Cheng Yan (001090569)

Program Structures & Algorithms

Fall 2021

Assignment No. 2

⊙ Task (List down the tasks performed in the Assignment)

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.
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.
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.
4. Report on your observations and show screenshots of the runs and also the unit tests.

⊙ Relationship Conclusion:

1. Time cost t's relation with different array types, under same array length, is:

$$t(ordered) < t(partially - ordered) < t(random) < t(reverse - ordered)$$

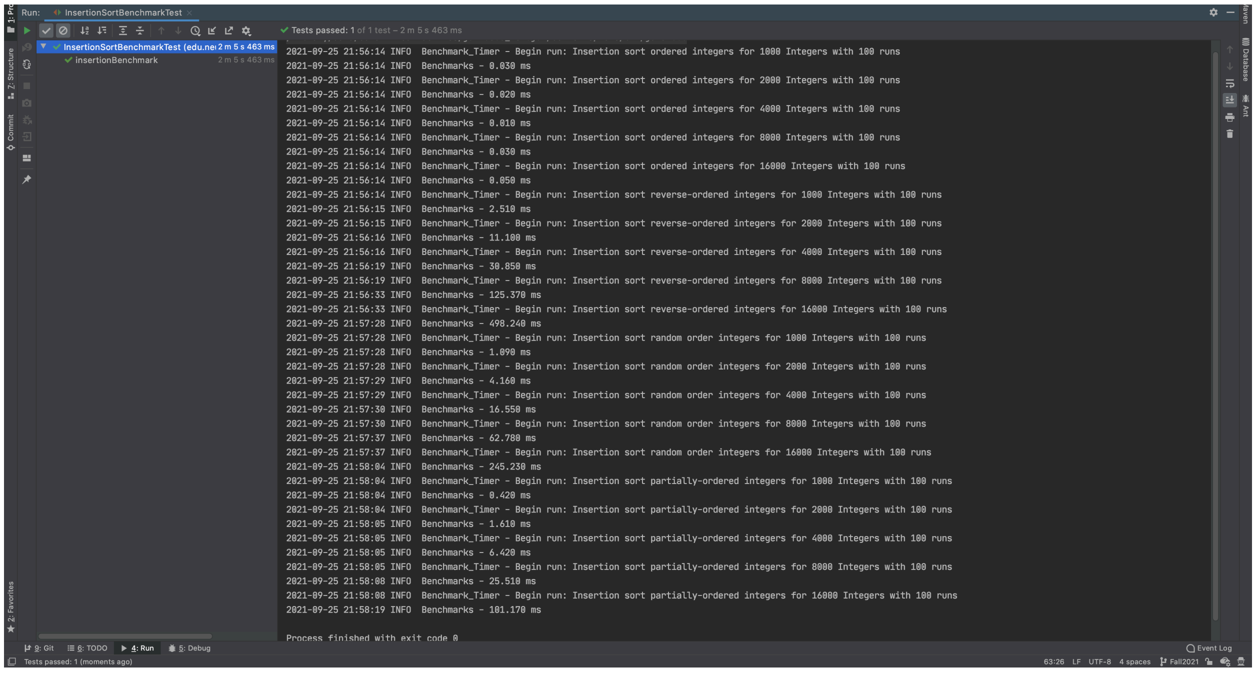
2. When we are using the doubling method for choosing N, log t increases linearly with log N, and we have the following k value, which is the growth rate of insertion sort algorithms:

$$\bar{k} = 0.51383778 \approx \frac{1}{2}$$

*As for the log t with ordered array on the log/log graph, array size and time t are both too small, it is not accurate. So, I do not take using it to calculate the average value of k.

⊙ Evidence to support the conclusion:

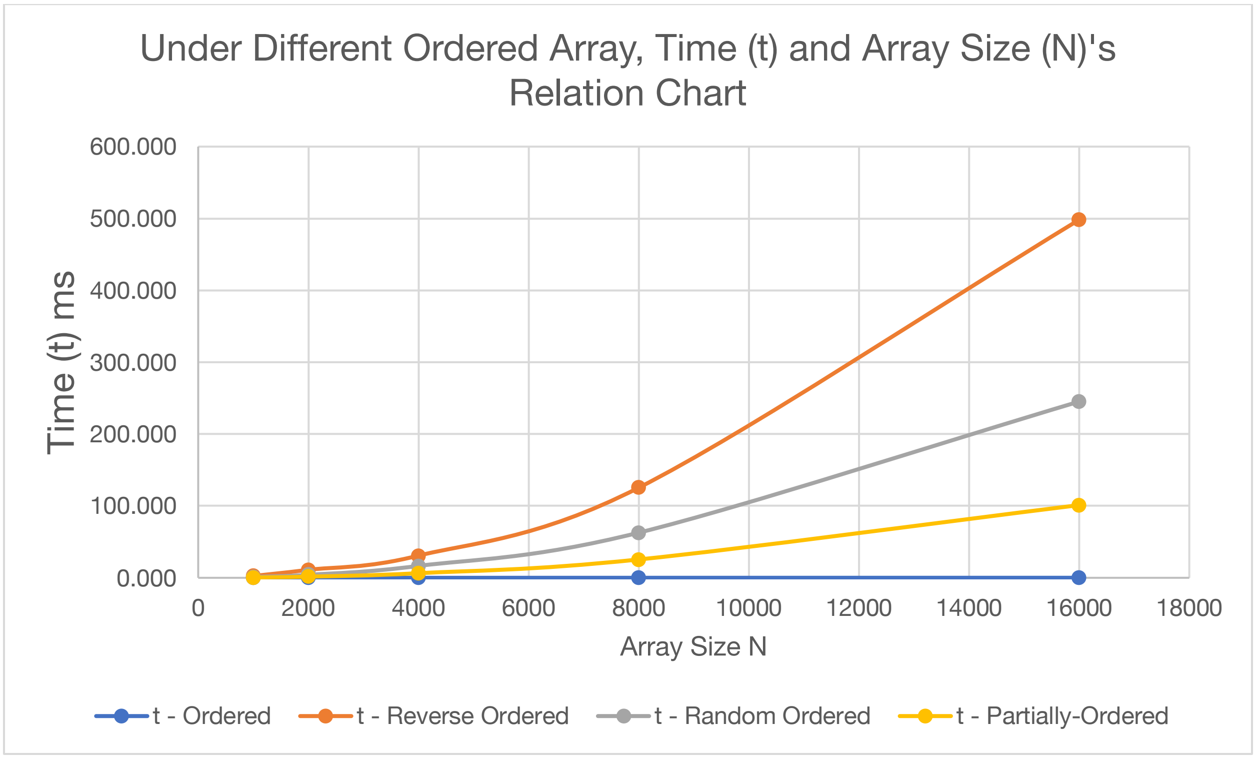
1. Output/Test Result (Snapshot of Code output in the terminal)


Run: InsertionSortBenchmarkTest
Tests passed: 1 of 1 test – 2 m 5 s 463 ms
InsertionSortBenchmarkTest (edu.ne 2 m 5 s 463 ms
insertionBenchmark 2 m 5 s 463 ms
Tests passed: 1 (moments ago)

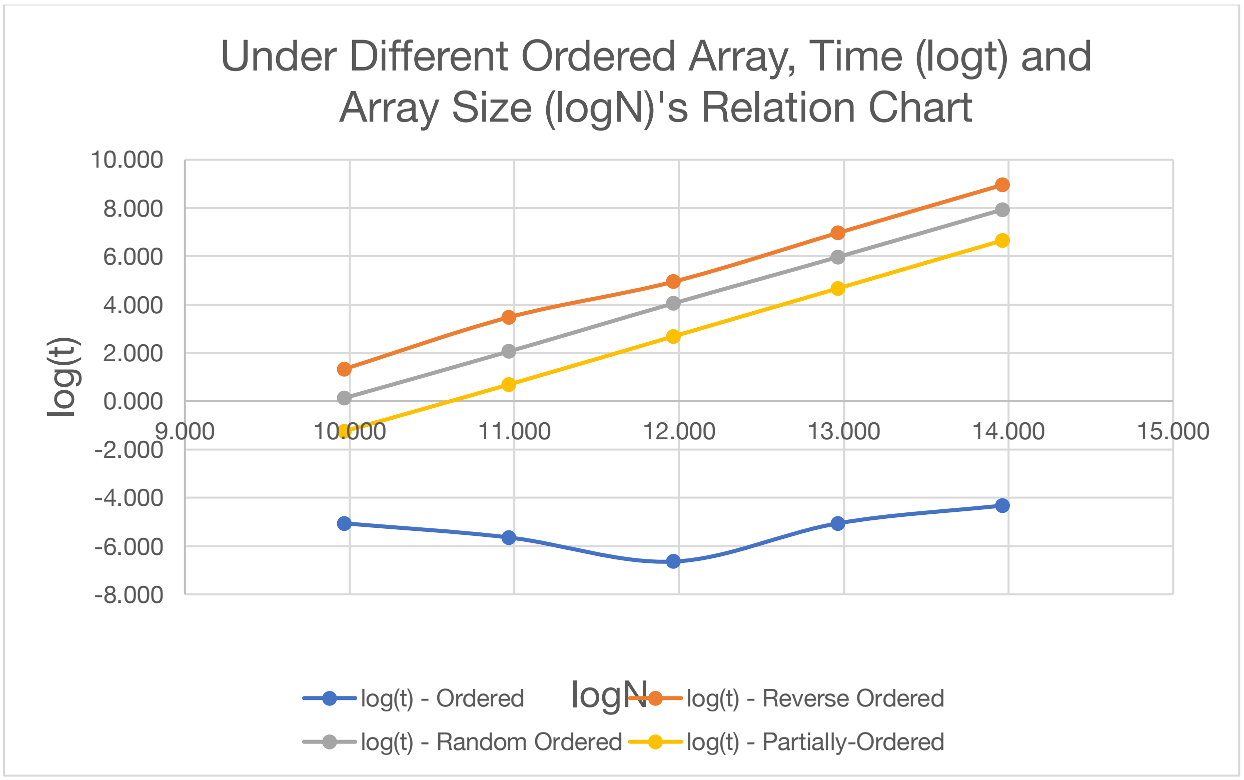
```
2021-09-25 21:56:14 INFO  Benchmark_Timer - Begin run: Insertion sort ordered integers for 1000 Integers with 100 runs
2021-09-25 21:56:14 INFO  Benchmarks - 0.030 ms
2021-09-25 21:56:14 INFO  Benchmark_Timer - Begin run: Insertion sort ordered integers for 2000 Integers with 100 runs
2021-09-25 21:56:14 INFO  Benchmarks - 0.020 ms
2021-09-25 21:56:14 INFO  Benchmark_Timer - Begin run: Insertion sort ordered integers for 4000 Integers with 100 runs
2021-09-25 21:56:14 INFO  Benchmarks - 0.010 ms
2021-09-25 21:56:14 INFO  Benchmark_Timer - Begin run: Insertion sort ordered integers for 8000 Integers with 100 runs
2021-09-25 21:56:14 INFO  Benchmarks - 0.030 ms
2021-09-25 21:56:14 INFO  Benchmark_Timer - Begin run: Insertion sort ordered integers for 16000 Integers with 100 runs
2021-09-25 21:56:14 INFO  Benchmarks - 0.050 ms
2021-09-25 21:56:14 INFO  Benchmark_Timer - Begin run: Insertion sort reverse-ordered integers for 1000 Integers with 100 runs
2021-09-25 21:56:15 INFO  Benchmarks - 2.510 ms
2021-09-25 21:56:15 INFO  Benchmark_Timer - Begin run: Insertion sort reverse-ordered integers for 2000 Integers with 100 runs
2021-09-25 21:56:16 INFO  Benchmarks - 11.100 ms
2021-09-25 21:56:16 INFO  Benchmark_Timer - Begin run: Insertion sort reverse-ordered integers for 4000 Integers with 100 runs
2021-09-25 21:56:19 INFO  Benchmarks - 30.850 ms
2021-09-25 21:56:19 INFO  Benchmark_Timer - Begin run: Insertion sort reverse-ordered integers for 8000 Integers with 100 runs
2021-09-25 21:56:33 INFO  Benchmarks - 125.370 ms
2021-09-25 21:56:33 INFO  Benchmark_Timer - Begin run: Insertion sort reverse-ordered integers for 16000 Integers with 100 runs
2021-09-25 21:57:28 INFO  Benchmarks - 498.240 ms
2021-09-25 21:57:28 INFO  Benchmark_Timer - Begin run: Insertion sort random order integers for 1000 Integers with 100 runs
2021-09-25 21:57:28 INFO  Benchmarks - 1.090 ms
2021-09-25 21:57:28 INFO  Benchmark_Timer - Begin run: Insertion sort random order integers for 2000 Integers with 100 runs
2021-09-25 21:57:29 INFO  Benchmarks - 4.160 ms
2021-09-25 21:57:29 INFO  Benchmark_Timer - Begin run: Insertion sort random order integers for 4000 Integers with 100 runs
2021-09-25 21:57:30 INFO  Benchmarks - 16.550 ms
2021-09-25 21:57:30 INFO  Benchmark_Timer - Begin run: Insertion sort random order integers for 8000 Integers with 100 runs
2021-09-25 21:57:37 INFO  Benchmarks - 62.780 ms
2021-09-25 21:57:37 INFO  Benchmark_Timer - Begin run: Insertion sort random order integers for 16000 Integers with 100 runs
2021-09-25 21:58:04 INFO  Benchmarks - 245.230 ms
2021-09-25 21:58:04 INFO  Benchmark_Timer - Begin run: Insertion sort partially-ordered integers for 1000 Integers with 100 runs
2021-09-25 21:58:04 INFO  Benchmarks - 0.420 ms
2021-09-25 21:58:04 INFO  Benchmark_Timer - Begin run: Insertion sort partially-ordered integers for 2000 Integers with 100 runs
2021-09-25 21:58:05 INFO  Benchmarks - 1.610 ms
2021-09-25 21:58:05 INFO  Benchmark_Timer - Begin run: Insertion sort partially-ordered integers for 4000 Integers with 100 runs
2021-09-25 21:58:05 INFO  Benchmarks - 6.420 ms
2021-09-25 21:58:05 INFO  Benchmark_Timer - Begin run: Insertion sort partially-ordered integers for 8000 Integers with 100 runs
2021-09-25 21:58:08 INFO  Benchmarks - 25.510 ms
2021-09-25 21:58:08 INFO  Benchmark_Timer - Begin run: Insertion sort partially-ordered integers for 16000 Integers with 100 runs
2021-09-25 21:58:19 INFO  Benchmarks - 101.170 ms

Process finished with exit code 0
```

2. **Graphical Representation(Observations from experiments should be tabulated and analyzed by plotting graphs(usually in excel) to arrive on the relationship conclusion)**

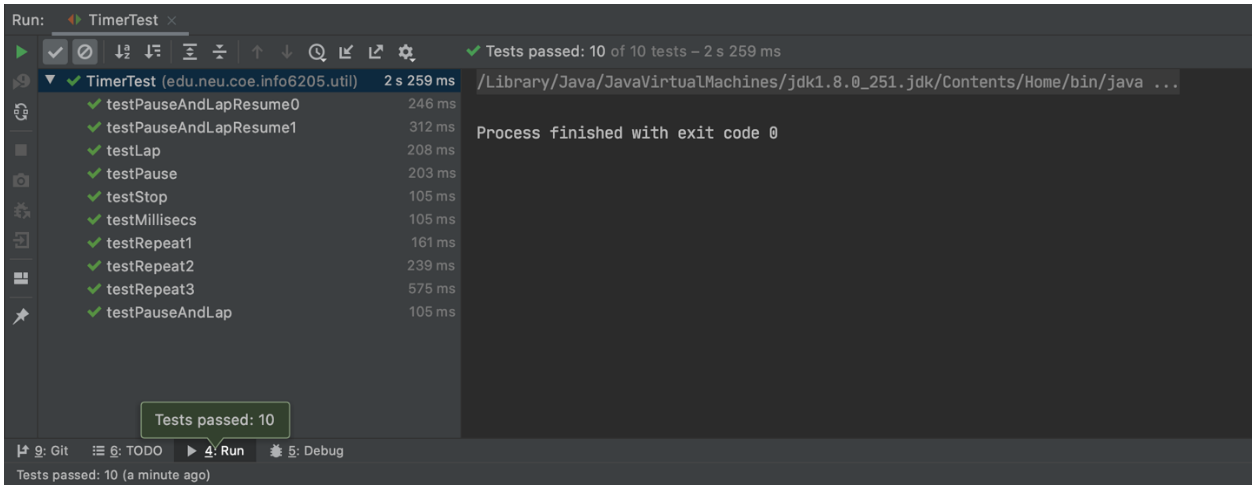
Under Different Ordered Array, Time (t) and Array Size (N)'s Relation Table				
N	**t - Ordered**	**t - Reverse Ordered**	**t - Random Ordered**	**t - Partially-Ordered**
1000	0.030	2.460	1.000	0.410
2000	0.020	9.140	4.040	1.580
4000	0.020	30.240	14.850	6.240
8000	0.030	120.670	59.560	24.250
16000	0.060	484.520	233.490	96.630



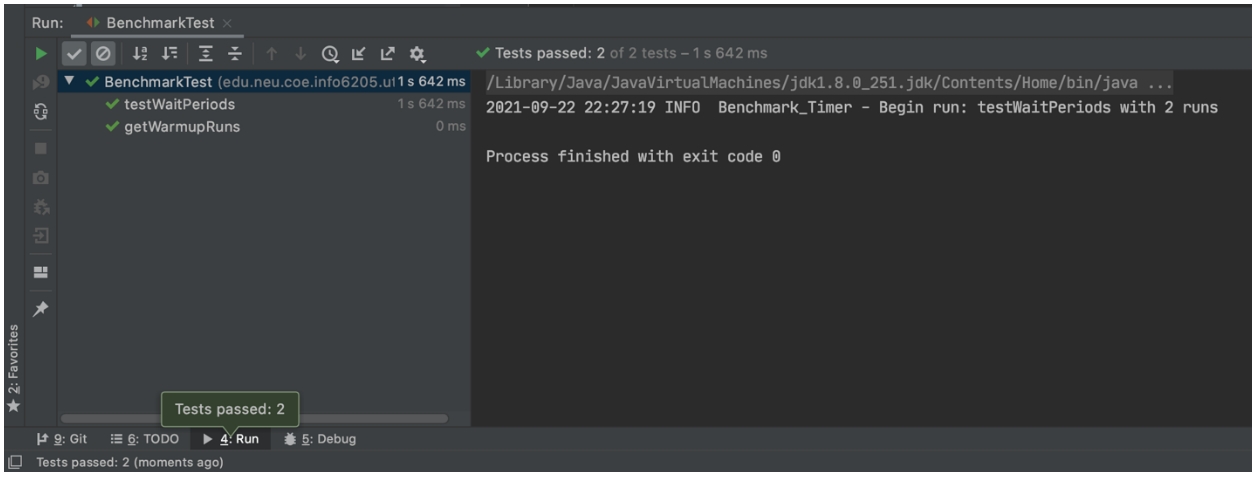
Under Different Ordered Array, Time (log t) and Array Size (log N)'s Relation Chart				
log N	**log t - Ordered**	**log t - Reverse Ordered**	**log t - Random Ordered**	**log t - Partially-Ordered**
9.966	-5.059	1.328	0.124	-1.252
10.966	-5.644	3.472	2.057	0.687
11.966	-6.644	4.947	4.049	2.683
12.966	-5.059	6.970	5.972	4.673
13.966	-4.322	8.961	7.938	6.661



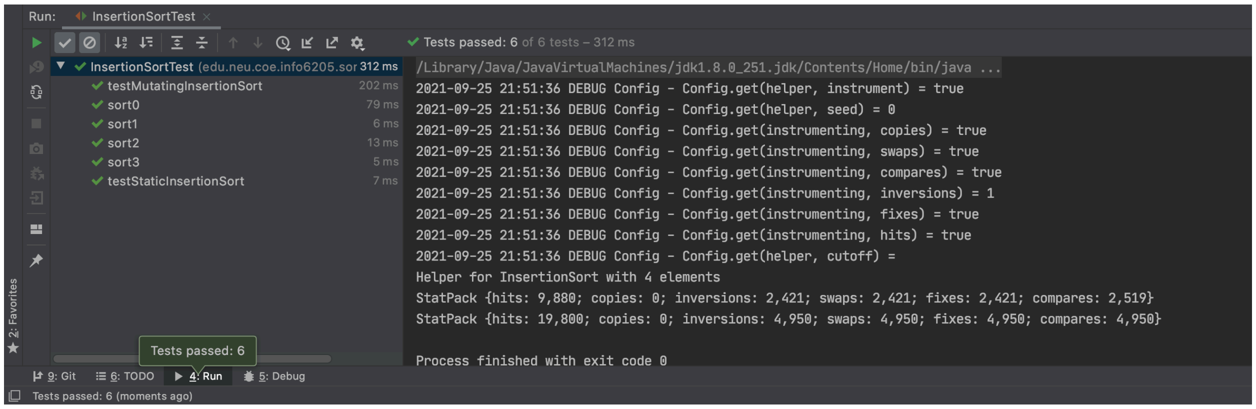
- **Unit tests result:(Snapshot of successful unit test run)**



```
Tests passed: 10 of 10 tests – 2 s 259 ms

TimerTest (edu.neu.coe.info6205.util)   2 s 259 ms
  testPauseAndLapResume0                246 ms
  testPauseAndLapResume1                312 ms
  testLap                               208 ms
  testPause                             203 ms
  testStop                              105 ms
  testMillisecs                         105 ms
  testRepeat1                           161 ms
  testRepeat2                           239 ms
  testRepeat3                           575 ms
  testPauseAndLap                       105 ms

/Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java ...

Process finished with exit code 0

Tests passed: 10
Tests passed: 10 (a minute ago)
```


Run: BenchmarkTest
Tests passed: 2 of 2 tests – 1 s 642 ms
BenchmarkTest (edu.neu.coe.info6205.ut 1 s 642 ms
testWaitPeriods 1 s 642 ms
getWarmupRuns 0 ms
Tests passed: 2
9: Git
6: TODO
4: Run
5: Debug
2: Favorites
Tests passed: 2 (moments ago)

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java ...
2021-09-22 22:27:19 INFO  Benchmark_Timer - Begin run: testWaitPeriods with 2 runs

Process finished with exit code 0
```


Run: InsertionSortTest
Tests passed: 6 of 6 tests – 312 ms
InsertionSortTest (edu.neu.coe.info6205.sor 312 ms
testMutatingInsertionSort 202 ms
sort0 79 ms
sort1 6 ms
sort2 13 ms
sort3 5 ms
testStaticInsertionSort 7 ms
Tests passed: 6
9: Git
6: TODO
4: Run
5: Debug
2: Favorites
Tests passed: 6 (moments ago)

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home/bin/java ...
2021-09-25 21:51:36 DEBUG Config - Config.get(helper, instrument) = true
2021-09-25 21:51:36 DEBUG Config - Config.get(helper, seed) = 0
2021-09-25 21:51:36 DEBUG Config - Config.get(instrumenting, copies) = true
2021-09-25 21:51:36 DEBUG Config - Config.get(instrumenting, swaps) = true
2021-09-25 21:51:36 DEBUG Config - Config.get(instrumenting, compares) = true
2021-09-25 21:51:36 DEBUG Config - Config.get(instrumenting, inversions) = 1
2021-09-25 21:51:36 DEBUG Config - Config.get(instrumenting, fixes) = true
2021-09-25 21:51:36 DEBUG Config - Config.get(instrumenting, hits) = true
2021-09-25 21:51:36 DEBUG Config - Config.get(helper, cutoff) =
Helper for InsertionSort with 4 elements
StatPack {hits: 9,880; copies: 0; inversions: 2,421; swaps: 2,421; fixes: 2,421; compares: 2,519}
StatPack {hits: 19,800; copies: 0; inversions: 4,950; swaps: 4,950; fixes: 4,950; compares: 4,950}

Process finished with exit code 0
```