\* Sorting: Sorting is the process of arranging the data in ascending and descending order.

Ex:- 3,1,5,4,2 ⇒ [1,2,3,4,5] ← ascending ordered Sorted

Bubble Sort:

>>> It is basically a Comparison-sort method.

[comparison sort means sorting step-by-step]

>>> It is also known as Sinking/Exchange Sort.

>>> In every Step, we are comparing adjacent elements.

\* Why Bubble Sort ?

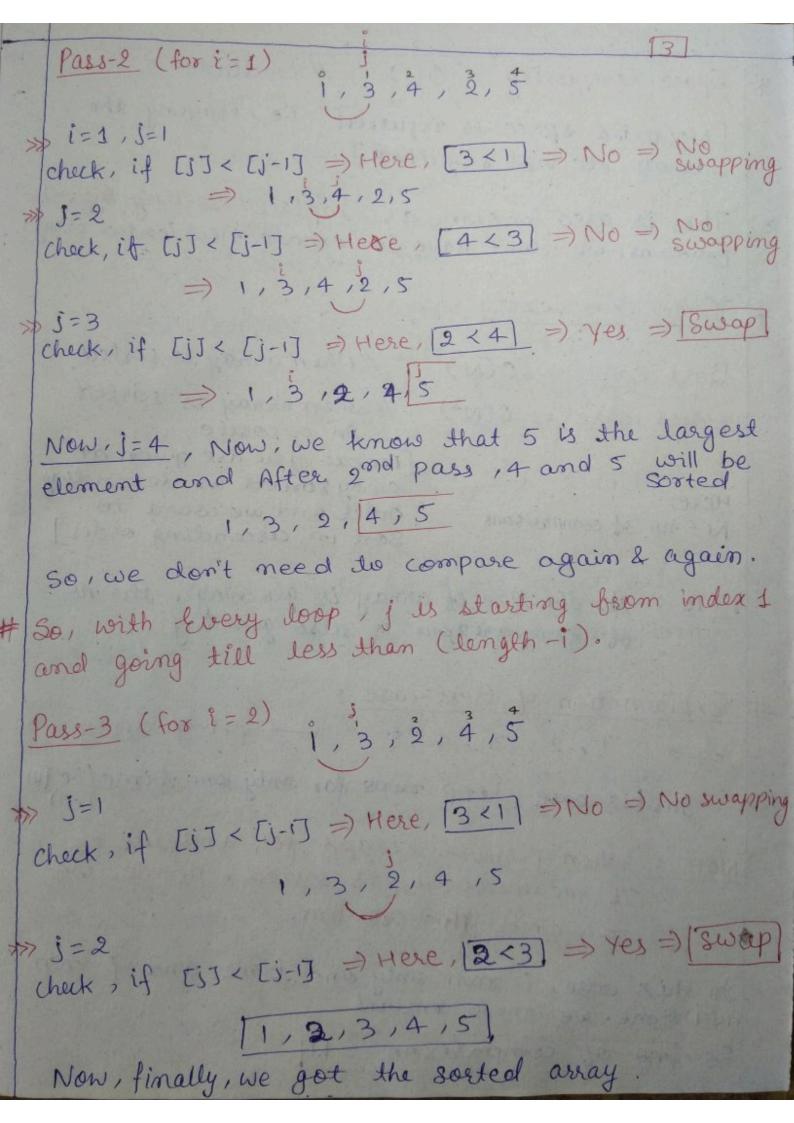
with Every Pass / Step. the largest element comes in the end.

Means

>>> With Pass no. 1, the largest element come at the end.

>>> with Pass no. 2, the 2<sup>nd</sup> largest element came at 2<sup>nd</sup> index from the last and 80 on ----

```
# Example i i 2 3 4 , 2
* Pass 1 (i=0)
>>> Initially, Li = 0, j=1
   check, if [i]<[i]-i] => Here, [1 < 3] => Yes => [Swap]
       \Rightarrow 1, 3, 5, 4, 2
>>> Now, [j=2]
   Check, if [j] < [j-1] => Here, [5 < 3] => No => No swapping
>>> Now, [j=3] ? 3, 5, 4, 2
   Check, if [j] < [j-1] =) Here, [4<5] =) Yes =) [Swap
      => 1, 3, 4, 5, 2
>>> Now, j=4
   check, if [j] < [j-1] => Here, [2<5], => Yes=) Swap
       =>1,3,4,3,5
    Now, j will index out of bound.
# Here, i = counter
        j = internal loop
· Now, j will start again for Second pass (i-e for i=1)
```



14

\* Space Complexity = O(1) // constant

[No extra space is required. i.e., copying the array, etc not required]

>>> This is also known as Inplace Sorting Algorithm. [means, we don't have to create new lentra array].

\* Time Complexity

Best case = O(N)

Worst case = O(N2)

Here, N = no of comparisons // when array is sorted

// when array is sorted

in opposite

[For ex:- you are given an

array sorted in descending

order and we want to Sort in ascending order]

NOTE: As the size of array is growing, the no. of comparisons is also growing.

# Explaination of Best case:

Ex= 1,2,3,4,5 = Sorted array

In this case, loop runs for only one time (i.e for i=0)

NOTE: when j never swaps for a value of i, it means array is sorted. Hence, we can end the program.

In this case, i van only one time and j van (N-1) time. we ignore constant.

So, no. of comparison = N

