Rahul Bhiwande(001040583)
Program Structures & Algorithms
Spring 2021
Assignment No.2

1) Task:

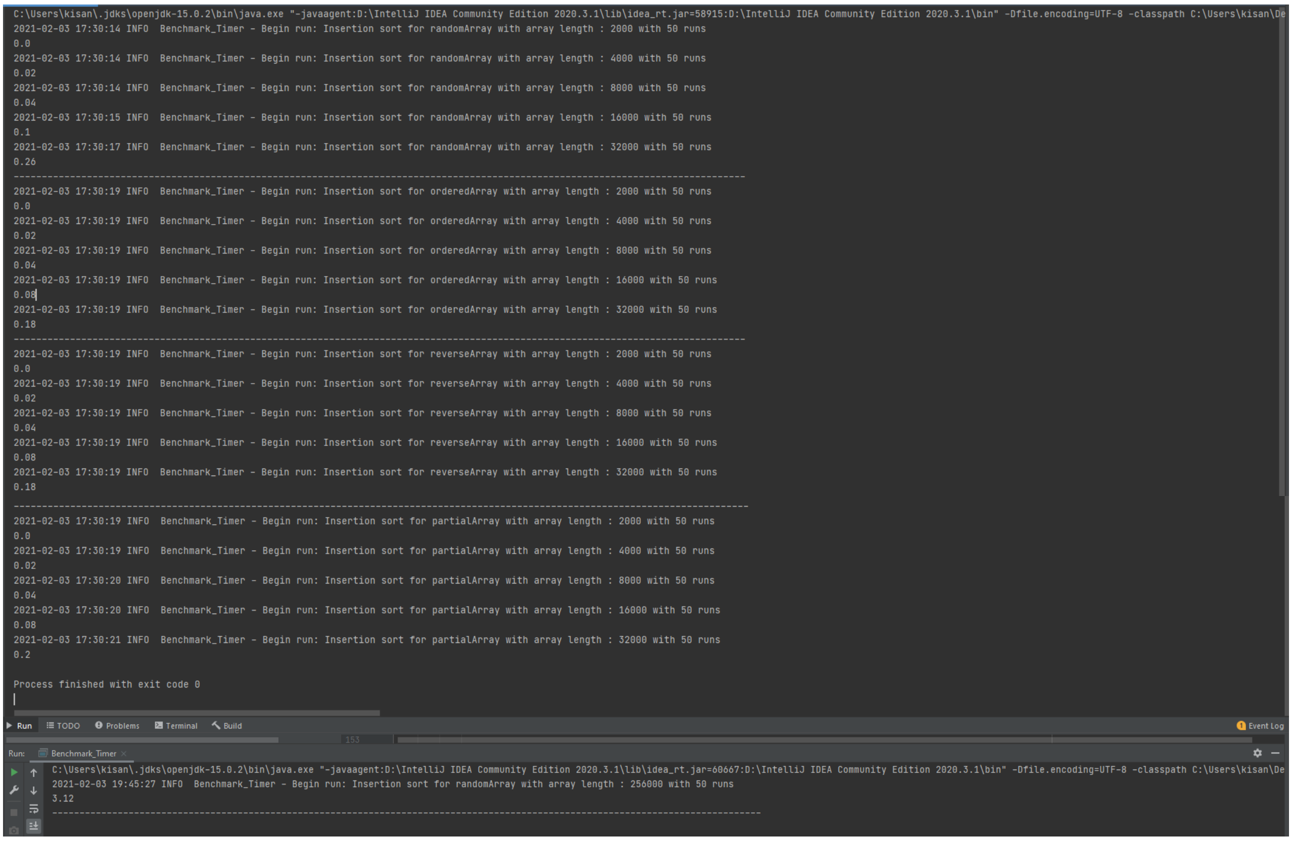
Your task for this assignment is in three parts.

(Part 1) You are to implement three methods of a class called Timer. Please see the skeleton class that I created in the repository. Timer is invoked from a class called *Benchmark_Timer* which implements the *Benchmark* interface. The function to be timed, hereinafter the "target" function, is the *Consumer* function *fRun* (or just *f*) passed in to one or other of the constructors. For example, you might create a function which sorts an array with *n* elements. The generic type *T* is that of the input to the target function.The first parameter to the first run method signature is the parameter that will, in turn, be passed to target function. In the second signature, *supplier* will be invoked each time to get a *t* which is passed to the other run method.The second parameter to the *run* function (*m)* is the number of times the target function will be called.The return value from *run* is the average number of milliseconds taken for each run of the target function.Don't forget to check your implementation by running the unit tests in *BenchmarkTest* and*TimerTest*.

(Part 2) Implement *InsertionSort* (in the *InsertionSort* class) by simply looking up the insertion code used by *Arrays.sort.* You should use the *helper.swap* method although you could also just copy that from the same source code. You should of course run the unit tests in *InsertionSortTest*.

(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. I suggest that your arrays to be sorted are of type *Integer*. Use the doubling method for choosing *n* and test for at least five values of *n.* Draw any conclusions from your observations regarding the order of growth.

2) Output:



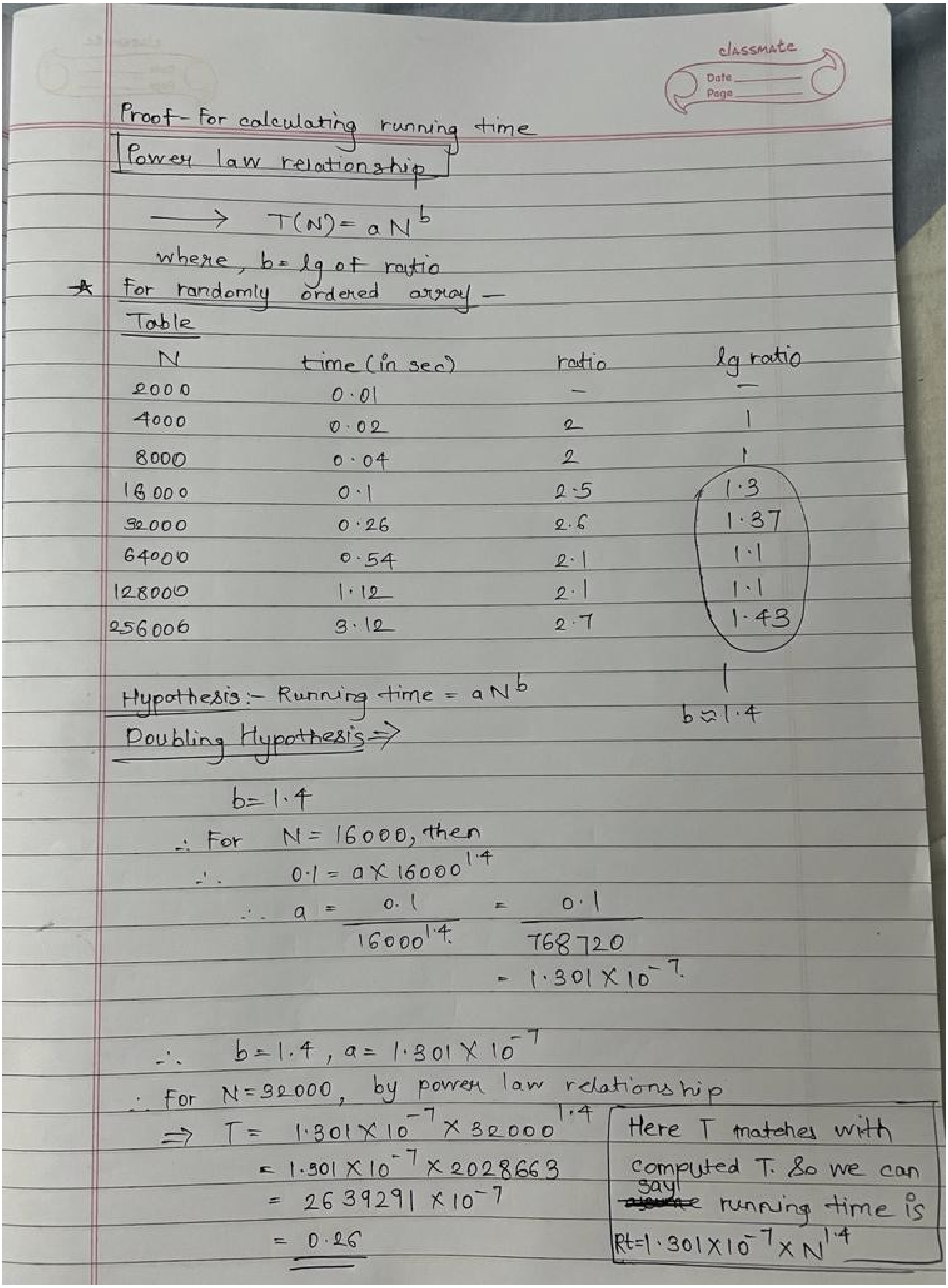
```
C:\Users\kisan\.jdks\openjdk-15.0.2\bin\java.exe "-javaagent:D:\IntelliJ IDEA Community Edition 2020.3.1\lib\idea_rt.jar=58915:D:\IntelliJ IDEA Community Edition 2020.3.1\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\kisan\De
2021-02-03 17:30:14 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 2000 with 50 runs
0.0
2021-02-03 17:30:14 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 4000 with 50 runs
0.02
2021-02-03 17:30:14 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 8000 with 50 runs
0.04
2021-02-03 17:30:15 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 16000 with 50 runs
0.1
2021-02-03 17:30:17 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 32000 with 50 runs
0.26
-------------------------------------------------------------------------------------------------------------------------------
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 2000 with 50 runs
0.0
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 4000 with 50 runs
0.02
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 8000 with 50 runs
0.04
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 16000 with 50 runs
0.08
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for orderedArray with array length : 32000 with 50 runs
0.18
-------------------------------------------------------------------------------------------------------------------------------
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 2000 with 50 runs
0.0
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 4000 with 50 runs
0.02
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 8000 with 50 runs
0.04
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 16000 with 50 runs
0.08
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for reverseArray with array length : 32000 with 50 runs
0.18
-------------------------------------------------------------------------------------------------------------------------------
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for partialArray with array length : 2000 with 50 runs
0.0
2021-02-03 17:30:19 INFO  Benchmark_Timer - Begin run: Insertion sort for partialArray with array length : 4000 with 50 runs
0.02
2021-02-03 17:30:20 INFO  Benchmark_Timer - Begin run: Insertion sort for partialArray with array length : 8000 with 50 runs
0.04
2021-02-03 17:30:20 INFO  Benchmark_Timer - Begin run: Insertion sort for partialArray with array length : 16000 with 50 runs
0.08
2021-02-03 17:30:21 INFO  Benchmark_Timer - Begin run: Insertion sort for partialArray with array length : 32000 with 50 runs
0.2

Process finished with exit code 0
```

```
C:\Users\kisan\.jdks\openjdk-15.0.2\bin\java.exe "-javaagent:D:\IntelliJ IDEA Community Edition 2020.3.1\lib\idea_rt.jar=60667:D:\IntelliJ IDEA Community Edition 2020.3.1\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\kisan\De
2021-02-03 19:45:27 INFO  Benchmark_Timer - Begin run: Insertion sort for randomArray with array length : 256000 with 50 runs
3.12
-------------------------------------------------------------------------------------------------------------------------------
```

3) Relationship conclusion:

- **The order of growth: N$^{1.4}$**

4) Evidence to support the conclusion:



Proof- For calculating running time

$\boxed{\text{Power law relationship}}$

$\longrightarrow \quad T(N) = aN^b$

where, b = lg of ratio

★ For randomly ordered array —

Table

N	time (in sec)	ratio	lg ratio
2000	0.01	–	–
4000	0.02	2	1
8000	0.04	2	1
16000	0.1	2.5	1.3
32000	0.26	2.6	1.37
64000	0.54	2.1	1.1
128000	1.12	2.1	1.1
256000	3.12	2.7	1.43

$b \simeq 1.4$

Hypothesis:- Running time = aN^b

Doubling Hypothesis ⇒

$b = 1.4$

∴ For $N = 16000$, then

$\therefore \quad 0.1 = a \times 16000^{1.4}$

$\therefore \quad a = \dfrac{0.1}{16000^{1.4}} = \dfrac{0.1}{768720}$

$= 1.301 \times 10^{-7}.$

$\therefore \quad b = 1.4, \ a = 1.301 \times 10^{-7}$

∴ For $N = 32000$, by power law relationship

$\Rightarrow T = 1.301 \times 10^{-7} \times 32000^{1.4}$

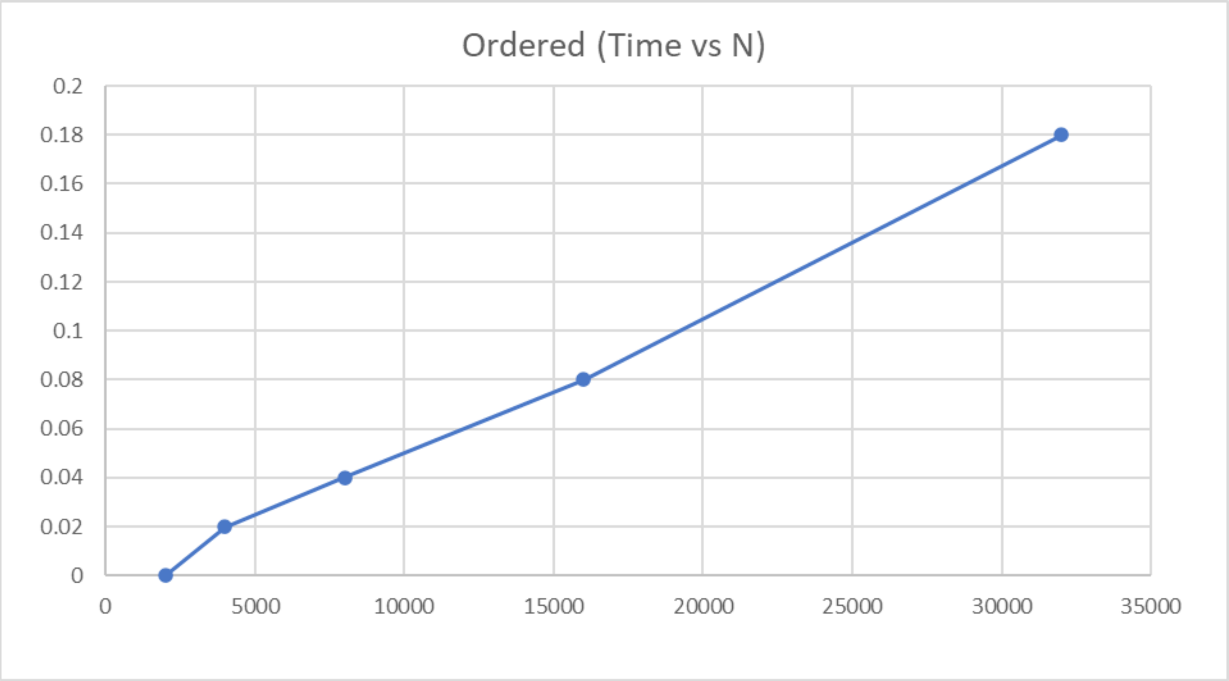
$= 1.301 \times 10^{-7} \times 2028663$

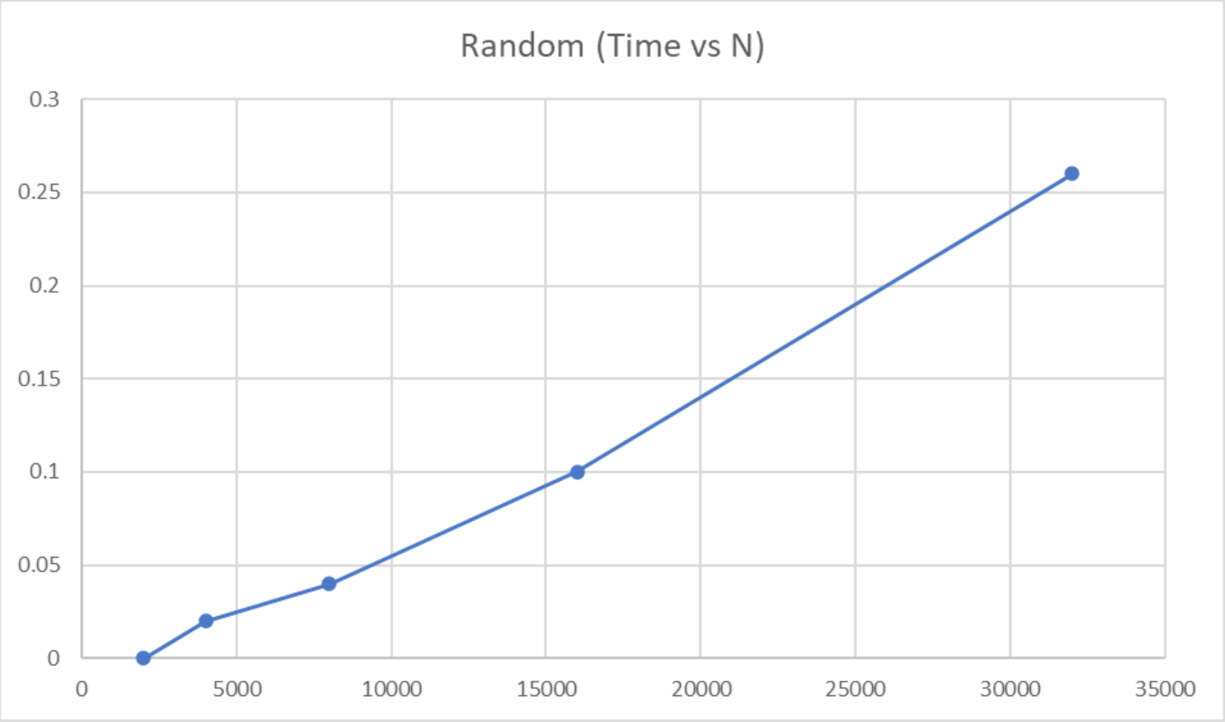
$= 2639291 \times 10^{-7}$

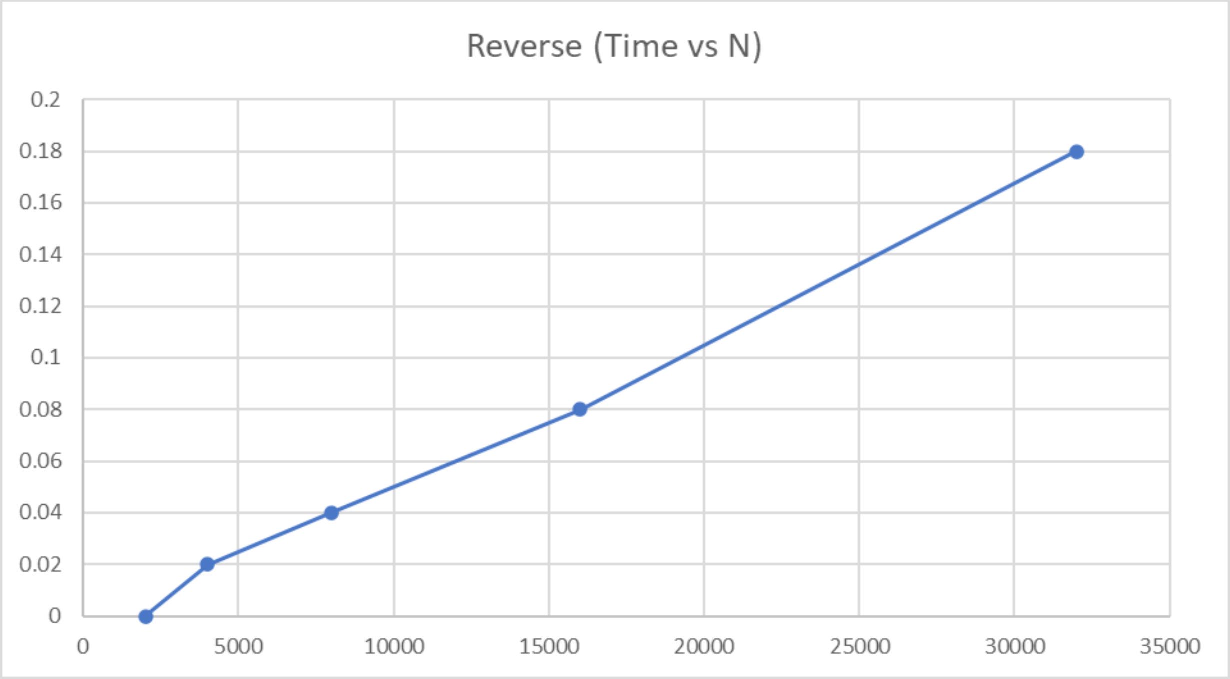
$= \underline{0.26}$

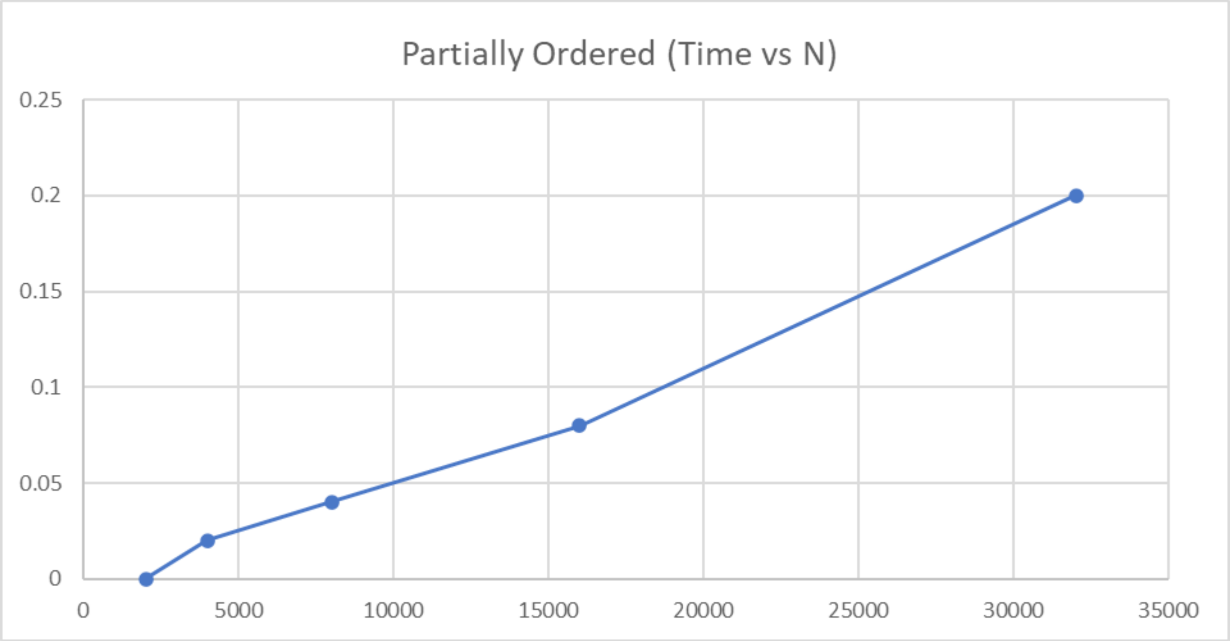
Here T matches with computed T. So we can say running time is
$Rt = 1.301 \times 10^{-7} \times N^{1.4}$

5) Graphical Evidence:



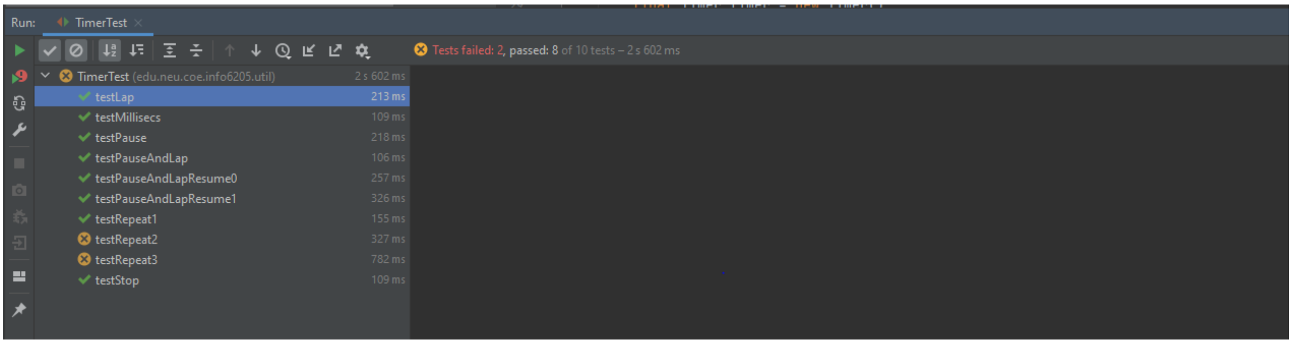



Reverse (Time vs N)
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
5000
10000
15000
20000
25000
30000
35000


Partially Ordered (Time vs N)
0.25
0.2
0.15
0.1
0.05
0
0
5000
10000
15000
20000
25000
30000
35000

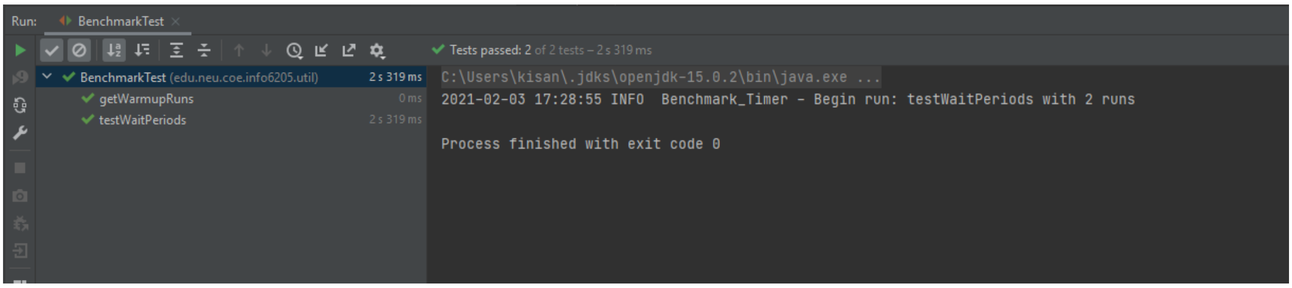
6) Unit tests screenshots:

Timer Test:



```
Tests failed: 2, passed: 8 of 10 tests – 2 s 602 ms

TimerTest (edu.neu.coe.info6205.util)    2 s 602 ms
    testLap                              213 ms
    testMillisecs                        109 ms
    testPause                            218 ms
    testPauseAndLap                      106 ms
    testPauseAndLapResume0               257 ms
    testPauseAndLapResume1               326 ms
    testRepeat1                          155 ms
    testRepeat2                          327 ms
    testRepeat3                          782 ms
    testStop                             109 ms
```

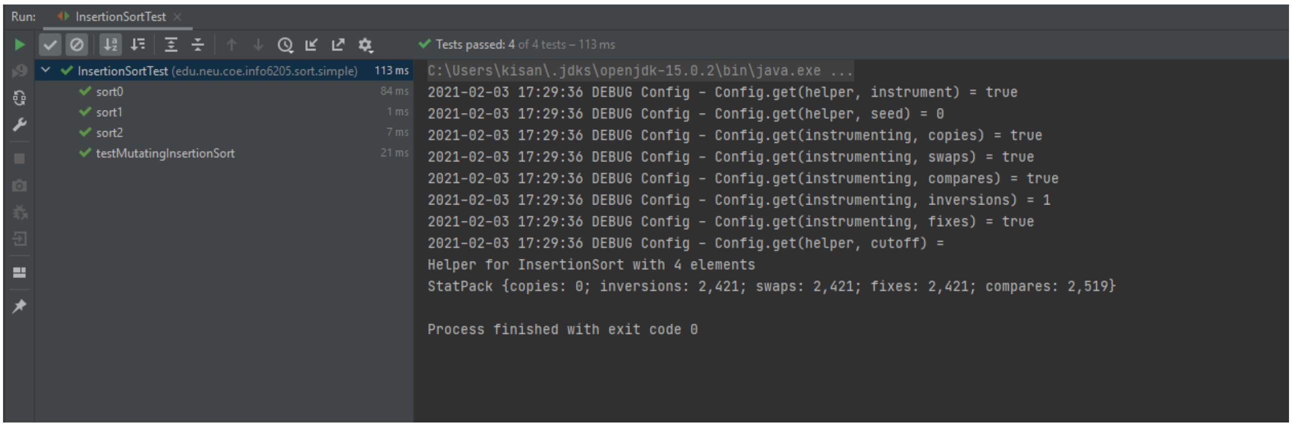
Benchmark Test:



```
Tests passed: 2 of 2 tests – 2 s 319 ms

BenchmarkTest (edu.neu.coe.info6205.util)    2 s 319 ms
    getWarmupRuns                            0 ms
    testWaitPeriods                          2 s 319 ms
```

```
C:\Users\kisan\.jdks\openjdk-15.0.2\bin\java.exe ...
2021-02-03 17:28:55 INFO  Benchmark_Timer - Begin run: testWaitPeriods with 2 runs

Process finished with exit code 0
```

InsertionSort Test:



```
Tests passed: 4 of 4 tests – 113 ms

InsertionSortTest (edu.neu.coe.info6205.sort.simple)    113 ms
    sort0                                               84 ms
    sort1                                               1 ms
    sort2                                               7 ms
    testMutatingInsertionSort                           21 ms
```

```
C:\Users\kisan\.jdks\openjdk-15.0.2\bin\java.exe ...
2021-02-03 17:29:36 DEBUG Config - Config.get(helper, instrument) = true
2021-02-03 17:29:36 DEBUG Config - Config.get(helper, seed) = 0
2021-02-03 17:29:36 DEBUG Config - Config.get(instrumenting, copies) = true
2021-02-03 17:29:36 DEBUG Config - Config.get(instrumenting, swaps) = true
2021-02-03 17:29:36 DEBUG Config - Config.get(instrumenting, compares) = true
2021-02-03 17:29:36 DEBUG Config - Config.get(instrumenting, inversions) = 1
2021-02-03 17:29:36 DEBUG Config - Config.get(instrumenting, fixes) = true
2021-02-03 17:29:36 DEBUG Config - Config.get(helper, cutoff) =
Helper for InsertionSort with 4 elements
StatPack {copies: 0; inversions: 2,421; swaps: 2,421; fixes: 2,421; compares: 2,519}

Process finished with exit code 0
```