Samarth Dare
2023300039

Page No.

Date

Report - Expt 1 B

✫ Space Complexity

Insertion Sort :- It is an in place algorithm, meaning it doesn't require additional memory aside from input array, ∴ Space Complexity is CONSTANT

Selection Sort :- like insertion sort, selection sort is also an in-place algorithm. Hence, Space Complexity is CONSTANT.

✫ Time Complexity

(i) Insertion Sort

- Best case :- $O(n)$:- The sorted array results in best case scenario. In this case ~~the~~ each element only need to be compared with previous
- Worst Case - $O(n^2)$:- when the array is sorted in reverse order, every element needs to be compared with ~~only~~ all the previously sorted elements

(ii) Selection Sort :-

- Best-Case (~~$O(n^2)$~~ [$O(n^2)$] :- selection always performs the same number of comparisions, regardless of the ~~it~~ initial order of the array. Even if it already sorted. It still performs $\frac{n(n-1)}{2}$ comparisions
- Worst-Case $O(n^2)$:- It is same as best case scenario. The best case, average, worst case all perform $\Theta \frac{n(n-1)}{2}$ comparisions.

Conclusion :- ~~for~~ ~~P~~ From the plotted graph, we can clearly see that despite them having ~~small~~ almost same ~~value~~ time at smaller inputs but still we see that Insertion Sort is better than selection sort in respect to time complexity. Where Space Complexity is same for both Algorithms.