Bench marking sorting algorithm

Conclusions

Array with random arrangement of numbers

Conclusion: In the case when array consist of random arrangement of numbers, as the length of the array increases sorting time factor between Selection Sort and Insertion Sort decreases.

For smaller array Insertion Sort is twice as efficient from Selection sort but as the length of array increases efficiency decreases but at any point Insertion Sort is faster than selection sort.

sno	Random	
n=	Selection	Insertion
100	138798.84	117824.28
200	440682.49	183296.75
400	552767.54	271212.74
600	780853.01	509553.37
700	759383.73	646596.48
800	916846.41	843195.46
1000	1225182.98	1161651.18

Sorted Array

Conclusion: When an ordered array is sorted there is a huge factor difference between Selection Sort and Insertion Sort.

For smaller array Insertion sort is faster compared to Selection Sort.

For big sorted array Insertion Sort is much faster.

sno	Sorted	
n=	Selection	Insertion
100	122687.41	5237.04
200	413651.46	9571.65
400	565843.15	16745.72
600	746089.12	17655.71
700	705223.48	23085.27
800	898598.04	32532.38
1000	1024445.68	27604.96

Reverse Array

Conclusion: For the reverse ordered array Insertion Sort is faster than Selection sort for the small and big arrays. But in for the medium size array Insertion Sort is a way faster than Selection sort.

sno	Reverse	
n=	Selection	Insertion
100	162031.44	156307.3
200	347922.2	212340.33
400	517924.39	424457.87
600	807631.05	907312.48
700	986423.36	1372628.43
800	1021266.6	1447819.93
1000	1212586.9	2275712.06