

080. Remove Duplicates from Sorted Array II

080 Remove Duplicates from Sorted Array II

- Two Pointers+Array

Description

Follow up for "Remove Duplicates":

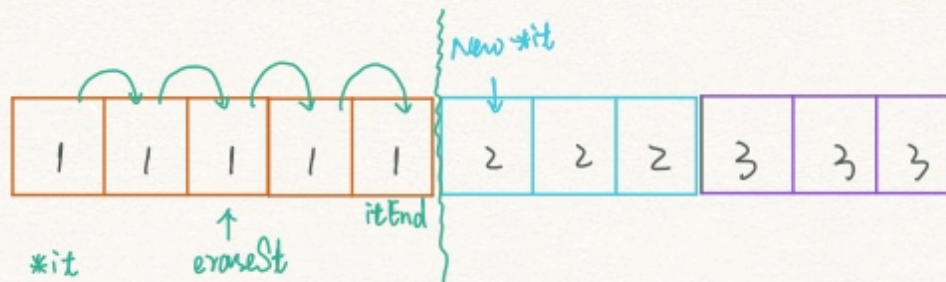
What if duplicates are allowed at most *twice*?

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 1st 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

```
class Solution {
public:
    int removeDuplicates(vector<int>& nums) {
        int result = 0;
        for (auto it = nums.begin(), eraseSt = it; it != nums.end(); eraseSt = it){
            int duplicateAct = 1;
            while (it + 1 != nums.end() && *(it+1) == *it){
                ++duplicateAct, ++it;
                if (duplicateAct == 3) eraseSt = it;
            }
            if (duplicateAct > 2) it = nums.erase(eraseSt, it+1);
            else ++it;
            result += duplicateAct < 2 ? duplicateAct : 2;
        }
        return result;
    }
};
```