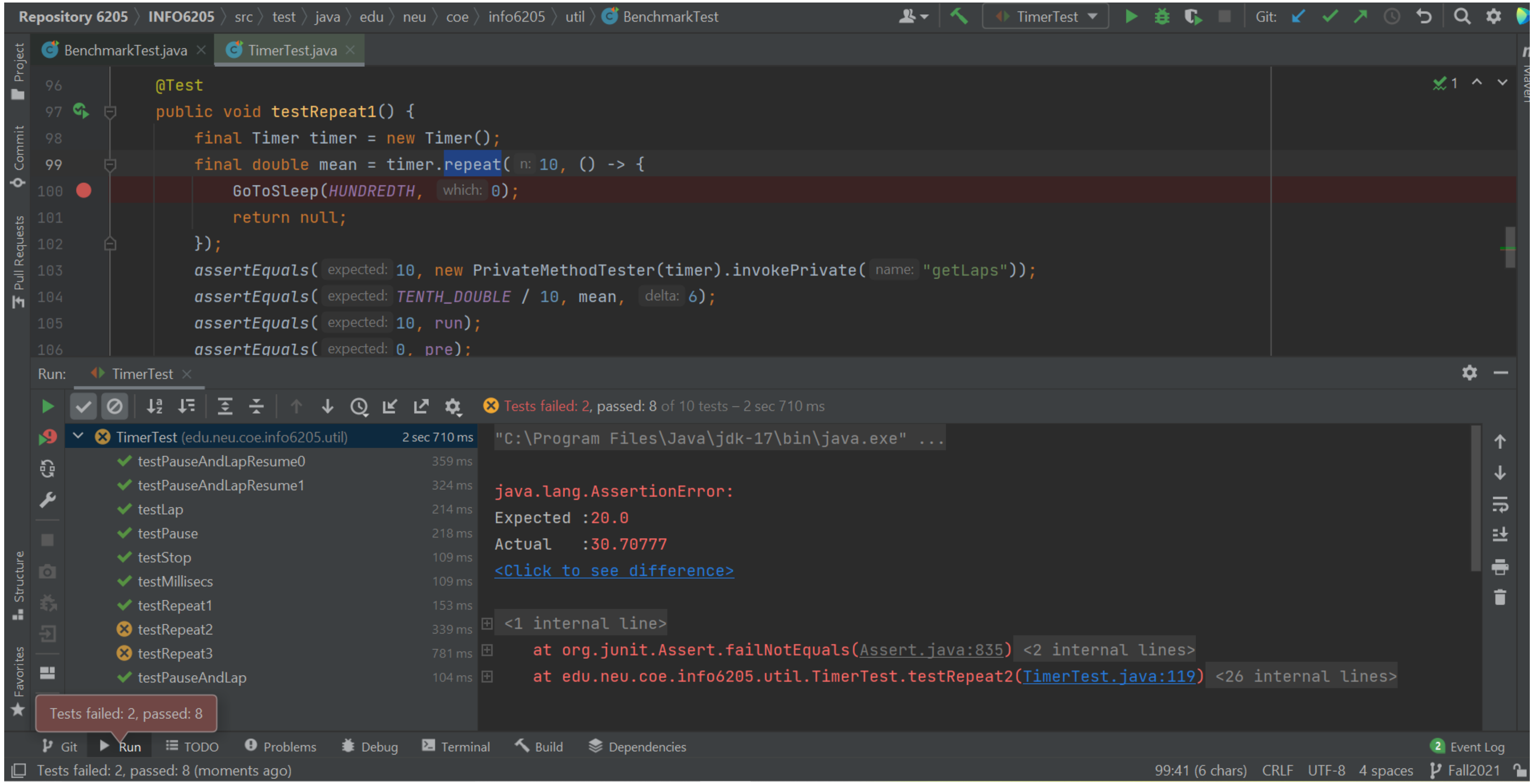
# NUID: 002175782
# PSA Fall 21

Assignment 1: Random Walk

DIKSHA BHATIA

# TASKS

- Part 1 – Fixed repeat function inTimer.java



```
    @Test
    public void testRepeat1() {
        final Timer timer = new Timer();
        final double mean = timer.repeat( n: 10, () -> {
            GoToSleep(HUNDREDTH, which: 0);
            return null;
        });
        assertEquals( expected: 10, new PrivateMethodTester(timer).invokePrivate( name: "getLaps"));
        assertEquals( expected: TENTH_DOUBLE / 10, mean, delta: 6);
        assertEquals( expected: 10, run);
        assertEquals( expected: 0, pre);
```

```
Tests failed: 2, passed: 8 of 10 tests – 2 sec 710 ms

TimerTest (edu.neu.coe.info6205.util)   2 sec 710 ms
  testPauseAndLapResume0                 359 ms
  testPauseAndLapResume1                 324 ms
  testLap                                214 ms
  testPause                              218 ms
  testStop                               109 ms
  testMillisecs                          109 ms
  testRepeat1                            153 ms
  testRepeat2                            339 ms
  testRepeat3                            781 ms
  testPauseAndLap                        104 ms
```

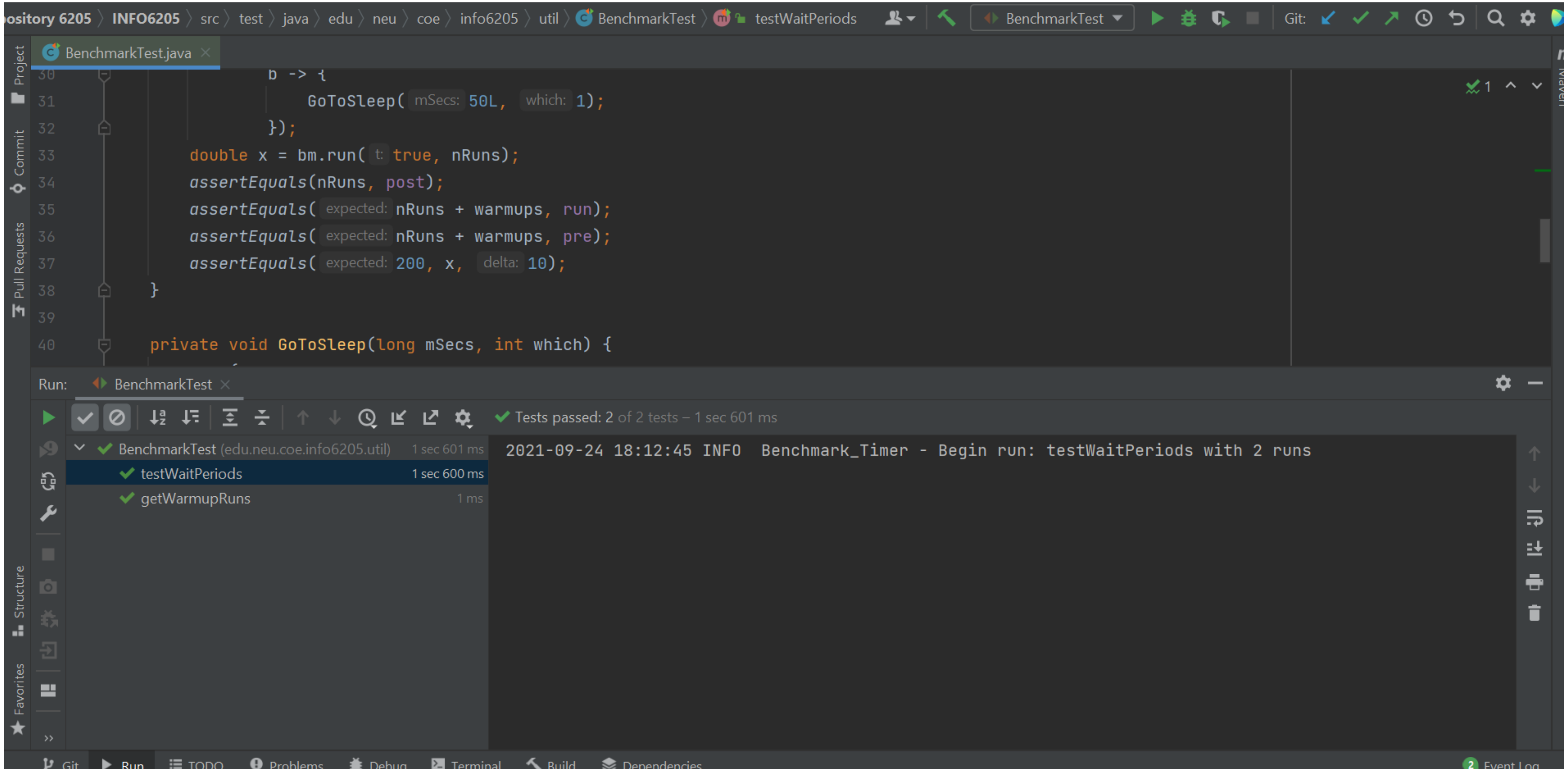
```
"C:\Program Files\Java\jdk-17\bin\java.exe" ...

java.lang.AssertionError:
Expected :20.0
Actual   :30.70777
<Click to see difference>

<1 internal line>
    at org.junit.Assert.failNotEquals(Assert.java:835) <2 internal lines>
    at edu.neu.coe.info6205.util.TimerTest.testRepeat2(TimerTest.java:119) <26 internal lines>
```

Tests failed: 2, passed: 8

Tests failed: 2, passed: 8 (moments ago)

# TASKS

- Ran BenchmarkTest.java



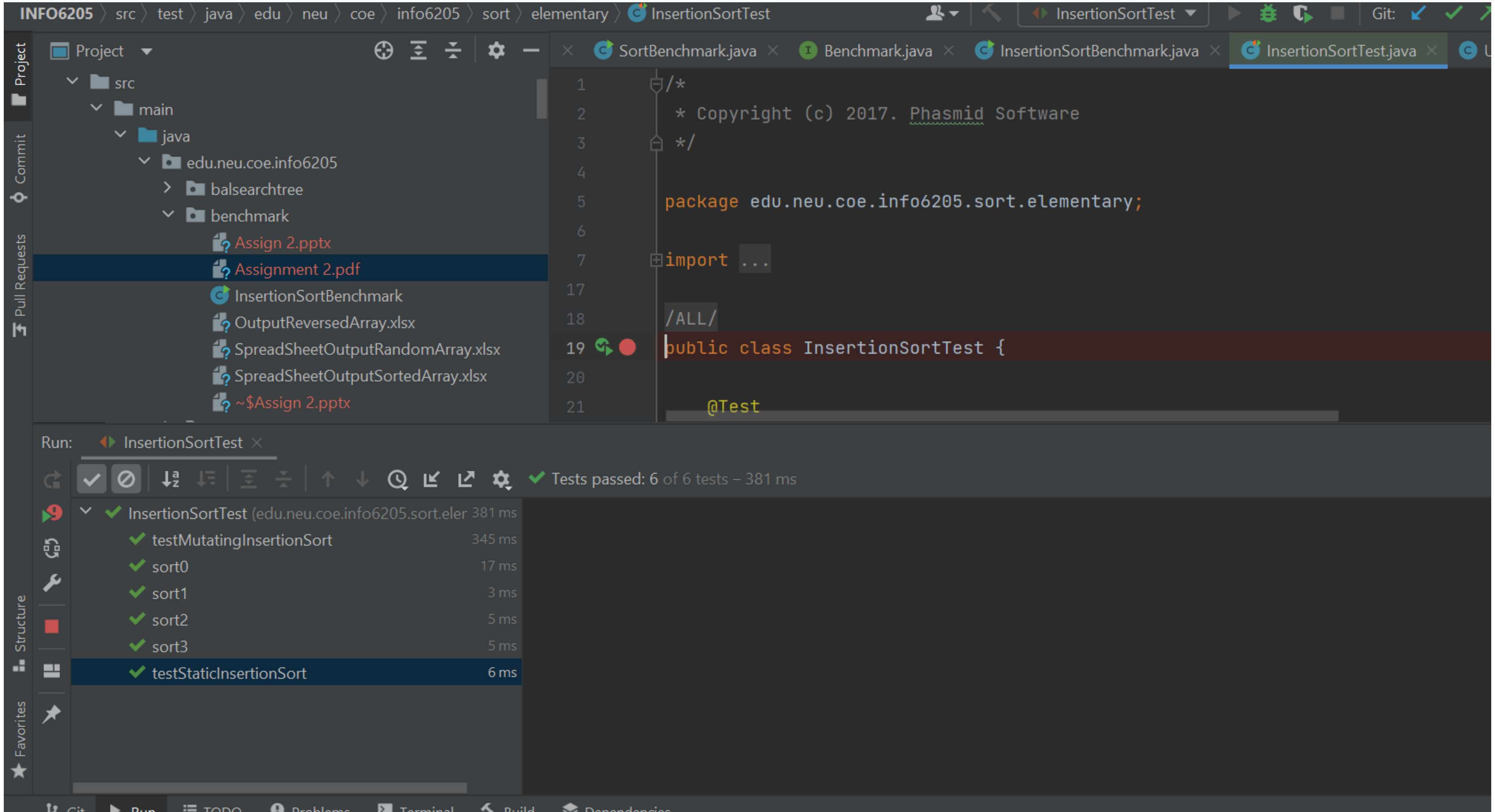
```
                b -> {
                    GoToSleep( mSecs: 50L,  which: 1);
                });
        double x = bm.run( t: true, nRuns);
        assertEquals(nRuns, post);
        assertEquals( expected: nRuns + warmups, run);
        assertEquals( expected: nRuns + warmups, pre);
        assertEquals( expected: 200, x,  delta: 10);
    }

    private void GoToSleep(long mSecs, int which) {
```

```
✓ Tests passed: 2 of 2 tests – 1 sec 601 ms

✓ BenchmarkTest (edu.neu.coe.info6205.util)   1 sec 601 ms
    ✓ testWaitPeriods                          1 sec 600 ms
    ✓ getWarmupRuns                                    1 ms
```

```
2021-09-24 18:12:45 INFO  Benchmark_Timer - Begin run: testWaitPeriods with 2 runs
```

# TASKS

- Implemented sort in InsertionSort.java and ran InsertionSort Test case



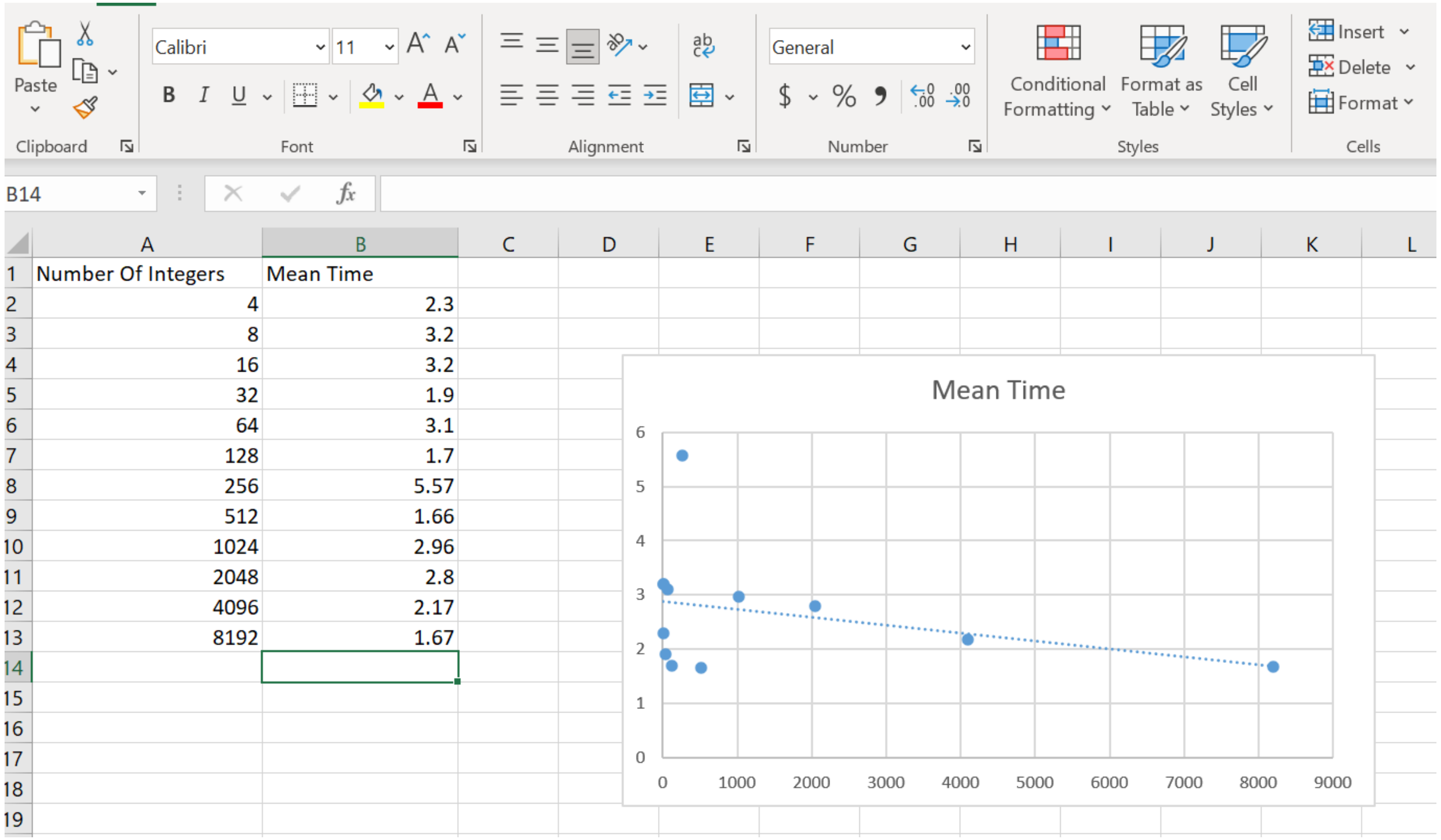
```
/*
 * Copyright (c) 2017. Phasmid Software
 */

package edu.neu.coe.info6205.sort.elementary;

import ...

/ALL/
public class InsertionSortTest {

    @Test
```

```
Tests passed: 6 of 6 tests – 381 ms

InsertionSortTest (edu.neu.coe.info6205.sort.eler 381 ms
    testMutatingInsertionSort                   345 ms
    sort0                                        17 ms
    sort1                                         3 ms
    sort2                                         5 ms
    sort3                                         5 ms
    testStaticInsertionSort                       6 ms
```

# Determining Order of Growth

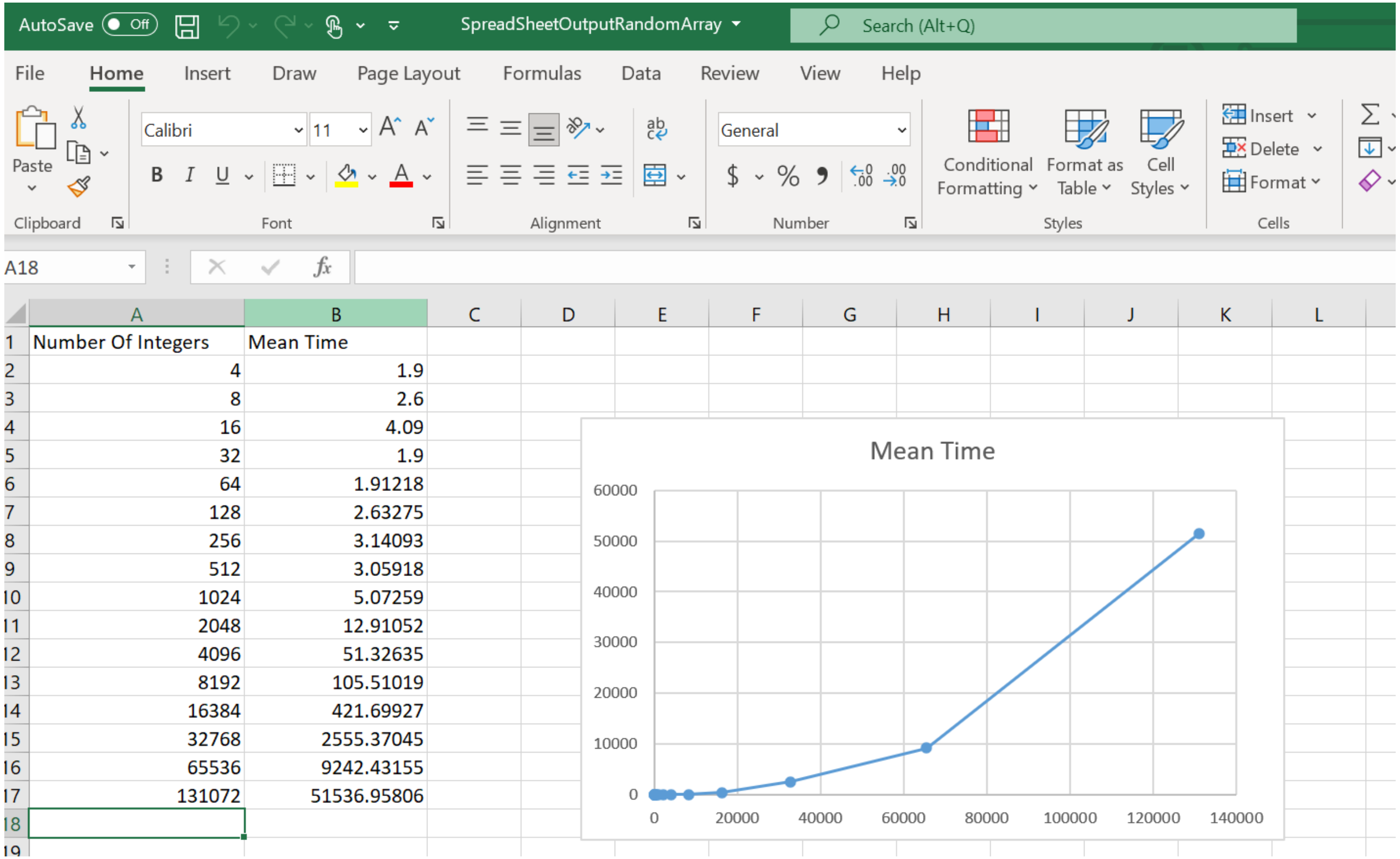
- Implemented sort in InsertionSortBenchmark.java in package benchmark.
- Created different excel files and plotted time taken for sorted / random and reversed array

# Sorted Array Plot – SpreadSheetOutputSortedArray.xls in pckg Benchmark



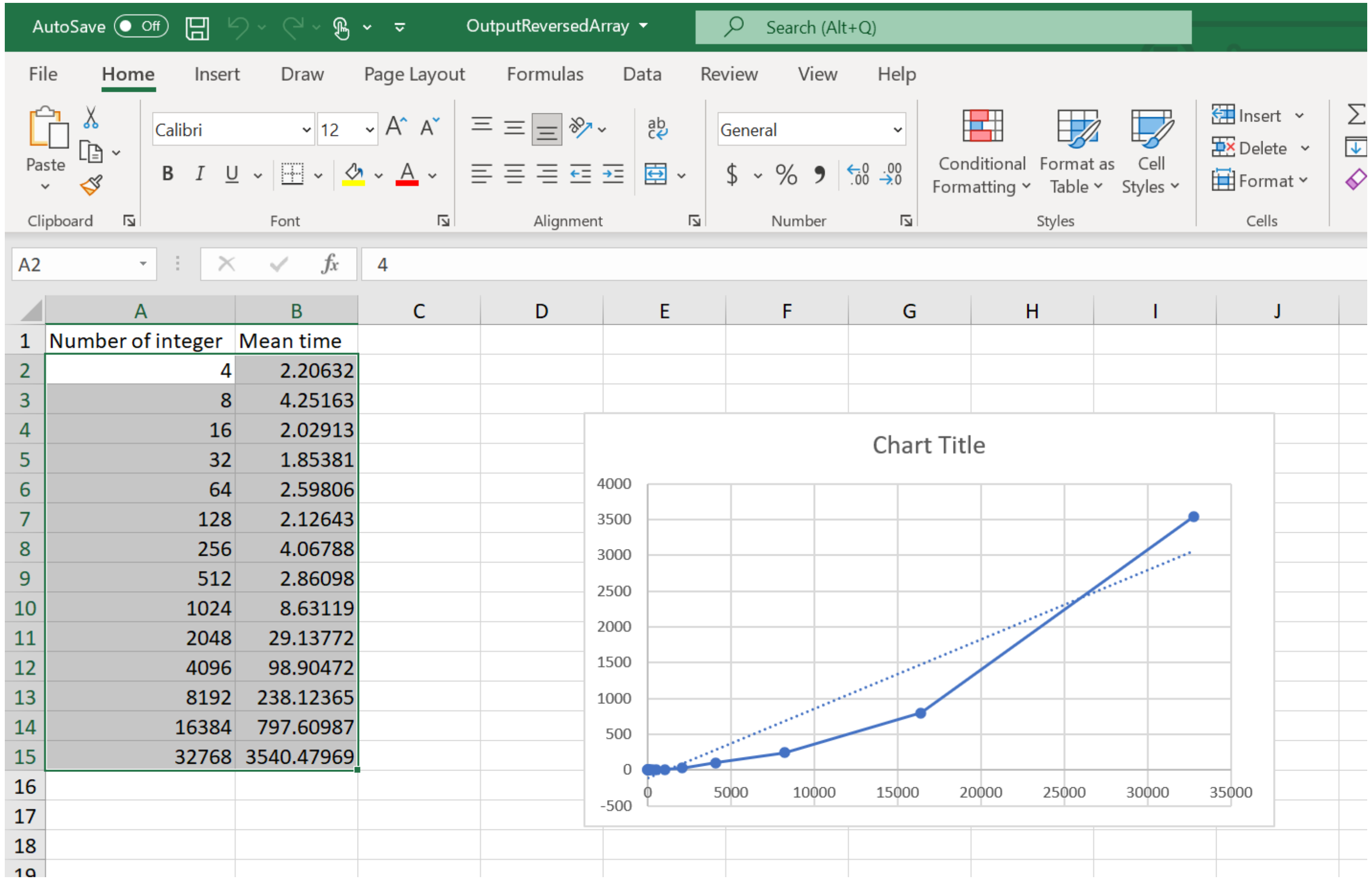
| Number Of Integers | Mean Time |
|---|---|
| 4 | 2.3 |
| 8 | 3.2 |
| 16 | 3.2 |
| 32 | 1.9 |
| 64 | 3.1 |
| 128 | 1.7 |
| 256 | 5.57 |
| 512 | 1.66 |
| 1024 | 2.96 |
| 2048 | 2.8 |
| 4096 | 2.17 |
| 8192 | 1.67 |

# Random Array Plot – SpreadSheetOutputRandomArray.xls in pckg Benchmark



| | A | B |
|---|---|---|
| 1 | Number Of Integers | Mean Time |
| 2 | 4 | 1.9 |
| 3 | 8 | 2.6 |
| 4 | 16 | 4.09 |
| 5 | 32 | 1.9 |
| 6 | 64 | 1.91218 |
| 7 | 128 | 2.63275 |
| 8 | 256 | 3.14093 |
| 9 | 512 | 3.05918 |
| 10 | 1024 | 5.07259 |
| 11 | 2048 | 12.91052 |
| 12 | 4096 | 51.32635 |
| 13 | 8192 | 105.51019 |
| 14 | 16384 | 421.69927 |
| 15 | 32768 | 2555.37045 |
| 16 | 65536 | 9242.43155 |
| 17 | 131072 | 51536.95806 |
| 18 | | |
| 19 | | |

# Reversed Array Plot – OutputReversedArray.xls in pckg Benchmark



| | A | B |
|---|---|---|
| 1 | Number of integer | Mean time |
| 2 | 4 | 2.20632 |
| 3 | 8 | 4.25163 |
| 4 | 16 | 2.02913 |
| 5 | 32 | 1.85381 |
| 6 | 64 | 2.59806 |
| 7 | 128 | 2.12643 |
| 8 | 256 | 4.06788 |
| 9 | 512 | 2.86098 |
| 10 | 1024 | 8.63119 |
| 11 | 2048 | 29.13772 |
| 12 | 4096 | 98.90472 |
| 13 | 8192 | 238.12365 |
| 14 | 16384 | 797.60987 |
| 15 | 32768 | 3540.47969 |

# ORDER OF GROWTH

- Based on the data points used: I can conclude that for sorted/random arrays, insertion sort behaves in a LINEAR fashion.
- Whereas for reversed array, its growth is in QUADRATIC fashion.