

Daily Temperatures :

Given an input array with each day temperature, find the next warm day from current day.
If you don't find next warm day then put as zero.

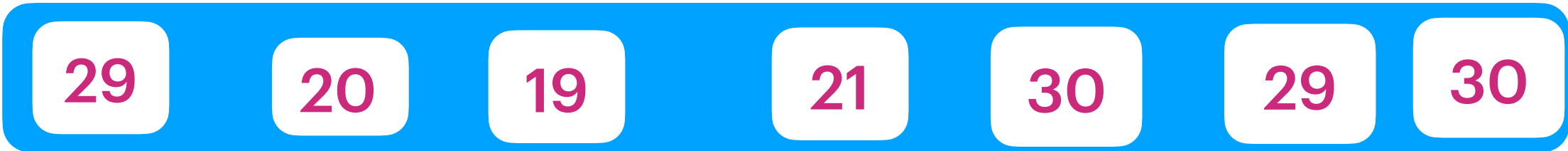
Input:



Output:



Input:



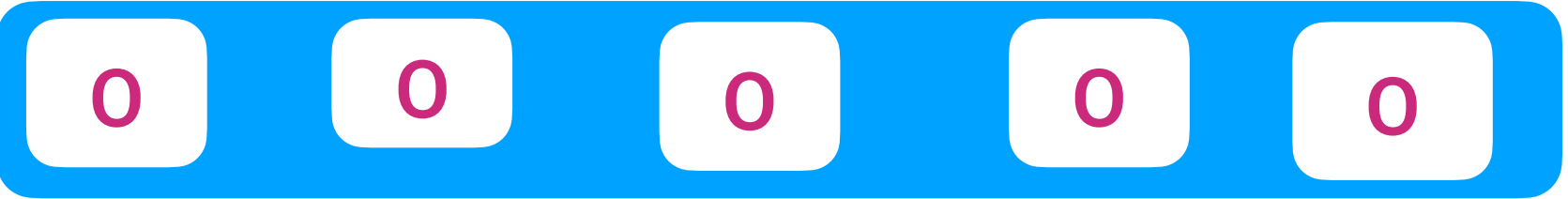
Output:



Input:

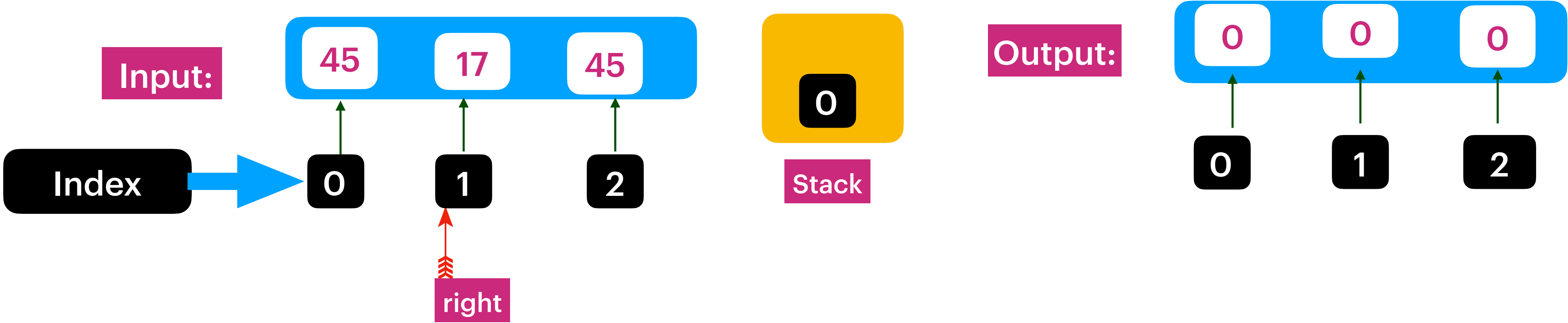
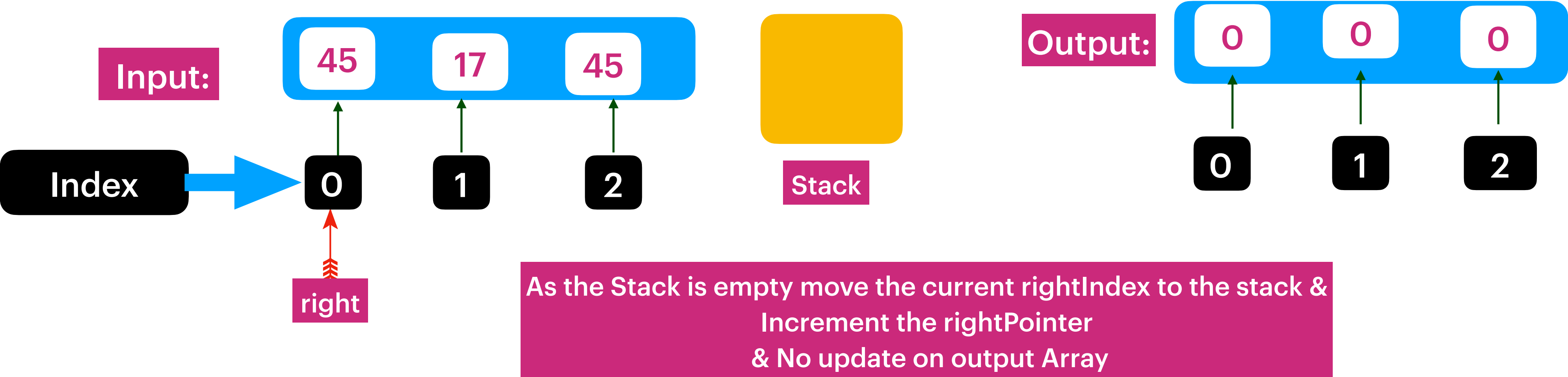


Output:



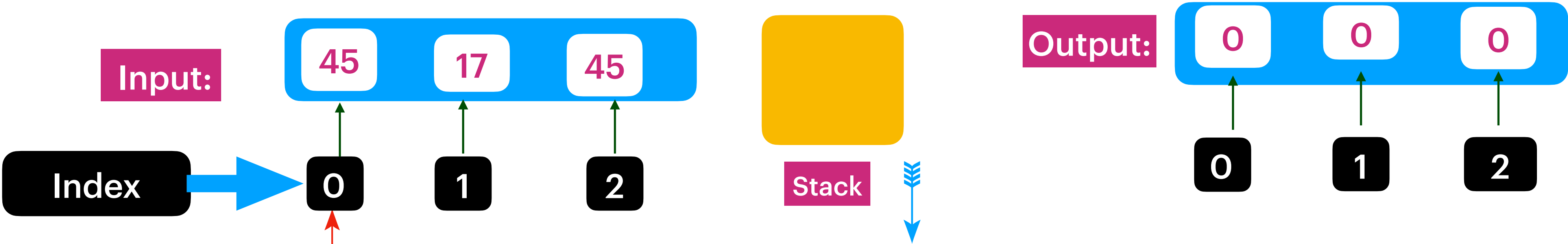
Increasing / Decreasing Stack : (Monotonic Stack Pattern)

Initiate the output array of same input size with default value as 0

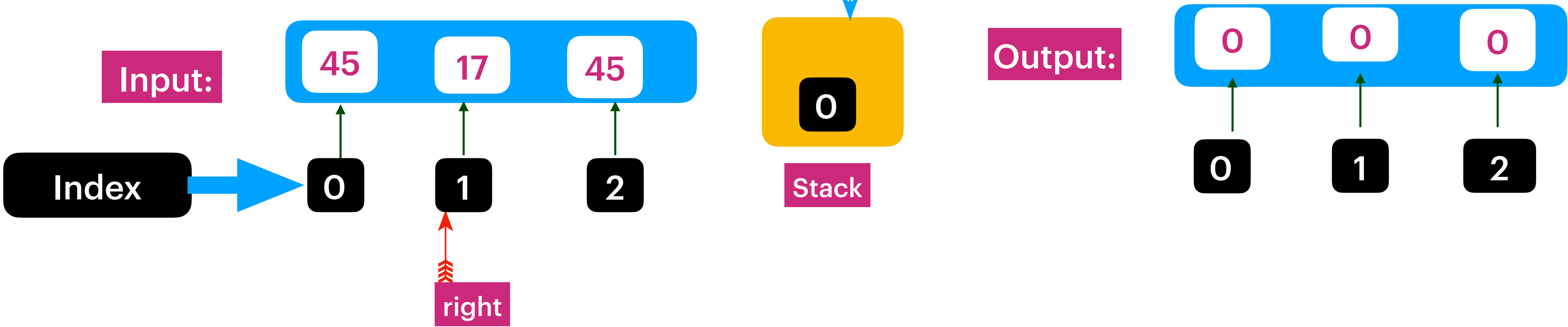


Solution with Increasing / Decreasing Stack : (Monotonic Stack Pattern)

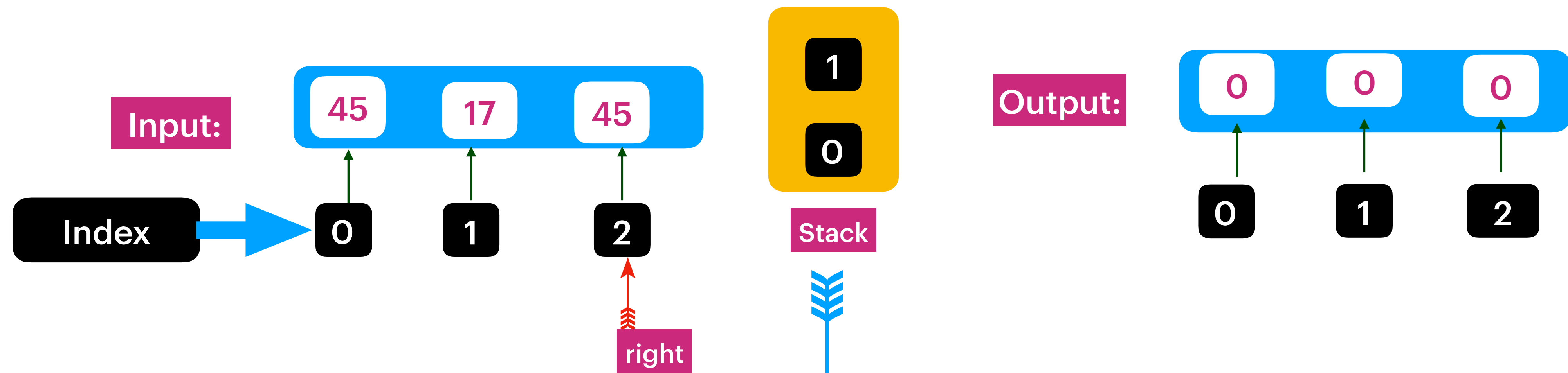
Initiate the output array of same input size with default value as 0



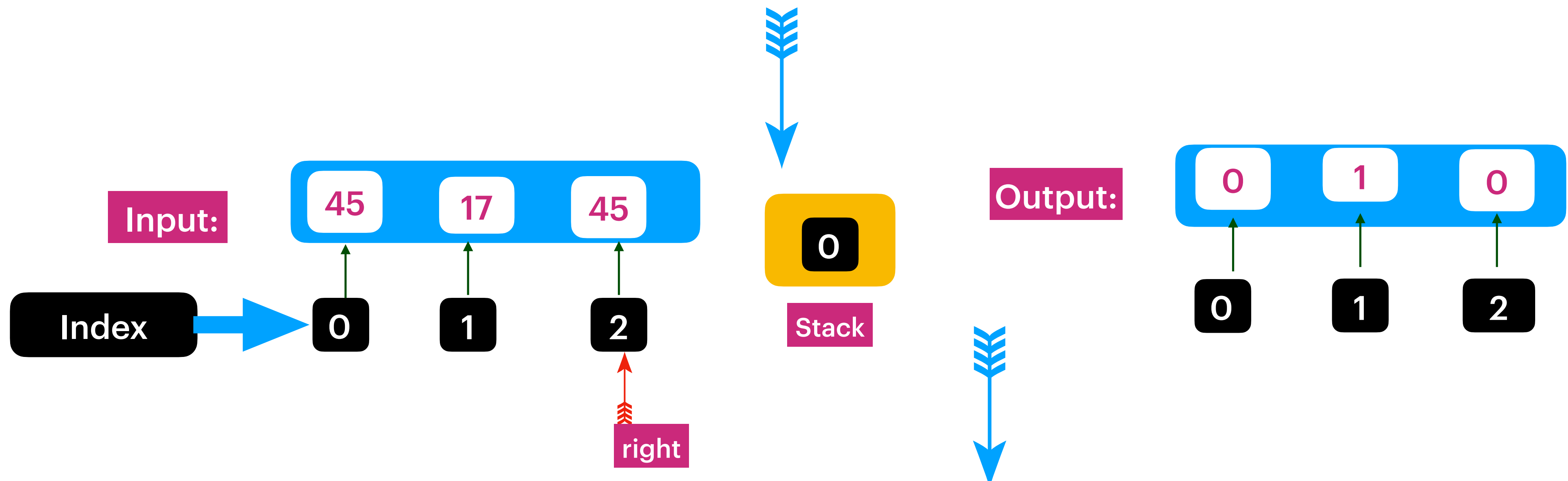
As the Stack is empty move the current rightIndex to the stack & Increment the rightPointer & No update on output Array



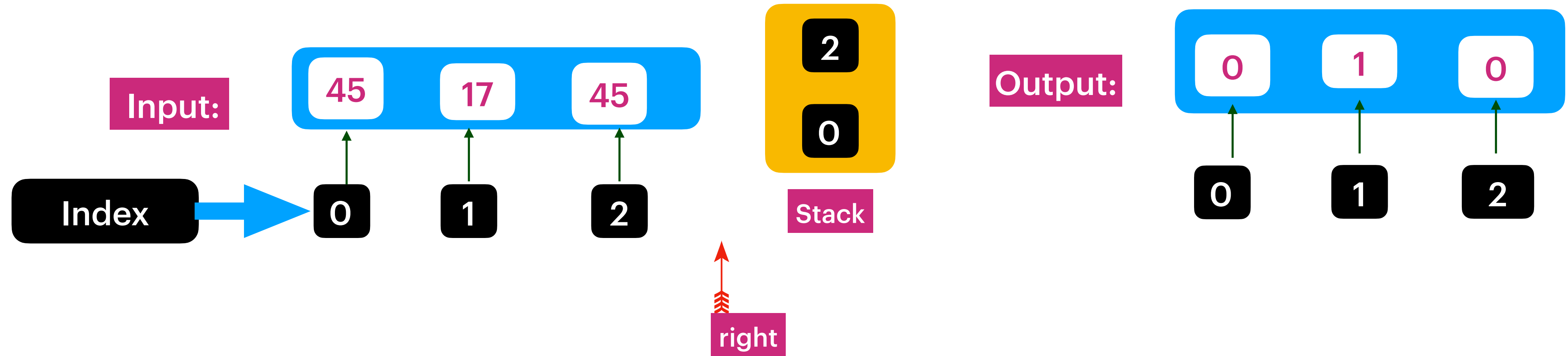
As my currentRight (17) index value is less than the top element(40) of the stack. i.e input[0] = 45
Push the currentRight Index to stack & Increment the rightPointer
& No update on output Array



As my currentRight (45) > top element (17) of the stack. i.e input[1] = 15
Pop the topIndex from the stack i.e 1
Update the output array :
 $\text{output}[\text{popIndex}] = \text{currentRightIndex} - \text{PopIndex}$
 $\text{output}[1] = 2 - 1$

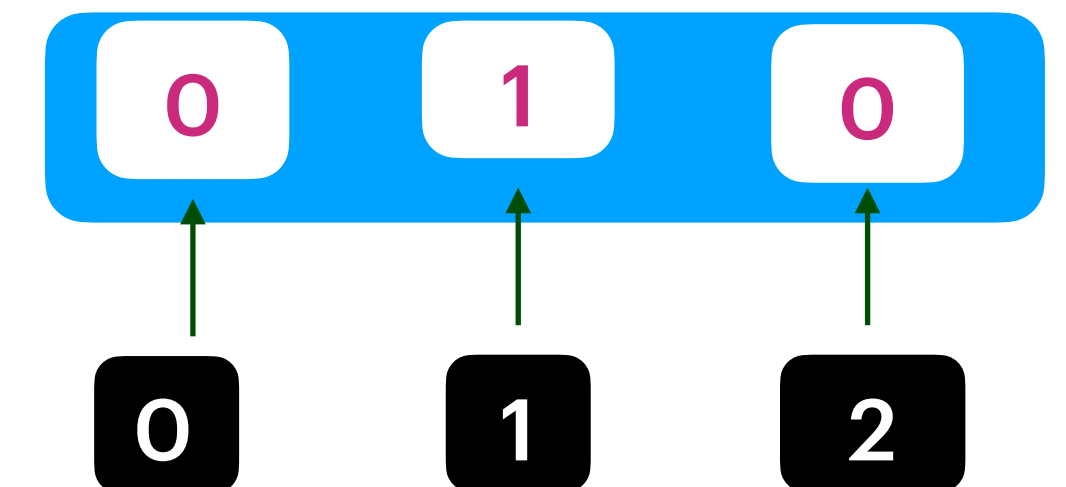


As my currentRight (45) is not greater than top element(45) of the stack.
Push the current index to stack & increment the rightPointer.



Now rightPointer reaching out of array size so we can return OutPut Array i.e

In worst cast each character would be visited twice :
1) By rightPointer in forward direction
2) Through stack in backward Direction
So the TimeComplexity would be $O(2n) = O(n)$
Space Complexity : $O(n)$



Solution with Back Tracking, Its Math Solution, refer to the notes.