Introduction to Sorting

Sorting means an arrangement of elements in an ordered sequence either in increasing(ascending) order or decreasing(descending) order. Sorting is very important and many software and programs use this. The major difference is the amount of space and time they consume while being performed in the program.

Different languages have their own in-built functions for sorting, which are basically hybrid sorts that are a combination of different basic sorting algorithms. For example, C++ uses introsort, which runs quicksort and switches to heapsort when the recursion gets too deep. This way, you get the fast performance of quicksort in practice while guaranteeing a worst-case time complexity of O(N*logN), where N is the number of elements.

The time complexity of Sorting Algorithms:

The time complexities of various sorting algorithms in the worst case, where N represents the number of elements in the array -

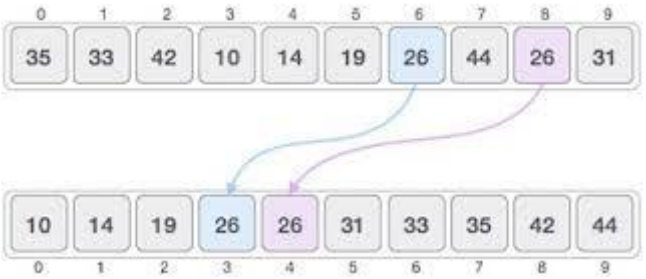
Sorting Algorithm	Time Complexity
Bubble Sort	O(N^2)
Selection Sort	O(N^2)
Insertion Sort	O(N^2)
Merge Sort	O(N*logN)
Quick Sort	O(N^2)

In-place sorting and Out place sorting

1. **In-place sorting:** In-place sorting does not require any extra space except the input space excluding the constant space which is used for variables or iterators. It also doesn't include the space used for stack in the recursive algorithms.
   For Example, Bubble Sort, Selection Sort, Insertion Sort, Quicksort, etc.
2. **Out-place sorting:** Out-place sorting requires extra space to sort the elements.
   For Example, Merge Sort, Counting sort, etc.

Stable and Unstable sorting

1. **Stable sorting:** A sorting algorithm when two objects with equal keys appear in the same order in the sorted output as they appear in the unsorted input.
   For Example, Insertion Sort, Merge Sort, Bubble Sort, Counting sort, etc.
2. **Unstable sorting:** A sorting algorithm is called unstable sorting if there are two or more objects with equal keys which don't appear in the same order before and after sorting.
   For Example, Quick Sort, Heap Sort, Selection Sort, etc.



Above is an example of stable sorting here number 26 is appearing two times at positions 6 and 8 respectively in an unsorted array and their order of occurrence is preserved in the sorted array as well and sorted array i.e. element 26 (blue) at position 6 in unsorted array appears first in sorted array followed by another 26 (red) at position 8.

Adaptive and Non-Adaptive sorting

1. **Adaptive sorting:** When the order of occurrence of elements in an array affects the time complexity of a sorting algorithm, then such an algorithm is known as an adaptive sorting algorithm.
   For Example, Bubble sort, Insertion Sort, Quick Sort, etc.
2. **Non-Adaptive sorting:** When the order of occurrence of elements in an array does not affect the time complexity of a sorting algorithm, then such an algorithm is known as a non-adaptive sorting algorithm.
   For Example, Selection Sort, Heap Sort, Merge Sort, etc.