* Sorting Algorithms *

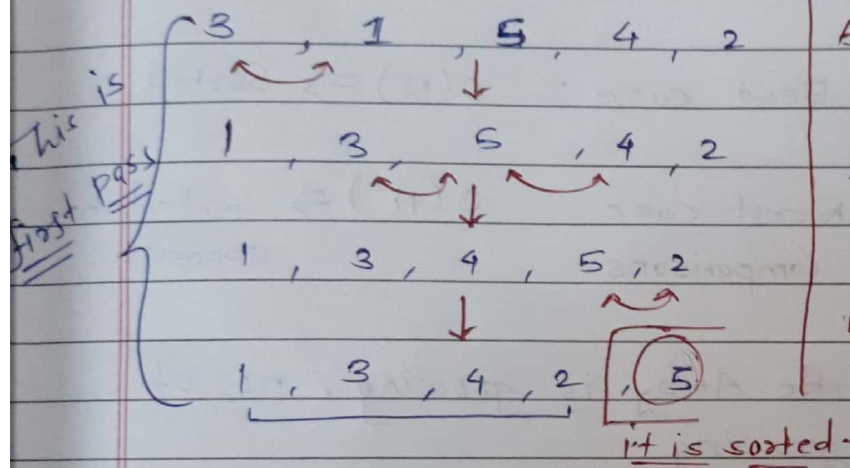## * Bubble sort :-

* What is bubble Sort ?

→ it is a sorting Algorithm in which array is will sorted through this Algorithm.

Example :→

This is first pass

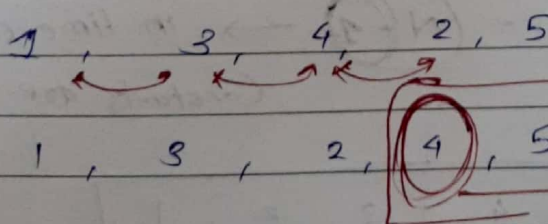| 3 , 1 , 5 , 4 , 2 |
| ↓ |
| 1 , 3 , 5 , 4 , 2 |
| ↓ |
| 1 , 3 , 4 , 5 , 2 |
| ↓ |
| 1 , 3 , 4 , 2 , (5) |

it is sorted.

In this algorithm, In array We make comparisons between adjecent elements. if element is smaller than previous one just "Swap it."

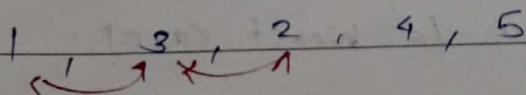We have to make comparisons till the last one.

* Why bubble Sort ?

→ with the first pass through the array, the largest element come to the end.

Now 2nd pass =>

1 , 3 , 4 , 2 , 5

1 , 3 , 2 , (4) , 5

1 , 3 , 2 , 4 , 5

1 , 2 , (3) , 4 , 5   => This is Sorted Array

with the pass no. 2nd, 2nd larget element is at the 2nd last. in index.

* imp } => Bubble Sort Also called as Sinking sort / Exchange sort.

* **Space Complexity:-** $O(1)$ // Constant.

i.e no extra space required

Aka inplane copying the array etc not

Sorting Algorithms required.

* **Time Complexity:-**

① Best case: $O(N) \Rightarrow$ Sorted.

② Worst case: $O(N^2) \Rightarrow$ Sorted in opposite.

Where N is no. of comparisons.

* As the size of the Array is growing, no. of Comparisons also growing.

$$\overset{j}{1}, \overset{j}{2}, \overset{j}{3}, \overset{j}{4}, \overset{j}{5}, \text{once.}$$

* **Best case** $\Rightarrow$

$i=0$

i^{th} pass   NOTE:- when j never swaps for a value of i means array is sorted.

Best case Comparison = $(N-1) \rightarrow$ in time complexity Constants are ignored.

* **Worst Case:-** $\boxed{5, 4, 3, 2, 1}$

This is worst caset.

Total Comparisons = $(N-1) + (N-2) + (N-3) + (N-4)$

$= 4N - (1+2+3+4)$

$= 4N - \left(\dfrac{N*(N+1)}{2}\right)$

$$= 4N - \left(\frac{N * (N+1)}{2}\right) \implies \frac{(1+2+3+4)}{2} =$$

$$= \frac{4 \times (4+1)}{2} = \frac{4 \times 5}{2}$$

$$= 4N - \left(\frac{N^2+N}{2}\right)$$

$$= \frac{20}{2} = \boxed{10}$$

$$= \frac{8N - N^2 + N}{2}$$

$$= O\left(\frac{4N - N^2}{2}\right) \implies \boxed{O(N^2)}$$

why., Constants are ignored in time Complexity ~~lectures~~.

---

 * **Stability :-**

⑩   ㉚   ⑳   ㉚   ⑩

↓ sort.

⑩   ⑩   ⑳   ⑳   ㉚
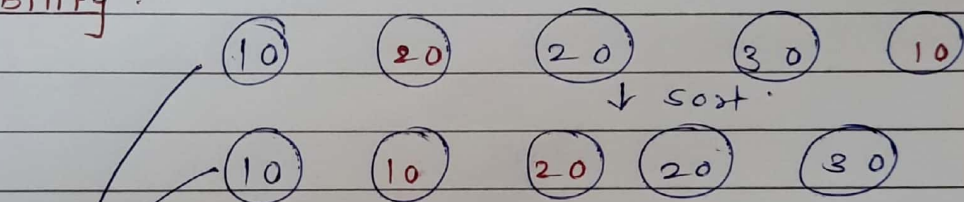
→ in original array black ball of 10 is before red ball of 10, And in the sorted one, this order is maintained

10 , 10 , 20 , 20 , 30

⇓

Unstable.