

INFO 6205-PROGRAM STRUCTURE AND ALGORITHMS

ASSIGNMENT-3

Sai Kashyap Cheruku
NU-ID- 002756849

Task:

The task at hand is to observe the running times of a Insertion Sort algorithm, for random, partially sorted, sorted and Reverse-sorted Arrays.

The timer used in the above analysis is to be configured by the programmer, by writing the methods to return system-tick count and mean millisecond time taken for completing an experiment.

The programmer is also required to write a main class for carrying out the benchmarking process.

Relationship Conclusion:

The Insertion Sort algorithm has an average time complexity of $O(N^2)$.

From the benchmark outputs, it is observed that the quadratic nature of insertion sort algorithm is somewhat agnostic to a sorted input. In case of the other input variations, the quadratic nature is visibly evident.

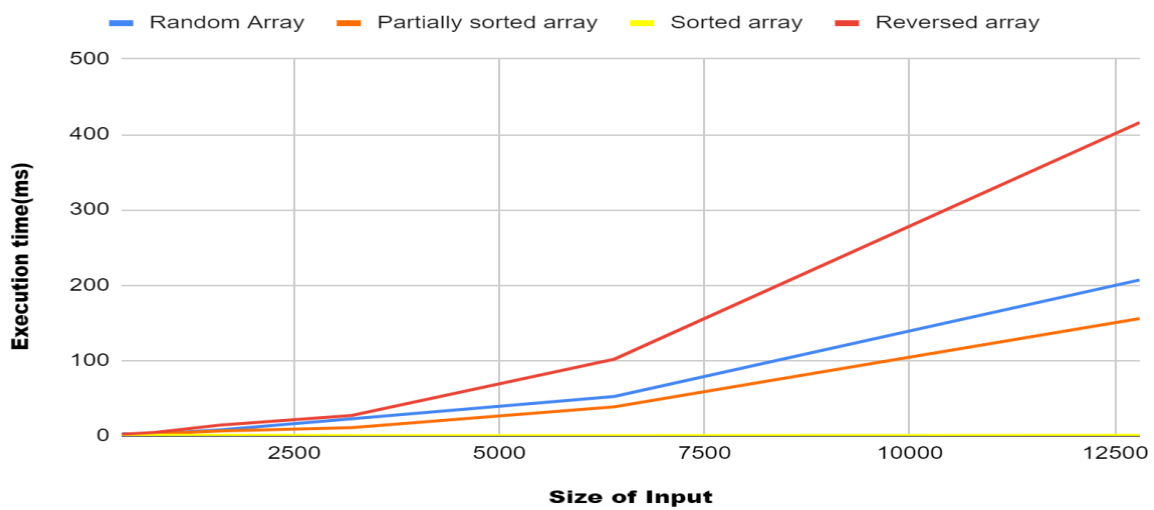
The most time consuming variant is a reverse-sorted array input in which the elements are inserted at their opposite ends to form the sorted output. Such an approach towards an already sorted(without considering the direction of growth) would prove extremely inefficient.

The partially sorted array input performance lies in between the random and reverse-sorted input performances of the algorithm. This is due to the fact that the traversing and swapping operation is only performed on the second half of the array.

Evidence:

N	Random Array	Partially sorted	Sorted array	Reversed array
400	2.8	1.975	1.25	2.225
800	2.95	2.9	0.8	4.575
1600	8.025	6.475	0.9	14.3
3200	22.575	10.85	0.35	27.125
6400	52.3	38.55	0.375	101.7
12800	206.95	155.7	0.525	415.825

Graphical Representations:



Unit Tests:

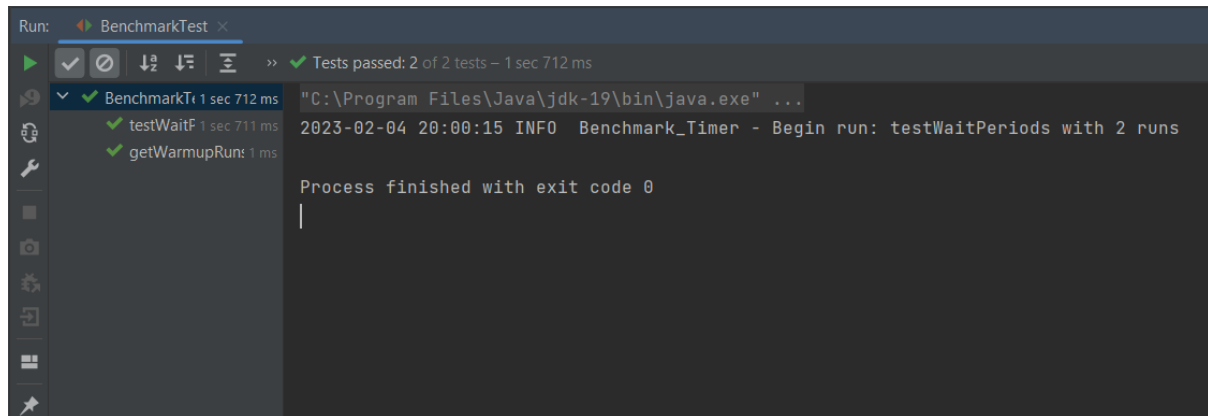
Timer test:

```
Run: TimerTest x
[Icons] [Check] [Close] [Run] [Debug] [Test] [Exit] [Help] [Filter] [Log] [Output] [Error] [Warning] [Info] [Debug] [Test] [Exit] [Help] [Filter] [Log] [Output] [Error] [Warning] [Info]
>> Tests passed: 11 of 11 tests - 3 sec 257 ms

TimerTest (e: 3 sec 257 ms) "C:\Program Files\Java\jdk-19\bin\java.exe" ...
  testPauseAndL 426 ms
  testPauseAndL 319 ms
  testLap 219 ms
  testPause 217 ms
  testStop 108 ms
  testMillisecs 109 ms
  testRepeat1 158 ms
  testRepeat2 311 ms
  testRepeat3 786 ms
  testRepeat4 488 ms
  testPauseAndL 116 ms

Process finished with exit code 0
```

Benchmark Test:



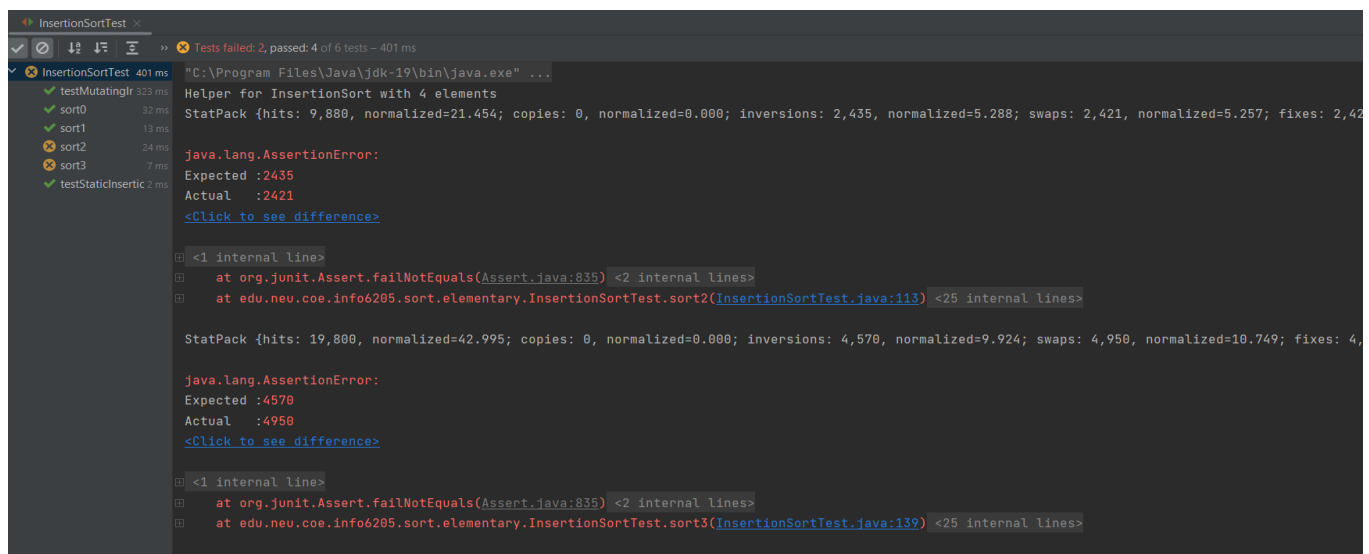
```
Run: BenchmarkTest x
>> Tests passed: 2 of 2 tests - 1 sec 712 ms

BenchmarkTest 1 sec 712 ms
  testWaitF 1 sec 711 ms
  getWarmupRun 1 ms

"C:\Program Files\Java\jdk-19\bin\java.exe" ...
2023-02-04 20:00:15 INFO Benchmark_Timer - Begin run: testWaitPeriods with 2 runs

Process finished with exit code 0
```

Insertion sort Test:



```
InsertionSortTest x
>> Tests failed: 2, passed: 4 of 6 tests - 401 ms

InsertionSortTest 401 ms
  testMutatingIn 323 ms
  sort0 32 ms
  sort1 13 ms
  sort2 24 ms
  sort3 7 ms
  testStaticInsert 2 ms

"C:\Program Files\Java\jdk-19\bin\java.exe" ...
Helper for InsertionSort with 4 elements
StatPack {hits: 9,880, normalized=21.454; copies: 0, normalized=0.000; inversions: 2,435, normalized=5.288; swaps: 2,421, normalized=5.257; fixes: 2,421}

java.lang.AssertionError:
Expected :2435
Actual   :2421
<Click to see difference>

<1 internal line>
at org.junit.Assert.failNotEquals(Assert.java:835) <2 internal lines>
at edu.neu.coe.info6205.sort.elementary.InsertionSortTest.sort2(InsertionSortTest.java:113) <25 internal lines>

StatPack {hits: 19,800, normalized=42.995; copies: 0, normalized=0.000; inversions: 4,570, normalized=9.924; swaps: 4,950, normalized=10.749; fixes: 4,950}

java.lang.AssertionError:
Expected :4570
Actual   :4950
<Click to see difference>

<1 internal line>
at org.junit.Assert.failNotEquals(Assert.java:835) <2 internal lines>
at edu.neu.coe.info6205.sort.elementary.InsertionSortTest.sort3(InsertionSortTest.java:139) <25 internal lines>
```

Main program output:

Started benchmarking procedure

Random Array Sorting 1 n = 400

2023-02-04 19:21:37 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs

Mean lap time: 2.8

Partially Sorted Array Sorting 1 n = 400

2023-02-04 19:21:37 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs

Mean lap time: 1.975

Fully Sorted Array Sorting 1 n = 400

2023-02-04 19:21:37 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs

Mean lap time: 1.25

Reverse Sorted Array Sorting 1 n = 400

2023-02-04 19:21:37 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs

Mean lap time: 2.225

Random Array Sorting 2 n = 800

2023-02-04 19:21:38 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs

Mean lap time: 2.95
Partially Sorted Array Sorting 2 n = 800
2023-02-04 19:21:38 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 2.9
Fully Sorted Array Sorting 2 n = 800
2023-02-04 19:21:38 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 0.8
Reverse Sorted Array Sorting 2 n = 800
2023-02-04 19:21:38 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 4.575
Random Array Sorting 3 n = 1600
2023-02-04 19:21:38 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 8.025
Partially Sorted Array Sorting 3 n = 1600
2023-02-04 19:21:38 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 6.475
Fully Sorted Array Sorting 3 n = 1600
2023-02-04 19:21:39 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 0.9
Reverse Sorted Array Sorting 3 n = 1600
2023-02-04 19:21:39 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 14.3
Random Array Sorting 4 n = 3200
2023-02-04 19:21:39 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 22.575
Partially Sorted Array Sorting 4 n = 3200
2023-02-04 19:21:40 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 10.85
Fully Sorted Array Sorting 4 n = 3200
2023-02-04 19:21:41 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 0.35
Reverse Sorted Array Sorting 4 n = 3200
2023-02-04 19:21:41 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 27.125
Random Array Sorting 5 n = 6400
2023-02-04 19:21:42 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 52.3
Partially Sorted Array Sorting 5 n = 6400
2023-02-04 19:21:44 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 38.55
Fully Sorted Array Sorting 5 n = 6400
2023-02-04 19:21:46 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 0.375
Reverse Sorted Array Sorting 5 n = 6400
2023-02-04 19:21:46 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs
Mean lap time: 101.7
Random Array Sorting 6 n = 12800
2023-02-04 19:21:51 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs

Mean lap time: 206.95

Partially Sorted Array Sorting 6 n = 12800

2023-02-04 19:22:00 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs

Mean lap time: 155.7

Fully Sorted Array Sorting 6 n = 12800

2023-02-04 19:22:07 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs

Mean lap time: 0.525

Reverse Sorted Array Sorting 6 n = 12800

2023-02-04 19:22:07 INFO Benchmark_Timer - Begin run: Insertion Sort with 40 runs

Mean lap time: 415.825

Benchmarking completed successfully