

## 75. Sort Colors

An array: `nums`

$n$  objects: `red, white, blue`  
                  ↓           ↓           ↓  
                  0           1           2

Goal: make some color adjacent

Example: ① `[2, 0, 2, 1, 1, 0]`

↓

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

② `[2, 0, 1]`

↓

`[0, 1, 2]`

Constraint: without using sort function

Analyse:

Brute-force:

2 placeholders, get 0, 1, 2 independent;

eg. , [ 2, 0, 2, 1, 1, 0 ]

1st [ 0, 0 ]

2nd [ 1, 1 ]

remove elements when append in  
a new list .

pseudo - code :

a = [ ] , b = [ ]

for i in len(nums):

if nums[i] == 0:

a.append(0)

nums[i] = -1

if nums[i] == 1:

b.append(1)

nums[i] = -1

nums = a + b + nums

for x in nums:

nums.remove(x == -1)

---

Partition :

three region

$[0 \dots \text{zero}] = 0$

$(\text{zero} \dots i) = 1$

$[\text{two} \dots \text{len}-1] = 2$

$\text{zero} = -1$

$\text{two} = \text{len}$

for  $i = 0, i < \text{two}$ :

if  $\text{nums}[i] == 0$ :

$\text{zero}++$

Helper  $\longleftrightarrow$  Swap  $(\text{nums}, i, \text{zero})$   
 $i++$

else if  $\text{nums}[i] == 1$ :

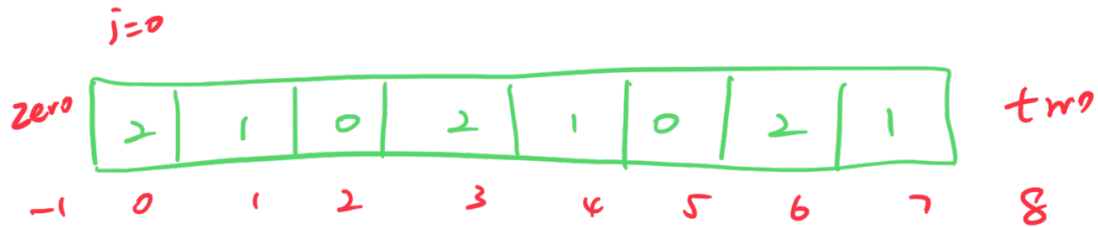
$i++$

else

$\text{two}--$

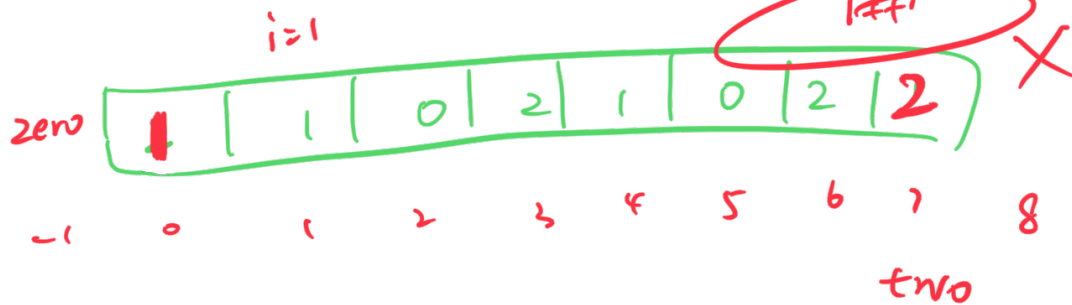
swap(nums, i, two)

Graph explain:

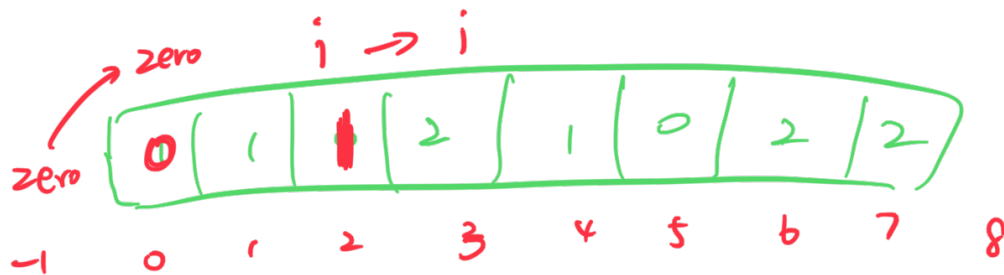


$j=0$ ,  $nums[i] = 2$ ,  $two--$

swap(i, two)



$j=1$ ,  $nums[i] = 1$ ,  $i++$



$i=2$ ,  $nums[i] = 0$

two

zero++

swap(i, zero)

$i++$



0 1 2 3 4 5 6 7

zero  $i \rightarrow i$  two

0	0	1	2	1	0	2	2
---	---	---	---	---	---	---	---

0 1 2 3 4 5 6 7

zero  $i \rightarrow i$  two

0	0	1	2	1	1	2	2
---	---	---	---	---	---	---	---

0 1 2 3 4 5 6 7

stop .

why wrong?

Redo

2

1	1	0	2	1	0	2	2
---	---	---	---	---	---	---	---

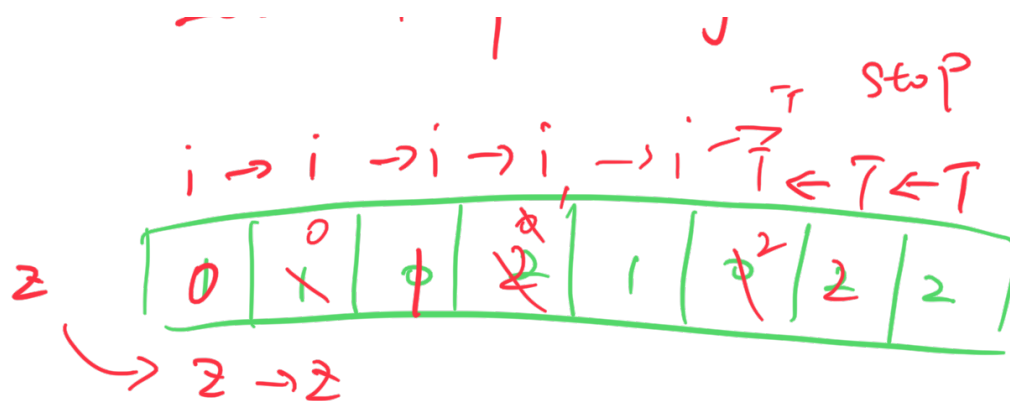
T

Find the problem .

should not move i when

$nums[i] = 2$

Let's keep moving :



work ✓