

Program Structures and Algorithms Spring 2024

NAME: Kartikey Vijayakumar Hebbar

NUID: 002276938

GITHUB LINK:

<https://github.com/kartikeyhebbbar/INFO6205/tree/Spring2024/src/main/java/edu/neu/coe/info6205/sort/elementary>

Task: Assignment 3 (Benchmark)

Relationship Conclusion:

Refer to the attached spreadsheet (Assignment_3_Statistical_Analysis.xlsx) with the detailed analysis of Time taken by Insertion Sort to sort a random array, an ordered array, partially ordered array and a reverse ordered array with respect to the increasing size of the array and number of repetitions performed on each array to get the mean time taken.

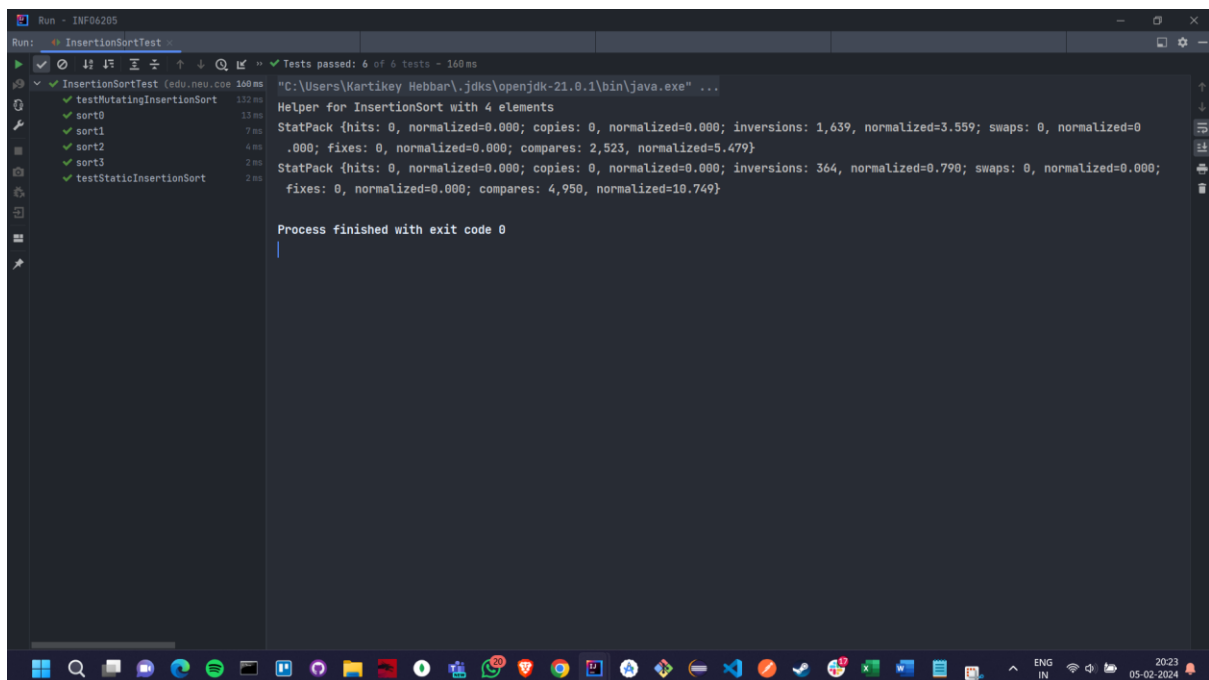
A general observation could be drawn that, with increasing number of repetitions (n), the mean time taken for any type of array starts decreasing. **The higher the value of n, the lower will be the mean time taken by the Insertion Sort algorithm for any given array.**

Another observation in this experiment is that with the increasing array size, the mean time taken by the algorithm reduces for the same value of n which shows that the algorithm might be performing better for larger array sizes.

Evidence to support that conclusion:

Refer to the attached spreadsheet (Assignment_3_Statistical_Analysis.xlsx) for a detailed report of the analysis to support the above conclusion.

Unit Test Screenshots: Screenshot of the successful unit tests:



The screenshot shows a Java IDE with a test runner window. The test runner displays the following output:

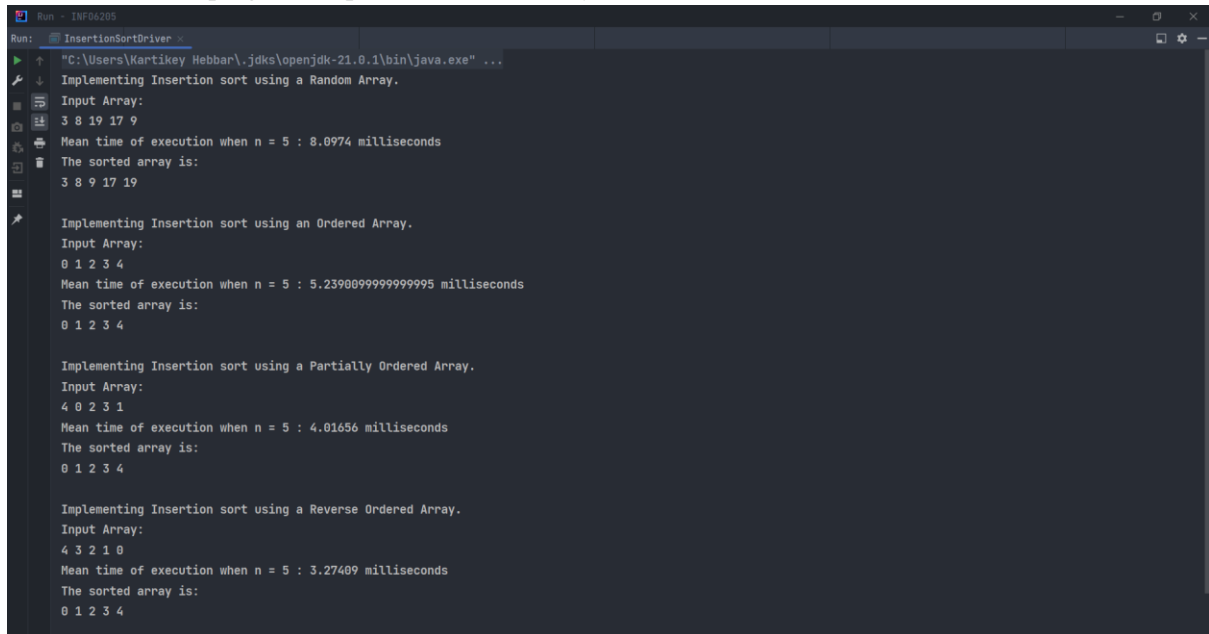
```
Run: InsertionSortTest
Tests passed: 6 of 6 tests - 160ms

InsertionSortTest (edu.neu.coe) 160ms
  testMutatingInsertionSort 132ms
    sort0 13ms
    sort1 7ms
    sort2 4ms
    sort3 2ms
  testStaticInsertionSort 2ms

Helper for InsertionSort with 4 elements
StatPack {hits: 0, normalized=0.000; copies: 0, normalized=0.000; inversions: 1,639, normalized=3.559; swaps: 0, normalized=0.000; fixes: 0, normalized=0.000; compares: 2,523, normalized=5.479}
StatPack {hits: 0, normalized=0.000; copies: 0, normalized=0.000; inversions: 364, normalized=0.790; swaps: 0, normalized=0.000; fixes: 0, normalized=0.000; compares: 4,950, normalized=10.749}

Process finished with exit code 0
```

Screenshot of the program output when size of array=5 and n=5,



```
Run: InsertionSortDriver
"C:\Users\Kartikey Hebbar\.jdk\openjdk-21.0.1\bin\java.exe" ...
Implementing Insertion sort using a Random Array.
Input Array:
3 8 19 17 9
Mean time of execution when n = 5 : 8.0974 milliseconds
The sorted array is:
3 8 9 17 19

Implementing Insertion sort using an Ordered Array.
Input Array:
0 1 2 3 4
Mean time of execution when n = 5 : 5.2390099999999995 milliseconds
The sorted array is:
0 1 2 3 4

Implementing Insertion sort using a Partially Ordered Array.
Input Array:
4 0 2 3 1
Mean time of execution when n = 5 : 4.01656 milliseconds
The sorted array is:
0 1 2 3 4

Implementing Insertion sort using a Reverse Ordered Array.
Input Array:
4 3 2 1 0
Mean time of execution when n = 5 : 3.27409 milliseconds
The sorted array is:
0 1 2 3 4
```