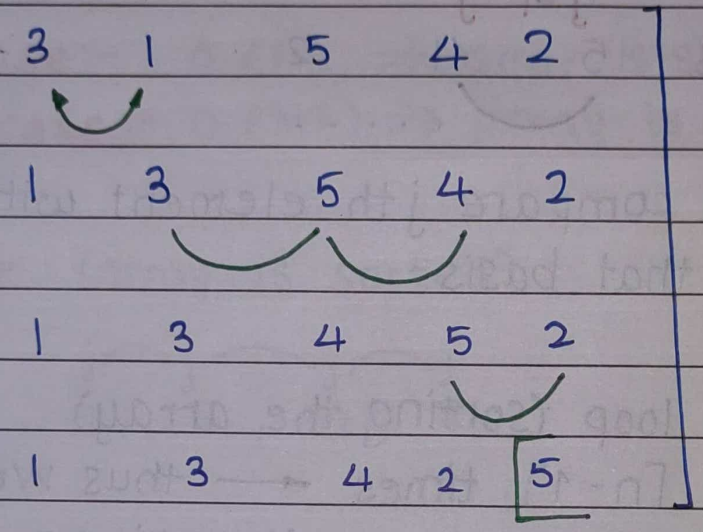
Bubble Sort Algorithm.

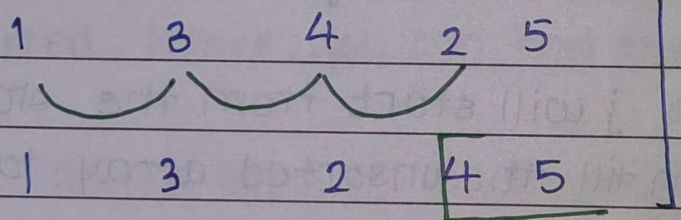
Sorting is the process or arranging items either in increasing or decreasing order.

Bubble Sort works by swapping the adjacent elements if they are in the wrong order.

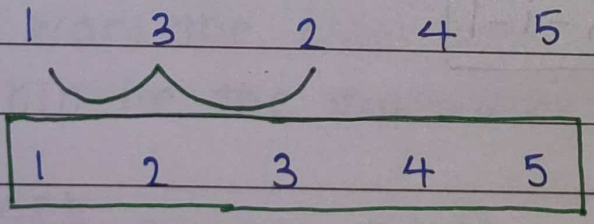
eg:-



PASS NO. 1
with the first pass through the entire array, the largest element came to the end.



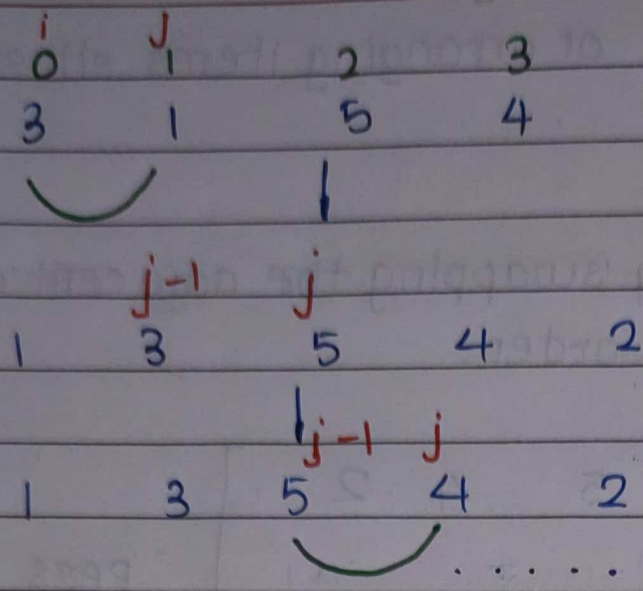
PASS NO 2
second largest ele. comes at second last index.



PASS NO 3

Sorted!

Also known as Sinking Sort or Exchange Sort.



Basically we compare jth element with j-1 th element and swap on that basis.

j :- internal loop (sorting the array)
it runs [n-1] times ← thus we keep i to count the no of iterations.

⇒ in every loop j will start from the start but will only run till the unsorted array lasts.

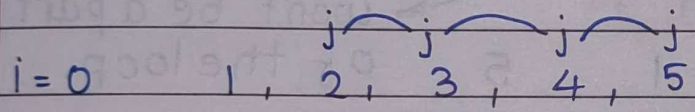
∴ $\boxed{j \leq \text{length} - i - 1}$

* COMPLEXITY

- Space Complexity :- O(1) //constant.
  ⇒ no extra space is required i.e copying the array etc is not required.
  ⇒ also known as inplace sorting algorithm.

- TIME COMPLEXITY :-
  * Best Case :- O(N) ⇒ array is sorted
  * Worst case :- O(N^2) ⇒ array is sorted in opp. (desc)

* Best Case (array is sorted)

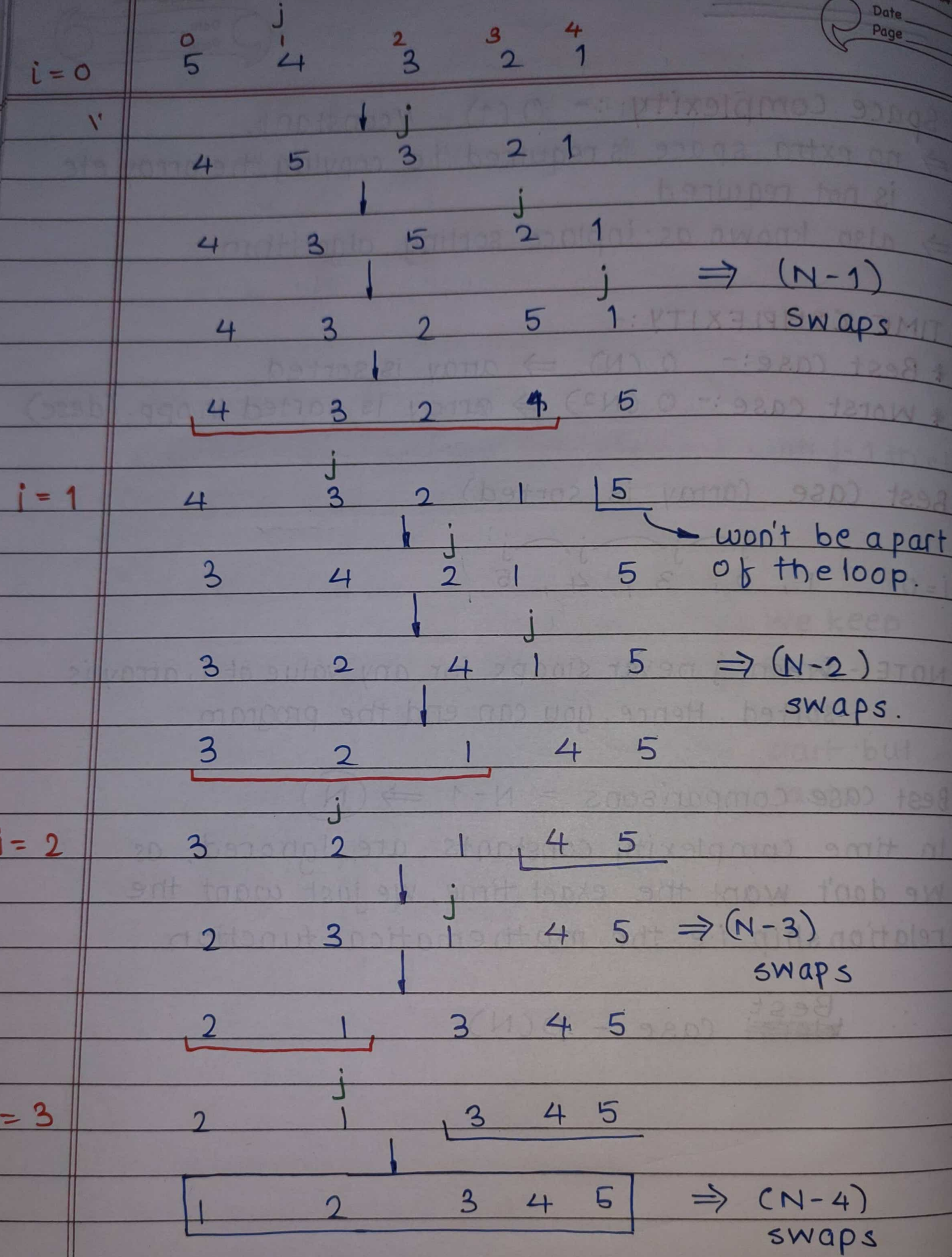


NOTE:- When j never swaps for any value of i, array is sorted. Hence, you can end the program.

Best case comparisons = N - 1 ⇒ (N)
In time complexity constants are ignored, as we don't want the exact time, we just want the relationship i.e the mathematical function.

∴ Best ~~Worst~~ case :- O(N).

* Worst Case :-


j
0 1 2 3 4
5 4 3 2 1
i=0
1'
j
4 5 3 2 1
j
4 3 5 2 1
j
⇒ (N-1) swaps
4 3 2 5 1
4 3 2 4 5
j
i=1
4 3 2 1 5
won't be a part of the loop.
j
3 4 2 1 5
j
3 2 4 1 5
⇒ (N-2) swaps.
3 2 1 4 5
j
i=2
3 2 1 4 5
j
2 3 1 4 5
⇒ (N-3) swaps
2 1 3 4 5
j
i=3
2 1 3 4 5
1 2 3 4 5
⇒ (N-4) swaps

∴ Total No. of Swaps =

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

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

$= 4N - \left[\frac{N \times (N+1)}{2}\right]$ ← N:- No of iterations.

$= 4N - \frac{N^2+N}{2}$

$= \frac{7N - N^2}{2} \Rightarrow O\left(\frac{7N-N^2}{2}\right)$

↳ In O Notation, constants are cancelled and less dominating terms are removed i.e highest degree remains.

∴ Worst case :- $O(N^2)$

* Stable Vs Unstable Algorithm.

Stable:- (10) (20) (20) (30) (10)

↓ after sorting

(10) (10) (20) (20) (30)

original order is maintained for same values.

⇒ order is maintained after sorting.

Unstable :- (10) (20) (20) (30) (10)

(10) (10) (20) (20) (30)

BUBBLE SORT is a stable Algorithm.