



Analysis of Sorting Algorithms

Objective:

To calculate the execution time of different sorting algorithms and on the basis of that analyze and find the best and worst algorithm in different cases.

Array Size:

100000

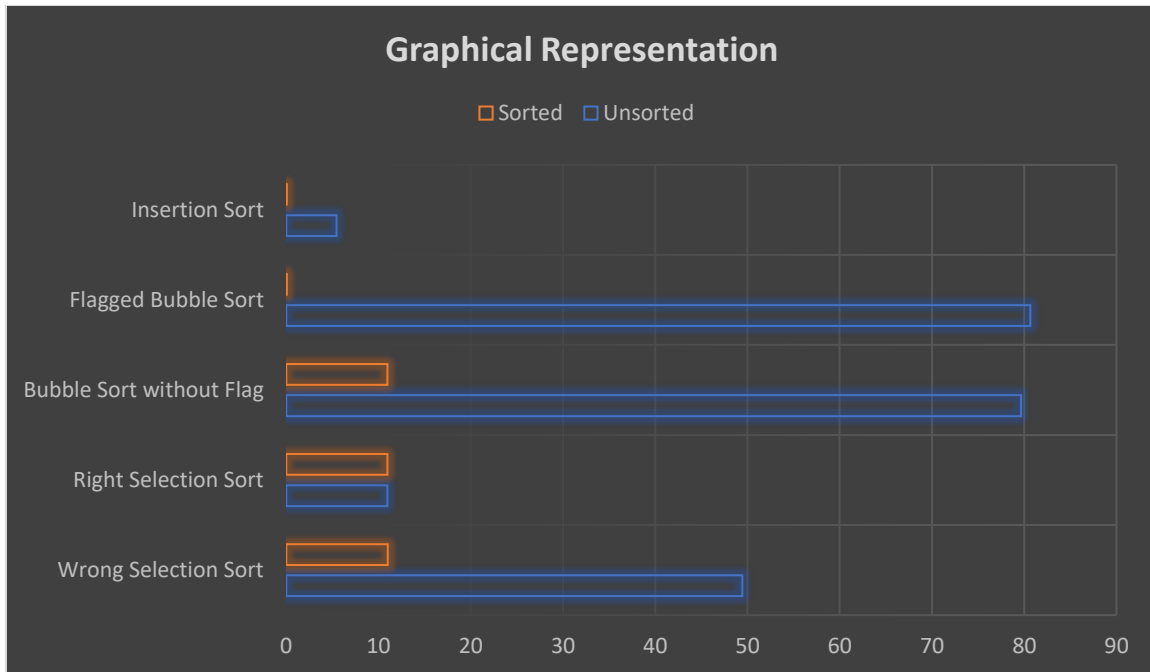
Algorithms:

- ✚ Wrong Selection Sort
- ✚ Right Selection Sort
- ✚ Bubble Sort without Flag
- ✚ Flagged Bubble Sort
- ✚ Insertion Sort

Observations:

Tabular Representation:

Algorithm	Array Size	Array Type	Number of Runs	Avg. Time Taken(s)
Wrong Selection Sort	100000	Unsorted	10	49.4335
Wrong Selection Sort	100000	Sorted	10	11.0186
Right Selection Sort	100000	Unsorted	10	10.9685
Right Selection Sort	100000	Sorted	10	10.9642
Bubble Sort without Flag	100000	Unsorted	10	79.6545
Bubble Sort without Flag	100000	Sorted	10	10.9627
Flagged Bubble Sort	100000	Unsorted	10	80.6557
Flagged Bubble Sort	100000	Sorted	10	0.0002473
Insertion Sort	100000	Unsorted	10	5.44709
Insertion Sort	100000	Sorted	10	0.00031715



Note: The results may vary based on machine's performance.

Discussion:

Best Case Analysis:

In the best case (sorted array), the insertion and flagged bubble sort take much lesser time as compared to the other algorithms because their asymptotic notation is $\Omega(n)$ while the other algorithms' asymptotic notation is $\Omega(n^2)$. In algorithms, except insertion and flagged bubble sort, there is no check or flag or the inner loop and it will always execute despite the condition but in insertion and flagged bubble sort, there are checks on inner loop and it will only execute if some part of array is unsorted, otherwise it will not enter the loop. That's why insertion and flagged bubble sort are more efficient than other algorithms in best case.

Worst Case Analysis:

In the worst case (unsorted array), the insertion sort takes much lesser time as compared to the other algorithms. In other algorithms, except insertion sort, there are at most $n - 1$ swappings in the inner loop while insertion sort uses assignment statements and it will definitely take lesser time than swapping as there are three assignment statements in one swap. That's why insertion sort is more efficient than other algorithms in worst case.

Conclusion:

After the analysis of both best and worst cases of these algorithms, I concluded that the insertion sort is the best sorting algorithm to use for either case, among these five algorithms as it takes least time to sort an array.