

# 080. Remove Duplicates from Sorted Array II

## 080 Remove Duplicates from Sorted Array II

- Two Pointers+Array

### Description

Follow up for "Remove Duplicates":

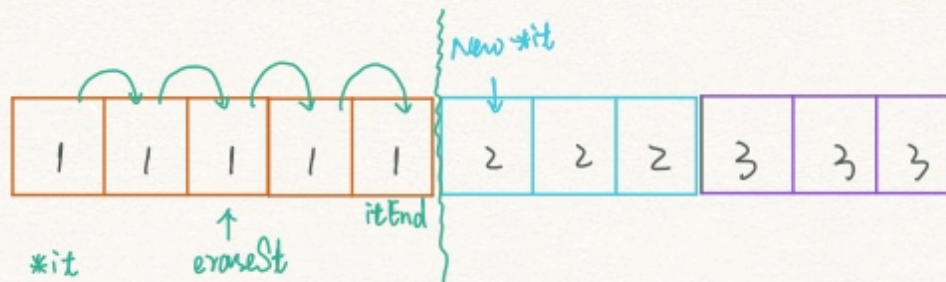
What if duplicates are allowed at most *twice*?

For example,

Given sorted array *nums* = [1,1,1,2,2,3] ,

Your function should return length = 5 , with the first five elements of *nums* being 1 , 1 , 2 , 2 and 3 . It doesn't matter what you leave beyond the new length.

### 1. Thought line



After 1<sup>st</sup> while loop,

it is at "itEnd"

⇒ Erase [eraseSt, itEnd]

★ vector.erase(a, b)

= erase [a, b)

⇒ nums.erase(eraseSt, itEnd+1)

New \*it = nums.erase(eraseSt, itEnd+1)

Because erase-~~for~~'s feature.

## 2. Two Pointers+Array

```

1 class Solution {
2 public:
3     int removeDuplicates(vector<int>& nums) {
4         int result = 0;
5         for (auto it = nums.begin(), eraseSt = it; it != nums.end(); eraseSt = it){
6             int duplicateAct = 1;
7             while (it + 1 != nums.end() && *(it+1) == *it){
8                 ++duplicateAct, ++it;
9                 if (duplicateAct == 3) eraseSt = it;
10            }
11            if (duplicateAct > 2) it = nums.erase(eraseSt, it+1);
12            else ++it;
13            result += duplicateAct < 2 ? duplicateAct : 2;
14        }
15        return result;
16    }
17 };

```