Venkata Srinivas Kompally (NUID : 002137855)

Program Structures & Algorithms

Fall 2021

Assignment No. 2

- **Task:**

  Part 1) Implement three methods of class called Timer. Timer is invoked from a class called Benchmark_Timer which implements the Benchmark interface.

  Part 2) Implement InsertionSort (in the InsertionSort class) by simply looking up the insertion code used by Arrays.sort.

  Part 3) Implement a main program (or you could do it via your own unit tests) 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.

- **Relationship Conclusion:**

  The Order of Growth for Randomly Ordered Array of Size N is $\approx N^{1.36}$

  The Order of Growth for Ordered Array of Size N is $\approx N^{0.80}$

The Order of Growth for Reverse Ordered Array of Size N is ≈ $N^{1.53}$

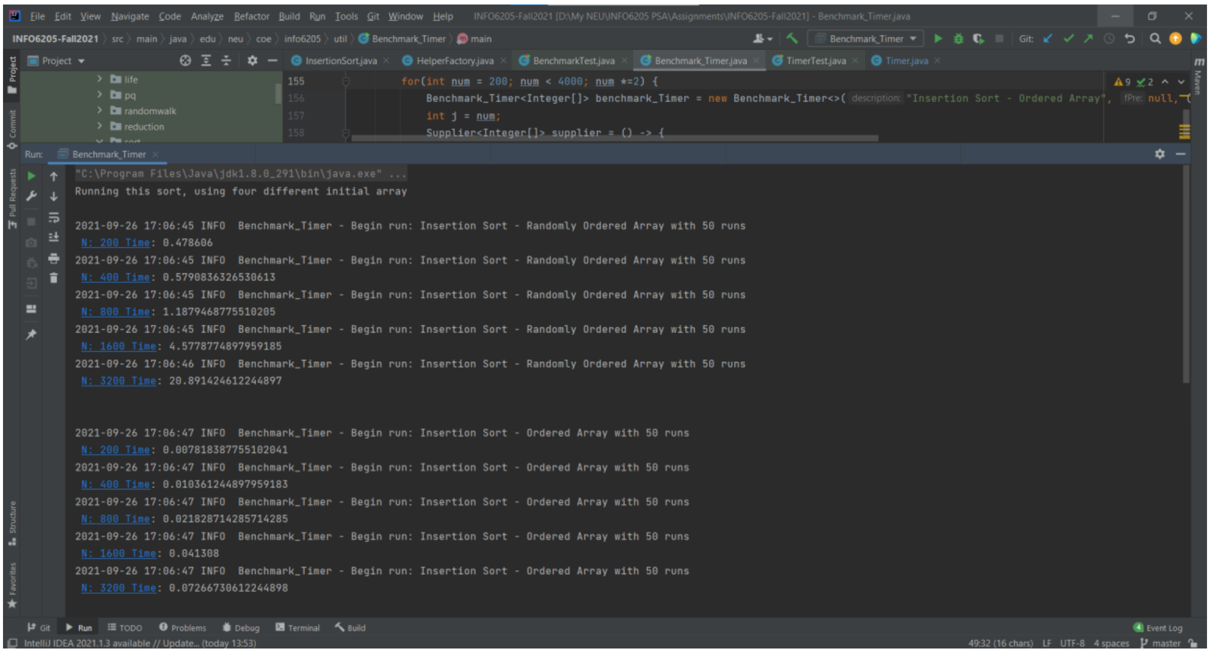
The Order of Growth for Partially Ordered Array of Size N is ≈ $N^{1.09}$

The Order of growth, based on running time of insertion sort is:

Ordered < Partially Ordered < Randomly Ordered < Reverse Ordered

- ⊙ **Evidence to support the conclusion:**

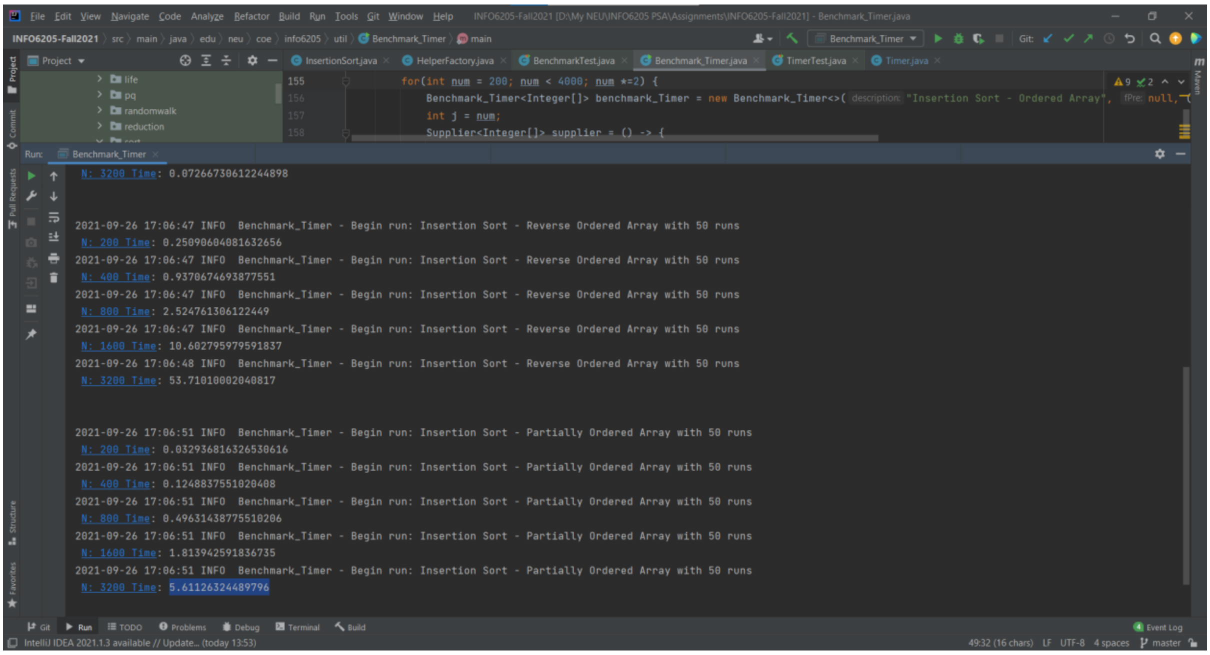
1. **Output (Snapshot of Code output in the terminal)**



File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help
INFO6205-Fall2021 [D:\My NEU\INFO6205 PSA\Assignments\INFO6205-Fall2021] - Benchmark_Timer.java
INFO6205-Fall2021 › src › main › java › edu › neu › coe › info6205 › util › Benchmark_Timer › main
Project
InsertionSort.java
HelperFactory.java
BenchmarkTest.java
Benchmark_Timer.java
TimerTest.java
Timer.java
life
pq
randomwalk
reduction
Run: Benchmark_Timer
Git
Run
TODO
Problems
Debug
Terminal
Build
IntelliJ IDEA 2021.1.3 available // Update... (today 13:53)
Event Log
49:32 (16 chars) LF UTF-8 4 spaces master

```
for(int num = 200; num < 4000; num *=2) {
    Benchmark_Timer<Integer[]> benchmark_Timer = new Benchmark_Timer<>( description: "Insertion Sort - Ordered Array", fPre: null,
    int j = num;
    Supplier<Integer[]> supplier = () -> {
```

```
"C:\Program Files\Java\jdk1.8.0_291\bin\java.exe" ...
Running this sort, using four different initial array

2021-09-26 17:06:45 INFO  Benchmark_Timer - Begin run: Insertion Sort - Randomly Ordered Array with 50 runs
 N: 200 Time: 0.478606
2021-09-26 17:06:45 INFO  Benchmark_Timer - Begin run: Insertion Sort - Randomly Ordered Array with 50 runs
 N: 400 Time: 0.5790836326530613
2021-09-26 17:06:45 INFO  Benchmark_Timer - Begin run: Insertion Sort - Randomly Ordered Array with 50 runs
 N: 800 Time: 1.1879468775510205
2021-09-26 17:06:45 INFO  Benchmark_Timer - Begin run: Insertion Sort - Randomly Ordered Array with 50 runs
 N: 1600 Time: 4.5778774897959185
2021-09-26 17:06:46 INFO  Benchmark_Timer - Begin run: Insertion Sort - Randomly Ordered Array with 50 runs
 N: 3200 Time: 20.891424612244897


2021-09-26 17:06:47 INFO  Benchmark_Timer - Begin run: Insertion Sort - Ordered Array with 50 runs
 N: 200 Time: 0.007818387755102041
2021-09-26 17:06:47 INFO  Benchmark_Timer - Begin run: Insertion Sort - Ordered Array with 50 runs
 N: 400 Time: 0.010361244897959183
2021-09-26 17:06:47 INFO  Benchmark_Timer - Begin run: Insertion Sort - Ordered Array with 50 runs
 N: 800 Time: 0.021828714285714285
2021-09-26 17:06:47 INFO  Benchmark_Timer - Begin run: Insertion Sort - Ordered Array with 50 runs
 N: 1600 Time: 0.041308
2021-09-26 17:06:47 INFO  Benchmark_Timer - Begin run: Insertion Sort - Ordered Array with 50 runs
 N: 3200 Time: 0.07266730612244898
```



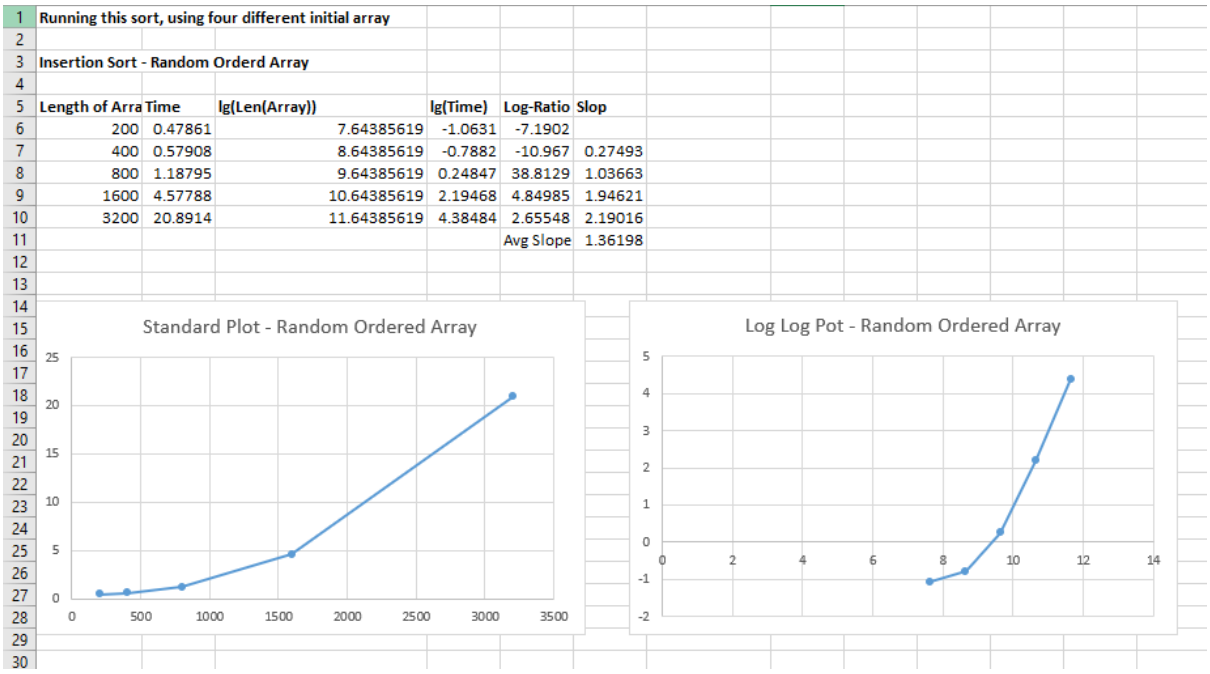
File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help
INFO6205-Fall2021 [D:\My NEU\INFO6205 PSA\Assignments\INFO6205-Fall2021] - Benchmark_Timer.java
INFO6205-Fall2021 src main java edu neu coe info6205 util Benchmark_Timer main
Benchmark_Timer
Git:
Project
life
pq
randomwalk
reduction
InsertionSort.java
HelperFactory.java
BenchmarkTest.java
Benchmark_Timer.java
TimerTest.java
Timer.java
Run:
Benchmark_Timer
Git
Run
TODO
Problems
Debug
Terminal
Build
Event Log
IntelliJ IDEA 2021.1.3 available // Update... (today 13:53)
49:32 (16 chars) LF UTF-8 4 spaces master

```
        for(int num = 200; num < 4000; num *=2) {
            Benchmark_Timer<Integer[]> benchmark_Timer = new Benchmark_Timer<>( description: "Insertion Sort - Ordered Array", fPre: null,
            int j = num;
            Supplier<Integer[]> supplier = () -> {
```

```
 N: 3200 Time: 0.07266730612244898


2021-09-26 17:06:47 INFO  Benchmark_Timer - Begin run: Insertion Sort - Reverse Ordered Array with 50 runs
 N: 200 Time: 0.25090604081632656
2021-09-26 17:06:47 INFO  Benchmark_Timer - Begin run: Insertion Sort - Reverse Ordered Array with 50 runs
 N: 400 Time: 0.9370674693877551
2021-09-26 17:06:47 INFO  Benchmark_Timer - Begin run: Insertion Sort - Reverse Ordered Array with 50 runs
 N: 800 Time: 2.524761306122449
2021-09-26 17:06:47 INFO  Benchmark_Timer - Begin run: Insertion Sort - Reverse Ordered Array with 50 runs
 N: 1600 Time: 10.602795979591837
2021-09-26 17:06:48 INFO  Benchmark_Timer - Begin run: Insertion Sort - Reverse Ordered Array with 50 runs
 N: 3200 Time: 53.71010002040817


2021-09-26 17:06:51 INFO  Benchmark_Timer - Begin run: Insertion Sort - Partially Ordered Array with 50 runs
 N: 200 Time: 0.032936816326530616
2021-09-26 17:06:51 INFO  Benchmark_Timer - Begin run: Insertion Sort - Partially Ordered Array with 50 runs
 N: 400 Time: 0.1248837551020408
2021-09-26 17:06:51 INFO  Benchmark_Timer - Begin run: Insertion Sort - Partially Ordered Array with 50 runs
 N: 800 Time: 0.49631438775510206
2021-09-26 17:06:51 INFO  Benchmark_Timer - Begin run: Insertion Sort - Partially Ordered Array with 50 runs
 N: 1600 Time: 1.813942591836735
2021-09-26 17:06:51 INFO  Benchmark_Timer - Begin run: Insertion Sort - Partially Ordered Array with 50 runs
 N: 3200 Time: 5.61126324489796
```

2. Graphical Representation

Random Ordered Array:



Running this sort, using four different initial array

Insertion Sort - Random Orderd Array

Length of Arra	Time	lg(Len(Array))	lg(Time)	Log-Ratio	Slop
200	0.47861	7.64385619	-1.0631	-7.1902	
400	0.57908	8.64385619	-0.7882	-10.967	0.27493
800	1.18795	9.64385619	0.24847	38.8129	1.03663
1600	4.57788	10.64385619	2.19468	4.84985	1.94621
3200	20.8914	11.64385619	4.38484	2.65548	2.19016
				Avg Slope	1.36198

Standard Plot - Random Ordered Array
25
20
15
10
5
0
0
500
1000
1500
2000
2500
3000
3500
Log Log Pot - Random Ordered Array
5
4
3
2
1
0
-1
-2
0
2
4
6
8
10
12
14

Initial input size is 200 and increased upto 3200. Calculated the running time of the soring algorithm Random Orderly.

Using Doubling Hypothesis:

For Function $T(N) = aN\text{^}b$

N = Input Size

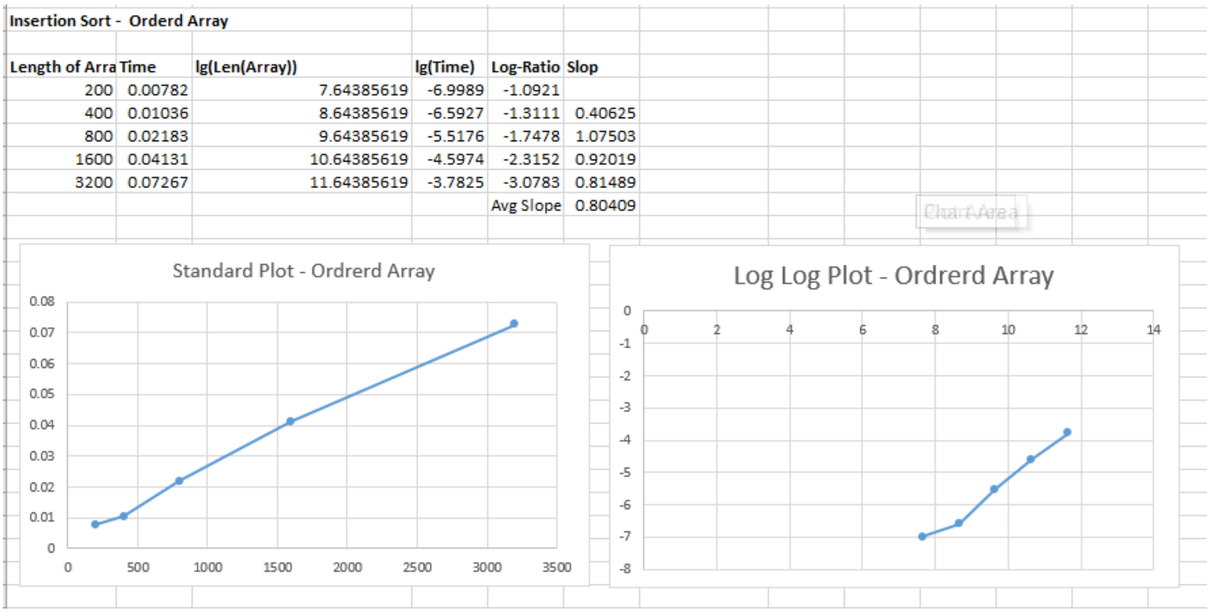
a = Constant

b = slope of the log-log graph

The average slope for Randomly Ordered Array is: 1.36

The Equation of such a line is:

llg$\left(T(N)\right) = \ aN^{1.36}$

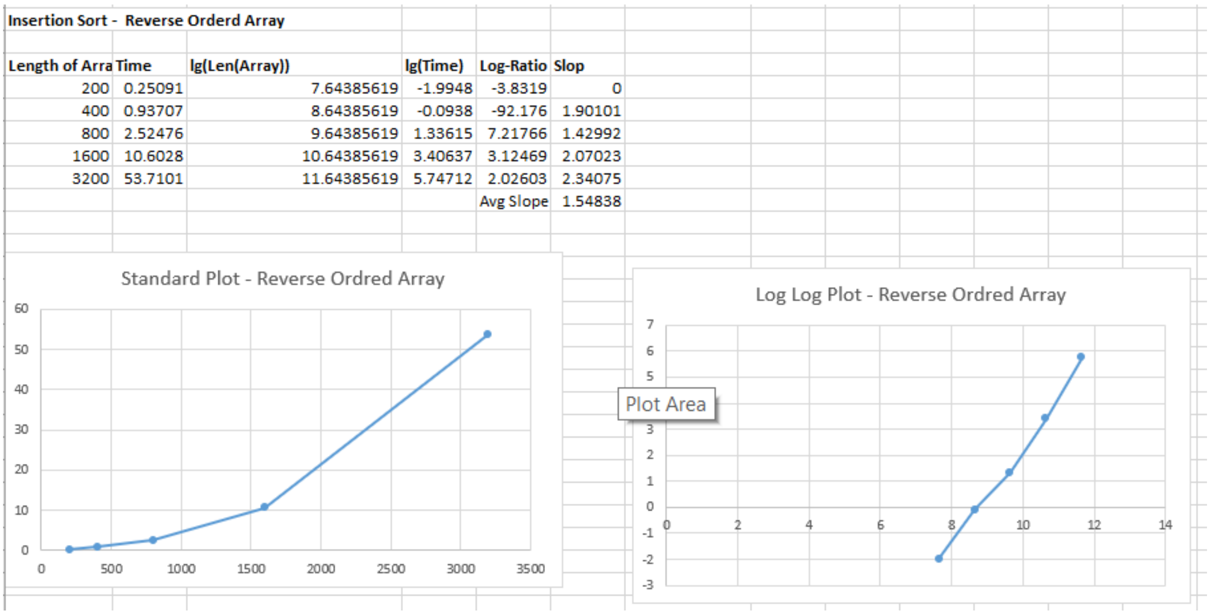


Insertion Sort - Orderd Array					
Length of Arra	Time	lg(Len(Array))	lg(Time)	Log-Ratio	Slop
200	0.00782	7.64385619	-6.9989	-1.0921	
400	0.01036	8.64385619	-6.5927	-1.3111	0.40625
800	0.02183	9.64385619	-5.5176	-1.7478	1.07503
1600	0.04131	10.64385619	-4.5974	-2.3152	0.92019
3200	0.07267	11.64385619	-3.7825	-3.0783	0.81489
				Avg Slope	0.80409

The average slope for Ordered Array is: 0.80

The Equation of such a line is:

llg$\left(T(N)\right) = \ aN^{0.80}$



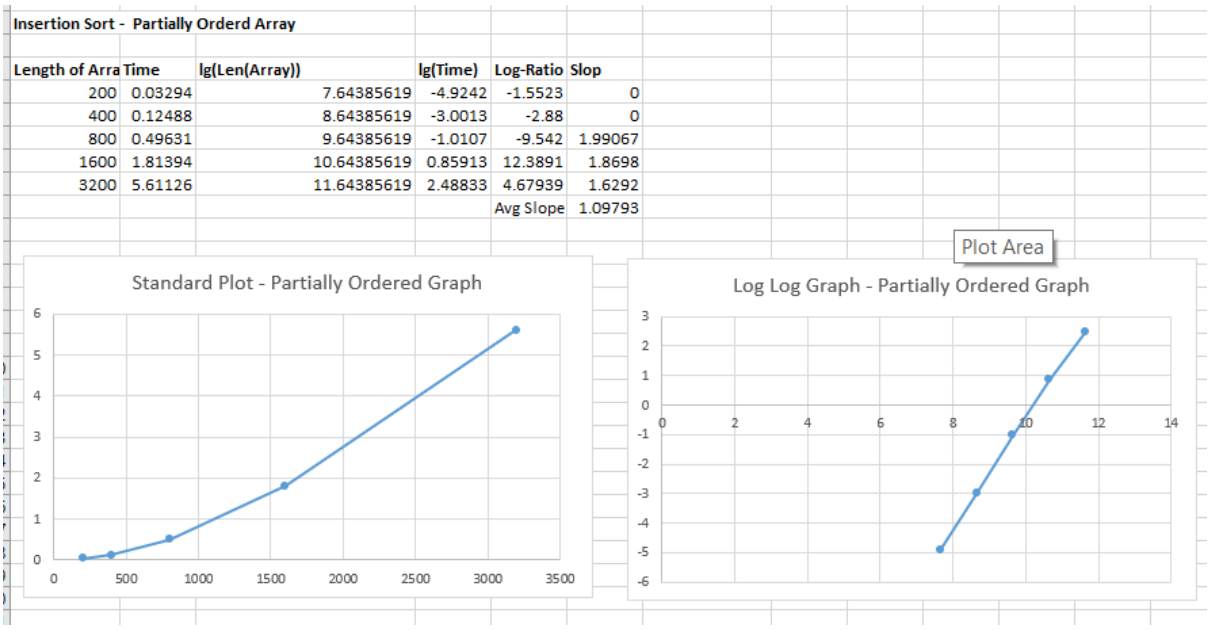
Insertion Sort - Reverse Orderd Array

Length of Arra	Time	lg(Len(Array))	lg(Time)	Log-Ratio	Slop
200	0.25091	7.64385619	-1.9948	-3.8319	0
400	0.93707	8.64385619	-0.0938	-92.176	1.90101
800	2.52476	9.64385619	1.33615	7.21766	1.42992
1600	10.6028	10.64385619	3.40637	3.12469	2.07023
3200	53.7101	11.64385619	5.74712	2.02603	2.34075
				Avg Slope	1.54838

The average slope for Reverse Ordered Array is: 1.54

The Equation of such a line is:

$$\mathrm{llg}\left(T(N)\right) = \ aN^{1.54}$$



Insertion Sort - Partially Orderd Array

Length of Arra	Time	lg(Len(Array))	lg(Time)	Log-Ratio	Slop
200	0.03294	7.64385619	-4.9242	-1.5523	0
400	0.12488	8.64385619	-3.0013	-2.88	0
800	0.49631	9.64385619	-1.0107	-9.542	1.99067
1600	1.81394	10.64385619	0.85913	12.3891	1.8698
3200	5.61126	11.64385619	2.48833	4.67939	1.6292
				Avg Slope	1.09793

Plot Area
Standard Plot - Partially Ordered Graph
6
5
4
3
2
1
0
0
500
1000
1500
2000
2500
3000
3500
Log Log Graph - Partially Ordered Graph
3
2
1
0
-1
-2
-3
-4
-5
-6
0
2
4
6
8
10
12
14

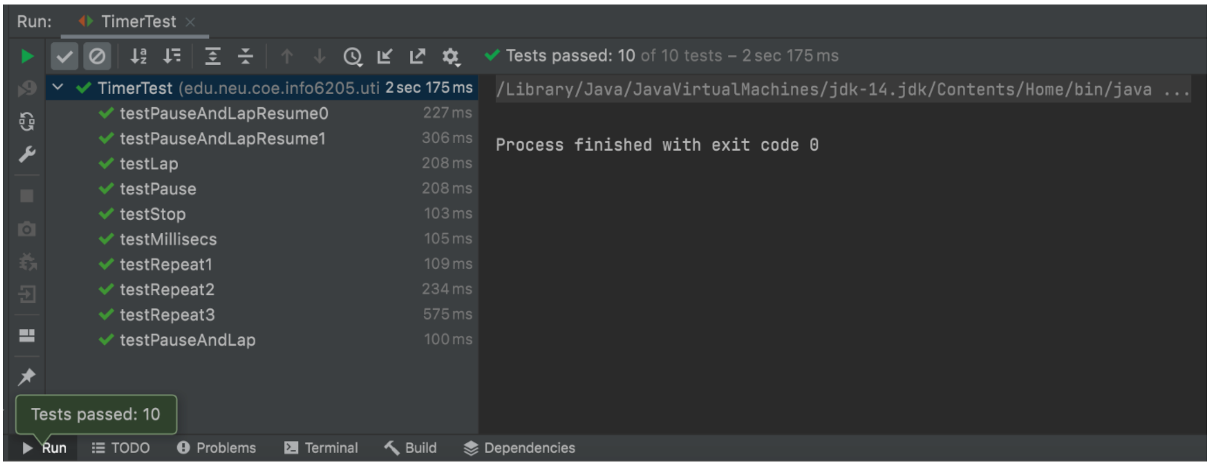
The average slope for Partially Ordered Array is: 1.09

The Equation of such a line is:

$$\mathrm{llg}\left(T(N)\right) = \ aN^{1.09}$$

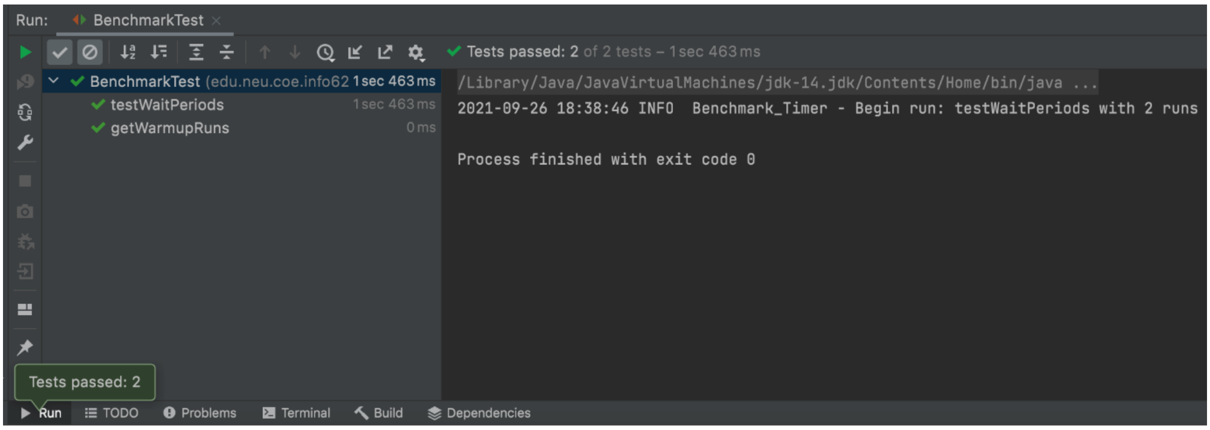
- **Unit tests result:**

Part 1 : Timer Test



```
✓ Tests passed: 10 of 10 tests – 2 sec 175 ms

✓ TimerTest (edu.neu.coe.info6205.uti 2 sec 175 ms
    ✓ testPauseAndLapResume0      227 ms
    ✓ testPauseAndLapResume1      306 ms
    ✓ testLap                     208 ms
    ✓ testPause                   208 ms
    ✓ testStop                    103 ms
    ✓ testMillisecs               105 ms
    ✓ testRepeat1                 109 ms
    ✓ testRepeat2                 234 ms
    ✓ testRepeat3                 575 ms
    ✓ testPauseAndLap             100 ms

Tests passed: 10
```

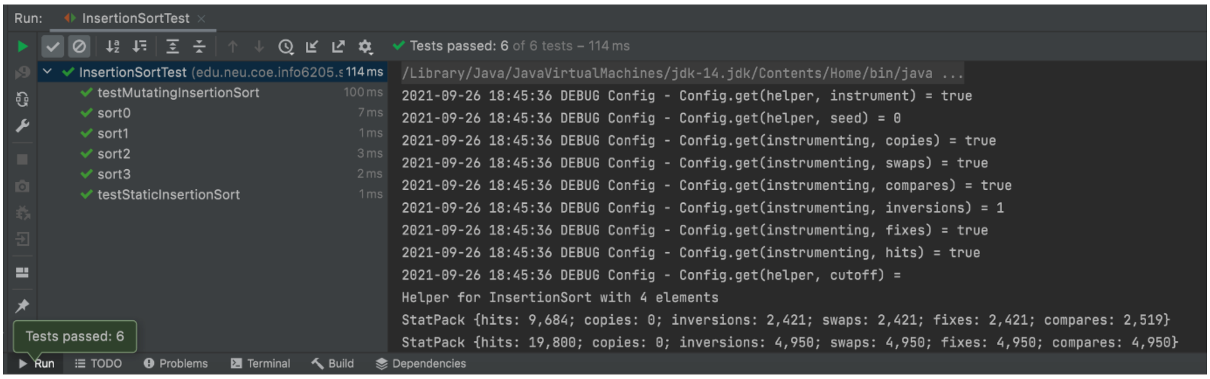
```
/Library/Java/JavaVirtualMachines/jdk-14.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```



```
✓ Tests passed: 2 of 2 tests – 1 sec 463 ms

✓ BenchmarkTest (edu.neu.coe.info62 1 sec 463 ms
    ✓ testWaitPeriods     1 sec 463 ms
    ✓ getWarmupRuns               0 ms

Tests passed: 2
```

```
/Library/Java/JavaVirtualMachines/jdk-14.jdk/Contents/Home/bin/java ...
2021-09-26 18:38:46 INFO  Benchmark_Timer - Begin run: testWaitPeriods with 2 runs

Process finished with exit code 0
```

Part 2: Insertion Sort



```
✓ Tests passed: 6 of 6 tests – 114 ms

✓ InsertionSortTest (edu.neu.coe.info6205.s 114 ms
    ✓ testMutatingInsertionSort   100 ms
    ✓ sort0                         7 ms
    ✓ sort1                         1 ms
    ✓ sort2                         3 ms
    ✓ sort3                         2 ms
    ✓ testStaticInsertionSort       1 ms

Tests passed: 6
```

```
/Library/Java/JavaVirtualMachines/jdk-14.jdk/Contents/Home/bin/java ...
2021-09-26 18:45:36 DEBUG Config - Config.get(helper, instrument) = true
2021-09-26 18:45:36 DEBUG Config - Config.get(helper, seed) = 0
2021-09-26 18:45:36 DEBUG Config - Config.get(instrumenting, copies) = true
2021-09-26 18:45:36 DEBUG Config - Config.get(instrumenting, swaps) = true
2021-09-26 18:45:36 DEBUG Config - Config.get(instrumenting, compares) = true
2021-09-26 18:45:36 DEBUG Config - Config.get(instrumenting, inversions) = 1
2021-09-26 18:45:36 DEBUG Config - Config.get(instrumenting, fixes) = true
2021-09-26 18:45:36 DEBUG Config - Config.get(instrumenting, hits) = true
2021-09-26 18:45:36 DEBUG Config - Config.get(helper, cutoff) =
Helper for InsertionSort with 4 elements
StatPack {hits: 9,684; copies: 0; inversions: 2,421; swaps: 2,421; fixes: 2,421; compares: 2,519}
StatPack {hits: 19,800; copies: 0; inversions: 4,950; swaps: 4,950; fixes: 4,950; compares: 4,950}
```