

Bubble sort

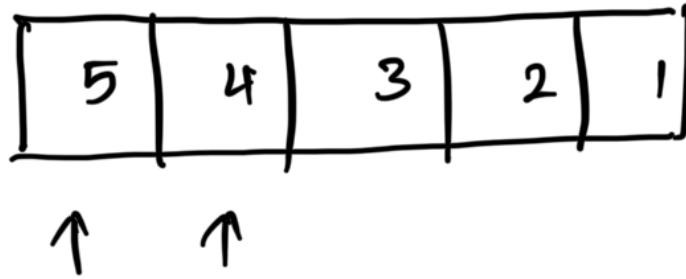
Q. what is Bubble sort? → ①

Q. what is Passthrough? → ②

A → ① Bubble sort is one of the sorting algorithm to sort array's element in ascending order.

How it sorts?

- Two pointers are used, start from left of the array.



- The two pointer's value is compared, if the second pointer's value is out of order, that is second pointer's value is less than first pointer's value, then

$$\therefore \boxed{5} < \boxed{4} = \text{false}$$

we swap the value.

- move the pointers forward, and repeat the above steps until the end of the array.

0

7⁽²⁾

∴ Each time we follow the above steps, it's known as one passthrough.

After each passthrough, the maximum is moved to the end.

- if No swapping is done in a passthrough then it means array is sorted and we can break, else follow the above steps.

eg:

4	2	7	1	3
---	---	---	---	---

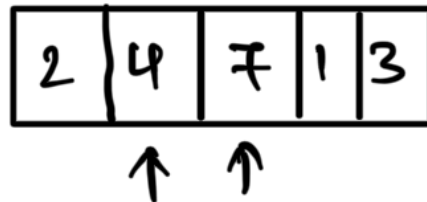
 → array

So we will start from left of array.

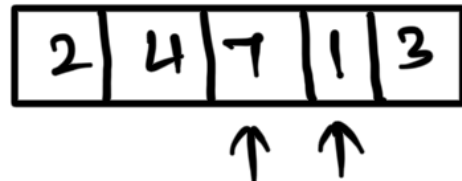


Here when compare we find that its false
as '4' is greater than '2', so we swap.

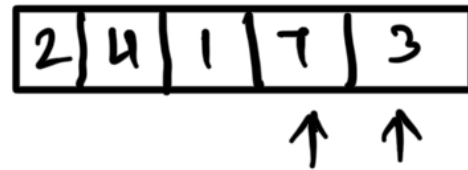
and move the pointer's
forward



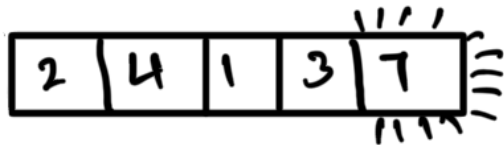
Here is true, No need of swapping. Hence
moving pointers further.



Here its false, Hence Swapping, and moving pointers furthers.



again follow the same.



we have travers till the end of the array.

we have completed our first passthrough.

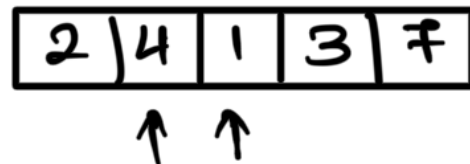
After each passthrough, the maximum element is "bubbled" at the end of

the array, and we can say for sure that element is in right position.

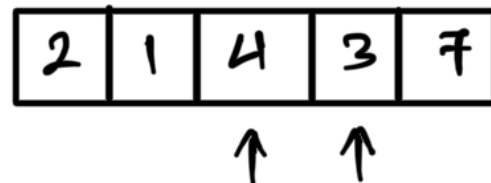
We are again going to follow the steps since we had at least one swap.



'2' is less than '4', so no need of swap.

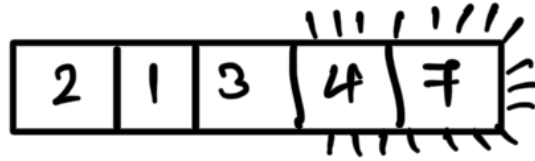


'1' is less than '4', so we swap.



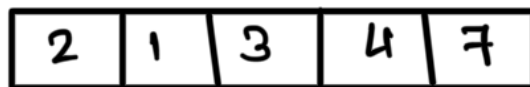
'2' is less than '4' on comparing

is less than 1, so swapping



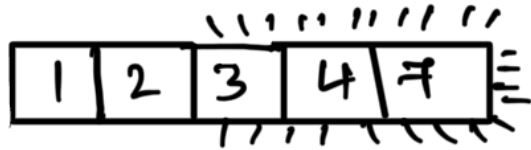
We know that, last element is already sorted, so no need to compare that, and by that we complete our 2nd pass through.

At the end of 2nd pass through, we can say for sure last 2 elements are in correct position. Since this pass through had swapping, we will continue to loop through array.



$\therefore 1 < 2$, so
compare

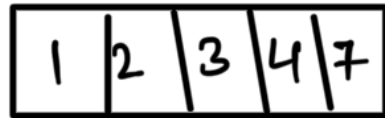
↑ ↑



swap

$\therefore 2 < 3$, swapping
not needed,

Since we had atleast one swap, will
continue,



↑ ↑

$\therefore 1 < 2$, No need
to swap.

All the other element are in correct
order, no need to process further, pass through
didn't have any swap, means array is
sorted Hence we can stop.