Program Structures and Algorithms
Spring 2023(SEC –8)
Assignment 3 (Benchmark)

NAME: Hao Zhou
NUID: 002749956

Task:

(Part 1) You are to implement three (3) methods (*repeat*, *getClock*, and *toMillisecs*) of a class called *Timer*.
(Part 2) Implement InsertionSort (in the InsertionSort class) by simply looking up the insertion code used by Arrays.sort. And running 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.

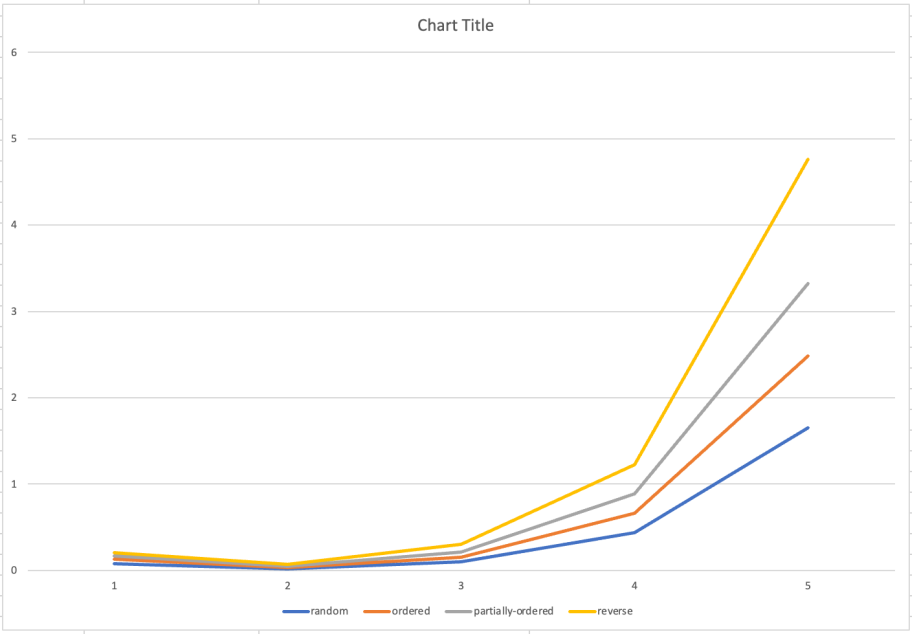
Relationship Conclusion:

The ordered array takes the least time and reverse-ordered takes the most time.

By running 4 different types of given arrays (random, ordered, partially-ordered, reveres-ordered) with 5 different n values in benchmark, I record the data into an Excel chart. I find that the ordered array takes the least time because it doesn't need to swap integers. And reverse-ordered takes the most time because it needs to swap every element. And with n increasing, the gap between each ordered array is becoming more and more obvious.

Evidence to support that conclusion:

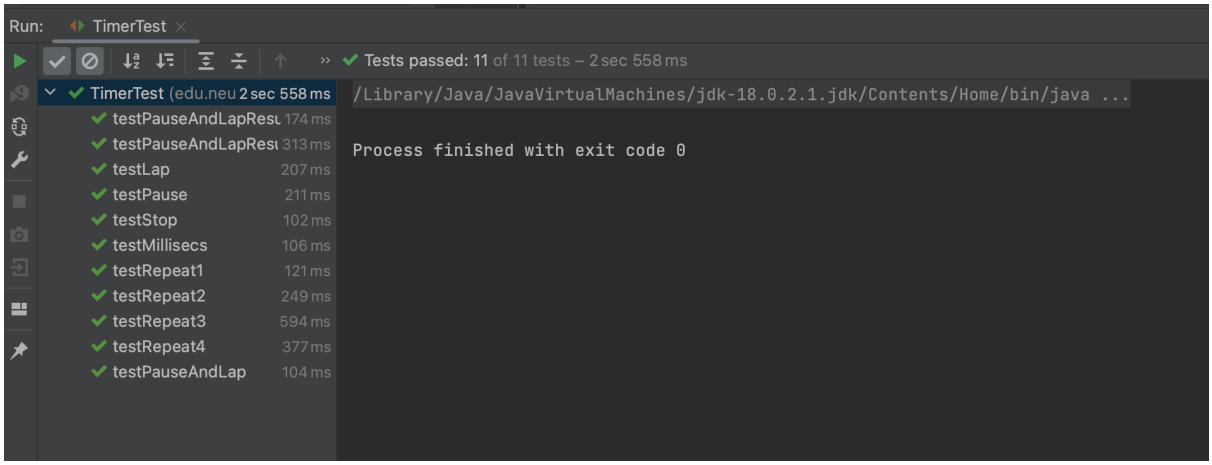
n	random	ordered	partially-ordered	reverse
1000	0.08	0.05	0.037	0.035
2000	0.02	0.015	0.013	0.025
4000	0.1	0.055	0.056	0.095
8000	0.44	0.225	0.22	0.3375
16000	1.65	0.835	0.837	1.4425



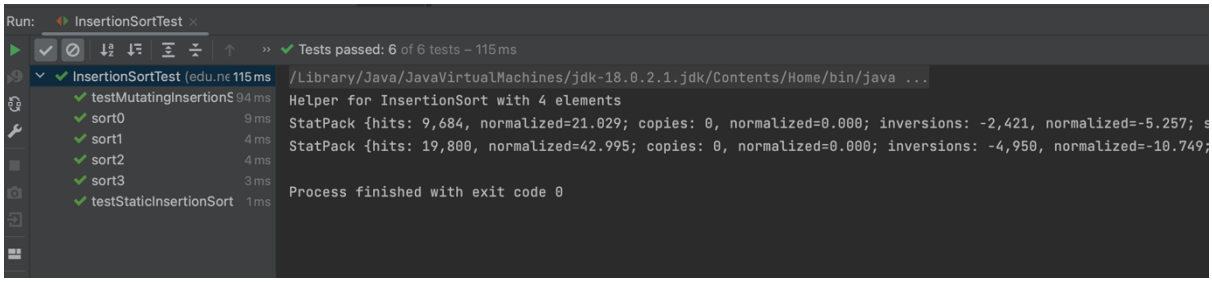
```
when N is : 1000
The insertion sort time of a random array is 0.08
The insertion sort time of an ordered array is 0.05
The insertion sort time of an partially-ordered array is 0.03666666666666667
The insertion sort time of an reverse-ordered array is 0.035
when N is : 2000
The insertion sort time of a random array is 0.02
The insertion sort time of an ordered array is 0.015
The insertion sort time of an partially-ordered array is 0.013333333333333334
The insertion sort time of an reverse-ordered array is 0.025
when N is : 4000
The insertion sort time of a random array is 0.1
The insertion sort time of an ordered array is 0.055
The insertion sort time of an partially-ordered array is 0.056666666666666664
The insertion sort time of an reverse-ordered array is 0.095
when N is : 8000
The insertion sort time of a random array is 0.44
The insertion sort time of an ordered array is 0.225
The insertion sort time of an partially-ordered array is 0.22
The insertion sort time of an reverse-ordered array is 0.3775
when N is : 16000
The insertion sort time of a random array is 1.65
The insertion sort time of an ordered array is 0.835
The insertion sort time of an partially-ordered array is 0.8366666666666667
The insertion sort time of an reverse-ordered array is 1.4425

Process finished with exit code 0
```

Unit Test Screenshots:

TimerTest:



InsertionSortTest:


Run:
InsertionSortTest
Tests passed: 6 of 6 tests – 115 ms
InsertionSortTest (edu.ne 115 ms
testMutatingInsertionS 94 ms
sort0 9 ms
sort1 4 ms
sort2 4 ms
sort3 3 ms
testStaticInsertionSort 1 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Helper for InsertionSort with 4 elements
StatPack {hits: 9,684, normalized=21.029; copies: 0, normalized=0.000; inversions: -2,421, normalized=-5.257;
StatPack {hits: 19,800, normalized=42.995; copies: 0, normalized=0.000; inversions: -4,950, normalized=-10.749;
Process finished with exit code 0