Xiaoran Li (NUID: 001023070)

INFO 6205

Program Structures & Algorithms

Fall 2020

Assignment No.2

GitHub:https://github.com/XiaoranCS-IS/INFO6205.git

- **Task**

Your task for this assignment is in three parts.

(Part 1) You are to implement four 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.

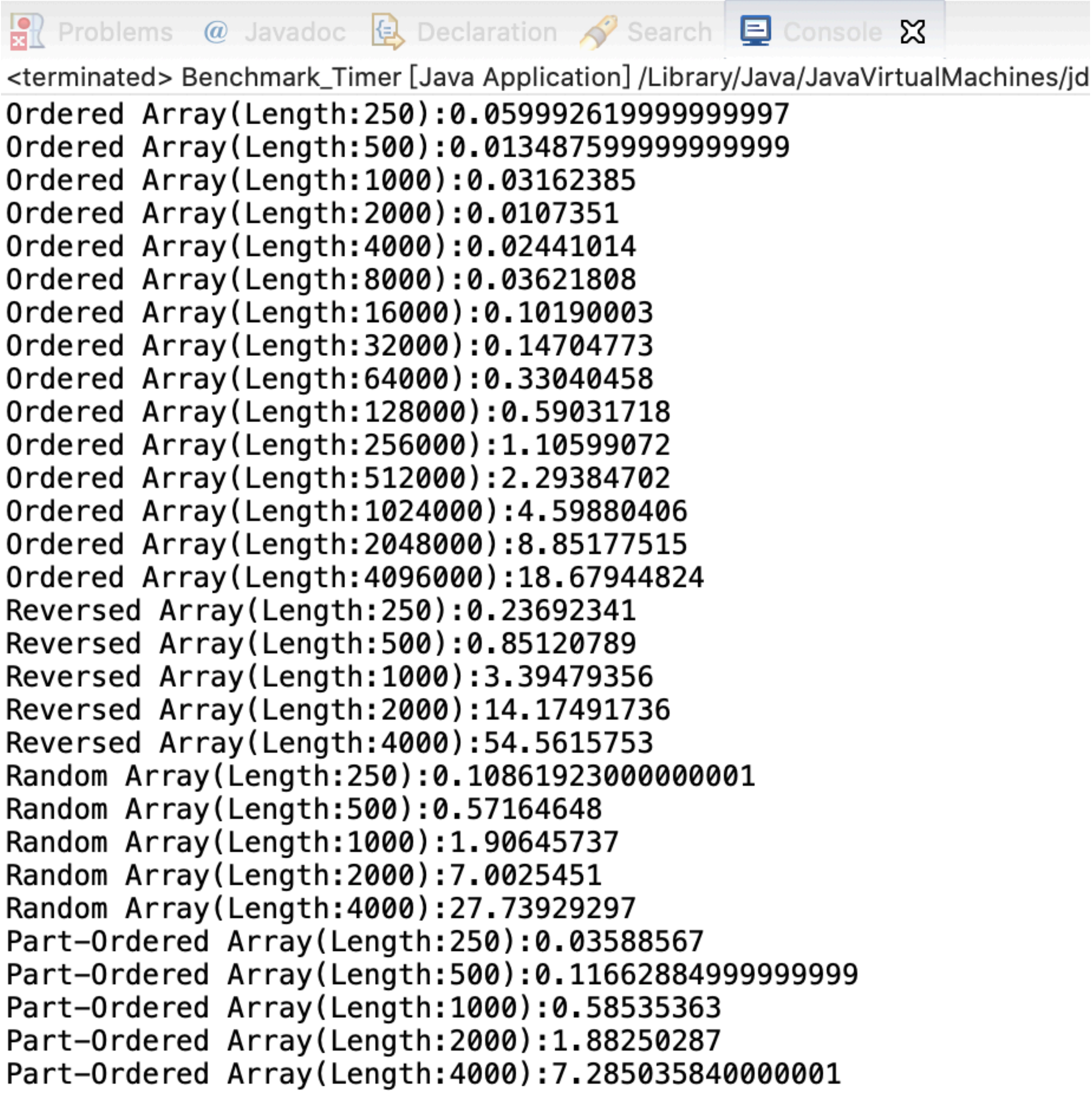
(Part 2) Implement InsertionSort (in the InsertionSort class) by simply looking up the insertion code used by Arrays.sort. You should use the helper.swap method although you could also just copy that from the same source code. In the main method of Benchmark, remove the reference to SelectionSort.

(Part 3) 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.

As usual, the submission will be your entire project (clean, i.e. without the target and project folders). There are stubs and unit tests in the repository.

Report on your observations and show screenshots of the runs and also the unit tests. Please note that you may have to adjust the required execution time for the insertion sort unit test(s) because your computer may not run at the same speed as mine.

- **Output** (few outputs to prove relationship)



```
<terminated> Benchmark_Timer [Java Application] /Library/Java/JavaVirtualMachines/jd
Ordered Array(Length:250):0.059992619999999997
Ordered Array(Length:500):0.013487599999999999
Ordered Array(Length:1000):0.03162385
Ordered Array(Length:2000):0.0107351
Ordered Array(Length:4000):0.02441014
Ordered Array(Length:8000):0.03621808
Ordered Array(Length:16000):0.10190003
Ordered Array(Length:32000):0.14704773
Ordered Array(Length:64000):0.33040458
Ordered Array(Length:128000):0.59031718
Ordered Array(Length:256000):1.10599072
Ordered Array(Length:512000):2.29384702
Ordered Array(Length:1024000):4.59880406
Ordered Array(Length:2048000):8.85177515
Ordered Array(Length:4096000):18.67944824
Reversed Array(Length:250):0.23692341
Reversed Array(Length:500):0.85120789
Reversed Array(Length:1000):3.39479356
Reversed Array(Length:2000):14.17491736
Reversed Array(Length:4000):54.5615753
Random Array(Length:250):0.10861923000000001
Random Array(Length:500):0.57164648
Random Array(Length:1000):1.90645737
Random Array(Length:2000):7.0025451
Random Array(Length:4000):27.73929297
Part-Ordered Array(Length:250):0.03588567
Part-Ordered Array(Length:500):0.11662884999999999
Part-Ordered Array(Length:1000):0.58535363
Part-Ordered Array(Length:2000):1.88250287
Part-Ordered Array(Length:4000):7.285035840000001
```

- **Relationship conclusion**

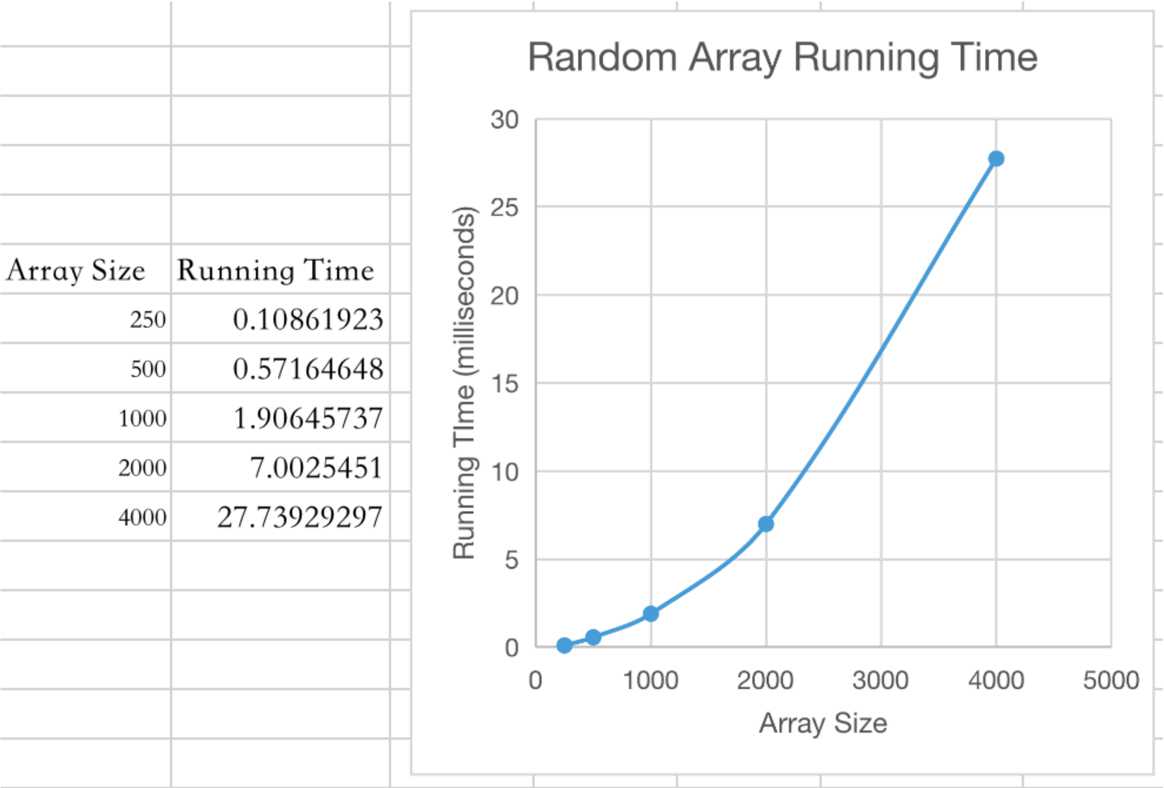
Random: Time =$1.74 * 10^{-6} * n^2$

Ordered: Time =$4.46 * 10^{-6} * n$

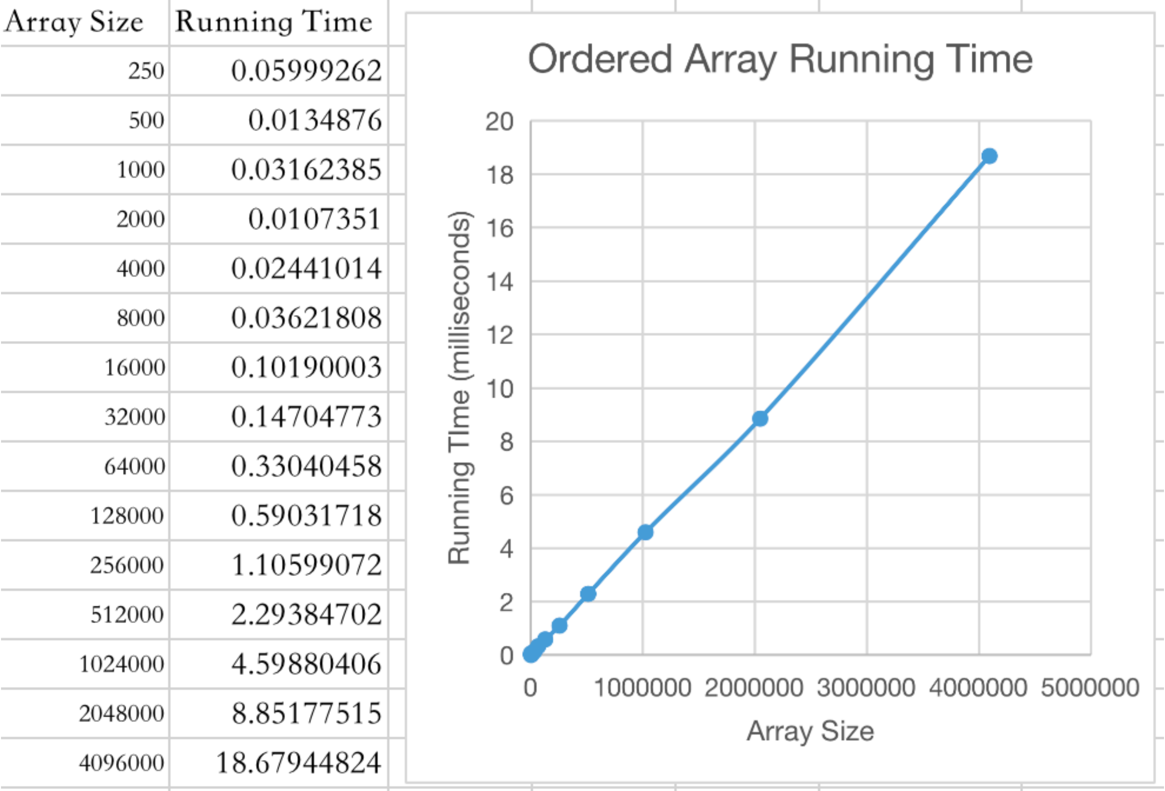
Partially-ordered: Time = $4.7 * 10^{-7} * n^2$ (Half sorted and half random)

Reverse-ordered: Time = $3.4 * 10^{-6} * n^2$

- **Evidence to support relationship** (screen shot and/or graph and/or spreadsheet)

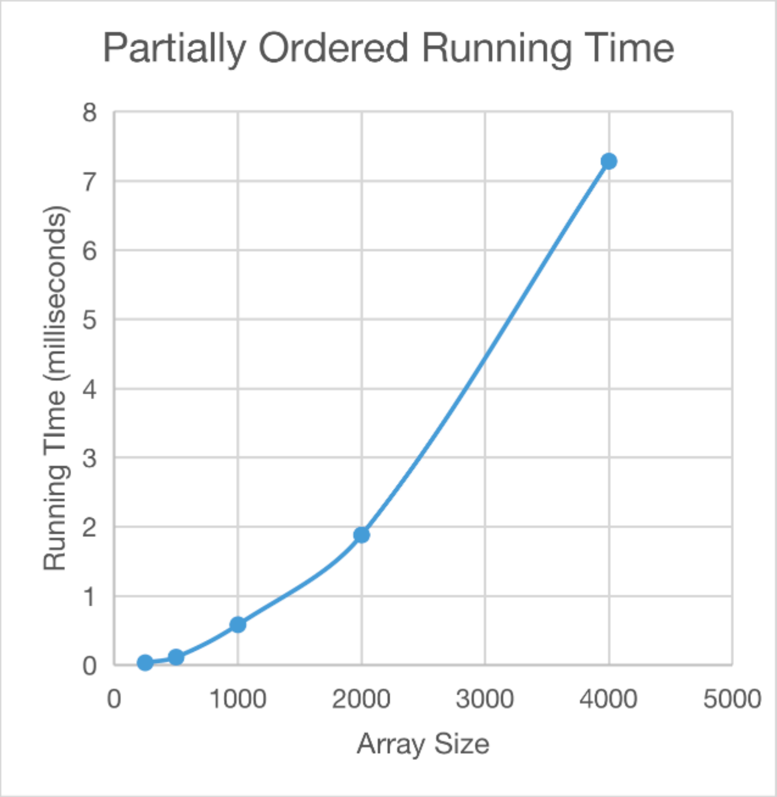


Array Size	Running Time
250	0.10861923
500	0.57164648
1000	1.90645737
2000	7.0025451
4000	27.73929297

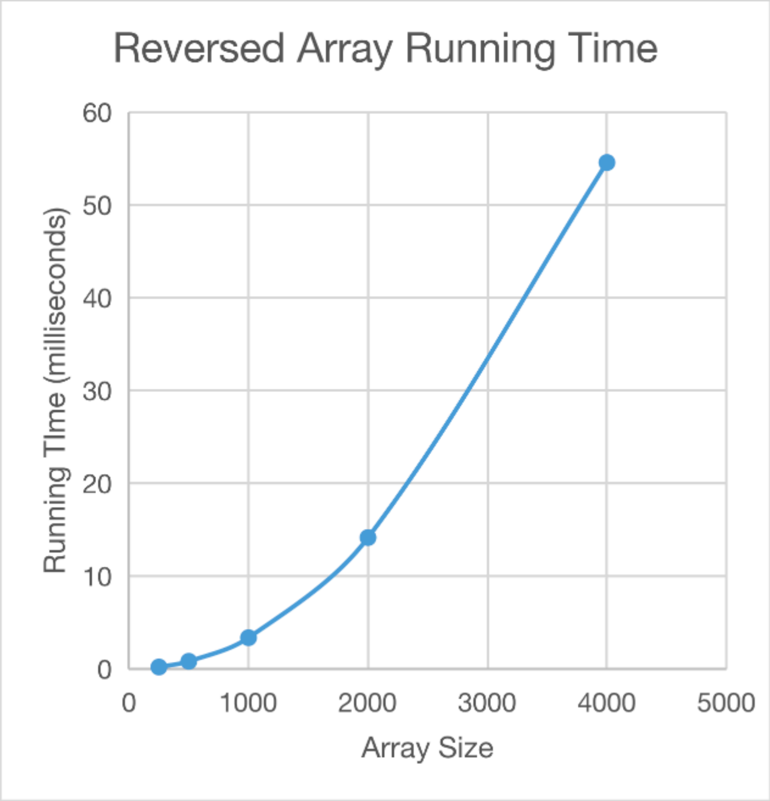


Array Size	Running Time
250	0.05999262
500	0.0134876
1000	0.03162385
2000	0.0107351
4000	0.02441014
8000	0.03621808
16000	0.10190003
32000	0.14704773
64000	0.33040458
128000	0.59031718
256000	1.10599072
512000	2.29384702
1024000	4.59880406
2048000	8.85177515
4096000	18.67944824

Array Size	Running Time
250	0.03588567
500	0.11662885
1000	0.58535363
2000	1.88250287
4000	7.28503584

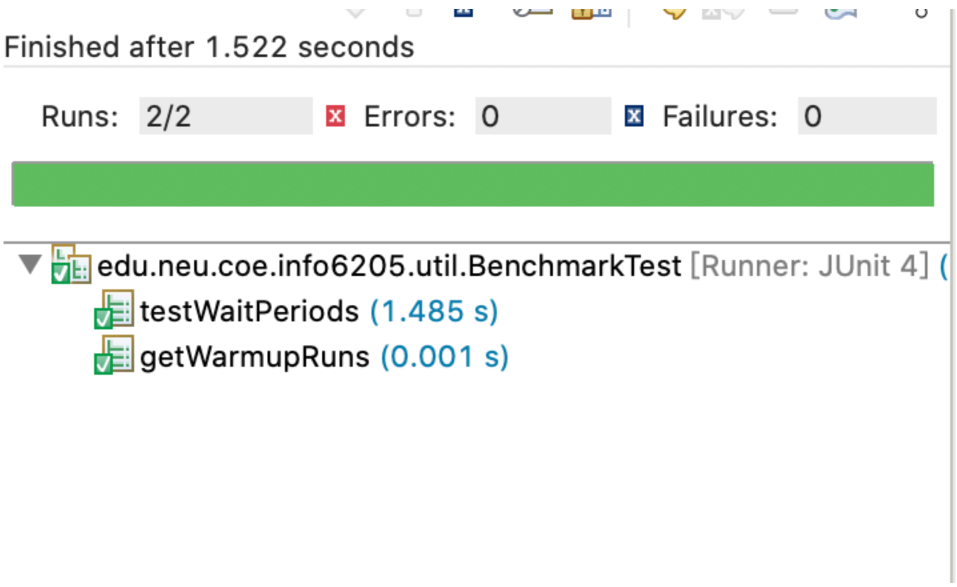


Array Size	Running Time
250	0.23692341
500	0.85120789
1000	3.39479356
2000	14.17491736
4000	54.5615753



- **Screenshot of Unit test passing**

BenchmarkTest:

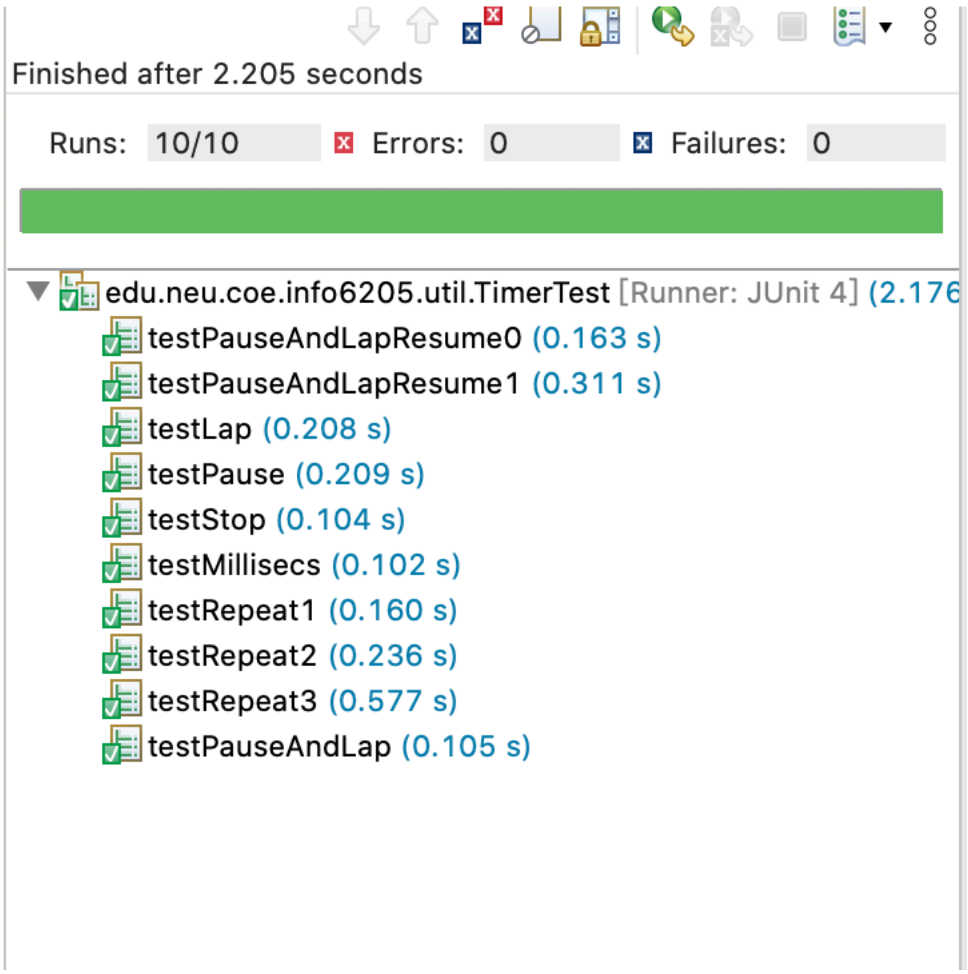


```
Finished after 1.522 seconds

Runs:  2/2     Errors:  0     Failures:  0

edu.neu.coe.info6205.util.BenchmarkTest [Runner: JUnit 4] (
    testWaitPeriods (1.485 s)
    getWarmupRuns (0.001 s)
```

TimerTest:

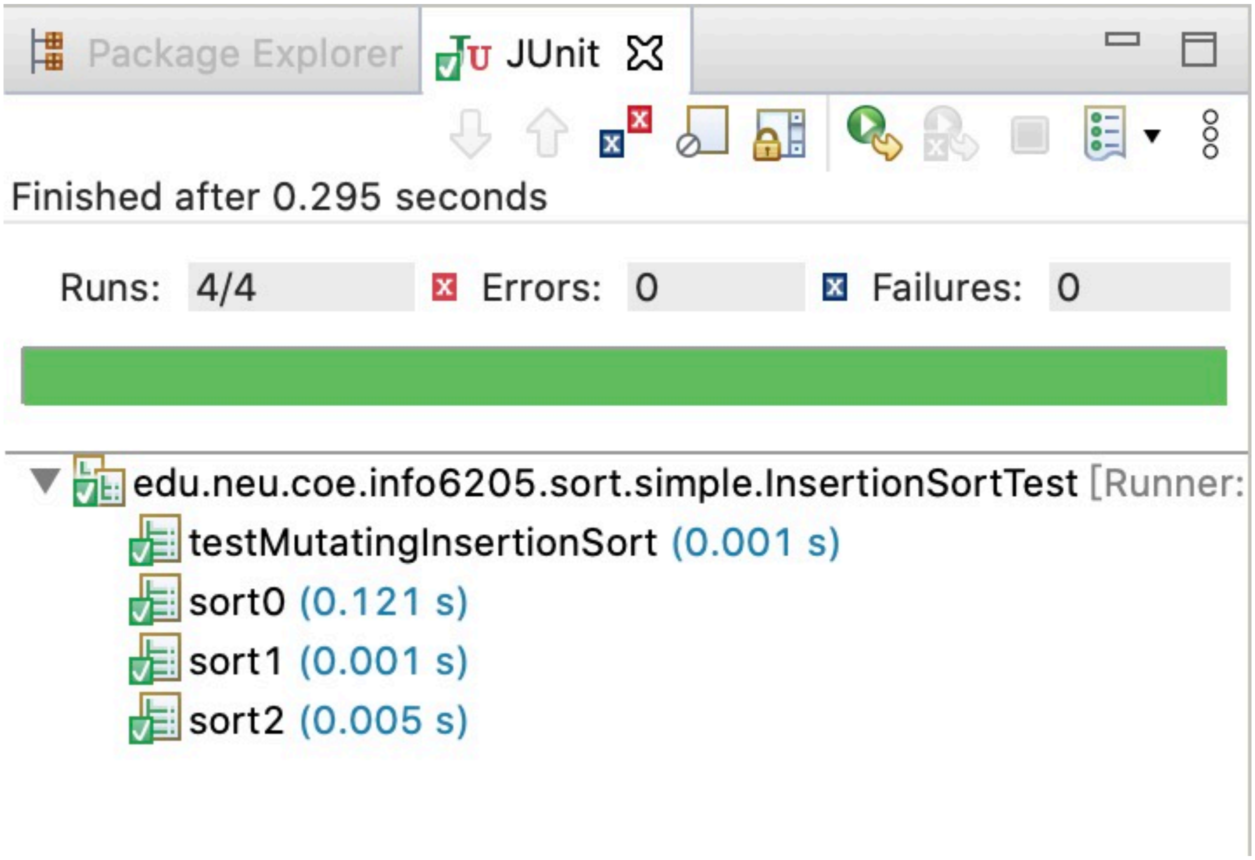


```
Finished after 2.205 seconds

Runs:  10/10     Errors:  0     Failures:  0

edu.neu.coe.info6205.util.TimerTest [Runner: JUnit 4] (2.176
    testPauseAndLapResume0 (0.163 s)
    testPauseAndLapResume1 (0.311 s)
    testLap (0.208 s)
    testPause (0.209 s)
    testStop (0.104 s)
    testMillisecs (0.102 s)
    testRepeat1 (0.160 s)
    testRepeat2 (0.236 s)
    testRepeat3 (0.577 s)
    testPauseAndLap (0.105 s)
```

InsertionSortTest:



Finished after 0.295 seconds

Runs: 4/4 Errors: 0 Failures: 0

```
edu.neu.coe.info6205.sort.simple.InsertionSortTest [Runner:
    testMutatingInsertionSort (0.001 s)
    sort0 (0.121 s)
    sort1 (0.001 s)
    sort2 (0.005 s)
```