Sort colors

↳ Red, green, blue

↓ ↓ ↓

0 1 2

[1, 1, 0, 2, 0, 2] ← Input

[0, 0, 1, 1, 2, 2] ← output

1) MergeSort/QuickSort

↳ $\Theta(n\log n)$

Optimized approach

2) Two pointers

- P_0 → to store 0's in extreme left
- P_2 → to store 2's in extreme right

$P_0 = 0$

$P_2 = 5$

0 1 2 3 4 5

[2, 0, 2, 1, 1, 0]

↑

n → size of an array

$\Theta(n)$

P_0, curr = 0

$P_2 = n - 1$

```
while curr <= p2:
    if nums(curr) == 0:
        swap(nums(curr), nums(p0))
        p0 += 1
        curr += 1
        elif nums(curr) = = 2:
            swap(num(curr), nums(P2))
            P2 -= 1

        else:
            cur += 1

    return nums
```