Space Complexity of Array VS Space Complexity of Stack

Now after learning the working of space complexity that why Push of Space Complexity takes n times and insertion of Array takes $O(1)$.

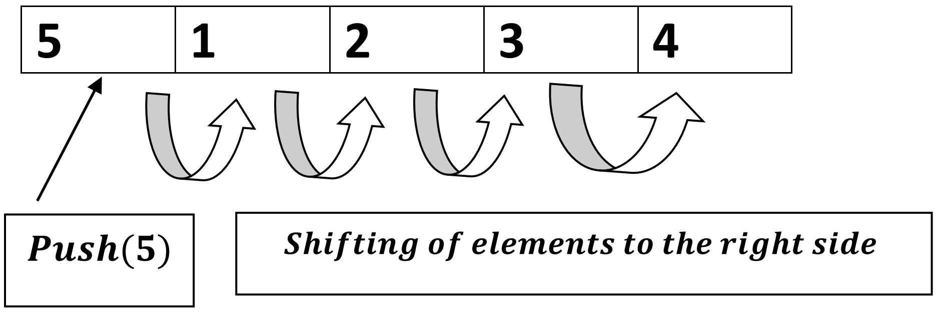
The reason is :

Push operation inserts elements at the begining of stack which shifts the rest of the element to the right.

Suppose we have stack :

1	2	3	4

***Now we push* 5 *in stack*:**



5	**1**	**2**	**3**	**4**

***Push*(5)**

Shifting of elements to the right side

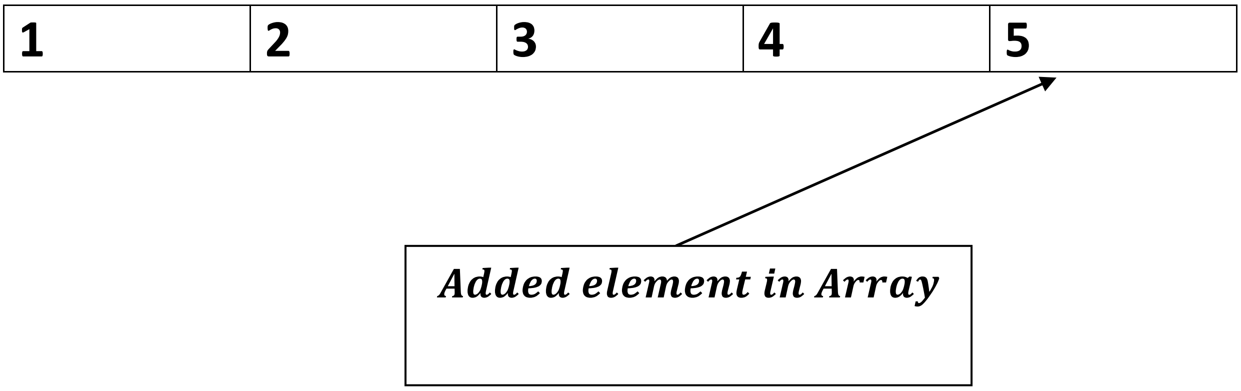
***Hence ,the whole operation takes O*(*n*) *space complexity*.**

***Where as inserting elements in array works like*:**

***Such as we have an array of size* 4 *i.e. a*[4]**

1	**2**	**3**	**4**

***Now we want to insert an element* 5**

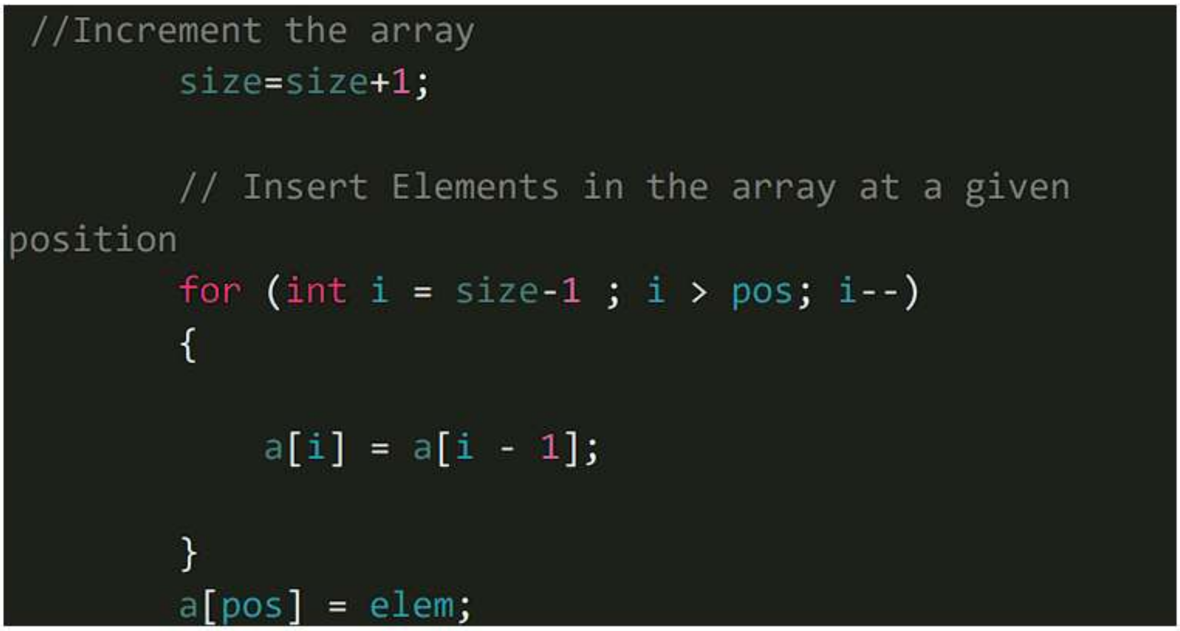


1	**2**	**3**	**4**	**5**

Added element in Array

That is just expanding the auxiliary space and insert element at the end of the array with no shifting activity.

Similary , if we talk of inserting an element in Array at a position:



```
 //Increment the array
        size=size+1;

        // Insert Elements in the array at a given
position
        for (int i = size-1 ; i > pos; i--)
        {

            a[i] = a[i - 1];

        }
        a[pos] = elem;
```

Here just we are swapping the elements from one index to another index :

$Suppose\ size = 5.$

$Now, size = size + 1 = 6$

such as:

$a[5] = a[4] = 4$

$a[4] = a[3] = 3$

$a[3] = a[2] = 2$

$a[2] = a[1] = 1$

$And\ now\ we\ insert\ a[0] = 5$

Though we are replacing or overriding elements

from one position to another and just inserting

at a given position.

Where as in push operation of a stack there is

no overdding or swapping of elements rather shifting of elements to the right and adding elements to the top creating a new data structure.

Where as in array no new data structure is created.

<u>When Stack Is Implemented Through Array</u>

As we can implement stack through array inserting element at top or begining will able to shift other element by 1 to the right side .

Hence Push operation takes $O(n)$ space complexity.