INSERTION SORT

By Prince Agarwal

[“ Hello World ”]

INSERTION SORT

Basic Idea Behind Insertion Sort :-

One element from input elements is consumed in each iteration

To find its correct position i.e. the position to which it belong in sorted array

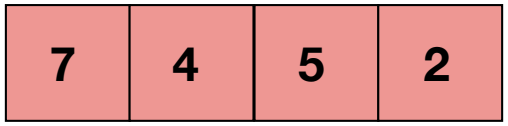
If the Current elements is greater than the all elements of his left side then leave it

Else

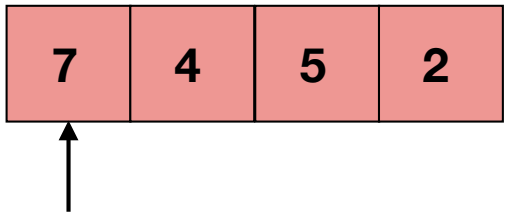
Shift all the elements which is larger than the current elements

INSERTION SORT

Take a example : Unsorted Array is —>

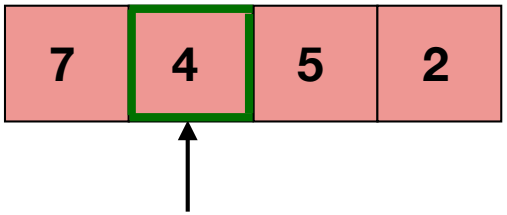


Step 1 :

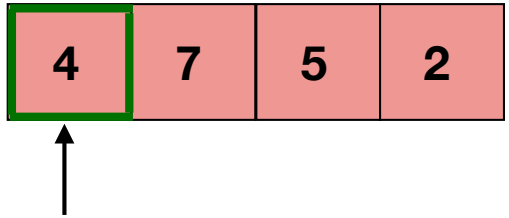


No element in left side of 7
so, no change in its position

Step 2 :



As 7>4 so 7 will move forward
And 4 will move on position of 7

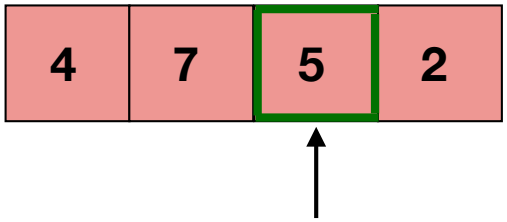


Here, checking on
Left side of 4

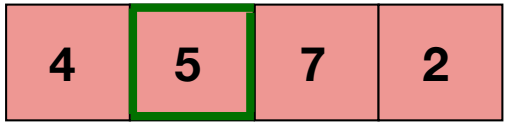
No element in left side of 4
so, no change in its position

INSERTION SORT

Step 3 :



As 7>5 so 7 will move forward
And 5 will move on position of 7

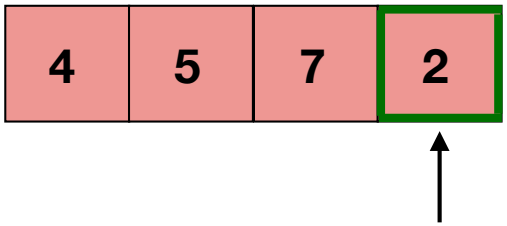


Here, checking on Left side of 5

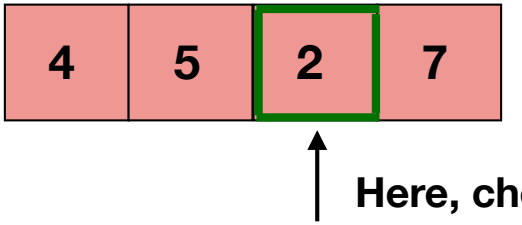
As 4<5 so 4 will not move forward
And 5 will remain in its position

INSERTION SORT

Step 4 :

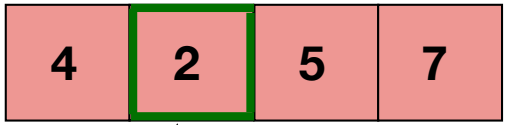


As 7>2 so 7 will move forward
And 2 will move on position of 7



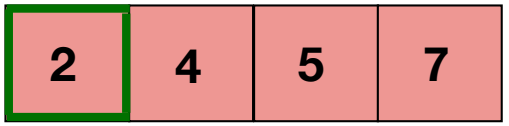
Here, checking on
Left side of 2

As 5>2 so 5 will move forward
And 2 will move on position of 5



Here, checking on
Left side of 2

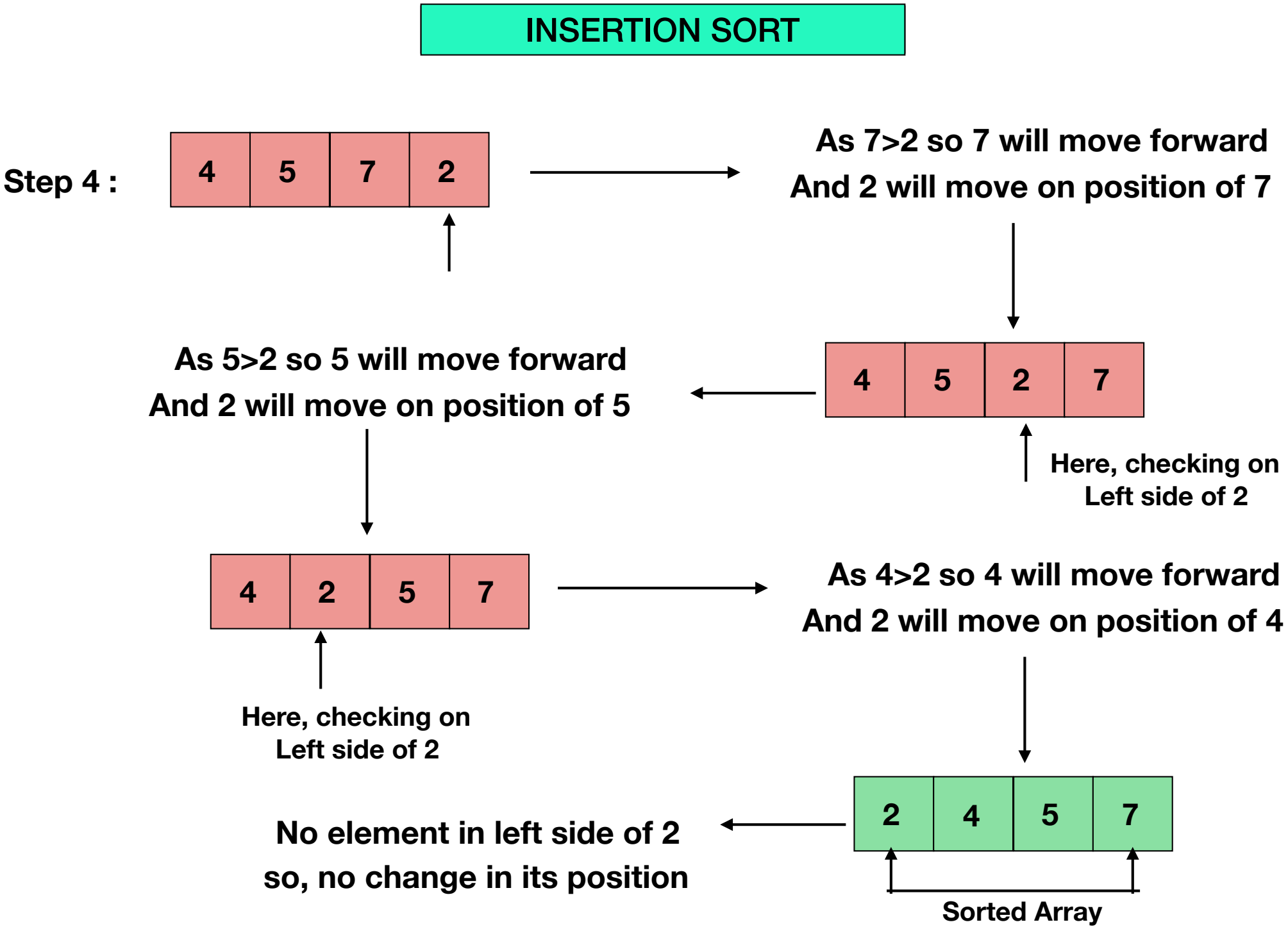
As 4>2 so 4 will move forward
And 2 will move on position of 4



Here, checking on
Left side of 2

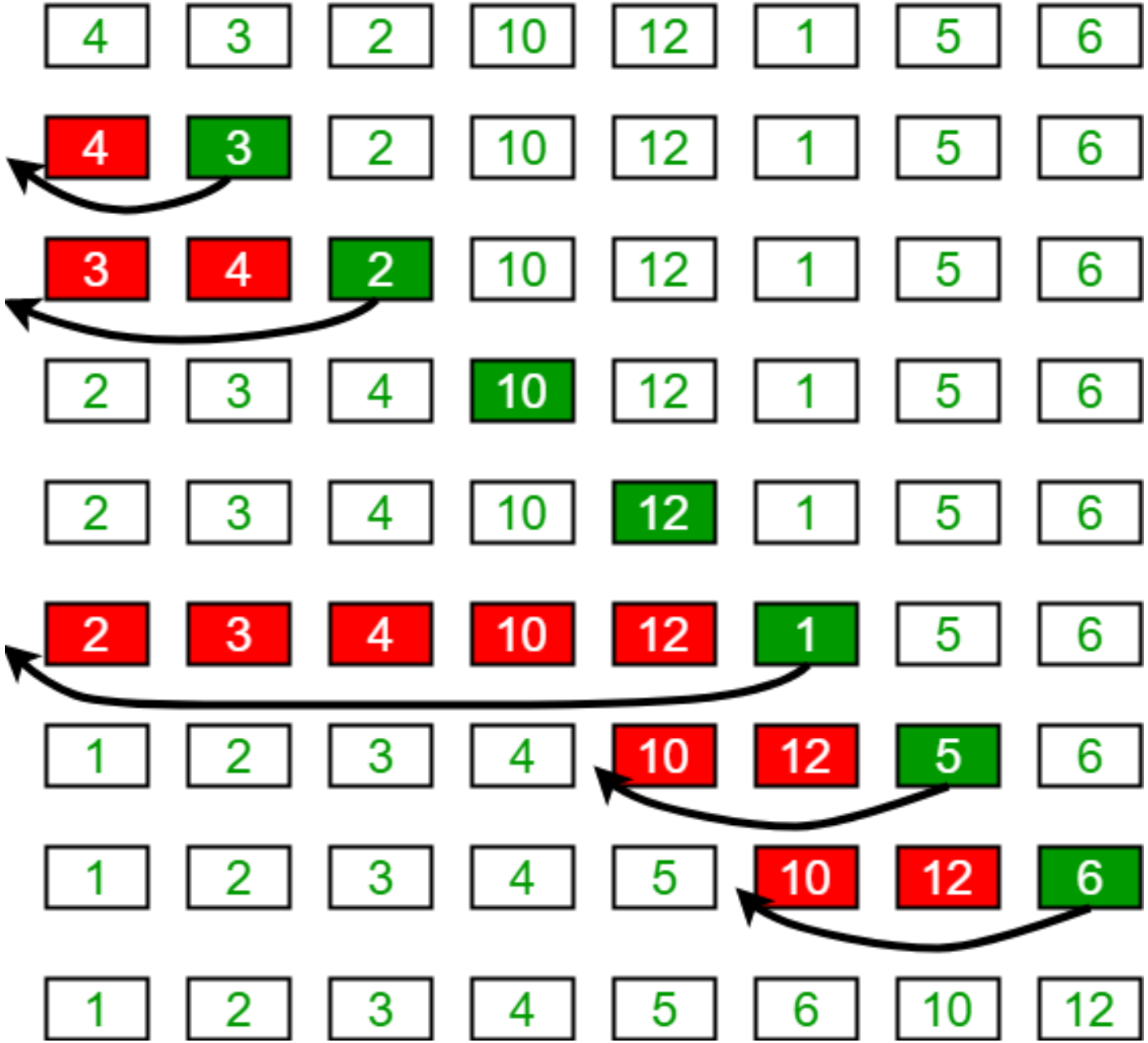
No element in left side of 2
so, no change in its position

INSERTION SORT



INSERTION SORT

Example 2 :-



INSERTION SORT

Time Complexity

Time Complexity = O(N^2)

Loop will run on N times ,
Where N is number of Elements in Array

Now, On Each Iteration There is a Worst case Possibility

That we have to compare all N-1 elements

Time Complexity = O (N * (N-1)) = O (N^2)

Subscribe, Like & Share

Hello World

“ If you feel any problem then comments in my video
I will reply as soon as possible “

- Prince Agarwal