26. Remove Duplicates from Sorted Array

Easy 2903 5704 Add to List Share

Given a sorted array *nums*, remove the duplicates **in-place** such that each element appear only *once* and return the new length.

Do not allocate extra space for another array, you must do this by **modifying the input array in-place** with O(1) extra memory.

Example 1:

```
Given nums = [1,1,2],

Your function should return length = 2, with the first two elements of nums being
1 and 2 respectively.

It doesn't matter what you leave beyond the returned length.
```

Example 2:

```
Given nums = [0,0,1,1,1,2,2,3,3,4],

Your function should return length = 5, with the first five elements of nums
being modified to 0, 1, 2, 3, and 4 respectively.

It doesn't matter what values are set beyond the returned length.
```

Clarification:

Confused why the returned value is an integer but your answer is an array?

Note that the input array is passed in by **reference**, which means modification to the input array will be known to the caller as well.

Internally you can think of this:

```
// nums is passed in by reference. (i.e., without making a copy)
int len = removeDuplicates(nums);

// any modification to nums in your function would be known by the caller.
```

C++ Autocomplete

```
1  class Solution {
2  public:
3      int removeDuplicates(vector<int>& a) {
4          if(a.size()==0)
5              return 0;
6          int sp=0;
7          for(int i=1;i<a.size();i++){
8              if(a[i]!=a[sp]){
9                  sp++;
10                 a[sp]=a[i];
11             }
12         }
13         return sp+1;
14
15     }
16 };
```

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

Your input [1,1,2]

Output [1,2] Diff

Expected [1,2]