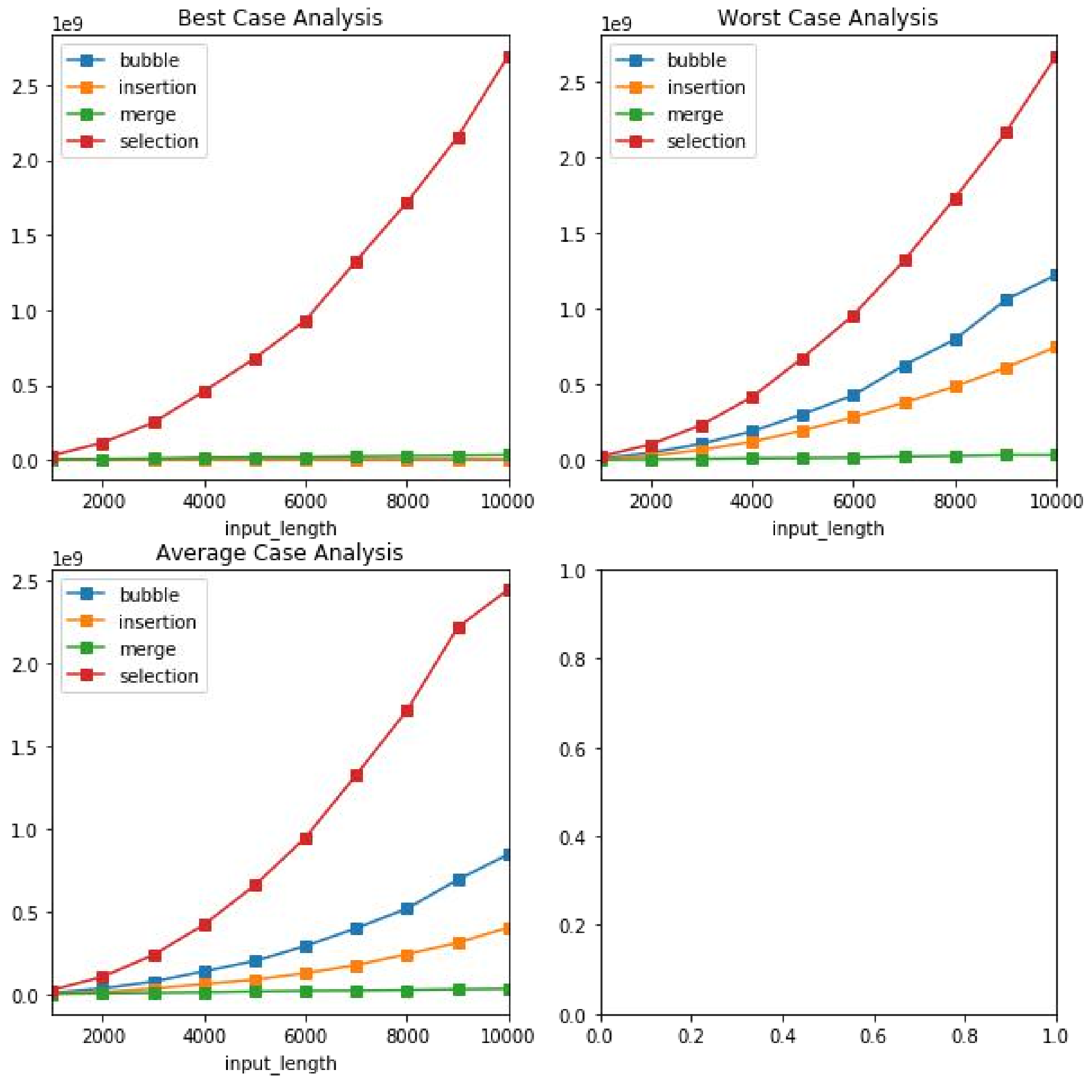
1. Empirical Analysis:

Best Case, Worst Case, Average Case comparisons for the 4 sorting algorithms.



2. Asymptotic analysis:

Sort Types	Best Case Performance	Average Case Performance	Worst Case Performance
Bubble Sort	O(n)	O(n^2)	O(n^2)
Insertion Sort	O(n^2)	O(n^2)	O(n^2)
Merge Sort	θ(n log(n))	θ(n log(n))	θ(n log(n))
Selection Sort	O(n^2)	O(n^2)	O(n^2)

Bubble Sort:

Analysis:

For Worst and Average case, there would be n-1 comparisons in the 1st pass, n-2 comparisons in the 2nd pass and so on.

$$T(n)= \sum_{k=1}^{n-1} k \;= n(n-1)/2 ==> O(n^2).$$

For Best case, the loop will run n times, T(n) ==> O(n)

Insertion Sort:

Analysis:

For Worst Case:

$$T(n)= \sum_{k=1}^{n} k \;= n(n-1)/2 ==> O(n^2).$$

For Average Case and Best Case:

$$T(n)= \sum_{k=1}^{n} k/2 \;= n(n-1)/4 ==> O(n^2).$$

Merge Sort:

Analysis:

For Best, Average, and Worst cases the sort behaves in the same manner i.e input list is divided into 2 parts and solved recursively.

$T(n)=2T(n/2)+ \theta(n)$
Using the master theorem, we get
$T(n) = \theta(n \log(n))$

Selection Sort:

Analysis:

For Best, Average, and Worst cases the sort behaves in the same manner i.e the selection sort has 2 nested for loops, so the time complexity is as mentioned.

$$T(n)= \sum_{k=1}^{n-1} k \;= n(n-1)/2 ==> O(n^2)$$