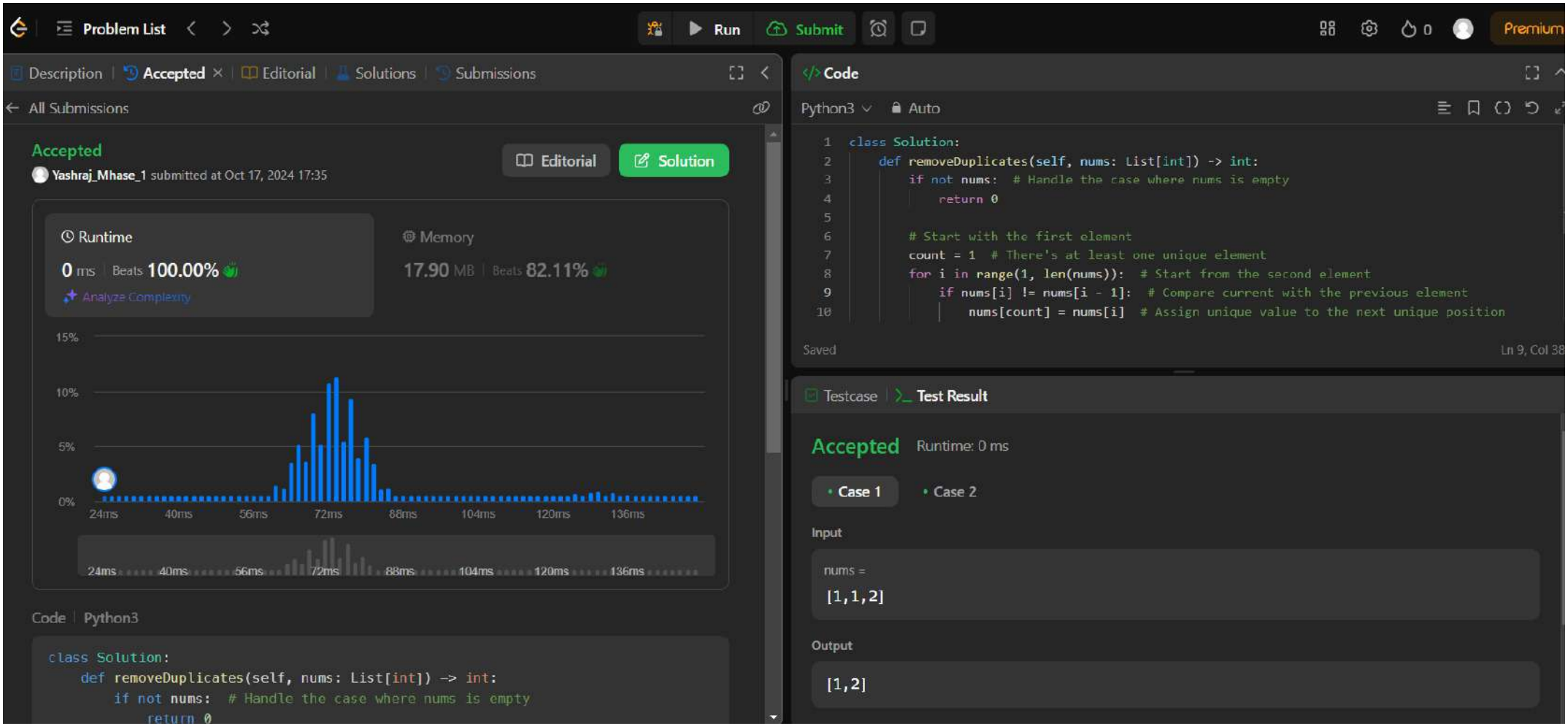
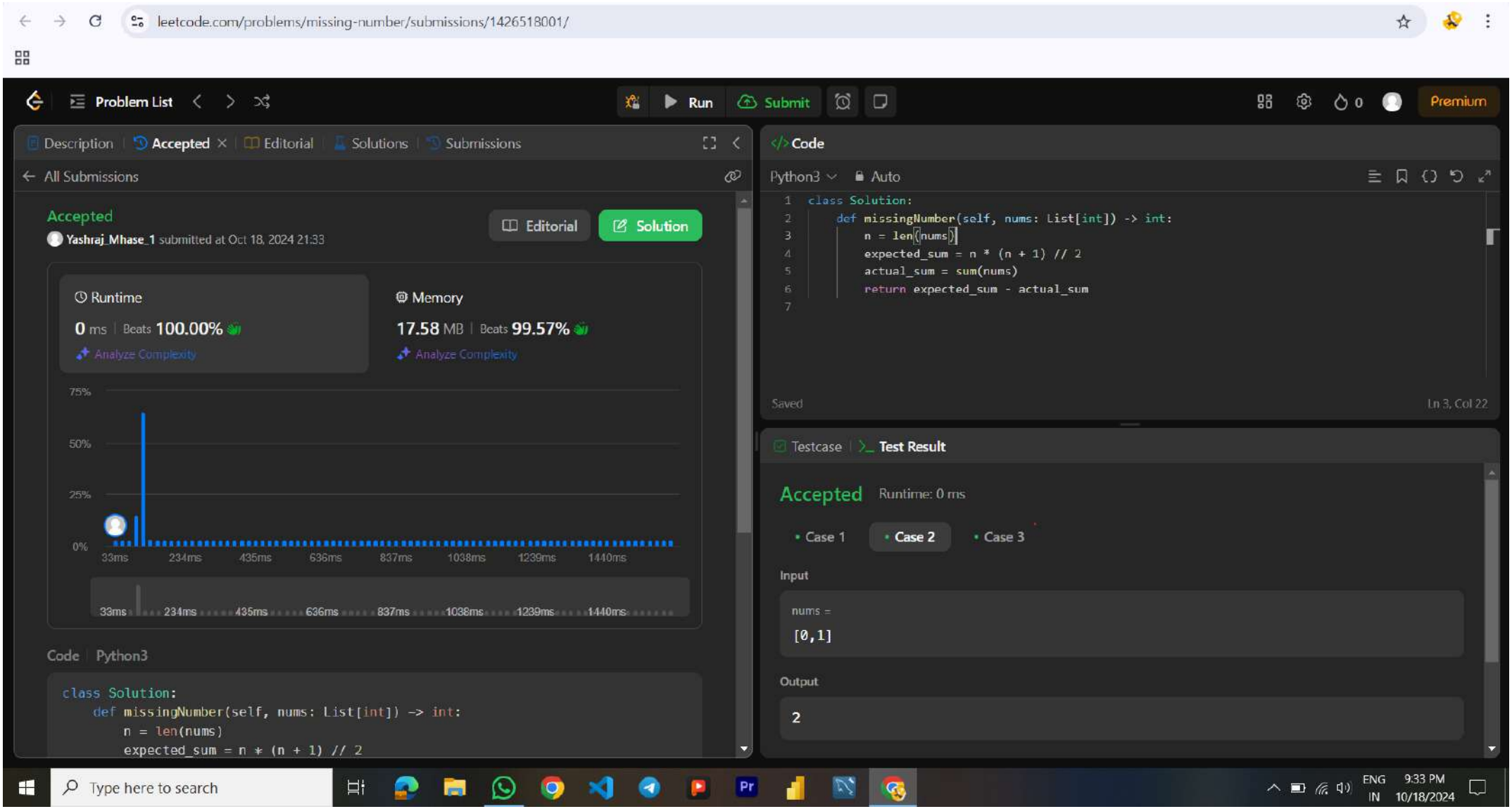

Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted
Yashraj_Mhase_1 submitted at Oct 17, 2024 17:35
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
17.90 MB | Beats 82.11%
15%
10%
5%
0%
24ms
40ms
56ms
72ms
88ms
104ms
120ms
136ms
Code | Python3
class Solution:
def removeDuplicates(self, nums: List[int]) -> int:
if not nums: # Handle the case where nums is empty
return 0
Code
Python3
Auto
class Solution:
def removeDuplicates(self, nums: List[int]) -> int:
if not nums: # Handle the case where nums is empty
return 0
# Start with the first element
count = 1 # There's at least one unique element
for i in range(1, len(nums)): # Start from the second element
if nums[i] != nums[i - 1]: # Compare current with the previous element
nums[count] = nums[i] # Assign unique value to the next unique position
Saved
Ln 9, Col 38
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
nums =
[1,1,2]
Output
[1,2]


leetcode.com/problems/missing-number/submissions/1426518001/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted
Yashraj_Mhase_1 submitted at Oct 18, 2024 21:33
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
17.58 MB | Beats 99.57%
Analyze Complexity
75%
50%
25%
0%
33ms
234ms
435ms
636ms
837ms
1038ms
1239ms
1440ms
Code | Python3
class Solution:
def missingNumber(self, nums: List[int]) -> int:
n = len(nums)
expected_sum = n * (n + 1) // 2
Code
Python3
Auto
class Solution:
def missingNumber(self, nums: List[int]) -> int:
n = len(nums)
expected_sum = n * (n + 1) // 2
actual_sum = sum(nums)
return expected_sum - actual_sum
Saved
Ln 3, Col 22
Testcase | Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums =
[0,1]
Output
2
Type here to search
ENG
IN
9:33 PM
10/18/2024