Program Structures & Algorithms

Spring 2022

Assignment Number 2

Name: Kanishk Bimalkumar Bhatia

NUID: 001580259

- Task:

1. Implement three methods of a class called timer.
    a. Implement method *getClock* by using function *System.nanoTime* function.
    b. Implement *toMillisecs* method by converting nanosecond to millisecond.
    c. Implement the *repeat* method by calculating the average milliseconds per repetition. This was calculated from *preFunction*, applying the function till post function.
2. InsertSort: Implement the sort method in *insertionSort* class by running nested for loop and swapping smaller numbers to the left and larger to the right.
3. Executing benchmark on Insertion Sort: Sorted and benchmarked timings of random, ordered, partially ordered and reverse-ordered arrays of different lengths and note the observation.

- Evidence:

  Insertion Sort is benchmarked for 4 types of arrays:
    a. Random Array
    b. Ordered Array
    c. Reversed Array
    d. Partially Ordered Array

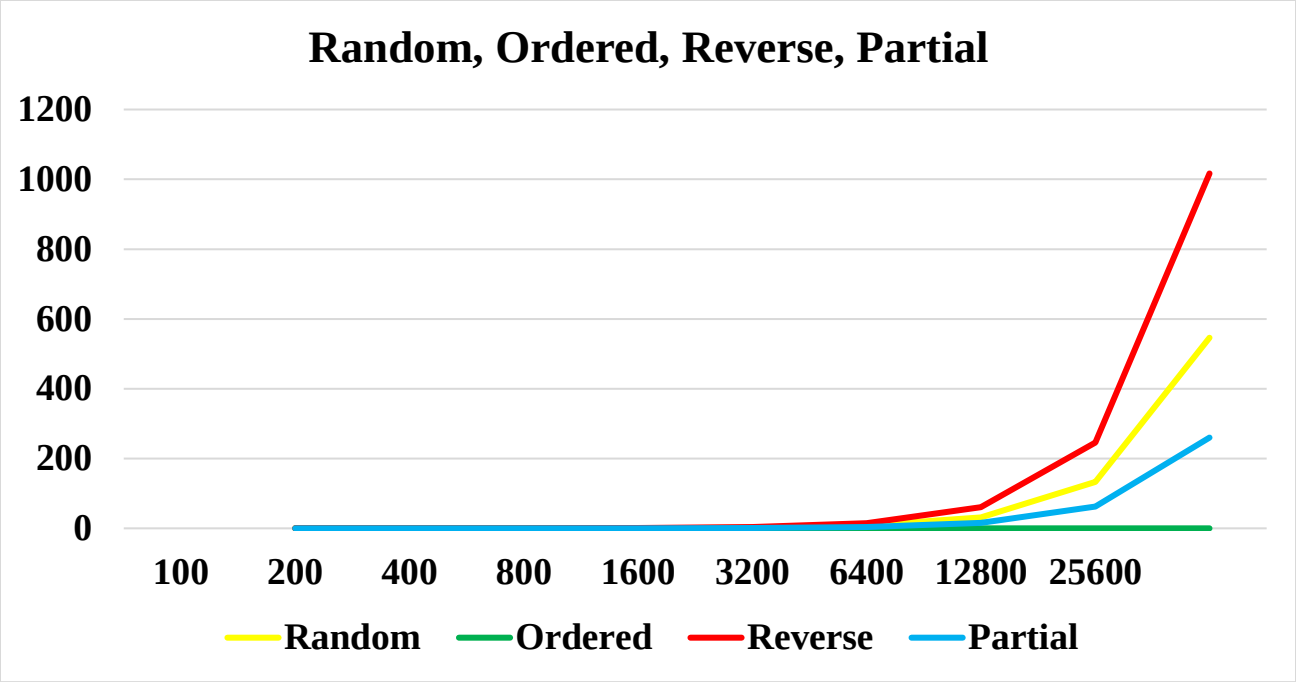
Each type of array is tested for different lengths, doubling it each time, hence making the lengths:

a. 100
b. 200
c. 400
d. 800
e. 1600
f. 3200
g. 6400
h. 12800
i. 25600

For each length variation, the experiment is run 10 times and the results are in the files named:

a. Assignment 2 Data.pdf
b. Assignment 2 Benchmark.xlsx

N	Random	Ordered	Reverse	Partial
100	0.1	0.0	0.2	0.0
200	0.1	0.0	0.2	0.0
400	0.1	0.0	0.2	0.0
800	0.4	0.0	1.0	0.2
1600	1.8	0.0	3.6	0.9
3200	8.0	0.0	14.3	3.4
6400	31.6	0.0	60.9	15.3
12800	132.7	0.0	245.4	62.3
25600	546.1	0.0	1016.8	260.0



- Conclusion:
  From the data presented above, we can conclude that the types of arrays are based on the time taken by each of them:

  Reversed Array > Random Array > Partially Ordered Array > Ordered Array

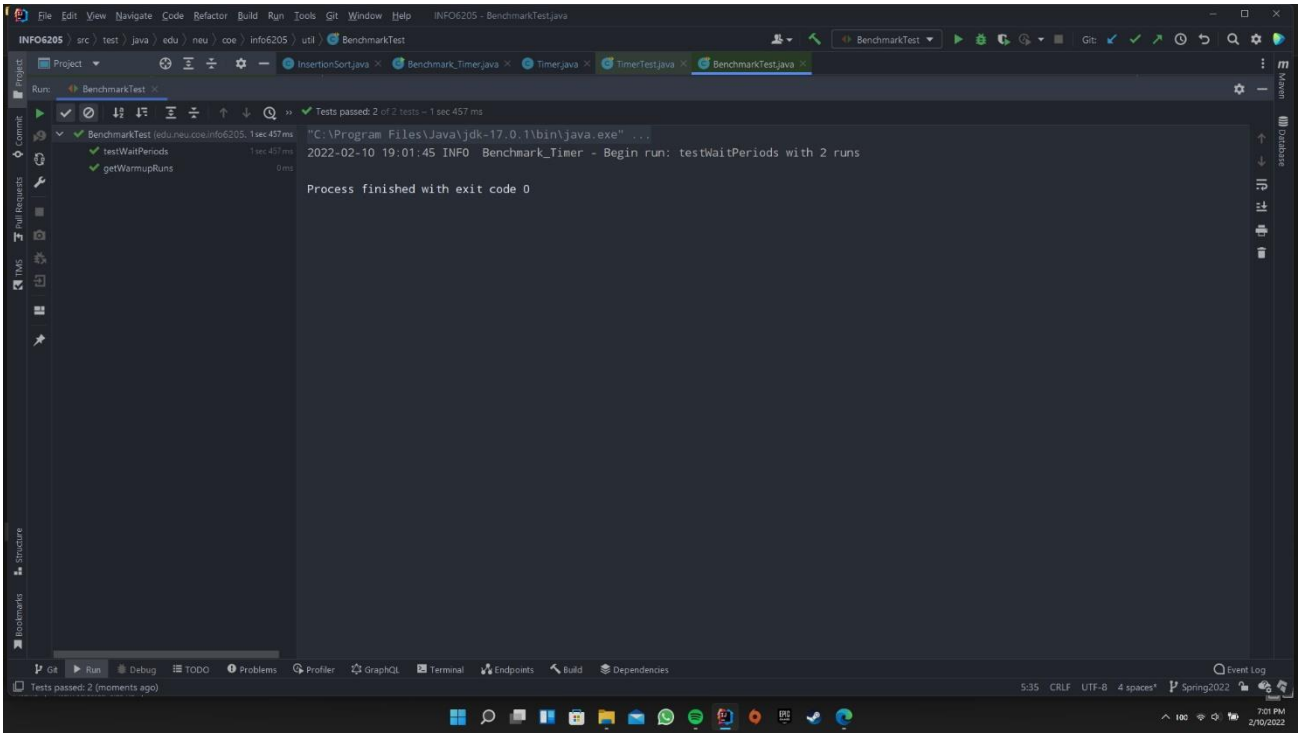
  Another conclusion drawn from the data is:
  As the length of an array doubles, the mean time-taken also increases drastically.

  For an Ordered Array, the time taken is directly proportional to O(n) time. As the array goes out of order, the time taken increases. It increases proportionally to the amount of disorderliness. This effect can be seen in the data for Partially Ordered and Random Arrays.

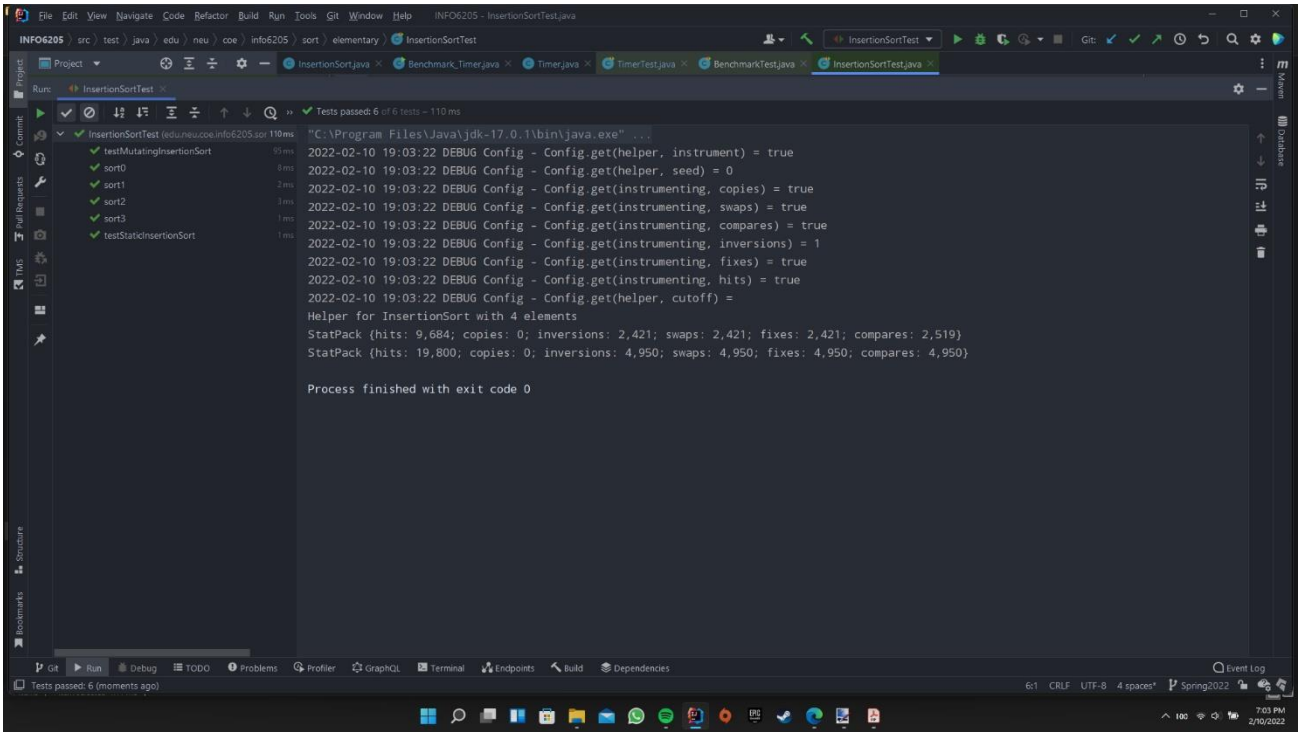
  For the worst-case scenario, that is the Reversed Array. The time taken is directly proportional to the time $O(n^2)$.

  For any array type, the mean time is proportional to the value (k*n), where k is a constant that varies from 1 to n.

- Test screenshot:
    1. Benchmark Test



    2. Insertion Sort Test



3. Timer Test on Windows and Mac