

* Sorting Algorithm :-

Sorting is a process of arranging items systematically. like alphabetically or ascending to descending or vice versa. e.g sort names, sorting numbers etc.

unsorted list

20	12	10	13	2
----	----	----	----	---

sorted list

2	10	12	13	20
---	----	----	----	----

* Types of sorting algorithm:

Elementary

- i> Bubble sort.
- ii> Selection sort.
- iii> Insertion sort.

Advanced

- i> Merge sort.
- ii> Quick sort.

* Bubble Sort :

Bubble sort is based on the idea of repeatedly comparing pairs of adjacent elements and then swapping their positions if they exist in the wrong order.

-2	45	0	11	-9
----	----	---	----	----

 \rightarrow

-9	-2	0	11	45
----	----	---	----	----

$i = 0$

-2	45	0	11	-9
----	----	---	----	----

 check $\text{index}(0) > \text{index}(4)$
No

$i = 1$

-2	45	0	11	-9
----	----	---	----	----

 $\text{index}(1) > \text{index}(0)$
Swap.

$i = 2$

-2	0	45	11	-9
----	---	----	----	----

 $\text{index}(2) > \text{index}(3)$
Swap.

$i = 3$

-2	0	11	45	-9
----	---	----	----	----

 $\text{index}(3) > \text{index}(4)$
Swap.

-2	0	11	-9	45
----	---	----	----	----

 Iteration 1

Completed.

Iteration 2 starts leaving

45

 out as it is already sorted in end.

$i = 0$

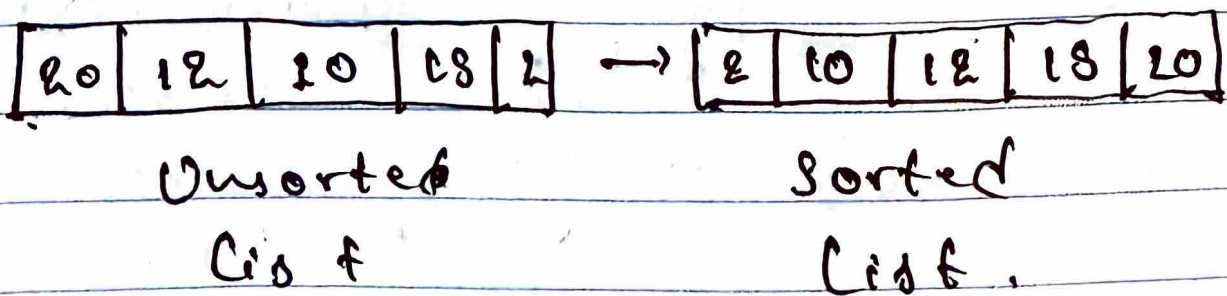
-2	0	11	-9
----	---	----	----

$\uparrow \quad \uparrow$

and soon.....

- * The complexity of bubble sort is $O(n^2)$. This is for the worst case. For best case it is $O(n)$ and in average case also it is $O(n^2)$.
- Best case when the list is already sorted.

* Selection Sort :- It selects the smallest element from the unsorted list in each iteration and try to place at the beginning of the unsorted list.



- The complexity of the selection sort is $O(n^2)$ for the worst case and $O(n^2)$ for the best case.

* Insertion Sort :- ^{1st} Insertion sort we divide our list into 2 parts the 1st part is sorted and the 2nd part is unsorted.

Complexity of insertion sort is $O(n^2)$ in worst case and in best case $O(n)$.

Problem Solving Pattern.

- * Sliding Window
- * Two pointer
- * Divide and Conquer.
- * Using Hash table.
- * Fast and Slow pointer.