**Jyoti Vaswani (NUID -001005711)**

**INFO 6205**

**Program Structures and Algorithms**

**Fall 2020**

**Assignment No -2**

**Note: If any of the inserted object/file fails to open, kindly check the zip. I have added all the supporting files.**

**Task :**

1. **I have added the code in Benchmark\_Timer.java for the methods listed below:**

* **main() –** changed the way input is taken. I have added the loop to generate doubled values of n i.e. number of elements in the array.
* **arrayInput()-** this method generates array of random numbers using getRandomArray() and arranges them as required for input (in 4 cases: randomly arranged, partially sorted, completely/already sorted, reversed sorted). Used for Part4 of assignment.

1. **I have added the code in Timer.java for the methods listed below:**

* **getClock () –** uses System.nanoTime() to get ticks.
* **tomillisecs ()-** converts the ticks into milliseconds.
* **Repeat() –** uses function to invoke sort() of InsertionSort.java. This is used for timing the sort() using benchmark concept.

1. **I have added the code in InsertionSort.java for the methods listed below:**

* **sort() –** sorts the input array using helper methods and is completely based on InsertionSort Algorithm.

**  **

** **

1. **Output:**

Below is the screenshot of output generated by Benchmark\_Timer.java over 10 iterations for different input values using doubling on n. (number of elements -n).

n goes like 2000, 4000, 8000,…128000.

****

****

****

****

****

****

****

1. **Relationship Conclusion:**

The average time taken by insertion sort (t) is proportional to the square of Number of elements (n) in the array.

The equation describing the above relation is :

**t= k.n^2**

The graph of t on y-axis and n on x-axis is of order 2. Refer the attached evidence sheet.

Hence the order of growth is n^2 for insertion sort.

1. **Evidence to support relationship:**

To support the relationship described above, I took 7 values of n – each was double of the previous value of n. I provided the input array containing n values in 4 ways (randomly arranged, partially sorted, completely/already sorted, reversed sorted) and graph for each of them is created in respective sheets.

Further, as mentioned in the requirements of assignment- all the values are mean of 10 repetitions.

All the observations are added in the below attached spreadsheet.

****

1. **Screenshot of Unit test passing:**

Delta values were changed for TimerTest.java and BenchmarkTest.java

Below are the attached screenshots of successful execution and results.



