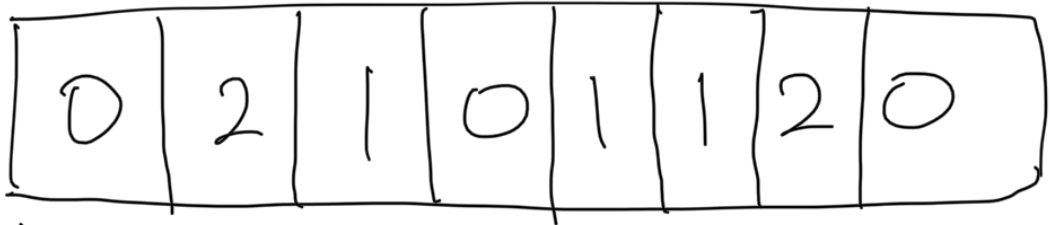


# 75 Sort Colors. [aka Dutch National Flag algorithm]

Sort an array of 0, 1 & 2s.



low  
mid

high

— everything before low are zeros.

— everything after high are twos.

(check arr[mid])

{

mid  $\leq$  high

if it is 0 : swap arr[low] & arr[mid]  
low++ & mid++

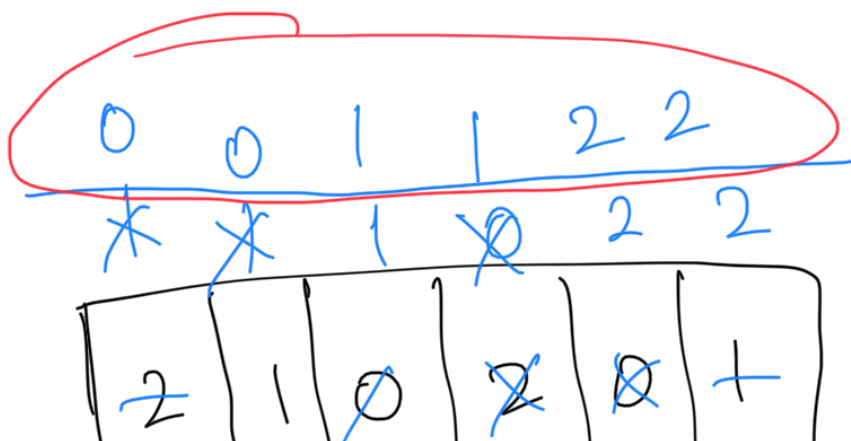
if 1 : mid++

if 2 : swap arr[mid] & arr[high]  
high-- (no mid++)

Dry run :

0  $\left[ \begin{matrix} a[l] \\ a[m] \end{matrix} \right]$   
 $l++, m++$

1  $m++$



2  $\begin{pmatrix} a[m] \\ a[r] \end{pmatrix}$   
 $r--$



$m > r$  ! break

l ~~0~~ \* 2  
m ~~0~~ \* ~~2~~ \* 4  
r ~~5~~ ~~4~~ 3

Code:

```
class Solution:
    def sortColors(self, nums: List[int]) -> None:
        """
        Do not return anything, modify nums in-place instead.
        """
        # aka Dutch National Flag
        low, mid, high = 0, 0, len(nums) - 1

        while mid <= high:
            # if element at mid is 0: swap element at low and mid; increment low and mid
            if nums[mid] == 0:
                nums[low], nums[mid] = nums[mid], nums[low]
                low += 1
                mid += 1
            # if element at mid is 1: increment mid
            elif nums[mid] == 1:
                mid += 1
            # if element at mid is 2: swap element at mid and high; increment high
            elif nums[mid] == 2:
                nums[mid], nums[high] = nums[high], nums[mid]
                high -= 1
```