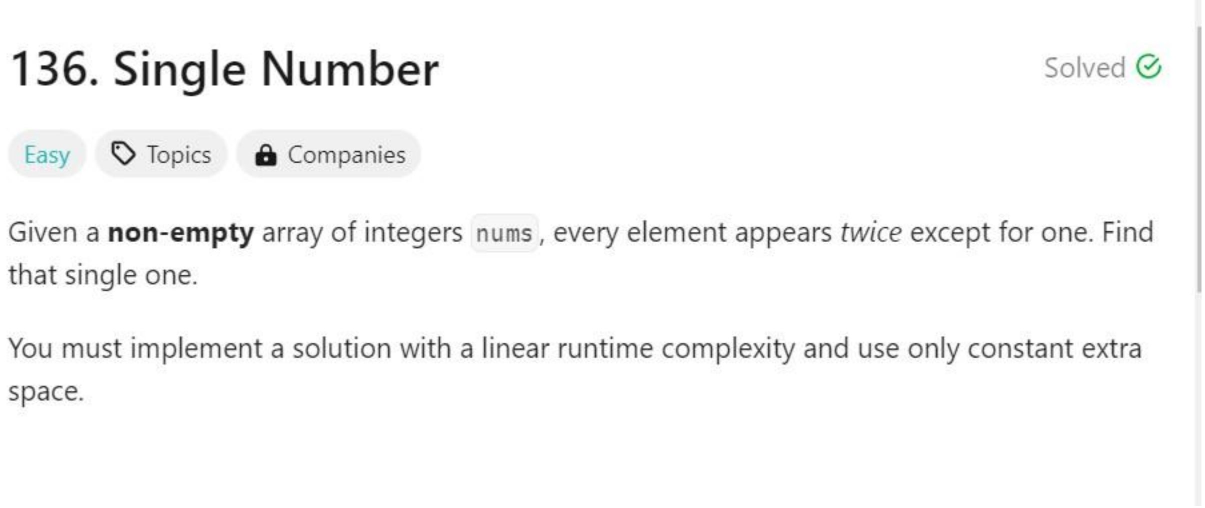
1)

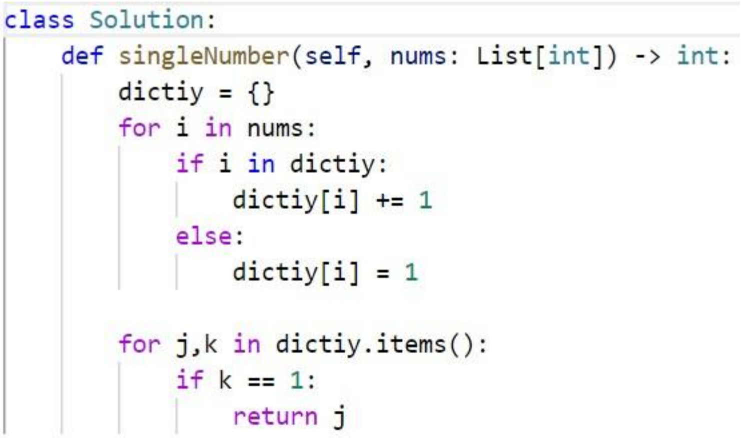


136. Single Number

Solved ✓

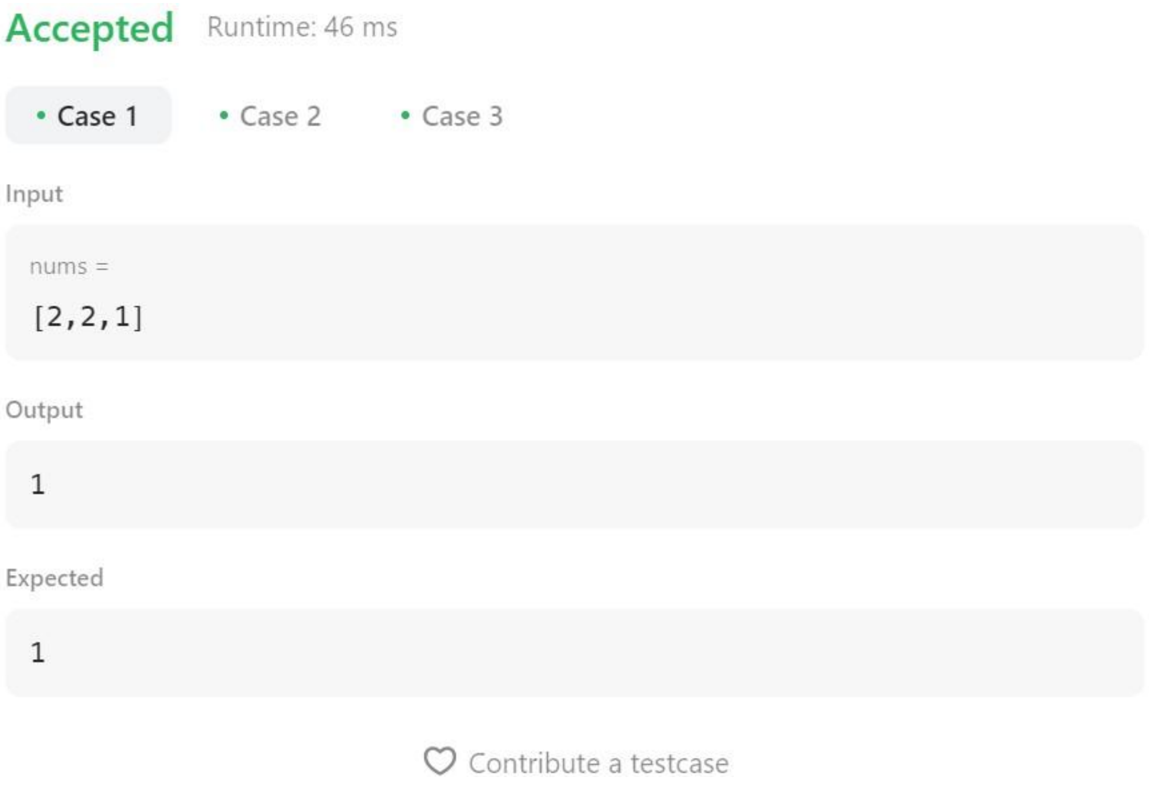
Given a **non-empty** array of integers `nums`, every element appears *twice* except for one. Find that single one.

You must implement a solution with a linear runtime complexity and use only constant extra space.



```
class Solution:
    def singleNumber(self, nums: List[int]) -> int:
        dictiy = {}
        for i in nums:
            if i in dictiy:
                dictiy[i] += 1
            else:
                dictiy[i] = 1

        for j,k in dictiy.items():
            if k == 1:
                return j
```



Accepted Runtime: 46 ms

Input

nums =

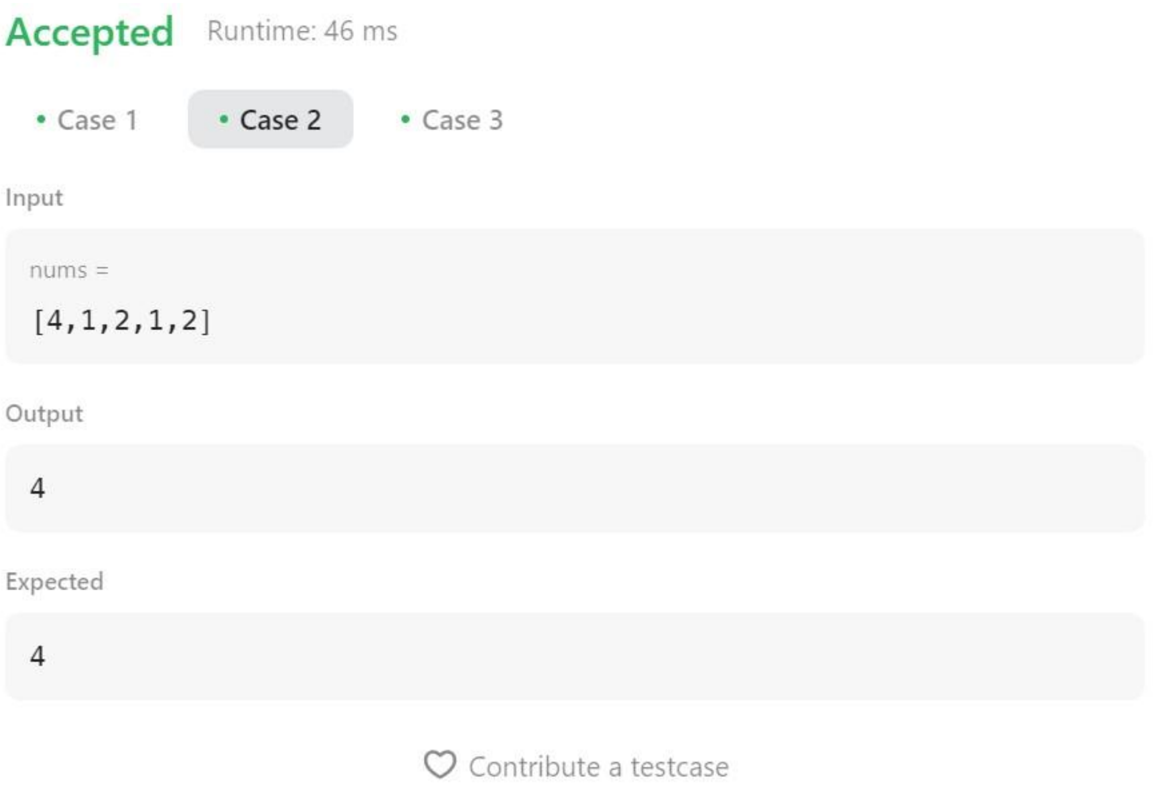
[2,2,1]

Output

1

Expected

1



Accepted Runtime: 46 ms

Input

nums =

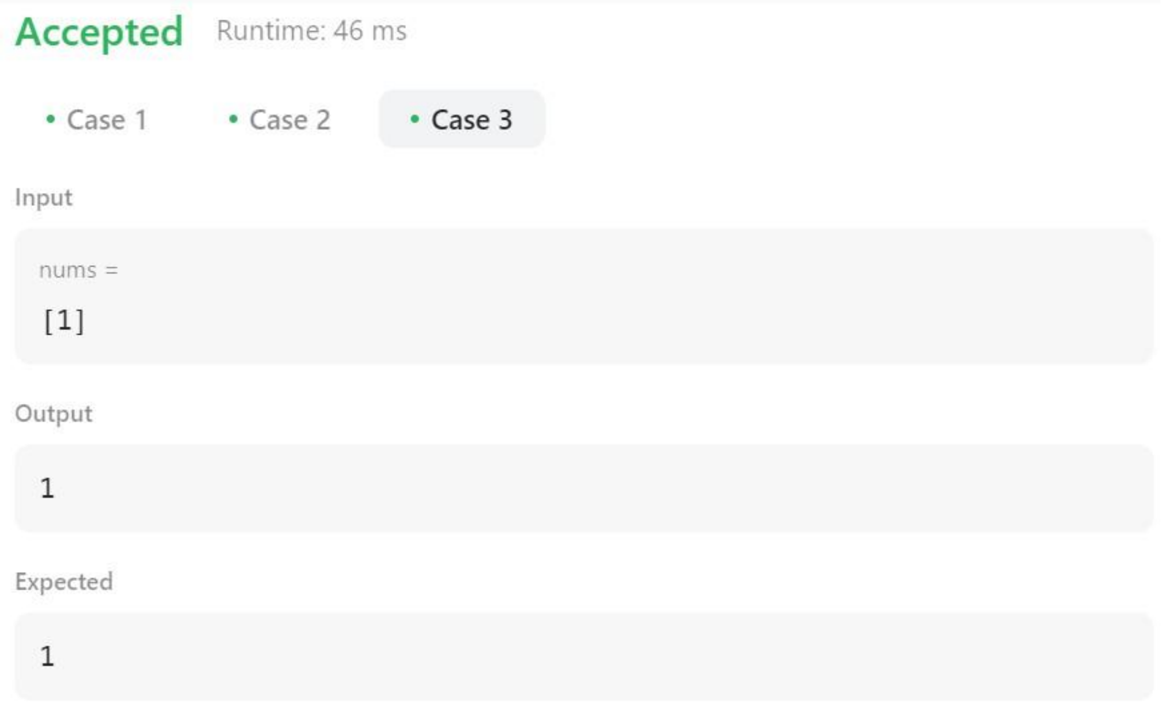
```
[4,1,2,1,2]
```

Output

```
4
```

Expected

```
4
```



Accepted Runtime: 46 ms

Input

nums =

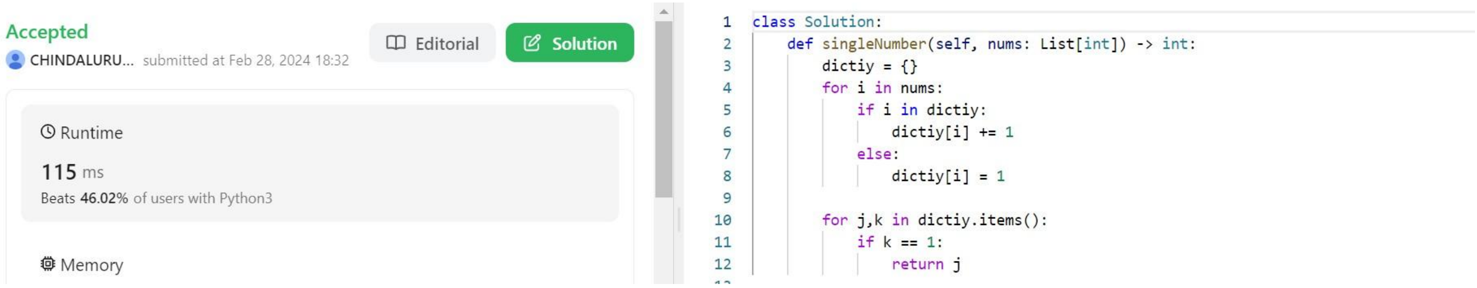
```
[1]
```

Output

```
1
```

Expected

```
1
```



Accepted

CHINDALURU... submitted at Feb 28, 2024 18:32

Runtime

115 ms

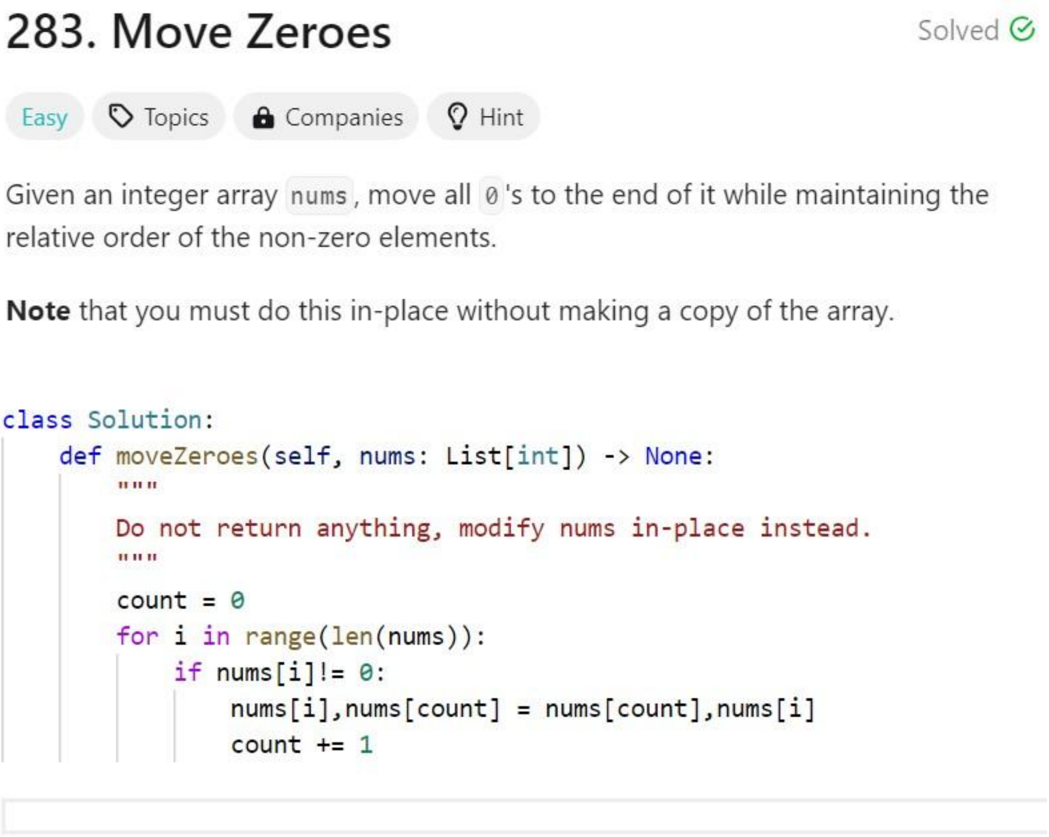
Beats 46.02% of users with Python3

Memory

```
class Solution:
    def singleNumber(self, nums: List[int]) -> int:
        dictiy = {}
        for i in nums:
            if i in dictiy:
                dictiy[i] += 1
            else:
                dictiy[i] = 1

        for j,k in dictiy.items():
            if k == 1:
                return j
```

2)

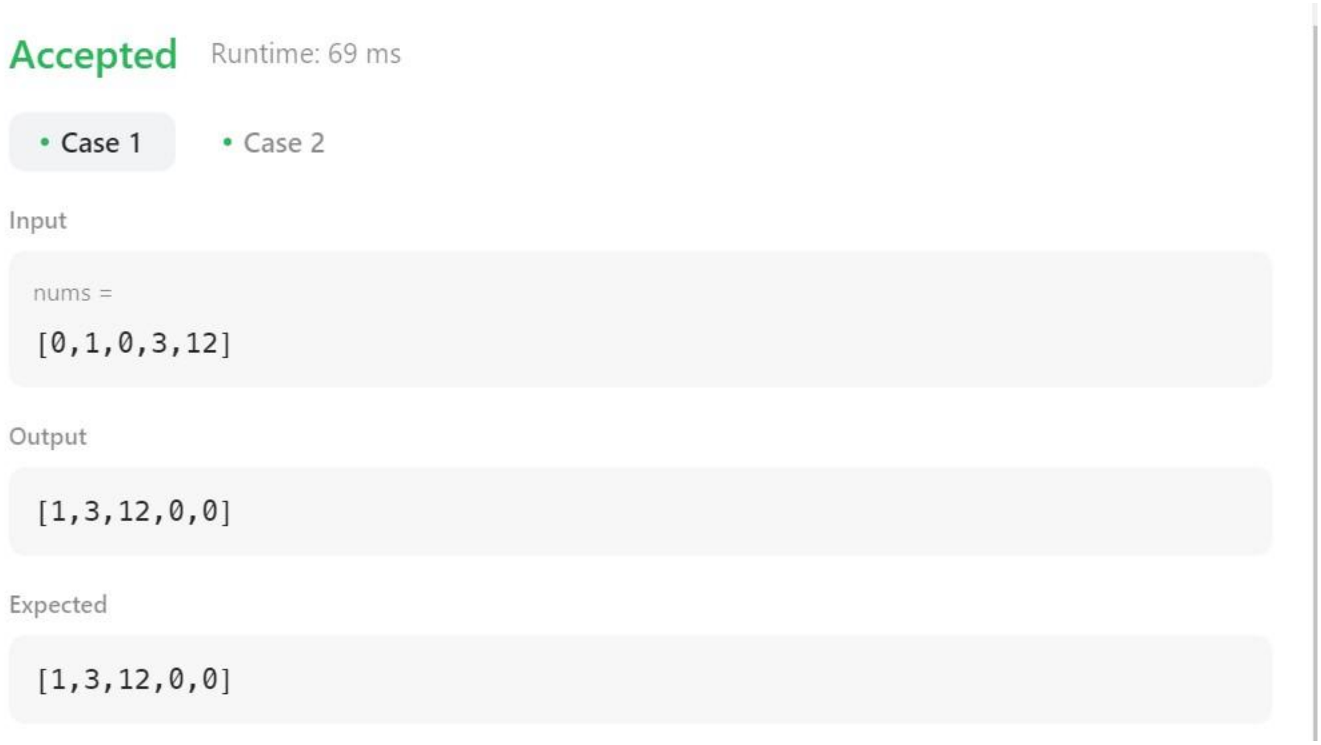


283. Move Zeroes

Given an integer array `nums`, move all `0`'s to the end of it while maintaining the relative order of the non-zero elements.

Note that you must do this in-place without making a copy of the array.

```
class Solution:
    def moveZeroes(self, nums: List[int]) -> None:
        """
        Do not return anything, modify nums in-place instead.
        """
        count = 0
        for i in range(len(nums)):
            if nums[i]!= 0:
                nums[i],nums[count] = nums[count],nums[i]
                count += 1
```



Accepted Runtime: 69 ms

Input

nums =

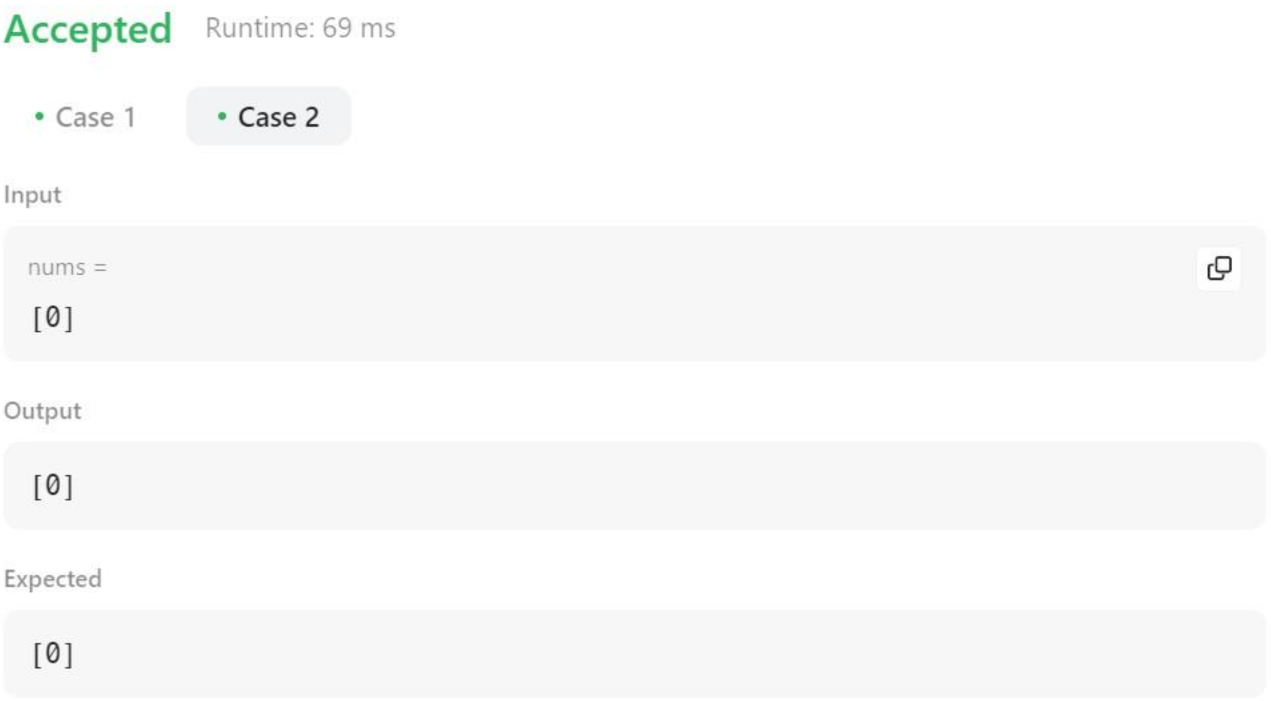
```
[0,1,0,3,12]
```

Output

```
[1,3,12,0,0]
```

Expected

```
[1,3,12,0,0]
```



Accepted Runtime: 69 ms

Input

nums =

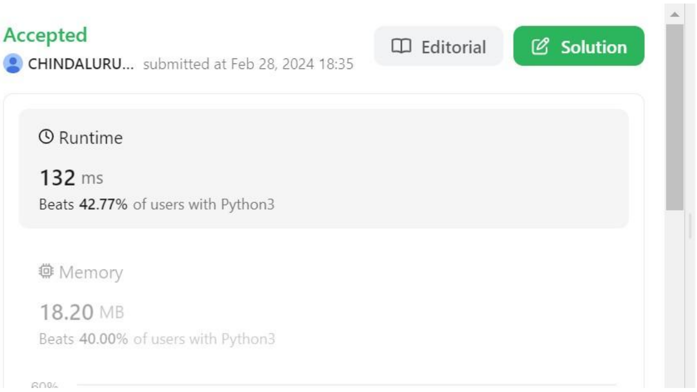
```
[0]
```

Output

```
[0]
```

Expected

```
[0]
```



Accepted

CHINDALURU... submitted at Feb 28, 2024 18:35

Runtime

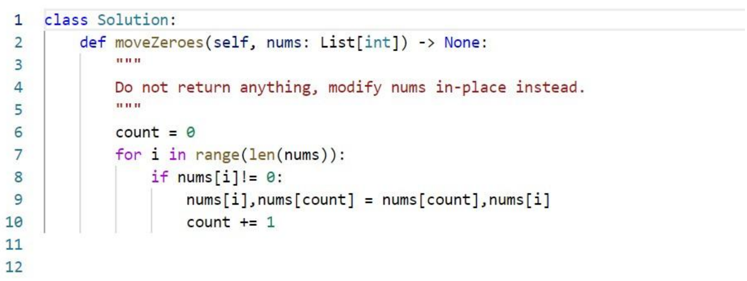
132 ms

Beats 42.77% of users with Python3

Memory

18.20 MB

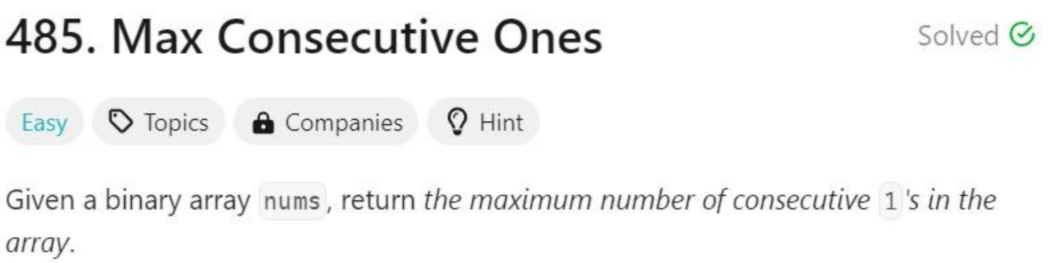
Beats 40.00% of users with Python3



```
class Solution:
    def moveZeroes(self, nums: List[int]) -> None:
        """
        Do not return anything, modify nums in-place instead.
        """
        count = 0
        for i in range(len(nums)):
            if nums[i]!= 0:
                nums[i],nums[count] = nums[count],nums[i]
                count += 1


```

3)

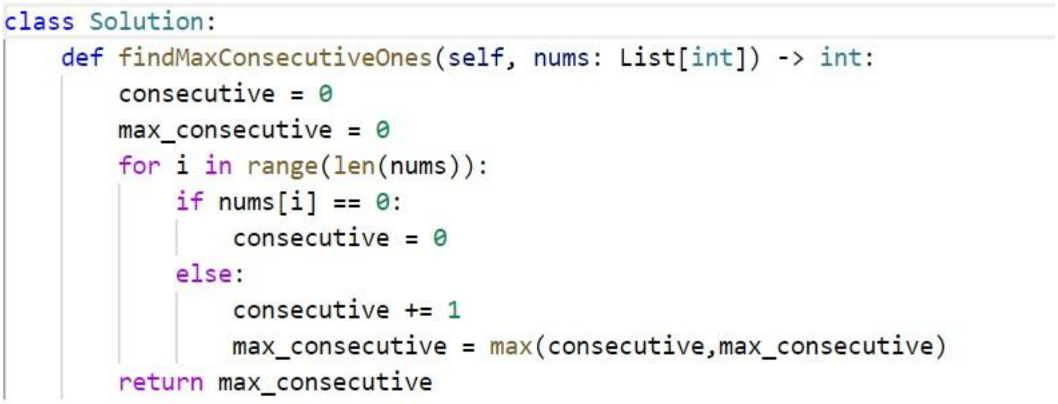


485. Max Consecutive Ones

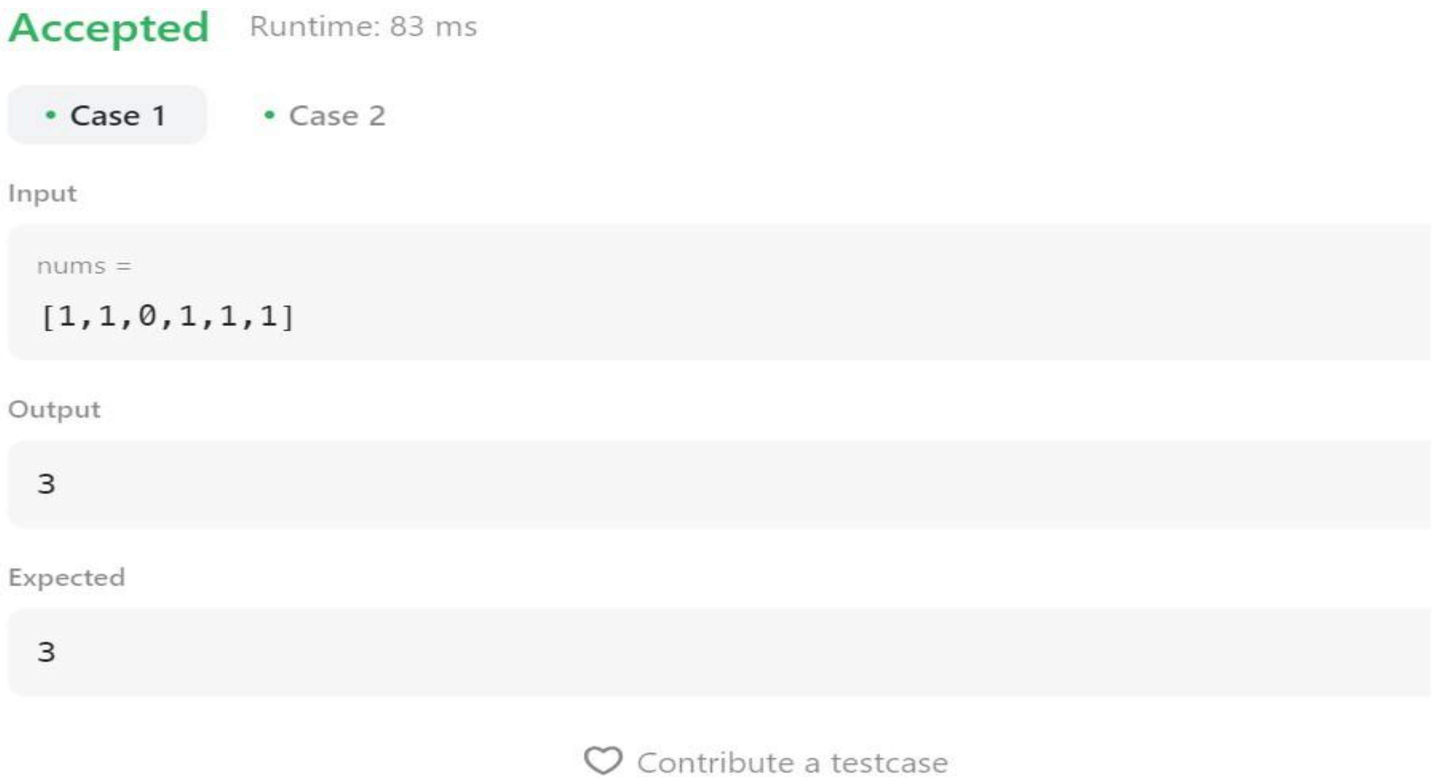
Solved ✓

Easy · Topics · Companies · Hint

Given a binary array `nums`, return *the maximum number of consecutive* `1`*'s in the array*.



```
class Solution:
    def findMaxConsecutiveOnes(self, nums: List[int]) -> int:
        consecutive = 0
        max_consecutive = 0
        for i in range(len(nums)):
            if nums[i] == 0:
                consecutive = 0
            else:
                consecutive += 1
                max_consecutive = max(consecutive,max_consecutive)
        return max_consecutive
```



Input

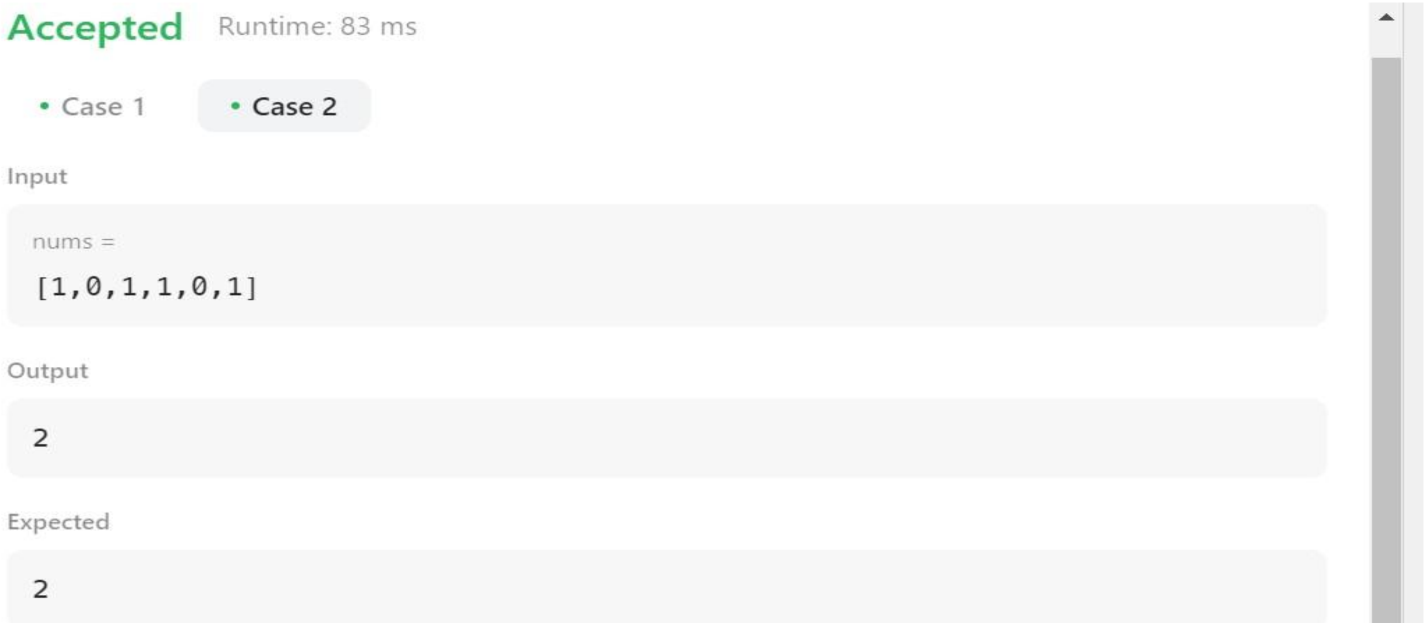
```
nums =
[1,1,0,1,1,1]
```

Output

```
3
```

Expected

```
3
```



Input

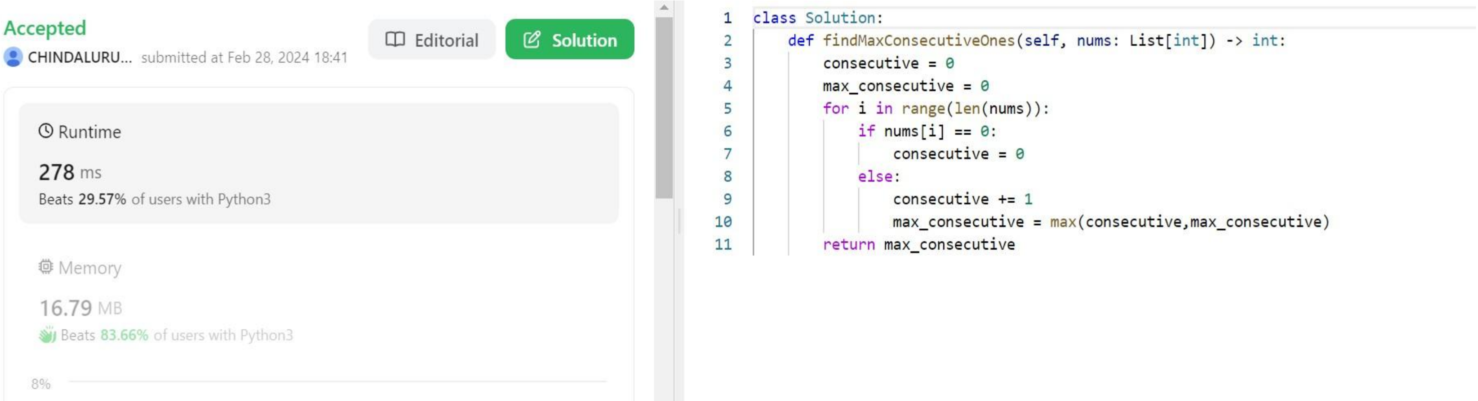
```
nums =
[1,0,1,1,0,1]
```

Output

```
2
```

Expected

```
2
```



CHINDALURU... submitted at Feb 28, 2024 18:41

Runtime

278 ms

Beats 29.57% of users with Python3

Memory

16.79 MB

Beats 83.66% of users with Python3

```python
class Solution:
    def findMaxConsecutiveOnes(self, nums: List[int]) -> int:
        consecutive = 0
        max_consecutive = 0
        for i in range(len(nums)):
            if nums[i] == 0:
                consecutive = 0
            else:
                consecutive += 1
                max_consecutive = max(consecutive,max_consecutive)
        return max_consecutive
```