Description | Editorial | Solutions | Submissions

1. Two Sum

Solved

Easy | Topics | Companies | Hint

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Example 2:

```
Input: nums = [3,2,4], target = 6
Output: [1,2]
```

Example 3:

Code

C++ | Auto

```cpp
class Solution
{
    public:
    vector<int> twoSum(vector<int>& nums, int target)
    {
        vector<int> res;
        for (int i=0;i<nums.size();i++)
        {
            for (int j=i+1;j<nums.size();j++)
            {
                int sum=nums[i]+nums[j];
                if (sum==target)
                {
                    res.push_back(i);
                    res.push_back(j);
                    break;
                }
            }
        }
        return res;
    }
};
```

Saved to local

Ln 8, Col 10

Description | Editorial | Solutions | Submissions

26. Remove Duplicates from Sorted Array

Solved

Easy | Topics | Companies | Hint

Given an integer array `nums` sorted in **non-decreasing order**, remove the duplicates **in-place** such that each unique element appears only **once**. The **relative order** of the elements should be kept the **same**. Then return *the number of unique elements in* `nums`.

Consider the number of unique elements of `nums` to be `k`, to get accepted, you need to do the following things:

- Change the array `nums` such that the first `k` elements of `nums` contain the unique elements in the order they were present in `nums` initially. The remaining elements of `nums` are not important as well as the size of `nums`.
- Return `k`.

Custom Judge:

The judge will test your solution with the following code:

```
int[] nums = [...]; // Input array
int[] expectedNums = [...]; // The expected answer with correct
length
```

Code

C++ | Auto

```cpp
class Solution
{
    public:
    int removeDuplicates(std::vector<int>& nums)
    {
        int count=0;
        for (int i=1;i<nums.size();i++)
        {
            if (nums[i]!=nums[count])
            {
                count+=1;
                nums[count]=nums[i];
            }
        }
        nums.resize(count+1);
        return count+1;
    }
};
```