

Description

Solution

Discuss

Submissions

C++

Autocomplete

i

{ }

↺

↻

⌵

## 27. Remove Element

Easy

1754

3112

Add to List

Share

Given an array *nums* and a value *val*, remove all instances of that value **in-place** 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.

The order of elements can be changed. It doesn't matter what you leave beyond the new 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 a 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 = removeElement(nums, val);

// any modification to nums in your function
// would be known by the caller.
// using the length returned by your function,
// it prints the first len elements.
for (int i = 0; i < len; i++) {
    print(nums[i]);
}
```

### Example 1:

**Input:** *nums* = [3,2,2,3], *val* = 3

**Output:** 2, *nums* = [2,2]

**Explanation:** Your function should return length = 2, with the first two elements of *nums* being 2.

It doesn't matter what you leave beyond the returned length. For example if you return 2 with *nums* = [2,2,3,3] or *nums* = [2,3,0,0], your answer will be accepted.

### Example 2:

**Input:** *nums* = [0,1,2,2,3,0,4,2], *val* = 2

**Output:** 5, *nums* = [0,1,4,0,3]

**Explanation:** Your function should return length = 5, with the first five elements of *nums* containing 0, 1, 3, 0, and 4. Note that the order of those five elements can be arbitrary. It doesn't matter what values are set beyond the returned length.

```
1 class Solution {
2 public:
3     int removeElement(vector<int>& nums, int val) {
4
5     }
6 };
```

Need more space? Try using full screen mode!

Got it!

1. 문제를 정확하게 파악할 수 있어야한다. 문제에서 주어진 조건이 왜 주어졌을 지 생각해본다. 출제자의 의도를 파악하면 알고리즘을 택하는데 있어서 시간을 크게 줄일 수 있다.
2. 시간복잡도:  $O(1)$  => 새로운 배열의 저장공간을 만들지 말고 인자로 주어진 배열 내에서 해결 해야한다.
3. 배열은 Arrays 클래스에 있는 메소드를 사용 할 수 있다. 하지만 이 클래스에는 값을 제거하는 메소드가 존재하지 않는다. => 제거하는 메소드가 존재하는 다른 종류의 컨테이너(리스트, 배열리스트 ...)로 변환하여 제거한 뒤에 다시 배열로 변환하는 알고리즘이 있지만, 시간복잡도의 조건에 맞지 않는다.
4. 문제를 읽다보면 여기서 제거의 의미는 배열에서 실제로 그 값을 제거하라는 의미가 아니다. 반환하는 값을 보면 원하는 값을 지웠을 때의 크기를 반환하면 된다. 그리고 주어진 배열 내에서 제거하고자 하는 값이 뒤로 밀려있으면 된다. 그 말은 단순히 배열 내의 값의 위치를 조정만 할 수 있으면 되고 제거하고자 하는 값이 배열 내에 몇 개가 있는지 파악하면 된다.
5. 배열을 새로운 컨테이너로 변환하는 것이 아닌 조건에만 맞게 문제를 풀려고 접근 했을 시 더 쉽게 끝낼 수 있다. 그것이 출제자가 원하는 특정한 이 문제에 대한 접근 방식이다.
6. 배열을 새로운 컨테이너로 변환하려고 할 때 생겼던 문제점들은 사실 상 이 문제를 훨씬 더 복잡하게 만든다. 왜냐하면 주어진 배열은 프리미티브(primitive) int 타입이기 때문에 리스트와 같은 컨테이너로 변환 하기전에 객체 타입으로 먼저 배열을 변환 해야한다. (int -> Integer) 그 과정에서 반복문이 필요하며 코드는 복잡해진다.