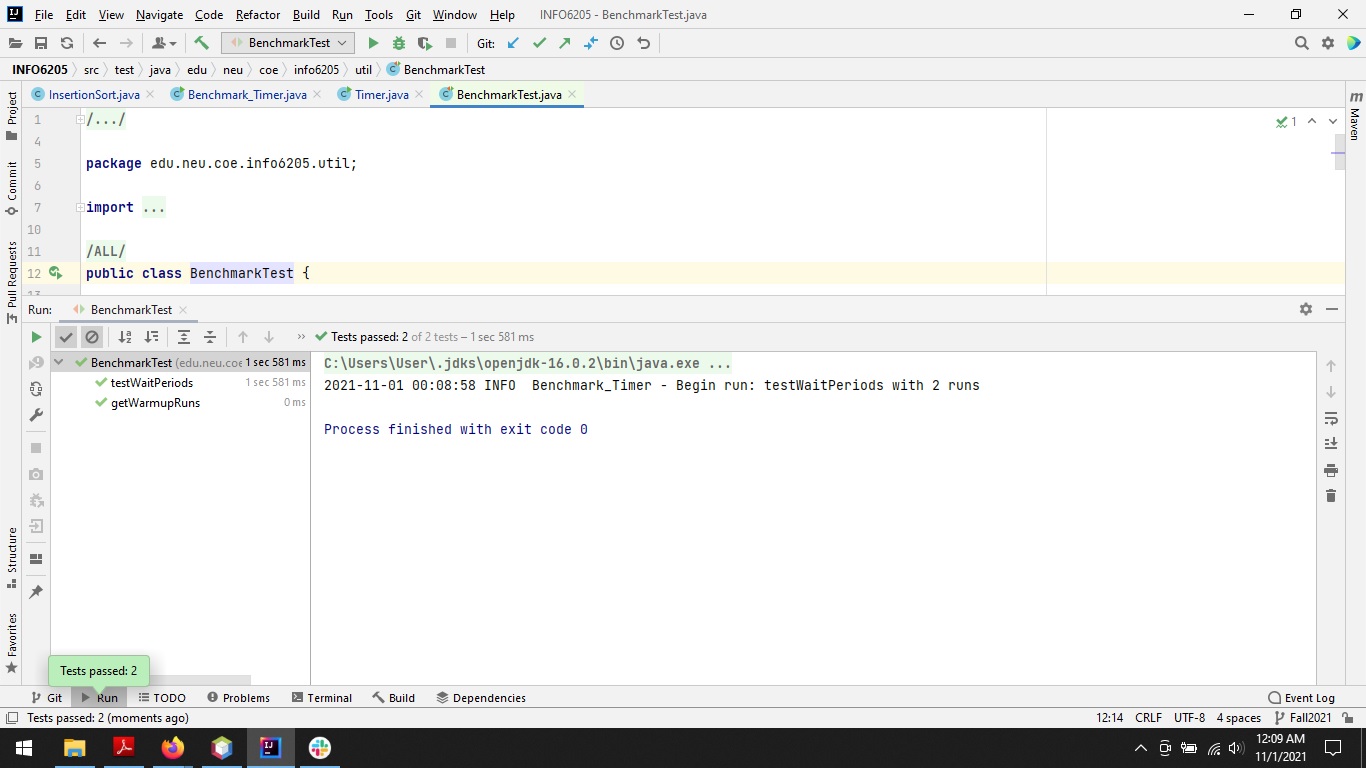
**Assignment No.2 (Benchmark)**

* **Task:**

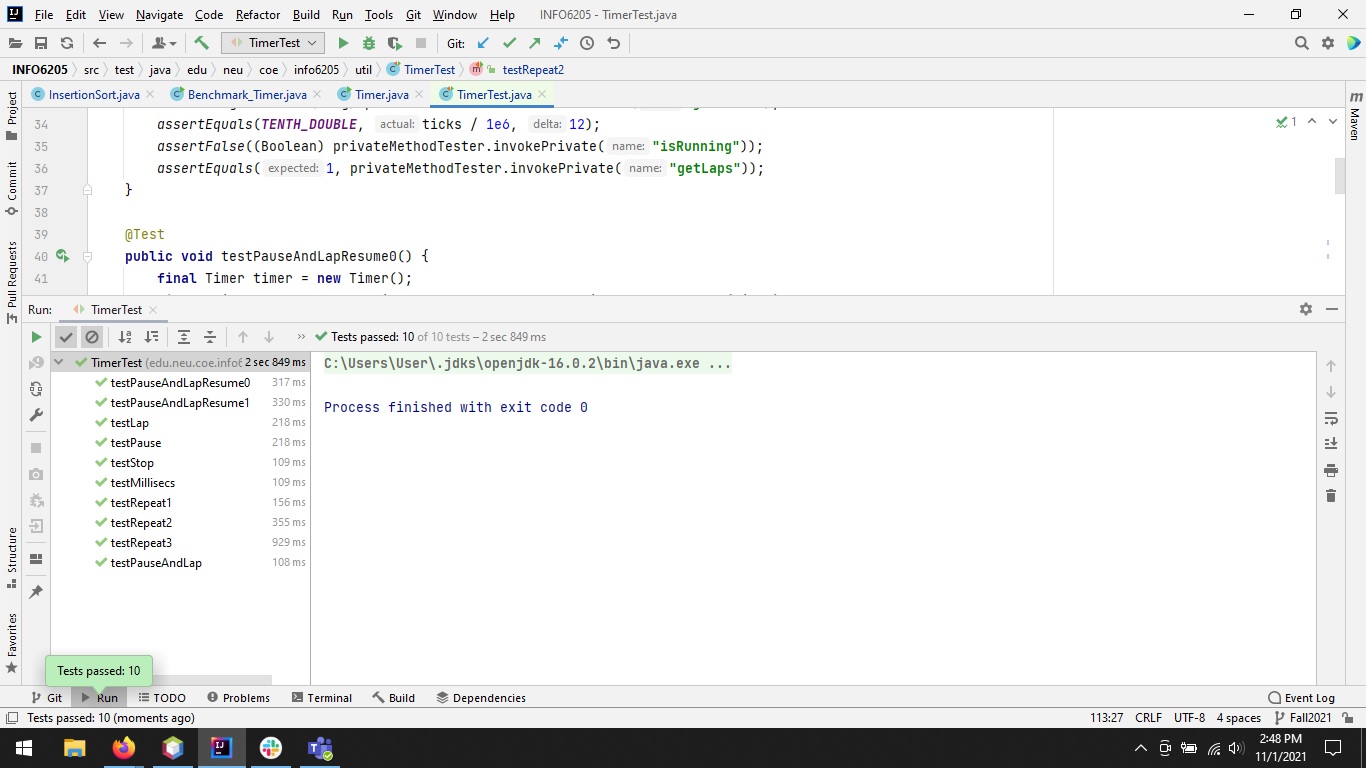
To Calculate benchmark timing for Insertion sort for differently sorted arrays (random, sorted, partially sorted and reverse ordered).

1. Part-1: Implement methods in Timer class.
2. Part-2: Implement InsertionSort in InsertionSort class.
3. Part-3: Implement main class in Benchmark\_Timer class where mean time for random, sorted, partially sorted and reversed array for various sizes will be calculated.

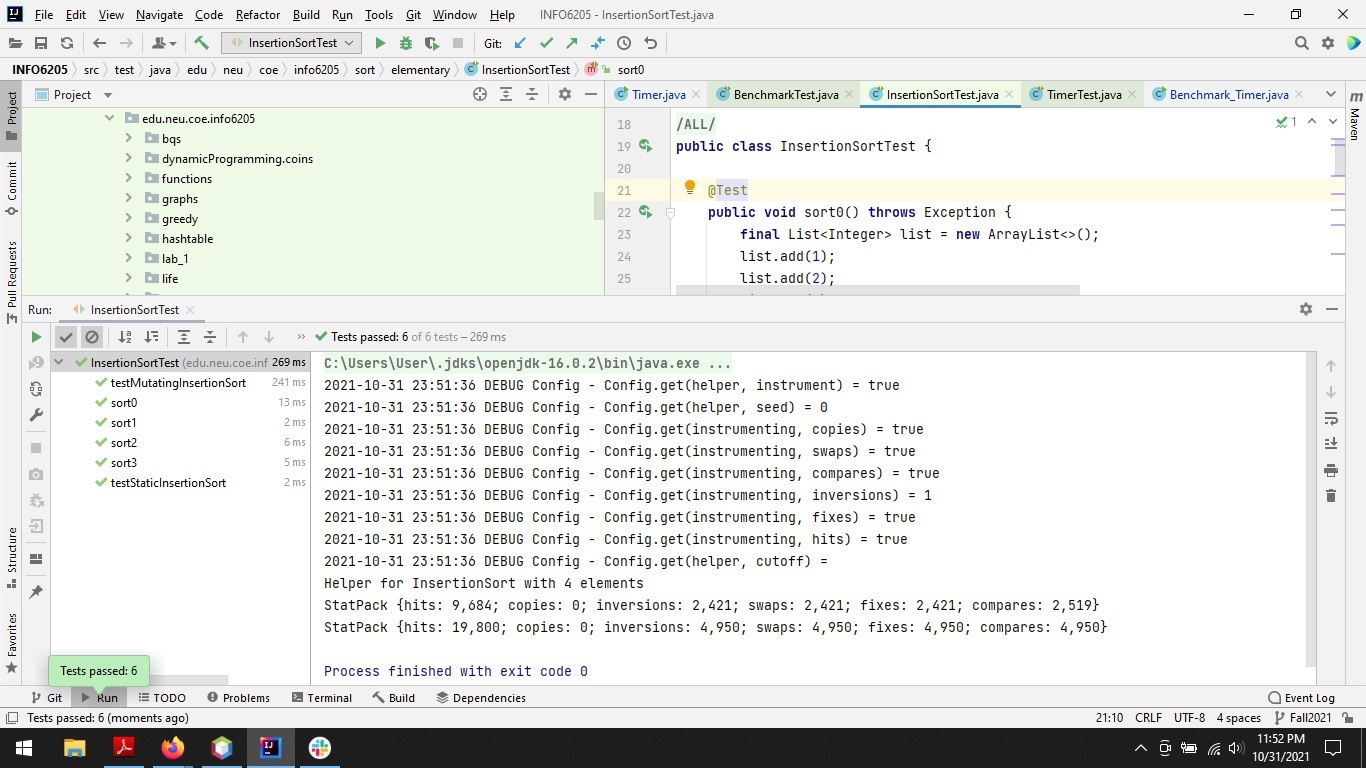
* **Test Cases for Benchmark Test:**

****

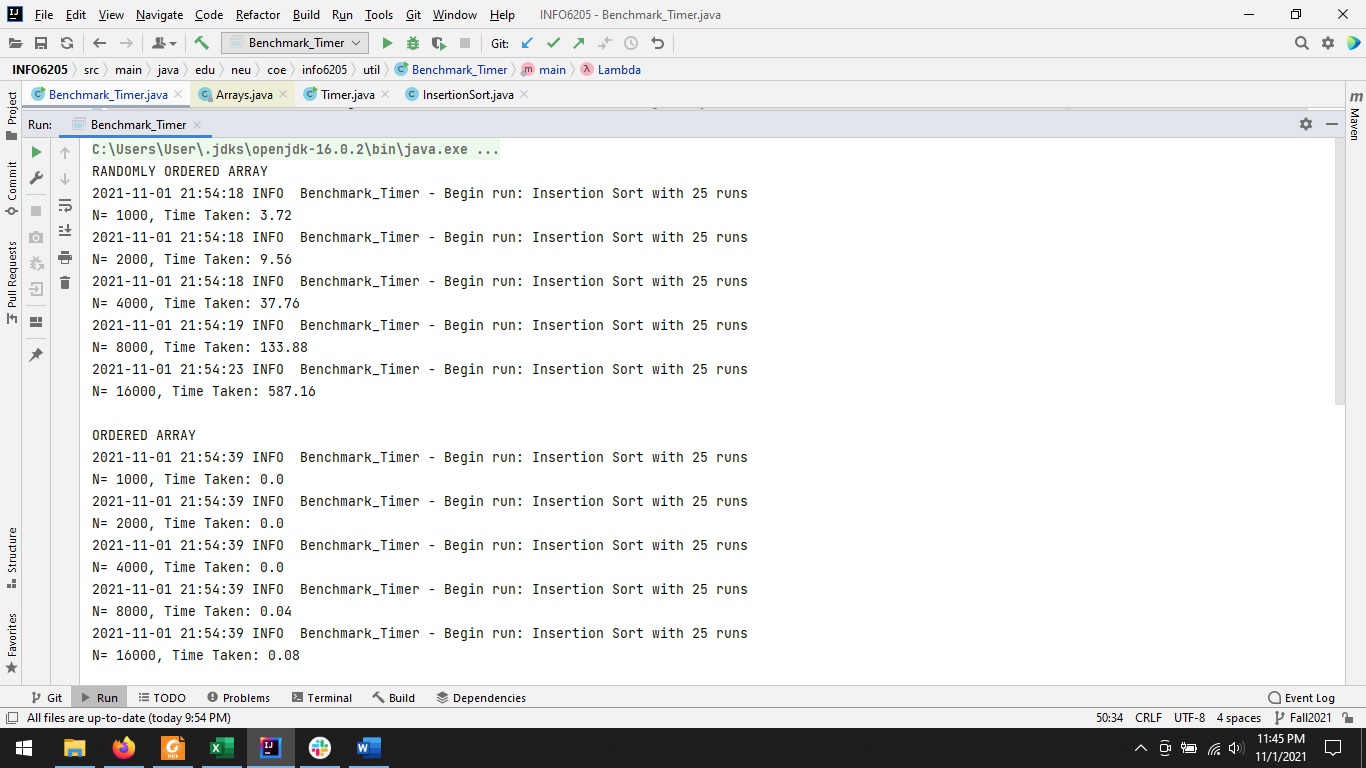
* **Test Cases for TimerTest:**

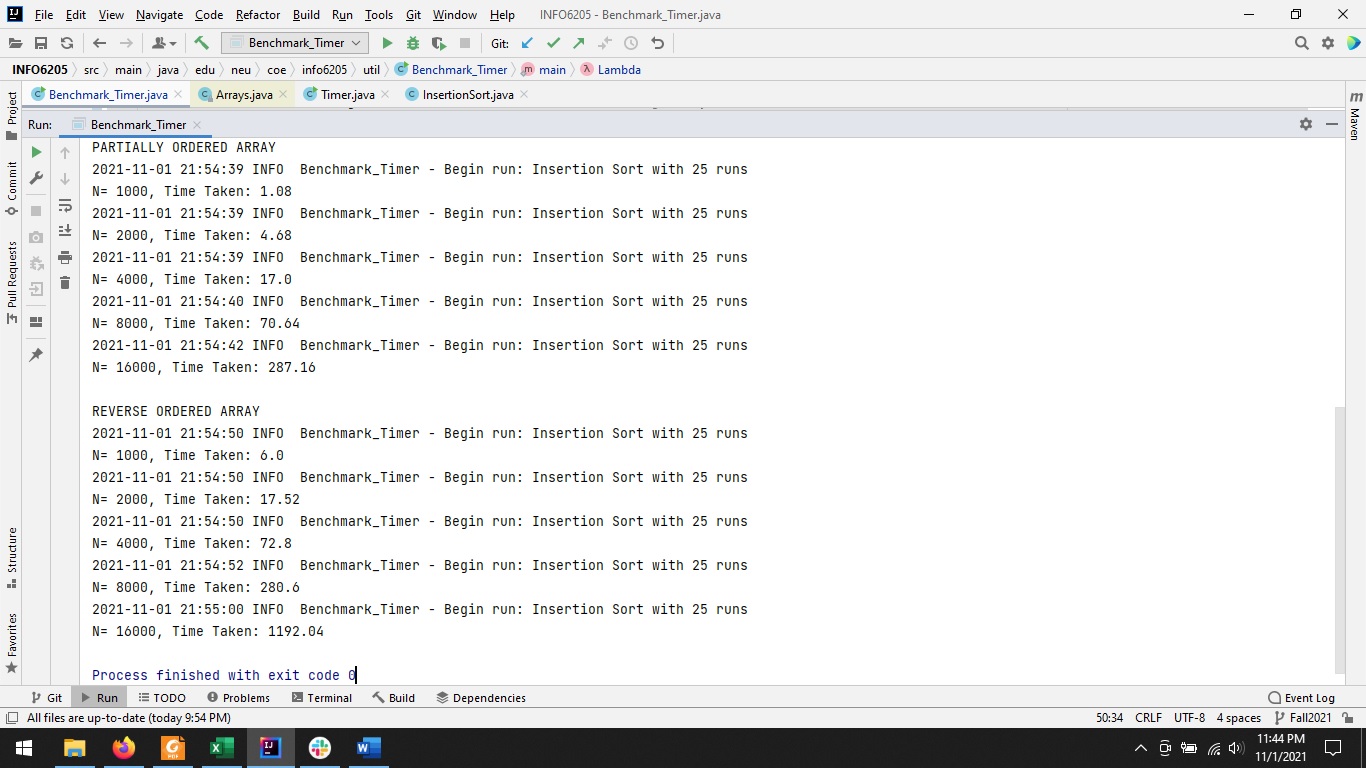
****

* **Test Cases for InsertionSortTest:**

****

* **Output:**

****

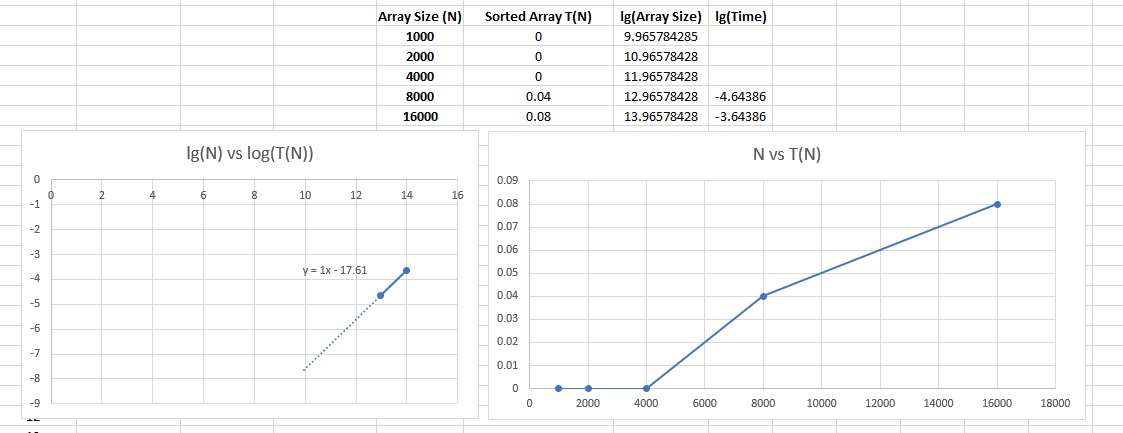
****

* **Relationship Conclusion:**

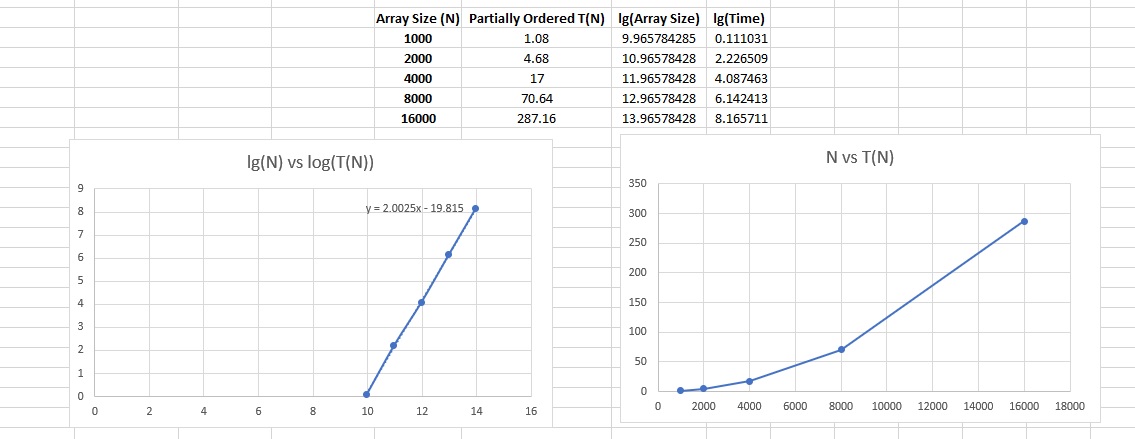
After performing tests on different types of arrays like: Sorted Array, Partially Sorted Array, Randomly Sorted Array and Reverse Ordered Array we can say that Insertion Sort takes the least time for sorting an already Sorted Array followed by Partially Sorted Array, which is followed by Randomly Sorted Array and finally, the most time is taken by a Reverse Ordered Array to be sorted.

* **Evidence:**

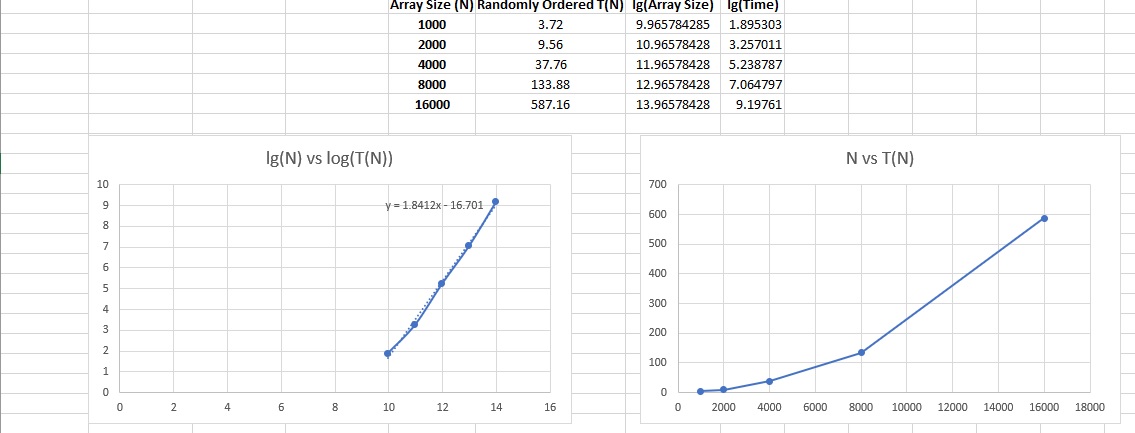
**For Sorted Array**

****

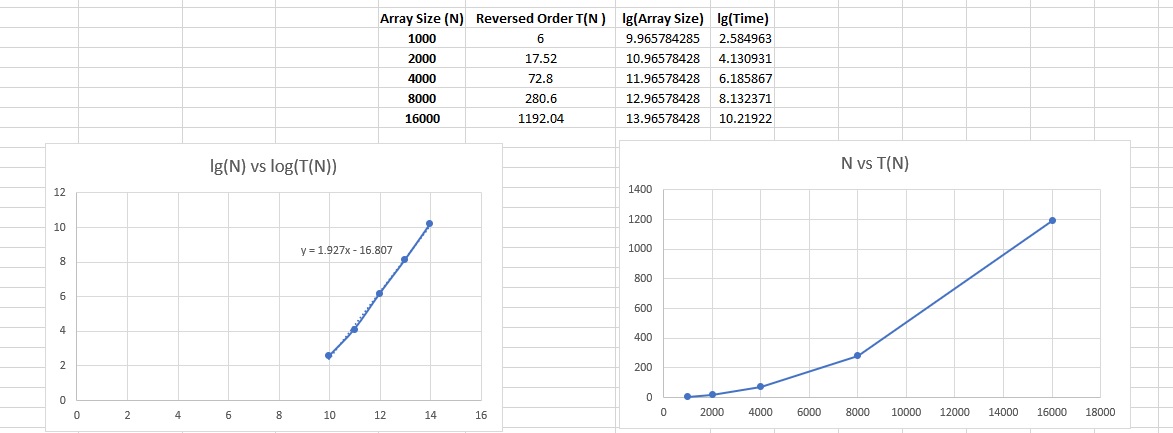
**For Partially Sorted Array**

****

**Randomly Sorted Array**

****

**For Reverse Ordered Array**

****

We can see that for Sorted Array the order of growth Time ∝ n and for everyone else the order of growth Time ∝ n2. Also from the equation of the graphs we can see that the slope of sorted array graph is approximately 1 and for the other its almost tending to 2, from that we can figure out that for sorted array it is linear time and for the rest its almost quadratic time.