Sort Colors

Given an array of integers values of 0, 1, and 2 (representing Red, White, and Blue), sort them in linear time.   
  
Input: Array of Integers where elements values belong to the set S : { 0, 1, 2 }  
Output: Sorted Array

# Example

Input: [2, 0, 1, 2, 1, 0] => Output: [0, 0, 1, 1, 2, 2]

Input: [1, 2, 2, 0] => Output: [0, 1, 2, 2]

# Constraints

Time Complexity: O(N)  
Auxiliary Space Complexity: O(1)  
  
Using a native sorting method is not allowed.

# Solution

1. Create a count array of zeros => [0, 0, 0]
2. Loop through the input array
3. For each element add one to the corresponding index in the count array
4. Instantiate a new pointer at 0
5. Now that the count array is populated, loop through input array again
6. If the count[pointer] > 0, decrement count[pointer] by one and replace current index of input to the pointer’s value
7. Otherwise if count[pointer] equals zero, then increment pointer and then execute step 6

# Notes

Using quicksort or mergesort will only be able to sort the array in O(NlogN) time. This approach uses counting sort which allows the sorting to be done in O(N) time due to its known/limited range.

# Resources

https://leetcode.com/problems/sort-colors/