

Yang Song (001003647)

Program Structures & Algorithms

Fall 2021

Assignment No. 2

◉ **Task (List down the tasks performed in the Assignment)**

1. Implement three methods of a class called Timer
2. Implement InsertionSort.java
3. Implement a main program 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:**

$f(\text{ordered}) = O(N)$

$f(\text{reverse-ordered}) = O(N^2)$

$f(\text{ordered}) < f(\text{partially ordered}) < f(\text{random}) < f(\text{reverse ordered})$

◉ **Evidence to support the conclusion:**

1. **Output**

Run main program in InsertionSort.java and get mean run time of different type of input arrays with doubling method

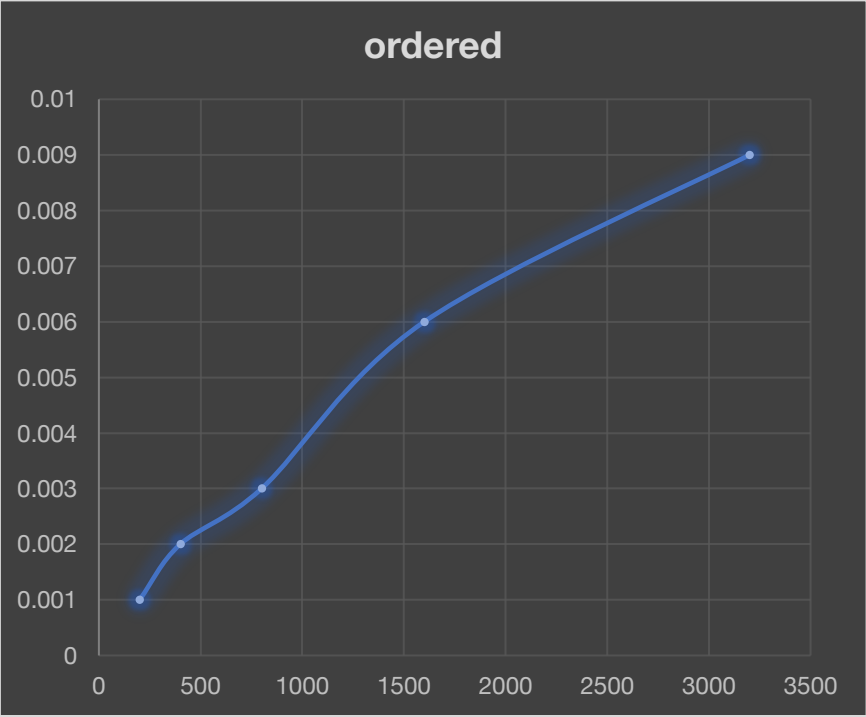
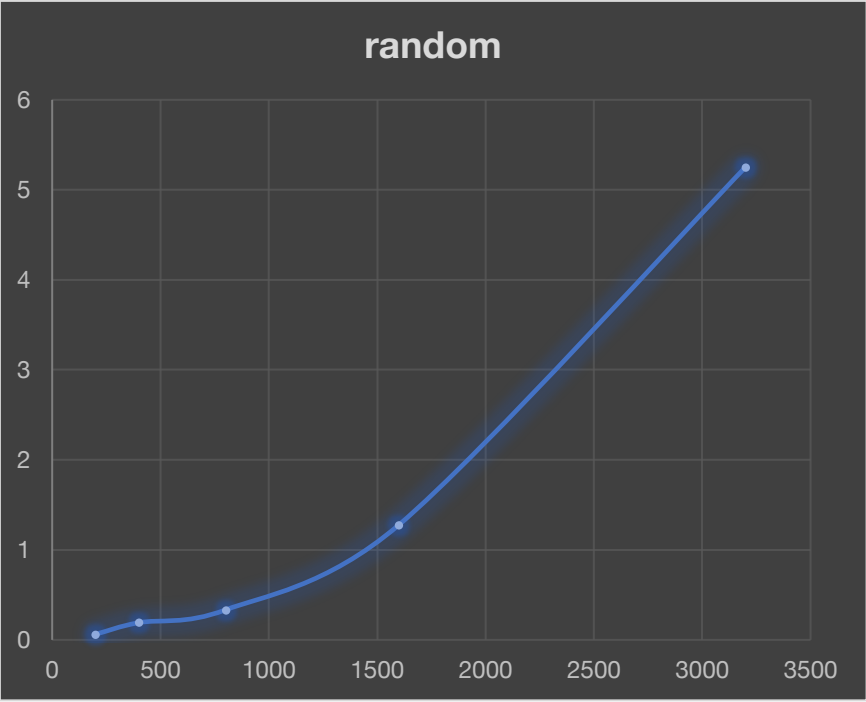
```
Run: InsertionSort x
/Library/Java/JavaVirtualMachines/jdk1.8.0_181.jdk/Contents/Home/bin/java ...
2021-09-24 20:18:07 INFO Benchmark_Timer - Begin run: random input insertion sort run time with 1,000 runs
when n is 200, run time is 0.062
2021-09-24 20:18:07 INFO Benchmark_Timer - Begin run: ordered input insertion sort run time with 1,000 runs
when n is 200, run time is 0.001
2021-09-24 20:18:07 INFO Benchmark_Timer - Begin run: partially ordered input insertion sort run time with 1,000 runs
when n is 200, run time is 0.034
2021-09-24 20:18:07 INFO Benchmark_Timer - Begin run: reverse ordered input insertion sort run time with 1,000 runs
when n is 200, run time is 0.096
2021-09-24 20:18:08 INFO Benchmark_Timer - Begin run: random input insertion sort run time with 1,000 runs
when n is 400, run time is 0.191
2021-09-24 20:18:08 INFO Benchmark_Timer - Begin run: ordered input insertion sort run time with 1,000 runs
when n is 400, run time is 0.002
2021-09-24 20:18:08 INFO Benchmark_Timer - Begin run: partially ordered input insertion sort run time with 1,000 runs
when n is 400, run time is 0.123
2021-09-24 20:18:08 INFO Benchmark_Timer - Begin run: reverse ordered input insertion sort run time with 1,000 runs
when n is 400, run time is 0.201
2021-09-24 20:18:08 INFO Benchmark_Timer - Begin run: random input insertion sort run time with 1,000 runs
when n is 800, run time is 0.333
2021-09-24 20:18:08 INFO Benchmark_Timer - Begin run: ordered input insertion sort run time with 1,000 runs
when n is 800, run time is 0.003
2021-09-24 20:18:08 INFO Benchmark_Timer - Begin run: partially ordered input insertion sort run time with 1,000 runs
when n is 800, run time is 0.238
2021-09-24 20:18:09 INFO Benchmark_Timer - Begin run: reverse ordered input insertion sort run time with 1,000 runs
when n is 800, run time is 0.625

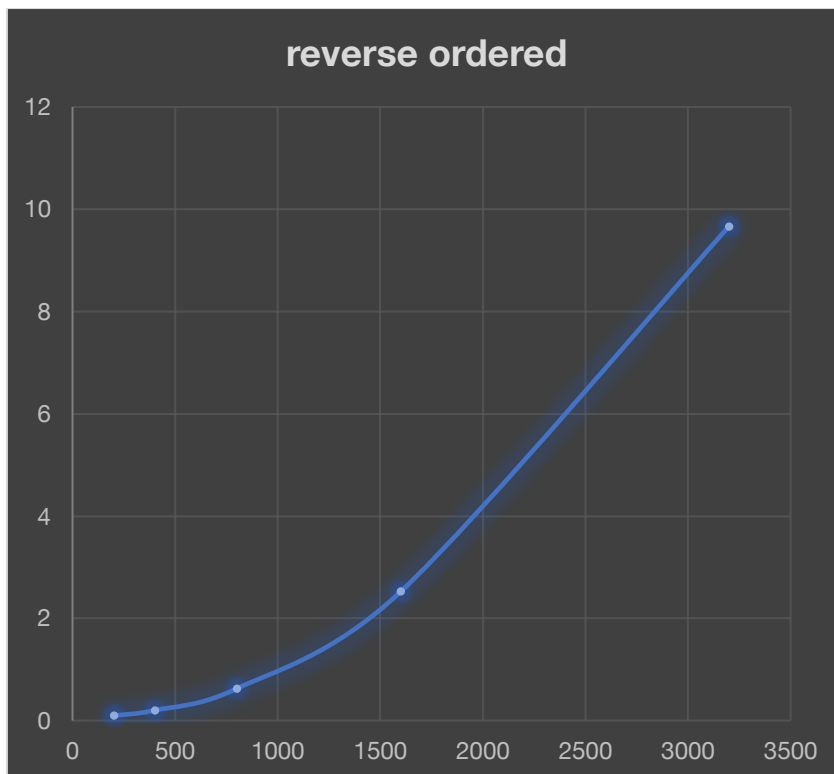
2021-09-24 20:18:09 INFO Benchmark_Timer - Begin run: random input insertion sort run time with 1,000 runs
when n is 1600, run time is 1.277
2021-09-24 20:18:11 INFO Benchmark_Timer - Begin run: ordered input insertion sort run time with 1,000 runs
when n is 1600, run time is 0.006
2021-09-24 20:18:11 INFO Benchmark_Timer - Begin run: partially ordered input insertion sort run time with 1,000 runs
when n is 1600, run time is 0.931
2021-09-24 20:18:12 INFO Benchmark_Timer - Begin run: reverse ordered input insertion sort run time with 1,000 runs
when n is 1600, run time is 2.524
2021-09-24 20:18:14 INFO Benchmark_Timer - Begin run: random input insertion sort run time with 1,000 runs
when n is 3200, run time is 5.252
2021-09-24 20:18:19 INFO Benchmark_Timer - Begin run: ordered input insertion sort run time with 1,000 runs
when n is 3200, run time is 0.009
2021-09-24 20:18:19 INFO Benchmark_Timer - Begin run: partially ordered input insertion sort run time with 1,000 runs
when n is 3200, run time is 3.699
2021-09-24 20:18:23 INFO Benchmark_Timer - Begin run: reverse ordered input insertion sort run time with 1,000 runs
when n is 3200, run time is 9.668

Process finished with exit code 0
```

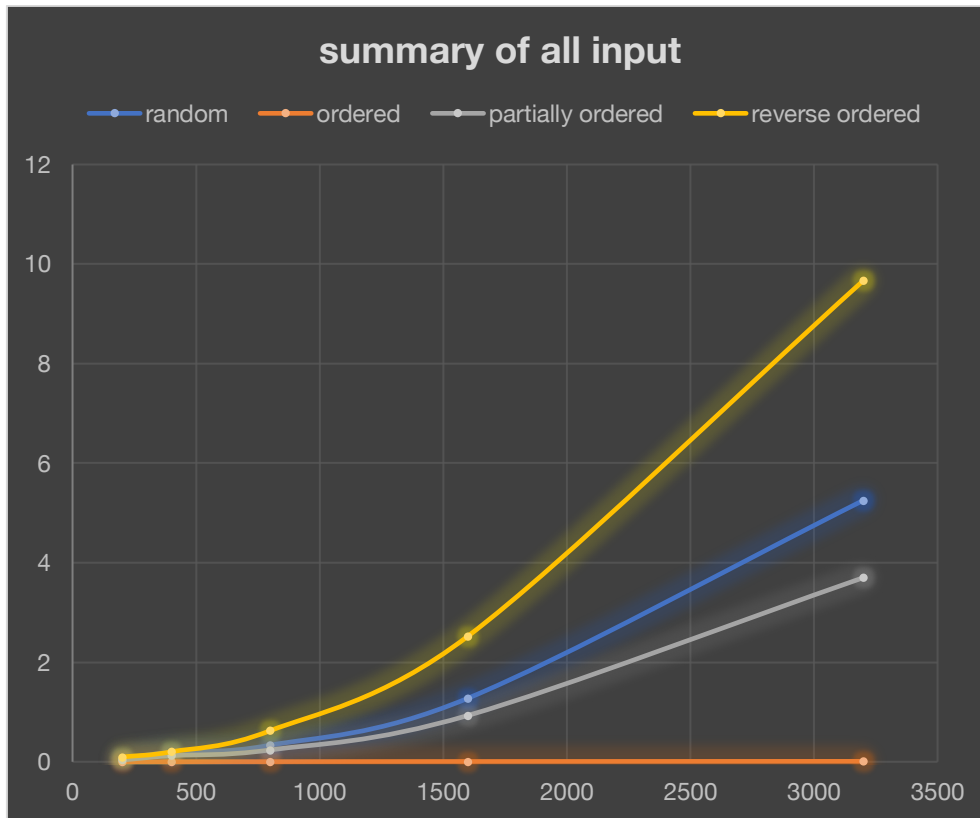
2. Graphical Representation

A		B	C	D	E
n	type	random	ordered	partially ordered	reverse ordered
200		0.062	0.001	0.034	0.096
400		0.191	0.002	0.123	0.201
800		0.333	0.003	0.238	0.625
1600		1.277	0.006	0.931	2.524
3200		5.252	0.009	3.699	9.668

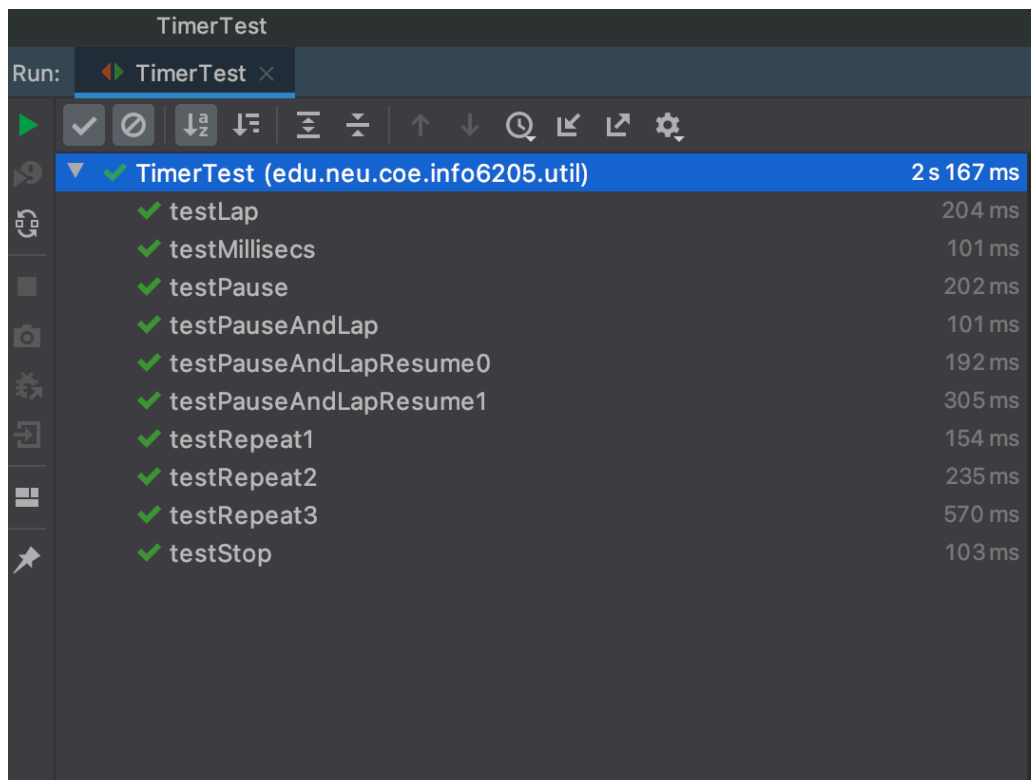





x-axis demonstrates n (size of array), y-axis demonstrates mean run time of the sort with each type of input. We can merge them into one graph to see the differences.



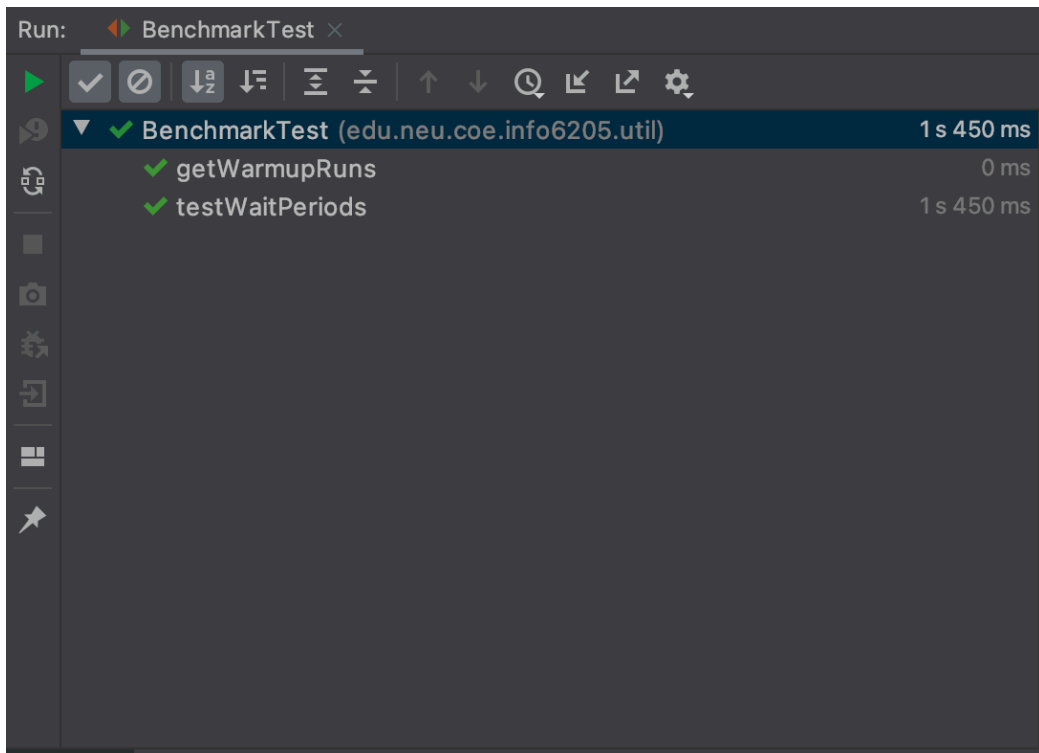
◉ **Unit tests result:**




Run:  BenchmarkTest x

✓ BenchmarkTest (edu.neu.coe.info6205.util) 1 s 450 ms

- ✓ getWarmupRuns 0 ms
- ✓ testWaitPeriods 1 s 450 ms



Run:  InsertionSortTest x

✓ InsertionSortTest (edu.neu.coe.info6205.sort.elementary) 151 ms

- ✓ sort0 48 ms
- ✓ sort1 1 ms
- ✓ sort2 6 ms
- ✓ sort3 2 ms
- ✓ testMutatingInsertionSort 92 ms
- ✓ testStaticInsertionSort 2 ms

