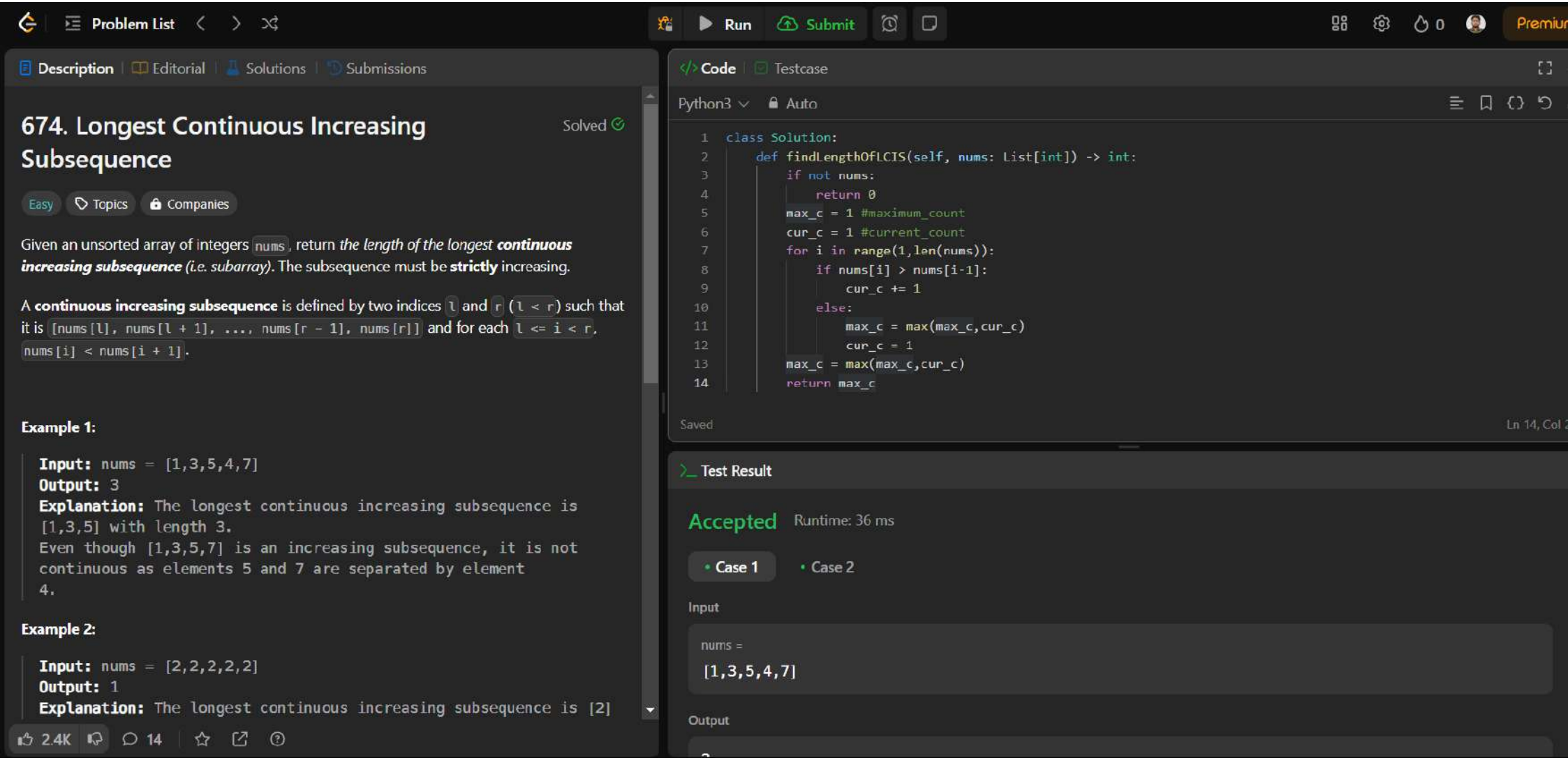
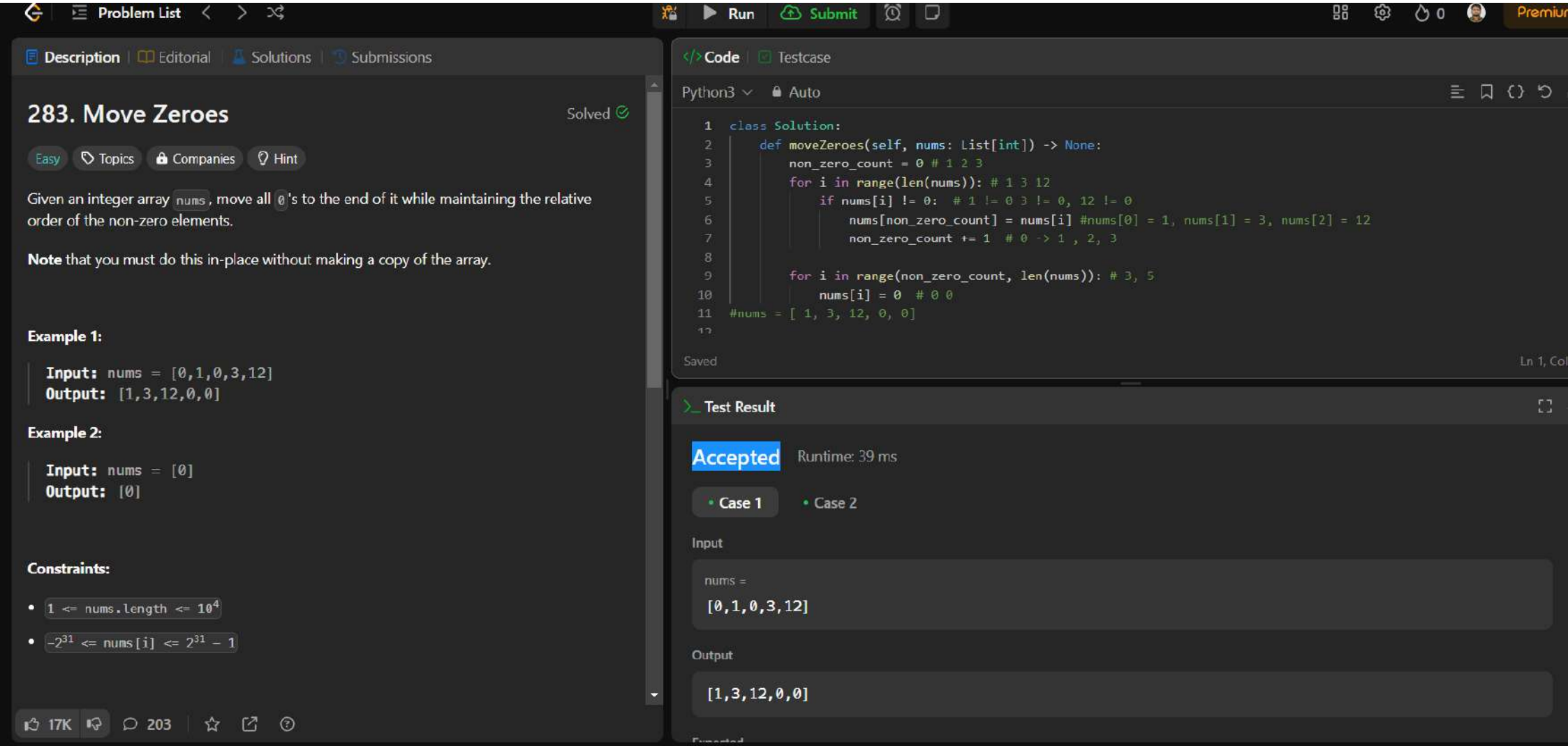

Problem List
Run
Submit
Premiu
Description
Editorial
Solutions
Submissions
674. Longest Continuous Increasing Subsequence
Solved
Easy
Topics
Companies
Given an unsorted array of integers nums, return the length of the longest continuous increasing subsequence (i.e. subarray). The subsequence must be strictly increasing.
A continuous increasing subsequence is defined by two indices l and r (l < r) such that it is [nums[l], nums[l + 1], ..., nums[r - 1], nums[r]] and for each l <= i < r, nums[i] < nums[i + 1].
Example 1:
Input: nums = [1,3,5,4,7]
Output: 3
Explanation: The longest continuous increasing subsequence is [1,3,5] with length 3.
Even though [1,3,5,7] is an increasing subsequence, it is not continuous as elements 5 and 7 are separated by element 4.
Example 2:
Input: nums = [2,2,2,2,2]
Output: 1
Explanation: The longest continuous increasing subsequence is [2]
2.4K
14
Code
Testcase
Python3
Auto
1 class Solution:
2 def findLengthOfLCIS(self, nums: List[int]) -> int:
3 if not nums:
4 return 0
5 max_c = 1 #maximum_count
6 cur_c = 1 #current_count
7 for i in range(1,len(nums)):
8 if nums[i] > nums[i-1]:
9 cur_c += 1
10 else:
11 max_c = max(max_c,cur_c)
12 cur_c = 1
13 max_c = max(max_c,cur_c)
14 return max_c
Saved
Ln 14, Col
Test Result
Accepted
Runtime: 36 ms
Case 1
Case 2
Input
nums =
[1,3,5,4,7]
Output


Problem List
Run
Submit
Premiu
Description
Editorial
Solutions
Submissions
283. Move Zeroes
Solved
Easy
Topics
Companies
Hint
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.
Example 1:
Input: nums = [0,1,0,3,12]
Output: [1,3,12,0,0]
Example 2:
Input: nums = [0]
Output: [0]
Constraints:
1 <= nums.length <= 10^4
-2^31 <= nums[i] <= 2^31 - 1
17K
203
Code
Testcase
Python3
Auto
class Solution:
def moveZeroes(self, nums: List[int]) -> None:
non_zero_count = 0 # 1 2 3
for i in range(len(nums)): # 1 3 12
if nums[i] != 0: # 1 != 0 3 != 0, 12 != 0
nums[non_zero_count] = nums[i] #nums[0] = 1, nums[1] = 3, nums[2] = 12
non_zero_count += 1 # 0 -> 1 , 2, 3
for i in range(non_zero_count, len(nums)): # 3, 5
nums[i] = 0 # 0 0
#nums = [1, 3, 12, 0, 0]
Saved
Ln 1, Co
Test Result
Accepted
Runtime: 39 ms
Case 1
Case 2
Input
nums =
[0,1,0,3,12]
Output
[1,3,12,0,0]

Problem List

Run Submit

Premium

Description | Editorial | Solutions | Submissions

66. Plus One

Solved

Easy Topics Companies

You are given a **large integer** represented as an integer array `digits`, where each `digits[i]` is the `i^th` digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading `0`'s.

Increment the large integer by one and return *the resulting array of digits*.

Example 1:

```
Input: digits = [1,2,3]
Output: [1,2,4]
Explanation: The array represents the integer 123.
Incrementing by one gives 123 + 1 = 124.
Thus, the result should be [1,2,4].
```

Example 2:

```
Input: digits = [4,3,2,1]
Output: [4,3,2,2]
Explanation: The array represents the integer 4321.
Incrementing by one gives 4321 + 1 = 4322.
Thus, the result should be [4,3,2,2].
```

Example 3:

9.6K 248

Code | Testcase

Python3 Auto

```python
class Solution:
    def plusOne(self, digits: List[int]) -> List[int]:
        s = int(''.join([str(i) for i in digits]))
        s += 1
        ans = list(str(s))
        return [int(i) for i in ans]
```

Saved

Ln 1, Col 1

Test Result

Accepted Runtime: 47 ms

• Case 1 • Case 2 • Case 3

Input

digits =

[1,2,3]

Output

[1,2,4]

Expected

[1,2,4]

Description | Editorial | Solutions | Submissions

169. Majority Element

Solved

Easy | Topics | Companies

Given an array `nums` of size `n`, return *the majority element*.

The majority element is the element that appears more than `⌊n / 2⌋` times. You may assume that the majority element always exists in the array.

Example 1:

```
Input: nums = [3,2,3]
Output: 3
```

Example 2:

```
Input: nums = [2,2,1,1,1,2,2]
Output: 2
```

Constraints:

- `n == nums.length`
- `1 <= n <= 5 * 10^4`
- `-10^9 <= nums[i] <= 10^9`

19.6K | 288

Code | Testcase

Python3 | Auto

```python
class Solution:
    def majorityElement(self, nums: List[int]) -> int:
        nums.sort()
        return nums[len(nums)//2]
```

Saved | Ln 1, Col 1

Test Result

Accepted Runtime: 44 ms

• Case 1 • Case 2

Input

nums =
[3,2,3]

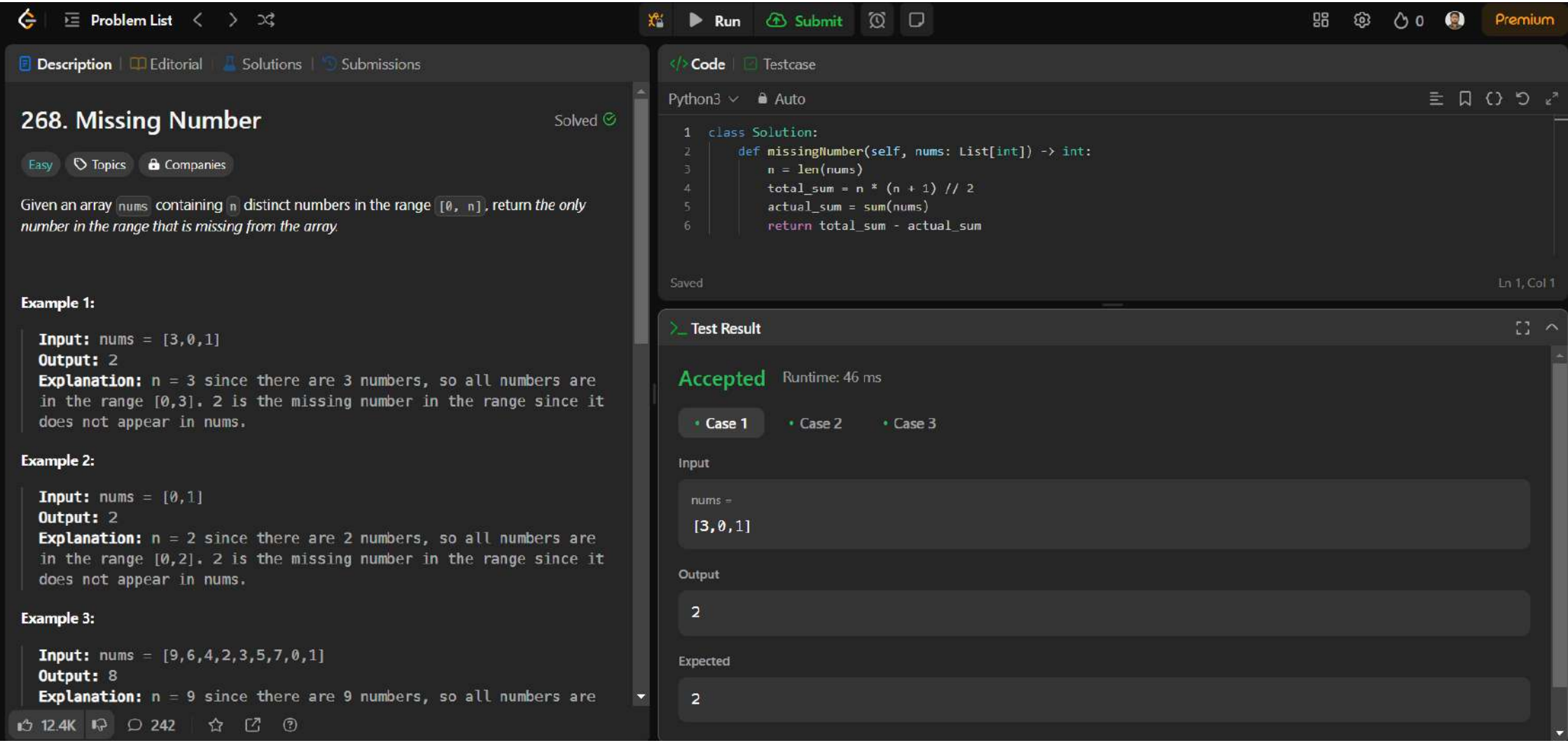
Output

3

Expected

3

Contribute a testcase


Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
268. Missing Number
Solved
Easy
Topics
Companies
Given an array nums containing n distinct numbers in the range [0, n], return the only number in the range that is missing from the array.
Example 1:
Input: nums = [3,0,1]
Output: 2
Explanation: n = 3 since there are 3 numbers, so all numbers are in the range [0,3]. 2 is the missing number in the range since it does not appear in nums.
Example 2:
Input: nums = [0,1]
Output: 2
Explanation: n = 2 since there are 2 numbers, so all numbers are in the range [0,2]. 2 is the missing number in the range since it does not appear in nums.
Example 3:
Input: nums = [9,6,4,2,3,5,7,0,1]
Output: 8
Explanation: n = 9 since there are 9 numbers, so all numbers are
12.4K
242
Code
Testcase
Python3
Auto
class Solution:
def missingNumber(self, nums: List[int]) -> int:
n = len(nums)
total_sum = n * (n + 1) // 2
actual_sum = sum(nums)
return total_sum - actual_sum
Saved
Ln 1, Col 1
Test Result
Accepted
Runtime: 46 ms
Case 1
Case 2
Case 3
Input
nums =
[3,0,1]
Output
2
Expected
2

Problem List

Description | Editorial | Solutions | Submissions

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.*

Example 1:

```
Input: nums = [1,1,0,1,1,1]
Output: 3
Explanation: The first two digits or the last three digits are consecutive 1s. The maximum number of consecutive 1s is 3.
```

Example 2:

```
Input: nums = [1,0,1,1,0,1]
Output: 2
```

Constraints:

- `1 <= nums.length <= 10^5`
- `nums[i]` is either `0` or `1`.

5.9K | 56

Run | Submit | 0 | Premium

Code | Testcase

Python3 | Auto

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

Saved | Ln 1, Col 1

Test Result

Accepted Runtime: 35 ms

Case 1 | Case 2

Input

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

Output

3