Description | Solution | Discuss (99...) | Submissions | Python3 | Autocomplete

27. Remove Element

Easy 2370 3804 Add to List Share

Given an integer array `nums` and an integer `val`, remove all occurrences of `val` in `nums` **in-place**. The relative order of the elements may be changed.

Since it is impossible to change the length of the array in some languages, you must instead have the result be placed in the **first part** of the array `nums`. More formally, if there are `k` elements after removing the duplicates, then the first `k` elements of `nums` should hold the final result. It does not matter what you leave beyond the first `k` elements.

Return `k` *after placing the final result in the first* `k` *slots of* `nums`.

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

Custom Judge:

The judge will test your solution with the following code:

```
int[] nums = [...]; // Input array
int val = ...; // Value to remove
int[] expectedNums = [...]; // The expected answer
with correct length.
                            // It is sorted with no
values equaling val.

int k = removeElement(nums, val); // Calls your
implementation

assert k == expectedNums.length;
sort(nums, 0, k); // Sort the first k elements of
```

```python
class Solution:
    def removeElement(self, nums: List[int], val: int) -> int:

        i=0
        while i < len(nums):

            if nums[i] == val:
                nums.remove(nums[i])

                i=i-1

            i=i+1

        return(len(nums))
```

Your previous code was restored from your local storage. Reset to default

Testcase | Run Code Result | Debugger

Accepted Runtime: 48 ms

Your input	[3,2,2,3] 3
Output	[2,2]
Expected	[2,2]

Diff