

Akash Sinha
NU ID: 001344425
Assignment 2

1. Code ->

<https://github.com/AkashSinha24/INFO6205>

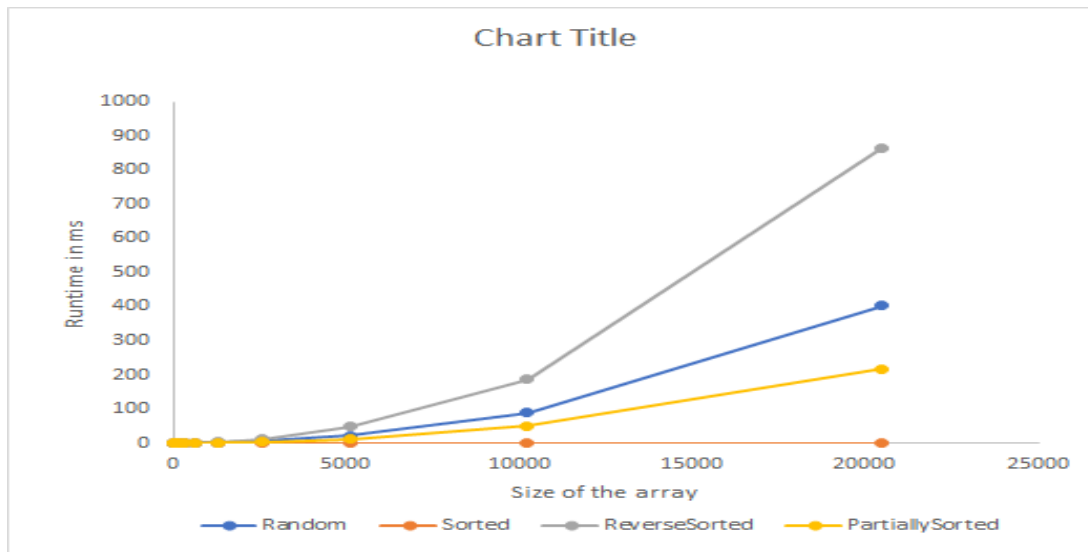
- *Assignment2.java File -> /src/main/java/edu/neu/coe/info6205/Assignment2.java*
- *Results -> resultsAssignment2/insertion_sort/insertionSortResults.csv*
- *Insertion Sort Code*
/src/main/java/edu/neu/coe/info6205/sort/simple/InsertionSort.java
- *Insertion Sort Basic Code*
/src/main/java/edu/neu/coe/info6205/sort/simple/InsertionSortBasic.java
- *Timer Code*
/src/main/java/edu/neu/coe/info6205/util/Timer.java

2. Screenshots and Evidences:

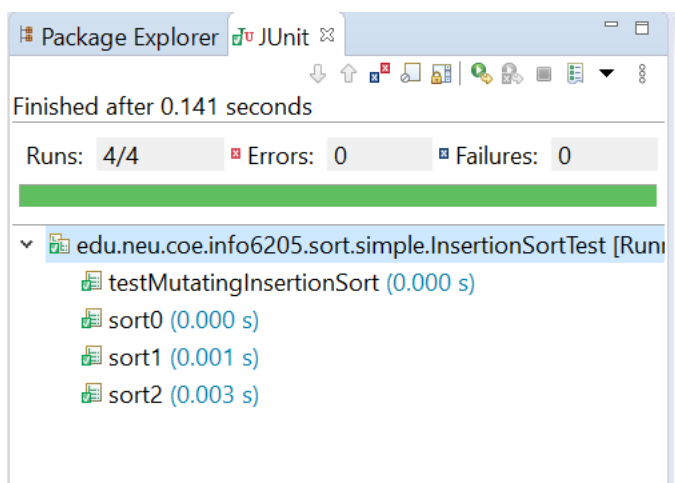
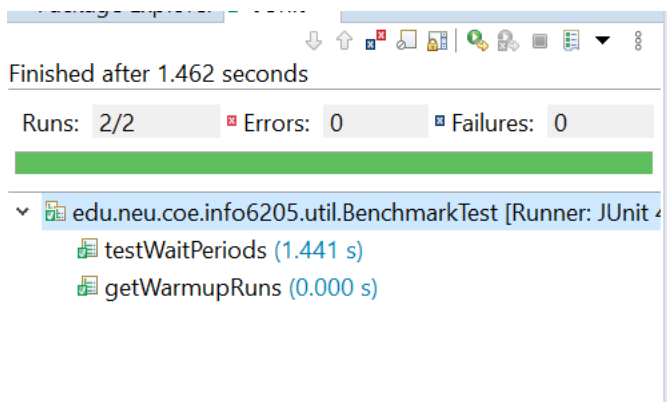
The following table shows the mean time taken by N size of input arrays when they are random, sorted, reverse or partially sorted.

n	Random	Sorted	ReverseSorted	PartiallySorted
20	0.017636	0.001957	0.012336	0.007843
40	0.027021	0.00255	0.034886	0.010443
80	0.030707	0.0017	0.08885	0.010579
160	0.0708	0.002221	0.150564	0.02155
320	0.259179	0.003714	0.248886	0.044364
640	0.339571	0.0028	0.692093	0.174114
1280	1.388864	0.004664	2.898821	0.709336
2560	5.599571	0.010286	11.01576	2.831021
5120	22.61807	0.017629	47.54827	12.34509
10240	89.20018	0.035193	187.6463	50.59708
20480	403.0968	0.089171	865.4882	217.5173

- The chart which we can understand from the above result of my code is:



- Unit Tests Result screenshots of Benchmark Test and Insertion Sort Test



3. Conclusion:

- Random sorted array also grows exponentially but not as much as that of reverse sorted.
- An already sorted array takes negligible amount of time to get sorted, even though the size of input array increases rapidly.
- Reverse sorted grow exponentially and takes most time to execute. It is of the order of N^2 .
- Partially sorted also grows exponentially to the order of N^2 , but depends its curve depends upon the number of initial elements sorted. The more percentage of elements sorted initially; the less duration taken to sort.