

# CSCI – B505 Applied Algorithms

## Assignment – 1

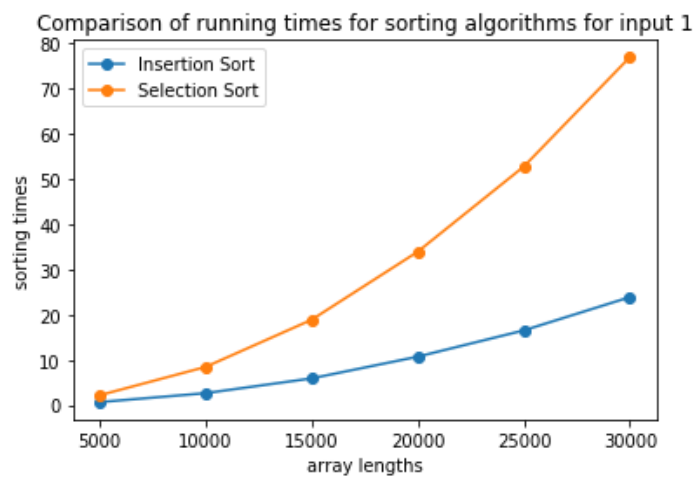
Submitted by – Vishal Singh (singhvis)

### Input 1

#### Runtimes-

Array Size	Insertion Sort runtimes (in sec)	Selection Sort runtimes (in sec)
5000	0.66907	2.1796
10000	2.64246	8.43661
15000	5.90244	18.84102
20000	10.69248	33.84813
25000	16.48671	52.62037
30000	23.82597	76.73722

#### Plot-

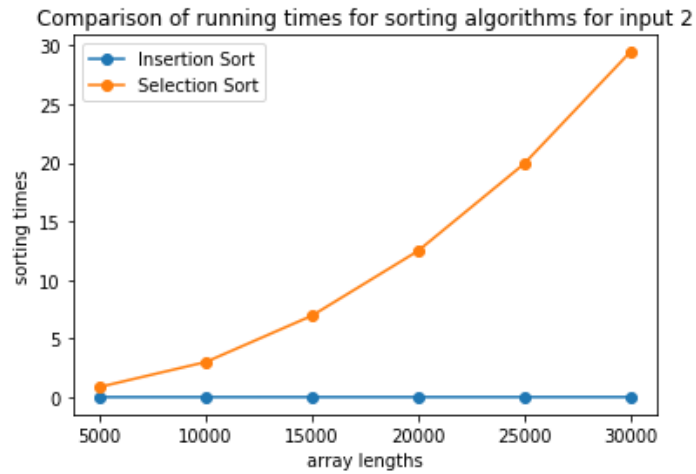


### Input 2

#### Runtimes-

Array Size	Insertion Sort runtimes (in sec)	Selection Sort runtimes (in sec)
5000	0.0009	0.87217
10000	0.00186	2.99814
15000	0.00272	6.95068
20000	0.00362	12.47518
25000	0.00457	19.9019
30000	0.00554	29.43347

Plot-

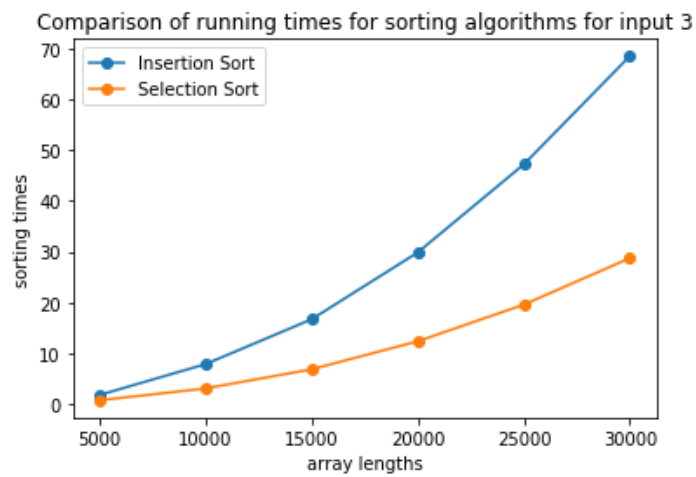


### Input 3

Runtime-

Array Size	Insertion Sort runtimes (in sec)	Selection Sort runtimes (in sec)
5000	1.81255	0.7658
10000	7.88961	3.10076
15000	16.71612	6.86434
20000	29.85541	12.37032
25000	47.17497	19.5861
30000	68.48699	28.7256

Plot-

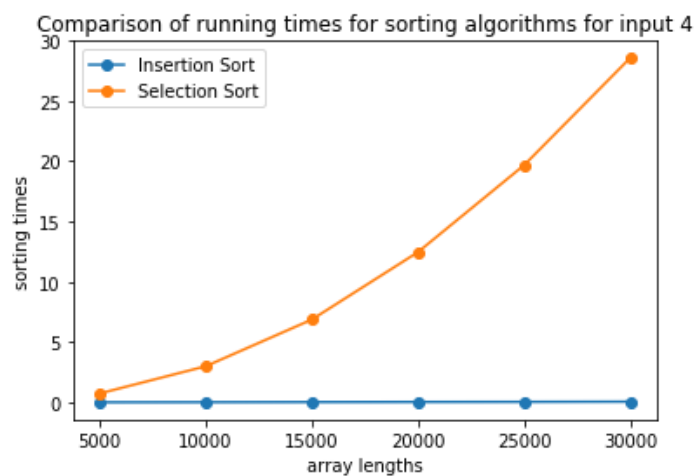


### Input 4

Runtimes-

Array Size	Insertion Sort runtimes (in sec)	Selection Sort runtimes (in sec)
5000	0.00964	0.7416
10000	0.01698	2.99993
15000	0.0268	6.86957
20000	0.03686	12.45686
25000	0.04564	19.654
30000	0.06084	28.5347

Plot-



## Input 5

Total insertion sort running time for input 5: 17.92108

Total selection sort running time for input 5: 23.00329

## Discussion

I choose Python for this programming assignment as being a data science student, it is my most preferred language. I'm also most familiar with visualization (plotting) techniques used in Python. I stored the data required to make plots as a list. I constructed separate lists for the two algorithms.

The first plot (for Input 1) shows an asymptotic quadratic plot for both the algorithms. The time taken for selection sort algorithm is more than insertion sort for all values of  $n$  and the difference increases as  $n$  increases.

In the second plot, the time taken for insertion sort is quite less ( $< 0.01\text{sec}$ ) as the array is already sorted. This is the best-case scenario for insertion sort algorithm. The algorithm executes in linear time. Selection sort runtime plot still appears asymptotic quadratic

In the third plot, the time taken for insertion sort is higher than selection sort. It is the worst possible case for the insertion sort. Both the algorithms run in quadratic time.

In the fourth plot few of the number are swapped from a sorted array this accounts for a very small share so the times are comparable to plot 2, we can see times for insertion sort are slightly higher than the ones in plot 2 due to the few numbers being swapped

In the output for input 5, we can see a quantifiable difference in runtimes of insertion sort and selection sort with insertion sort being faster

I would preferable not use these algorithms to sort large arrays as the running time is  $O(n^2)$  and it takes a significant amount of time to sort the arrays as we can see from plot 1 ( for random numbers)