Keyur Ashokbhai Barot (NU ID: 001568664)

Program Structures and Algorithms

Fall 2021

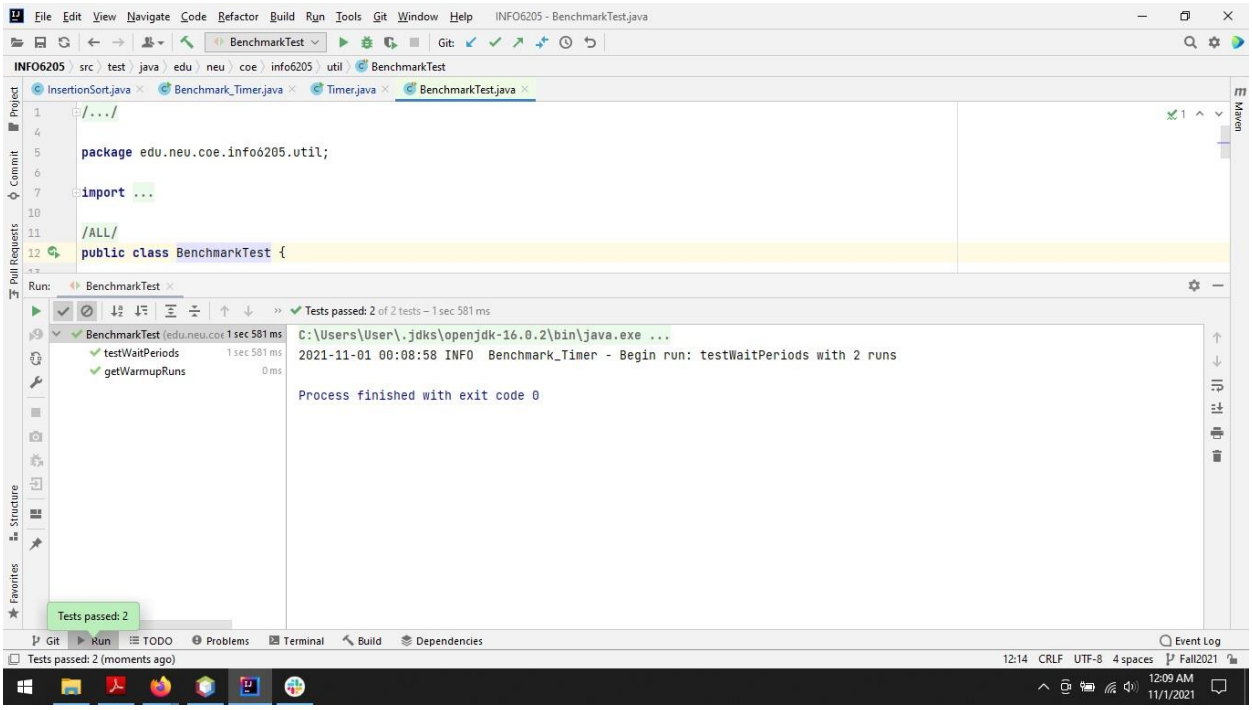
Assignment No.2 (Benchmark)

- **Task:**

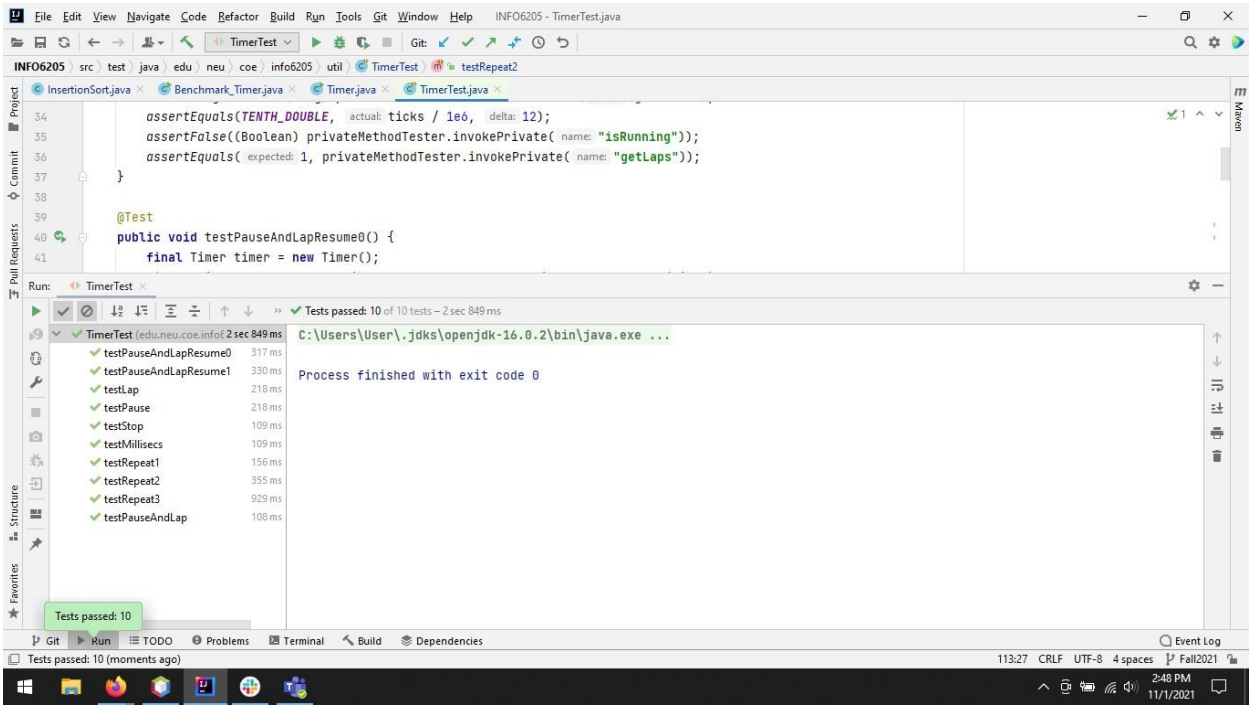
To Calculate benchmark timing for Insertion sort for differently sorted arrays (random, sorted, partially sorted and reverse ordered).

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.

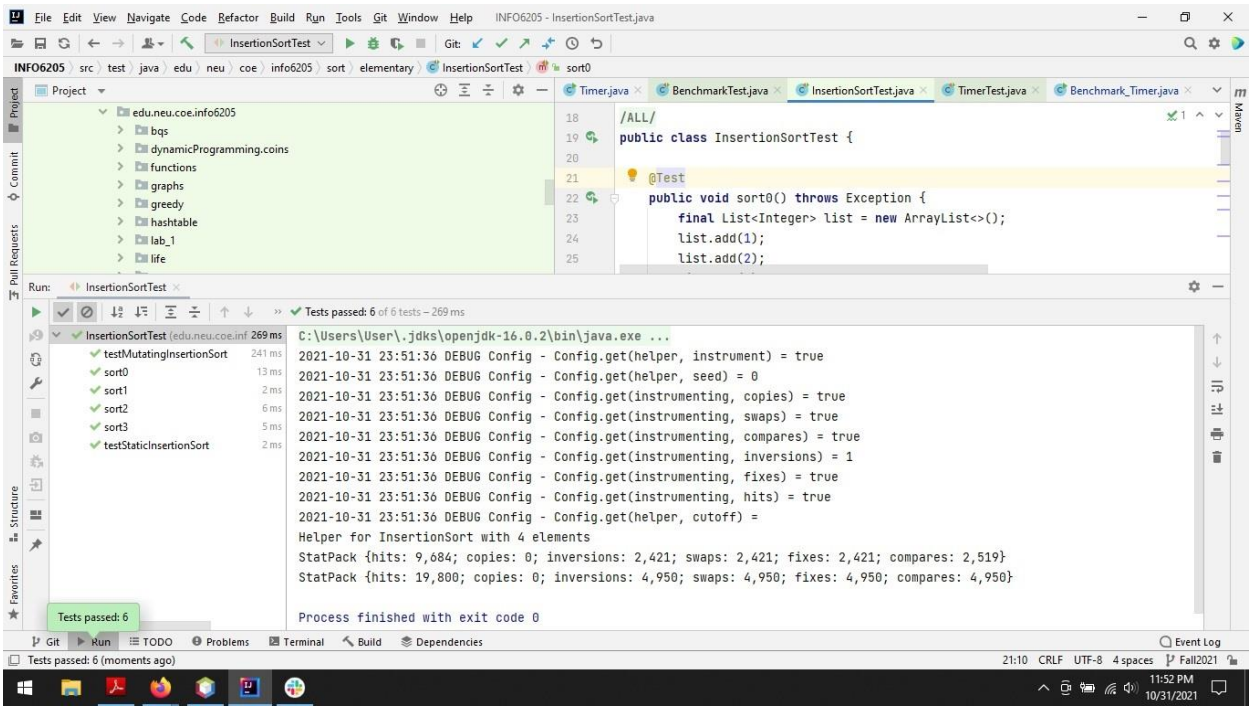
- **Test Cases for Benchmark Test:**



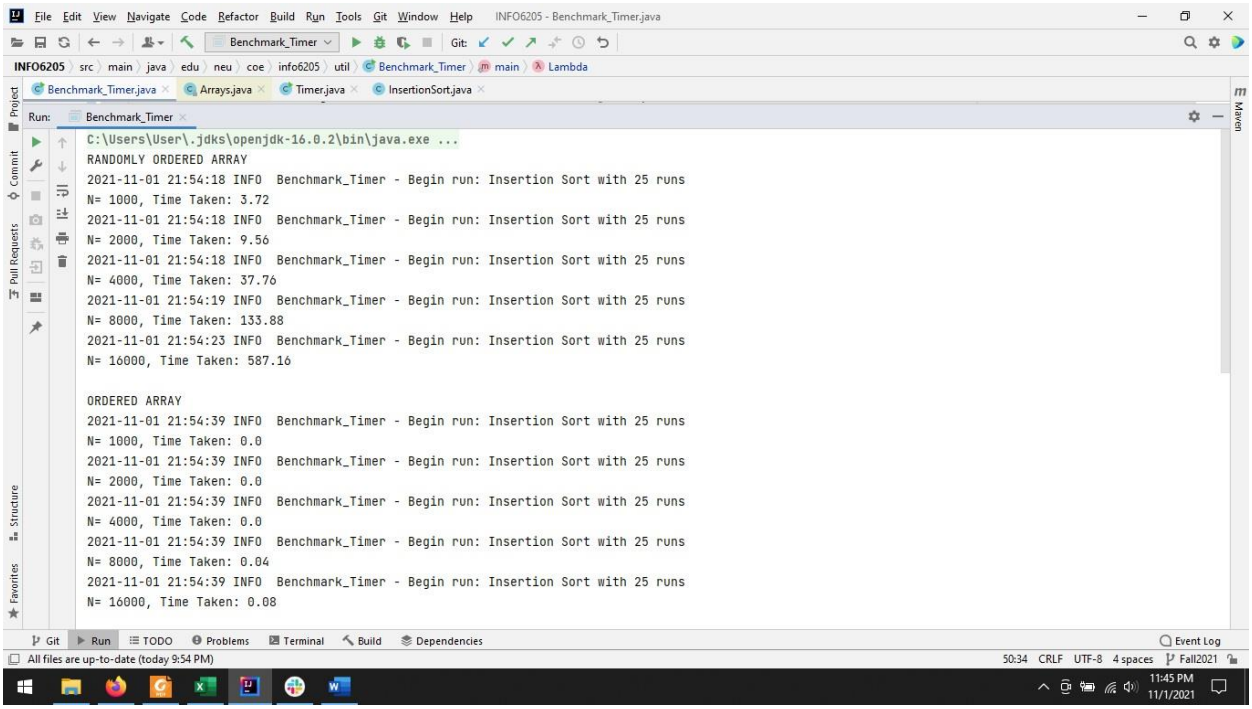
- **Test Cases for TimerTest:**

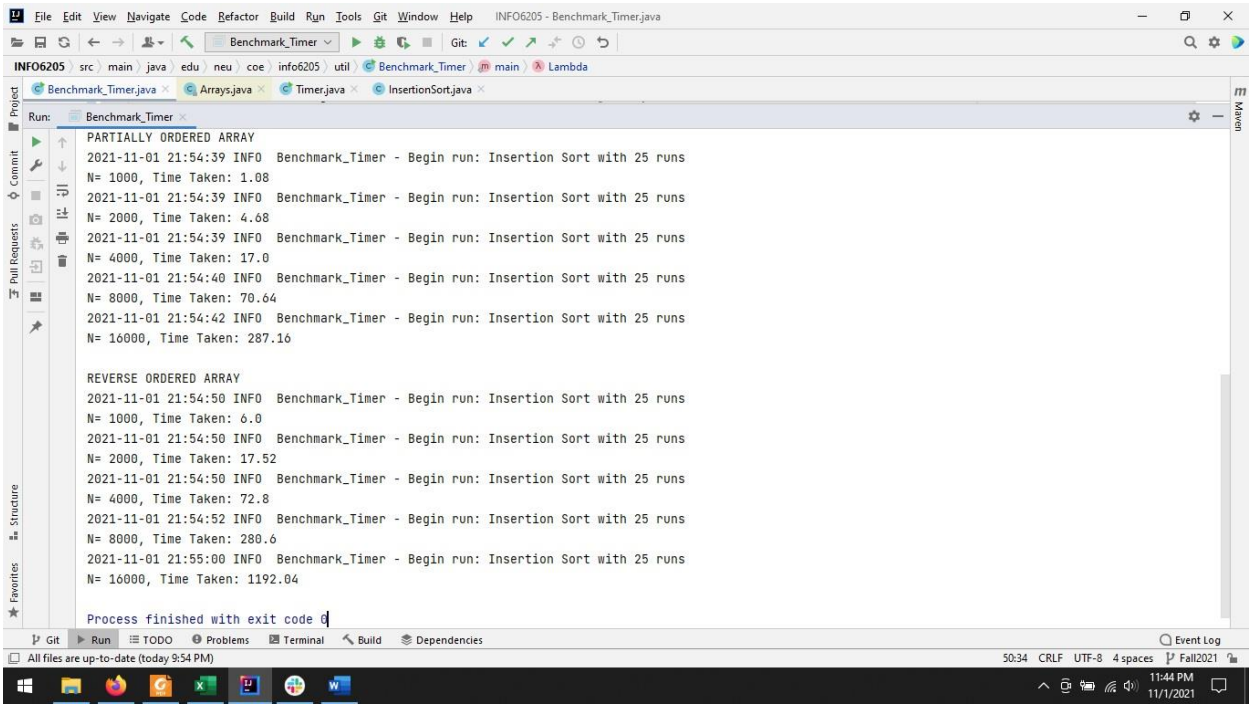


- **Test Cases for InsertionSortTest:**



- **Output:**


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
INFO6205 - Benchmark_Timer.java
Benchmark_Timer
INFO6205 src main java edu neu coe info6205 util Benchmark_Timer main Lambda
Benchmark_Timer.java Arrays.java Timer.java InsertionSort.java
Run: Benchmark_Timer
C:\Users\User\.jdks\openjdk-16.0.2\bin\java.exe ...
RANDOMLY ORDERED ARRAY
2021-11-01 21:54:18 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 1000, Time Taken: 3.72
2021-11-01 21:54:18 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 2000, Time Taken: 9.56
2021-11-01 21:54:18 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 4000, Time Taken: 37.76
2021-11-01 21:54:19 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 8000, Time Taken: 133.88
2021-11-01 21:54:23 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 16000, Time Taken: 587.16
ORDERED ARRAY
2021-11-01 21:54:39 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 1000, Time Taken: 0.0
2021-11-01 21:54:39 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 2000, Time Taken: 0.0
2021-11-01 21:54:39 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 4000, Time Taken: 0.0
2021-11-01 21:54:39 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 8000, Time Taken: 0.04
2021-11-01 21:54:39 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 16000, Time Taken: 0.08
Git Run TODO Problems Terminal Build Dependencies
Event Log
All files are up-to-date (today 9:54 PM)
50:34 CRLF UTF-8 4 spaces Fall2021
11:45 PM
11/1/2021


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
INFO6205 - Benchmark_Timer.java
Benchmark_Timer
INFO6205 src main java edu neu coe info6205 util Benchmark_Timer main Lambda
Benchmark_Timer.java Arrays.java Timer.java InsertionSort.java
Run: Benchmark_Timer
PARTIALLY ORDERED ARRAY
2021-11-01 21:54:39 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 1000, Time Taken: 1.08
2021-11-01 21:54:39 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 2000, Time Taken: 4.68
2021-11-01 21:54:39 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 4000, Time Taken: 17.0
2021-11-01 21:54:40 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 8000, Time Taken: 70.64
2021-11-01 21:54:42 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 16000, Time Taken: 287.16
REVERSE ORDERED ARRAY
2021-11-01 21:54:50 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 1000, Time Taken: 6.0
2021-11-01 21:54:50 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 2000, Time Taken: 17.52
2021-11-01 21:54:50 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 4000, Time Taken: 72.8
2021-11-01 21:54:52 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 8000, Time Taken: 280.6
2021-11-01 21:55:00 INFO Benchmark_Timer - Begin run: Insertion Sort with 25 runs
N= 16000, Time Taken: 1192.04
Process finished with exit code 0
Git Run TODO Problems Terminal Build Dependencies
Event Log
All files are up-to-date (today 9:54 PM)
50:34 CRLF UTF-8 4 spaces Fall2021
11:44 PM
11/1/2021

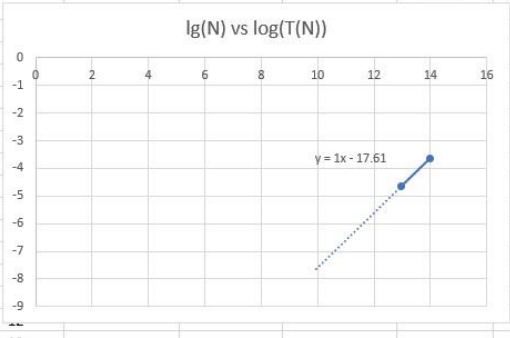
- **Relationship Conclusion:**

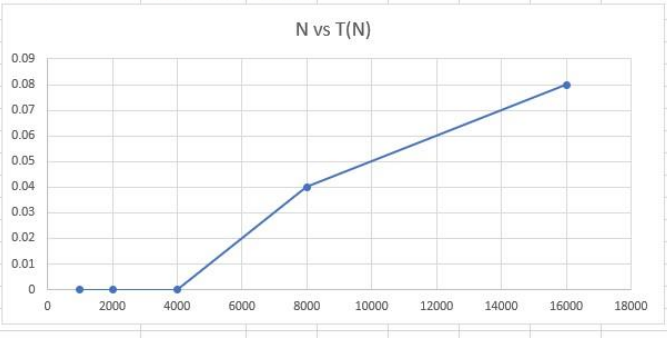
After performing tests on different types of arrays like: Sorted Array, Partially Sorted Array, Randomly Sorted Array and Reverse Ordered Array we can say that Insertion Sort takes the least time for sorting an already Sorted Array followed by Partially Sorted Array, which is followed by Randomly Sorted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

- **Evidence:**

For Sorted Array

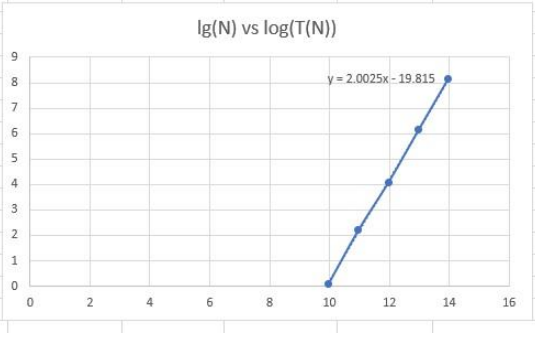
Array Size (N)	Sorted Array T(N)	lg(Array Size)	lg(Time)
1000	0	9.965784285	
2000	0	10.96578428	
4000	0	11.96578428	
8000	0.04	12.96578428	-4.64386
16000	0.08	13.96578428	-3.64386

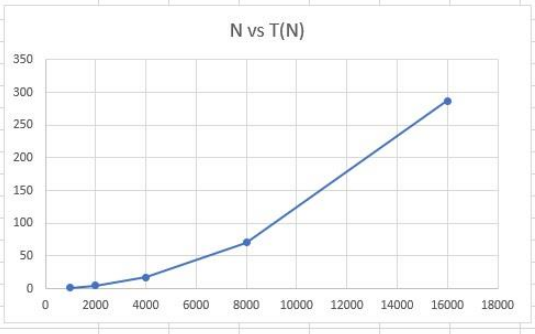




For Partially Sorted Array

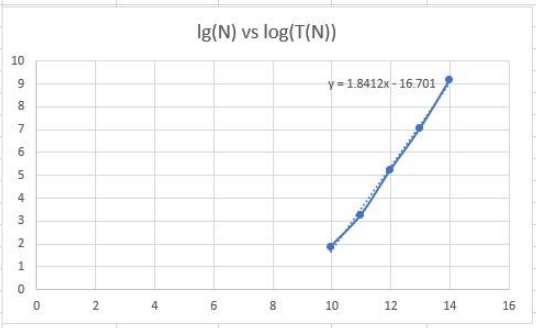
Array Size (N)	Partially Ordered T(N)	lg(Array Size)	lg(Time)
1000	1.08	9.965784285	0.111031
2000	4.68	10.96578428	2.226509
4000	17	11.96578428	4.087463
8000	70.64	12.96578428	6.142413
16000	287.16	13.96578428	8.165711

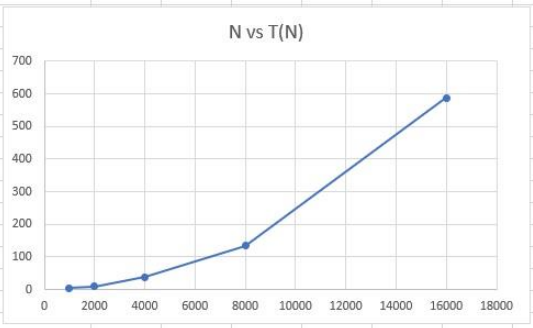




Randomly Sorted Array

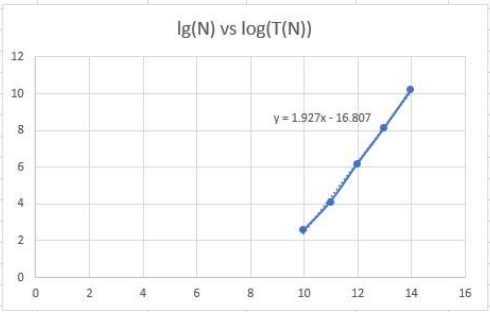
Array Size (N)	Randomly Ordered T(N)	lg(Array Size)	lg(Time)
1000	3.72	9.965784285	1.895303
2000	9.56	10.96578428	3.257011
4000	37.76	11.96578428	5.238787
8000	133.88	12.96578428	7.064797
16000	587.16	13.96578428	9.19761

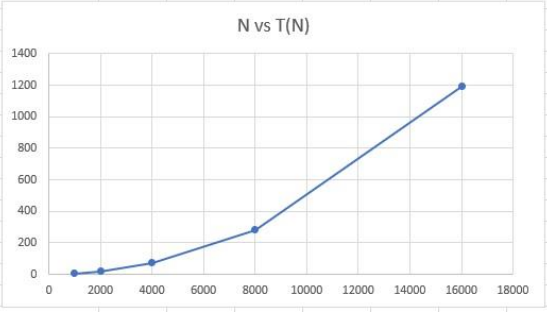




For Reverse Ordered Array

Array Size (N)	Reversed Order T(N)	lg(Array Size)	lg(Time)
1000	6	9.965784285	2.584963
2000	17.52	10.96578428	4.130931
4000	72.8	11.96578428	6.185867
8000	280.6	12.96578428	8.132371
16000	1192.04	13.96578428	10.21922





We can see that for Sorted Array the order of growth Time $\propto$ n and for everyone else the order of growth Time $\propto n^2$. 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.