

Sorting

STUDY NOTES

- The process of ordering or arranging a given collection of elements in some particular order is known as Sorting.
- In computer science, sorting is an important area of study. Several sorting algorithms have been developed and analysed from their performance.

Bubble Sort:

- Bubble sort sorts a given list of elements by simply repeatedly comparing the adjacent elements and changing their position with each other if they are not found to be in order.
- Every iteration through each element of the list is called a “pass”.
- For a list of size n , a total on $n-1$ passes are made to sort it.
- The largest element of the element is found and placed in the right position or ‘bubbled up’.
- The once sorted element is not considered in the remaining passes therefore the size of the list is reduced in successive passes.

Selection Sort:

- To sort a list having n elements, the selection sort makes $(n-1)$ number of passes.
- The list is considered to be divided into two lists -- the left list that has sorted elements, and the right list that has unsorted elements.
 - ❖ Initially, the left list is empty, and all the elements are in the right list.
 - ❖ In the first pass after traversing through all the elements the smallest element is identified and is swapped with the leftmost element of the unsorted list.
 - ❖ This element is not considered any further for sorting.
- The process continues till the list is sorted.
- Selection sort makes $n-1$ passes to sort a list of size n .

Insertion Sort:

- In Insertion sort, the list is divided into two parts - one of sorted elements and another of unsorted elements.
- Every element of the unsorted list is one by one evaluated and is inserted into the sorted list at its correct position.
- To find the position where the unsorted element should be inserted, the sorted list is traversed from the backward direction in each pass.

Time Complexity Of Algorithms:

- In the real-world applications, we use sorting algorithms to work on huge data sets.
- Total time utilisation on implementation of an algorithm becomes significant.
- It is important to consider the time complexity of an algorithm in order to use it in an application.