Description | Editorial | Solutions | Submissions

2. Add Two Numbers

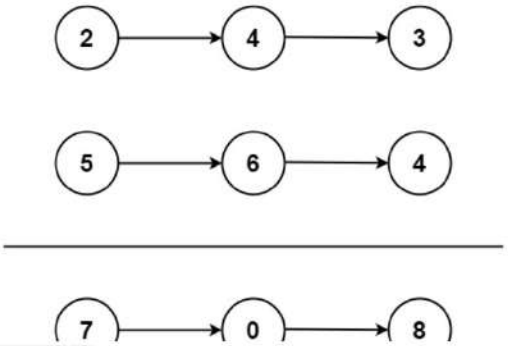
Solved

Medium | Topics | Companies

You are given two **non-empty** linked lists representing two non-negative integers. The digits are stored in **reverse order**, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Example 1:



30.8K | 576

Code

Python3 | Auto

```python
class Solution:
    def addTwoNumbers(
        self, l1: Optional[ListNode], l2: Optional[ListNode]
    ) -> Optional[ListNode]:
        dummyHead = ListNode(0)
        curr = dummyHead
        carry = 0
        while l1 != None or l2 != None or carry != 0:
            l1Val = l1.val if l1 else 0
            l2Val = l2.val if l2 else 0
            columnSum = l1Val + l2Val + carry
```

Saved | Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 36 ms

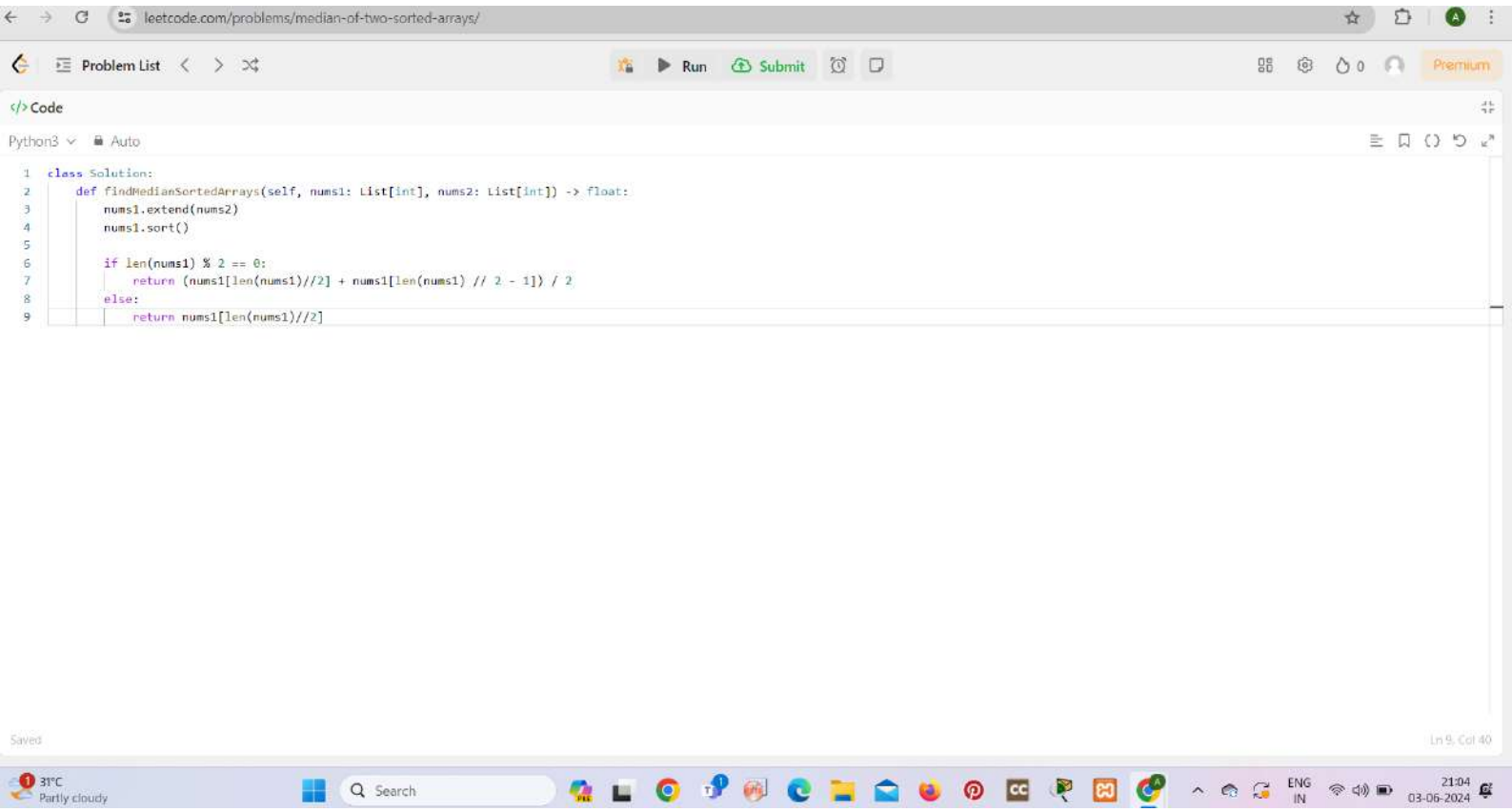
• Case 1 • Case 2 • Case 3

Input

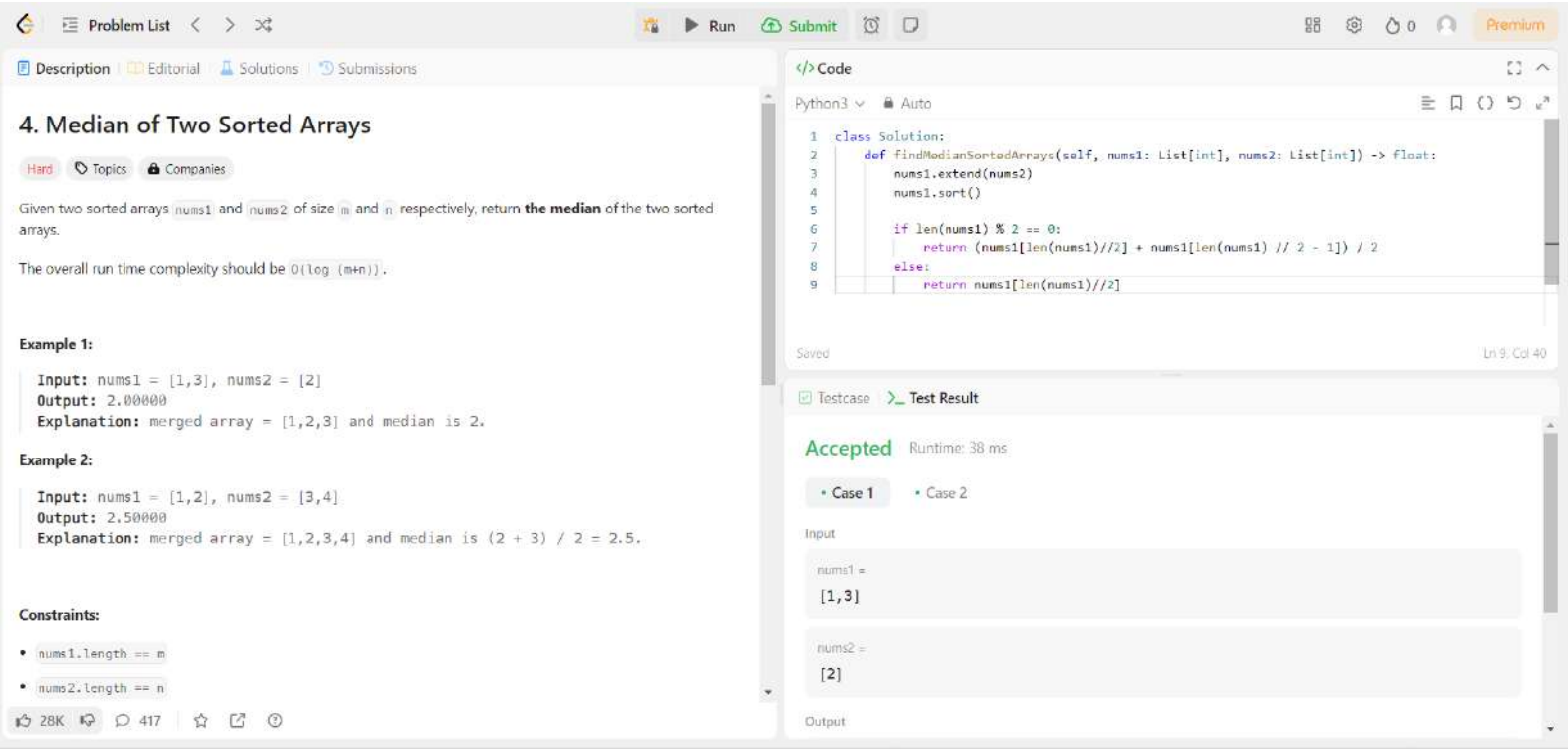
l1 =
[2,4,3]

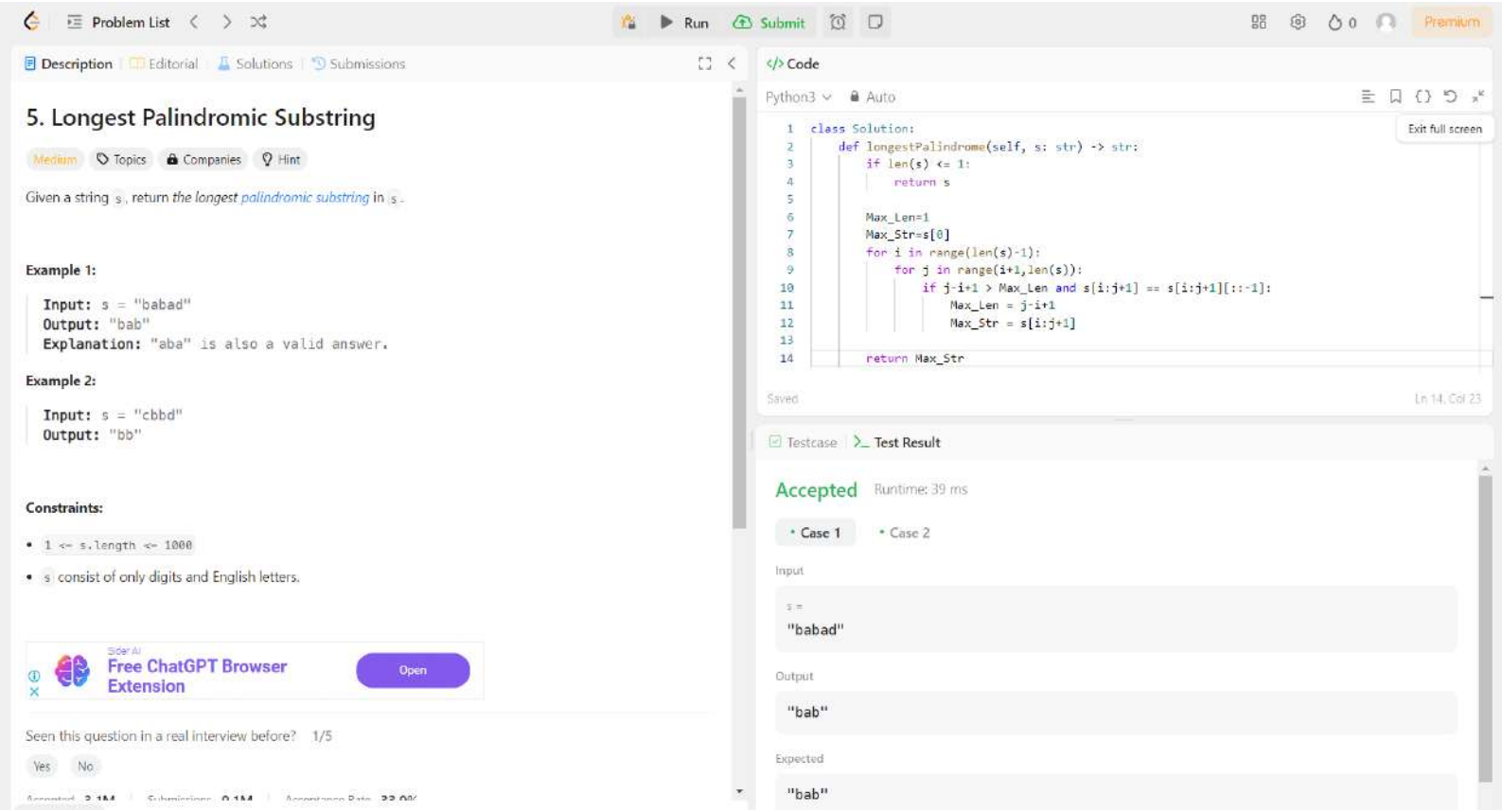
l2 =
[5,6,4]

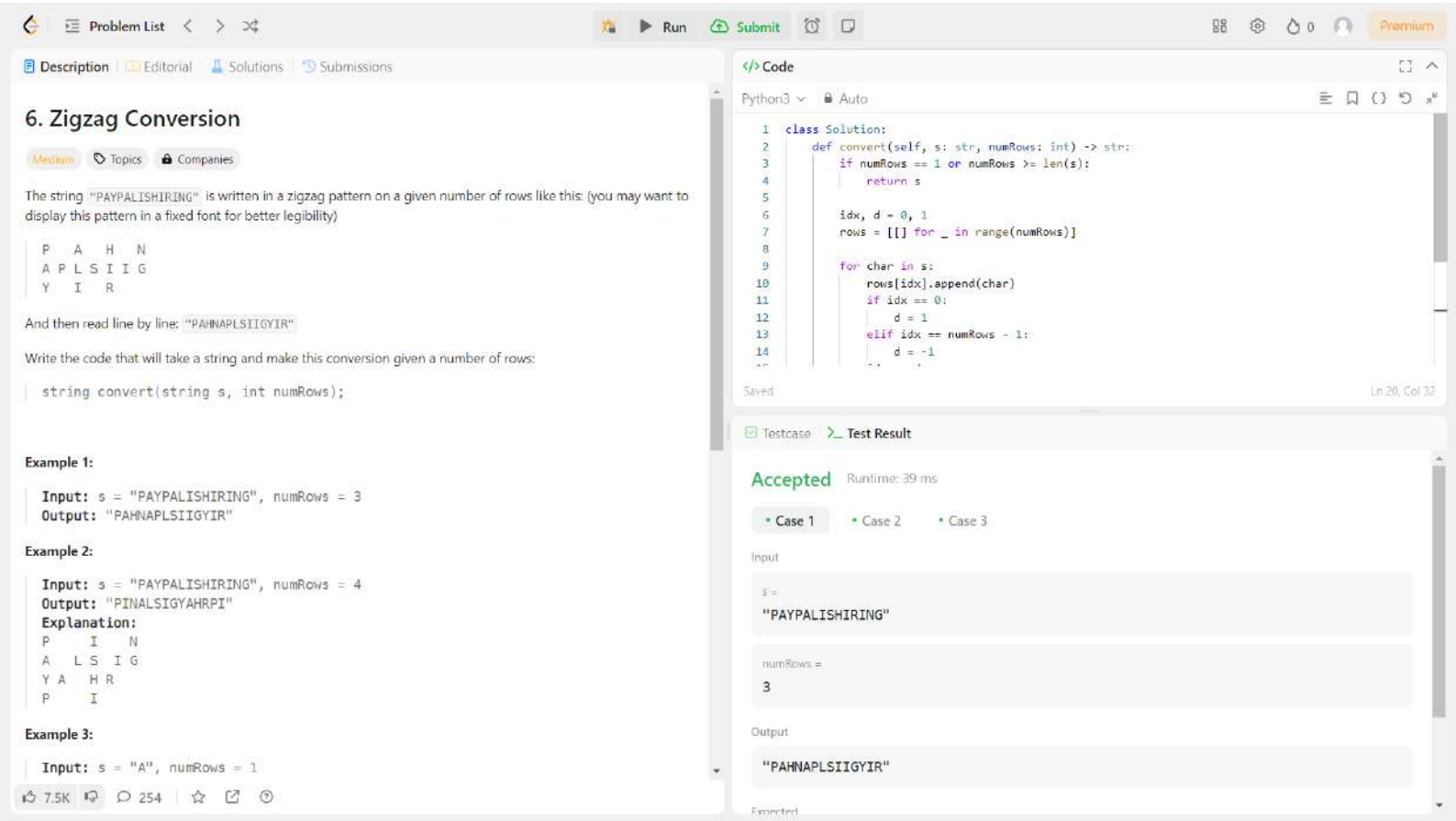
Output

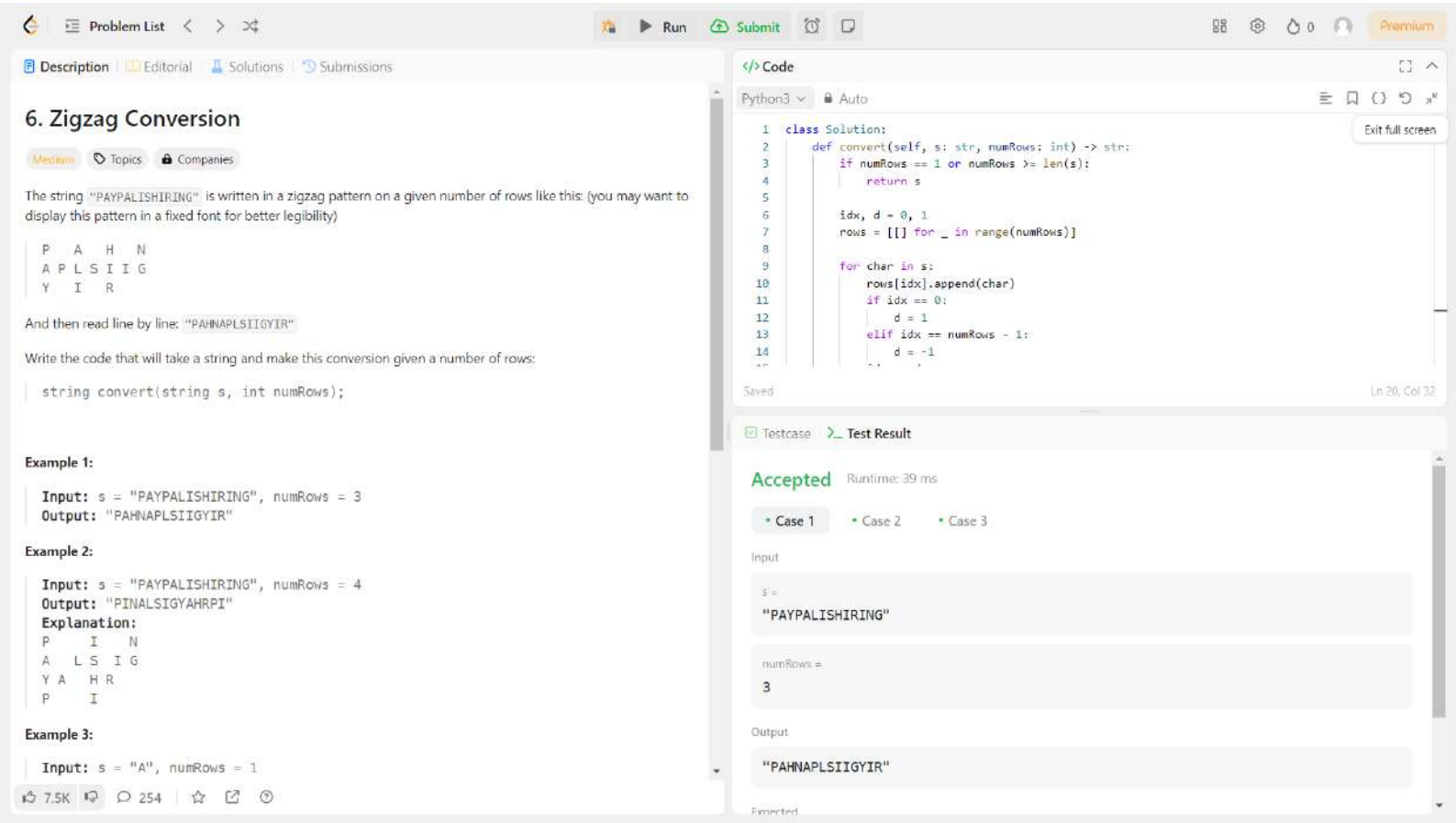

leetcode.com/problems/median-of-two-sorted-arrays/
Problem List
Run
Submit
0
Premium
Code
Python3
Auto
class Solution:
def findMedianSortedArrays(self, nums1: List[int], nums2: List[int]) -> float:
nums1.extend(nums2)
nums1.sort()
if len(nums1) % 2 == 0:
return (nums1[len(nums1)//2] + nums1[len(nums1) // 2 - 1]) / 2
else:
return nums1[len(nums1)//2]
Saved
Ln 9, Col 40

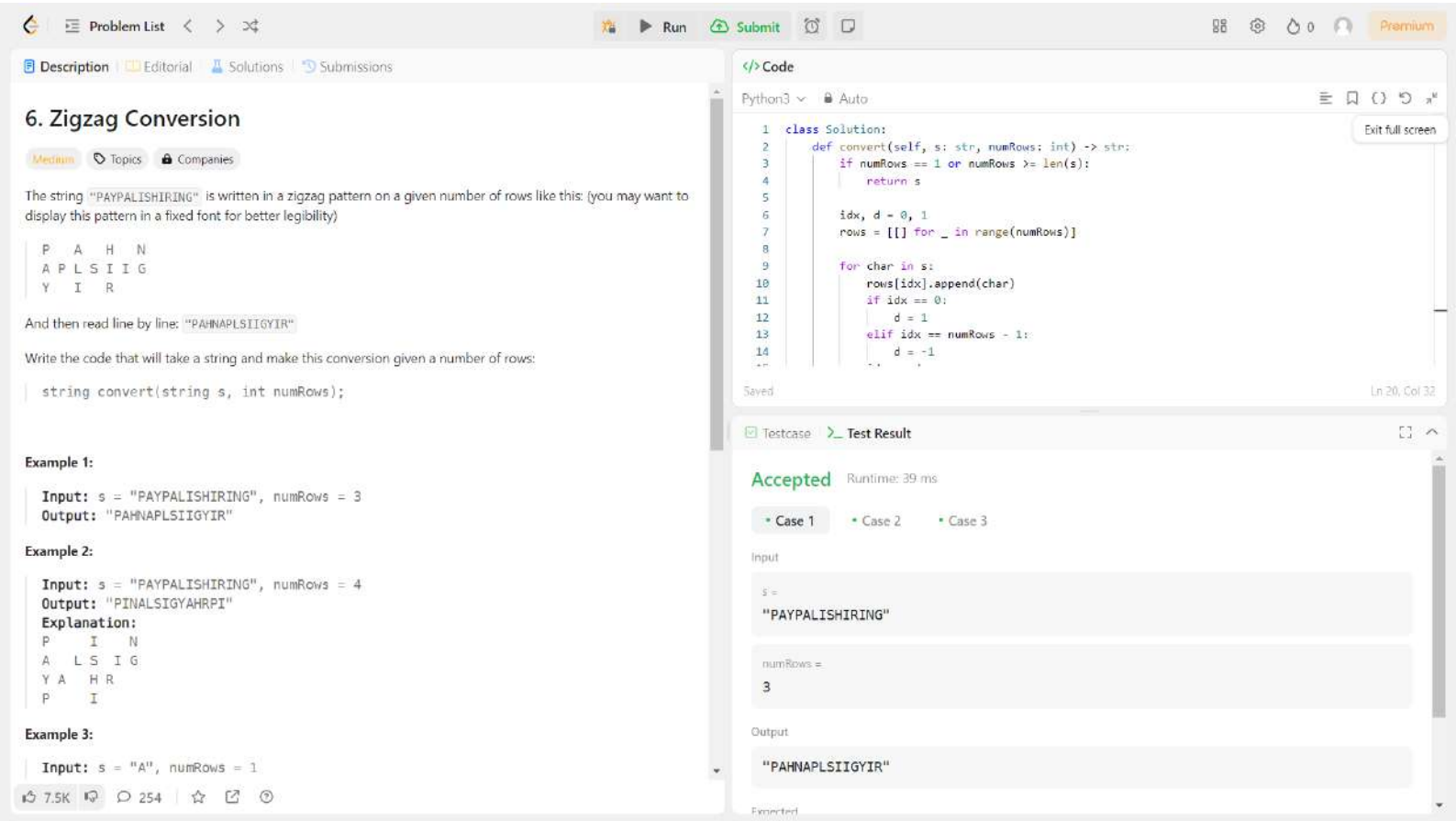
31°C
Partly cloudy
Search
ENG
IN
21:04
03-06-2024

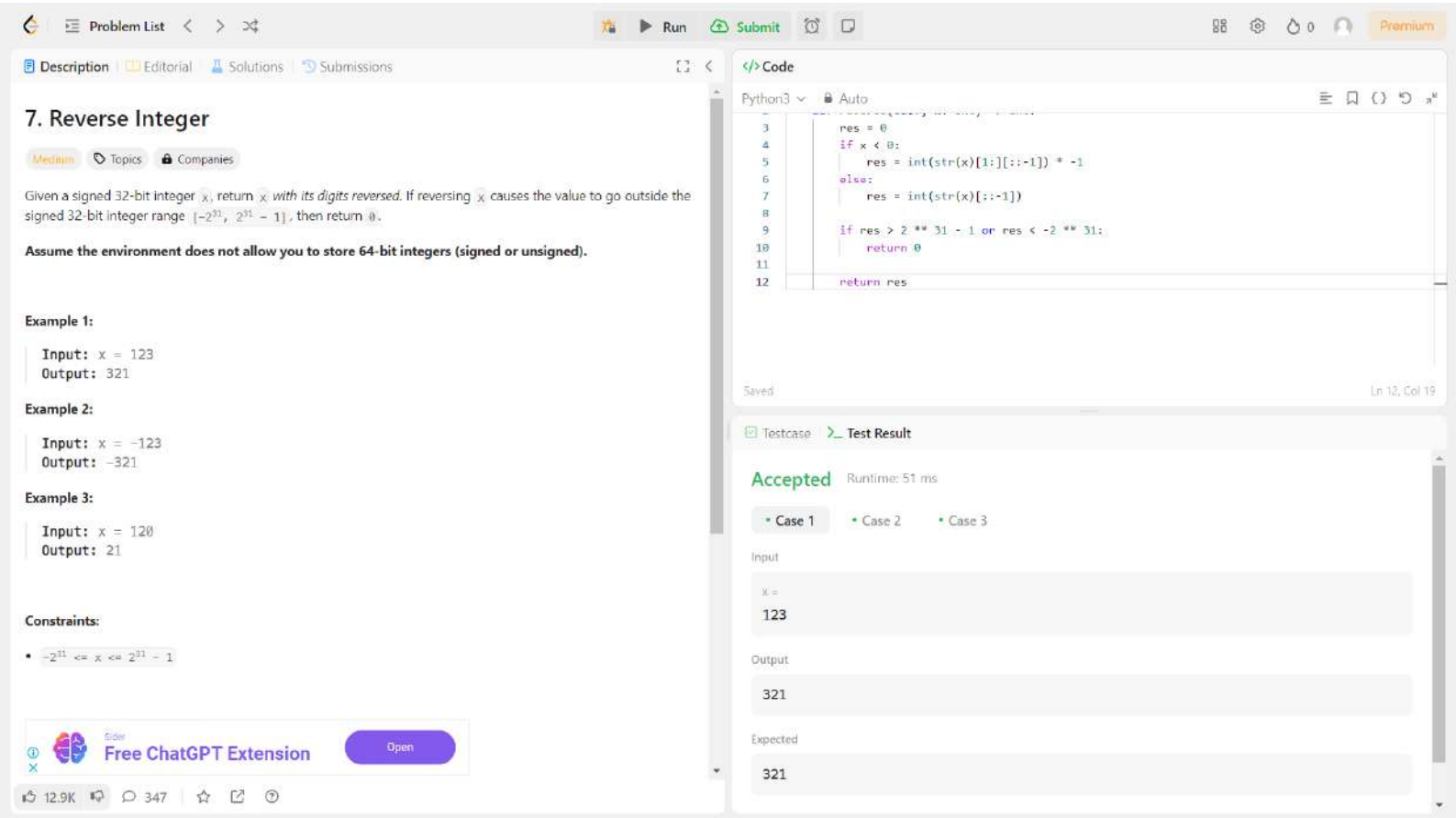

Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
4. Median of Two Sorted Arrays
Hard
Topics
Companies
Given two sorted arrays nums1 and nums2 of size m and n respectively, return the median of the two sorted arrays.
The overall run time complexity should be O(log (m+n)).
Example 1:
Input: nums1 = [1,3], nums2 = [2]
Output: 2.00000
Explanation: merged array = [1,2,3] and median is 2.
Example 2:
Input: nums1 = [1,2], nums2 = [3,4]
Output: 2.50000
Explanation: merged array = [1,2,3,4] and median is (2 + 3) / 2 = 2.5.
Constraints:
nums1.length == m
nums2.length == n
28K
417
Code
Python3
Auto
class Solution:
def findMedianSortedArrays(self, nums1: List[int], nums2: List[int]) -> float:
nums1.extend(nums2)
nums1.sort()
if len(nums1) % 2 == 0:
return (nums1[len(nums1)//2] + nums1[len(nums1) // 2 - 1]) / 2
else:
return nums1[len(nums1)//2]
Saved
Ln 9, Col 40
Testcase
Test Result
Accepted
Runtime: 38 ms
Case 1
Case 2
Input
nums1 =
[1,3]
nums2 =
[2]
Output

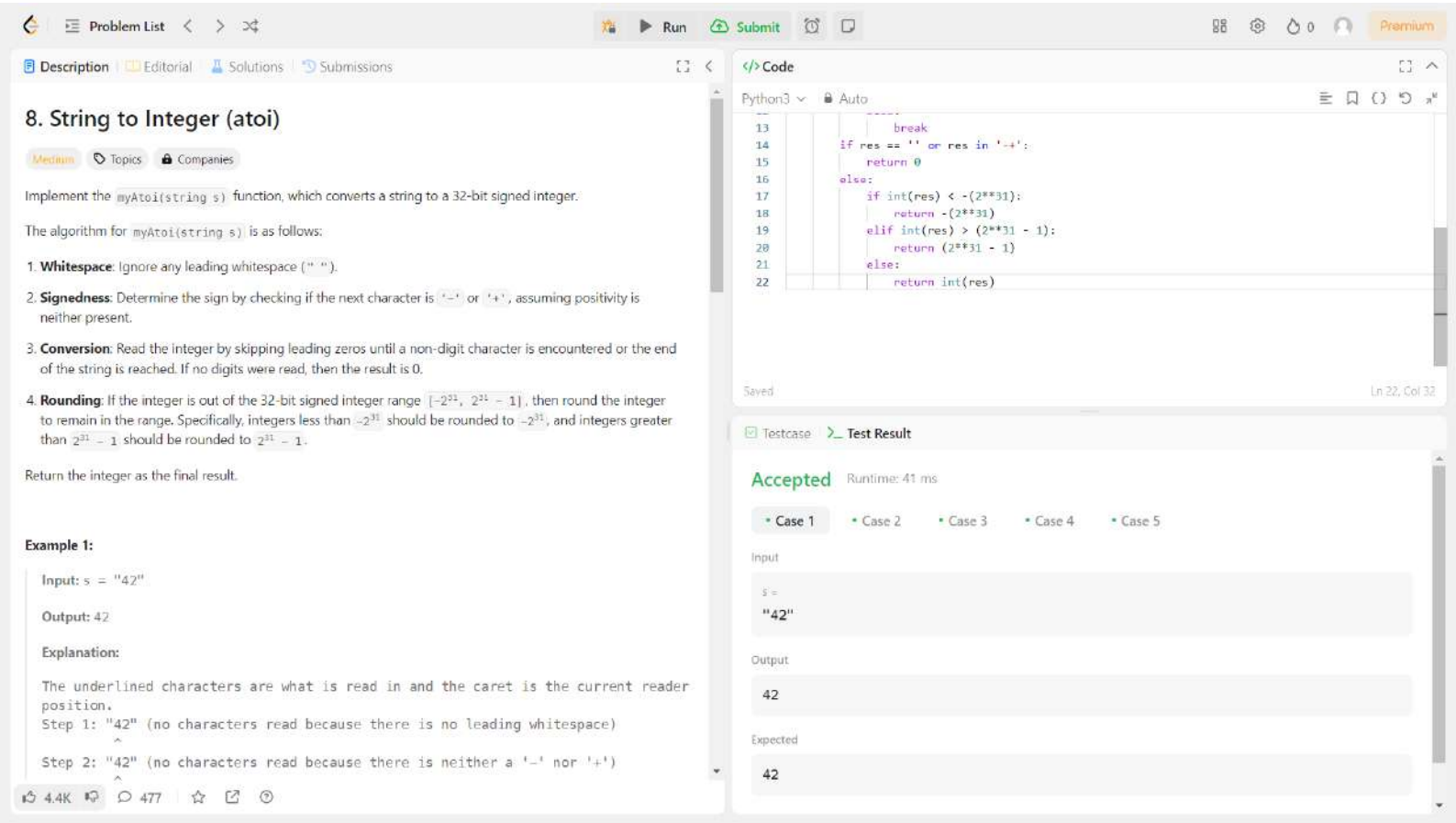

Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
5. Longest Palindromic Substring
Medium
Topics
Companies
Hint
Given a string s, return the longest palindromic substring in s.
Example 1:
Input: s = "babad"
Output: "bab"
Explanation: "aba" is also a valid answer.
Example 2:
Input: s = "cbbd"
Output: "bb"
Constraints:
1 <= s.length <= 1000
s consist of only digits and English letters.
Sider AI
Free ChatGPT Browser Extension
Open
Seen this question in a real interview before? 1/5
Yes
No
Code
Python3
Auto
Exit full screen
class Solution:
def longestPalindrome(self, s: str) -> str:
if len(s) <= 1:
return s
Max_Len=1
Max_Str=s[0]
for i in range(len(s)-1):
for j in range(i+1,len(s)):
if j-i+1 > Max_Len and s[i:j+1] == s[i:j+1][::-1]:
Max_Len = j-i+1
Max_Str = s[i:j+1]
return Max_Str
Saved
Ln 14, Col 23
Testcase
Test Result
Accepted
Runtime: 39 ms
Case 1
Case 2
Input
s =
"babad"
Output
"bab"
Expected
"bab"

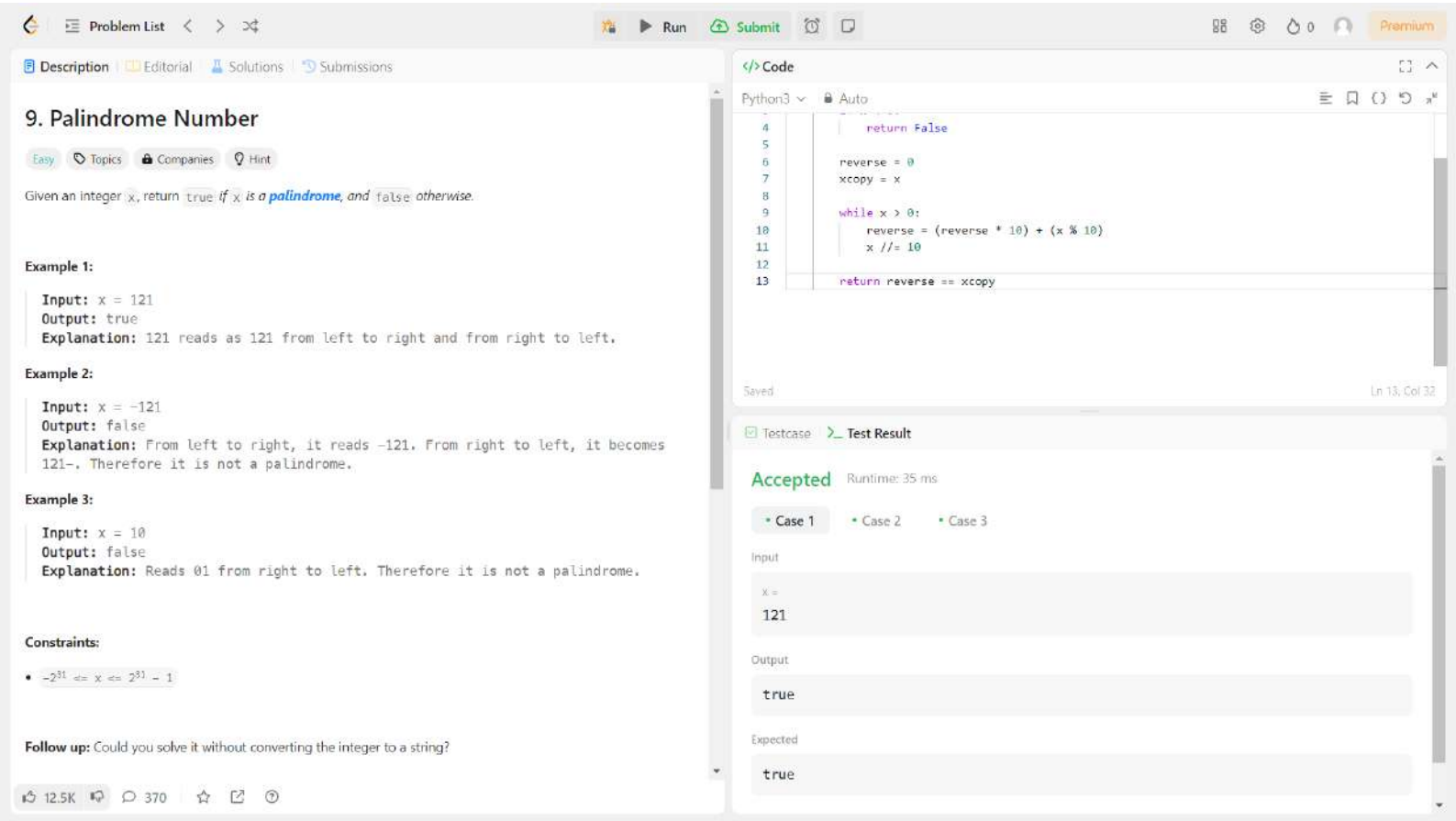

Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
6. Zigzag Conversion
Medium
Topics
Companies
The string "PAYPALISHIRING" is written in a zigzag pattern on a given number of rows like this: (you may want to display this pattern in a fixed font for better legibility)
P A H N
A P L S I I G
Y I R
And then read line by line: "PAHNAPLSIIGYIR"
Write the code that will take a string and make this conversion given a number of rows:
string convert(string s, int numRows);
Example 1:
Input: s = "PAYPALISHIRING", numRows = 3
Output: "PAHNAPLSIIGYIR"
Example 2:
Input: s = "PAYPALISHIRING", numRows = 4
Output: "PINALSIGYAHRPI"
Explanation:
P I N
A L S I G
Y A H R
P I
Example 3:
Input: s = "A", numRows = 1
7.5K
254
Code
Python3
Auto
class Solution:
def convert(self, s: str, numRows: int) -> str:
if numRows == 1 or numRows >= len(s):
return s
idx, d = 0, 1
rows = [[] for _ in range(numRows)]
for char in s:
rows[idx].append(char)
if idx == 0:
d = 1
elif idx == numRows - 1:
d = -1
Saved
Ln 20, Col 32
Testcase
Test Result
Accepted
Runtime: 39 ms
Case 1
Case 2
Case 3
Input
s =
"PAYPALISHIRING"
numRows =
3
Output
"PAHNAPLSIIGYIR"
Expected

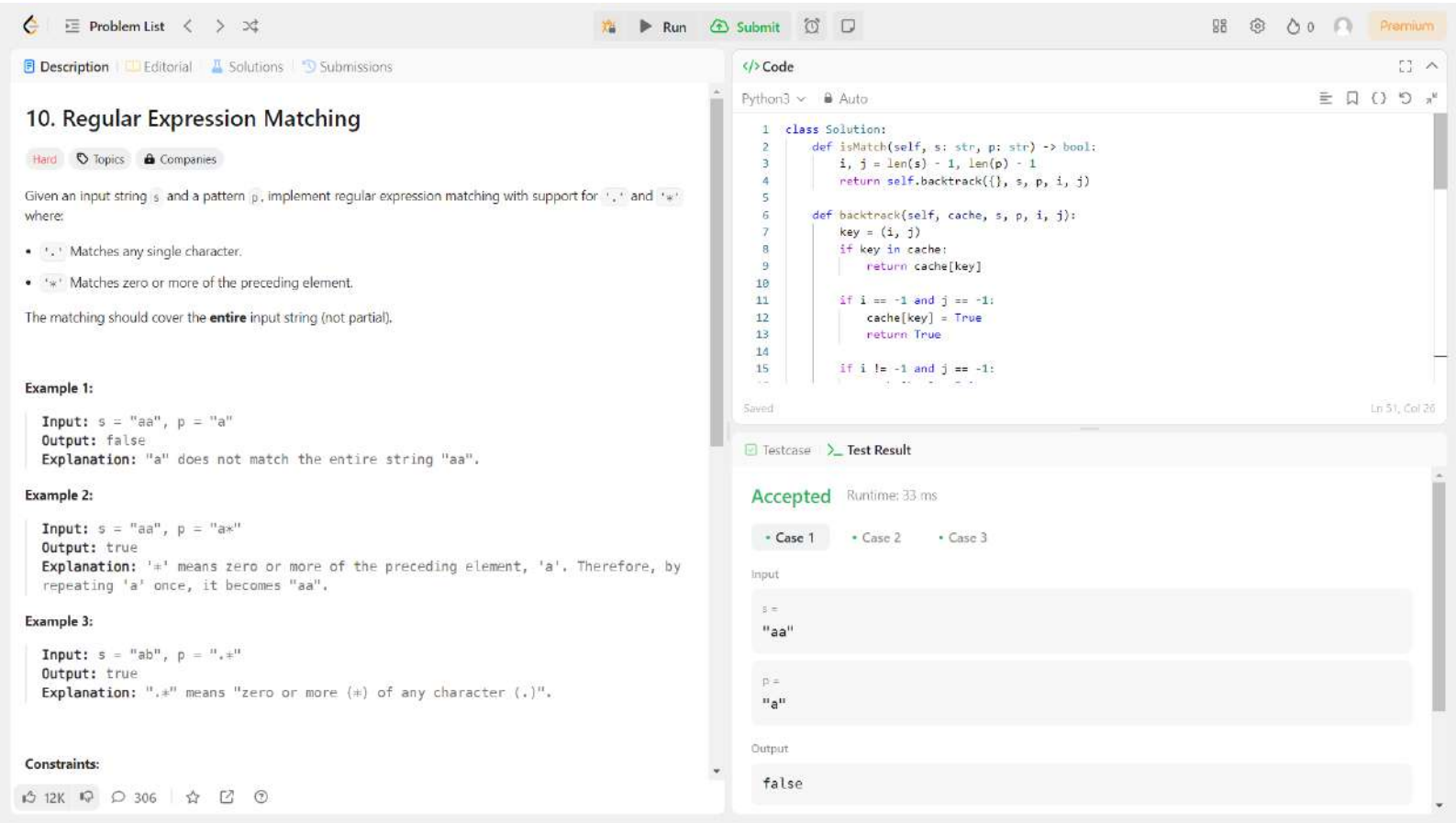

Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
6. Zigzag Conversion
Medium
Topics
Companies
The string "PAYPALISHIRING" is written in a zigzag pattern on a given number of rows like this: (you may want to display this pattern in a fixed font for better legibility)
P A H N
A P L S I I G
Y I R
And then read line by line: "PAHNAPLSIIGYIR"
Write the code that will take a string and make this conversion given a number of rows:
string convert(string s, int numRows);
Example 1:
Input: s = "PAYPALISHIRING", numRows = 3
Output: "PAHNAPLSIIGYIR"
Example 2:
Input: s = "PAYPALISHIRING", numRows = 4
Output: "PINALSIGYAHRPI"
Explanation:
P I N
A L S I G
Y A H R
P I
Example 3:
Input: s = "A", numRows = 1
7.5K
254
Code
Python3
Auto
Exit full screen
class Solution:
def convert(self, s: str, numRows: int) -> str:
if numRows == 1 or numRows >= len(s):
return s
idx, d = 0, 1
rows = [[] for _ in range(numRows)]
for char in s:
rows[idx].append(char)
if idx == 0:
d = 1
elif idx == numRows - 1:
d = -1
Saved
Ln 20, Col 32
Testcase
Test Result
Accepted
Runtime: 39 ms
Case 1
Case 2
Case 3
Input
s =
"PAYPALISHIRING"
numRows =
3
Output
"PAHNAPLSIIGYIR"


Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
6. Zigzag Conversion
Medium
Topics
Companies
The string "PAYPALISHIRING" is written in a zigzag pattern on a given number of rows like this: (you may want to display this pattern in a fixed font for better legibility)
P A H N
A P L S I I G
Y I R
And then read line by line: "PAHNAPLSIIGYIR"
Write the code that will take a string and make this conversion given a number of rows:
string convert(string s, int numRows);
Example 1:
Input: s = "PAYPALISHIRING", numRows = 3
Output: "PAHNAPLSIIGYIR"
Example 2:
Input: s = "PAYPALISHIRING", numRows = 4
Output: "PINALSIGYAHRPI"
Explanation:
P I N
A L S I G
Y A H R
P I
Example 3:
Input: s = "A", numRows = 1
7.5K
254
Code
Python3
Auto
Exit full screen
class Solution:
def convert(self, s: str, numRows: int) -> str:
if numRows == 1 or numRows >= len(s):
return s
idx, d = 0, 1
rows = [[] for _ in range(numRows)]
for char in s:
rows[idx].append(char)
if idx == 0:
d = 1
elif idx == numRows - 1:
d = -1
Saved
Ln 20, Col 32
Testcase
Test Result
Accepted
Runtime: 39 ms
Case 1
Case 2
Case 3
Input
s =
"PAYPALISHIRING"
numRows =
3
Output
"PAHNAPLSIIGYIR"


Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
Code
Python3
Auto
7. Reverse Integer
Medium
Topics
Companies
Given a signed 32-bit integer x, return x with its digits reversed. If reversing x causes the value to go outside the signed 32-bit integer range [-2^31, 2^31 - 1], then return 0.
Assume the environment does not allow you to store 64-bit integers (signed or unsigned).
Example 1:
Input: x = 123
Output: 321
Example 2:
Input: x = -123
Output: -321
Example 3:
Input: x = 120
Output: 21
Constraints:
-2^31 <= x <= 2^31 - 1
res = 0
if x < 0:
res = int(str(x)[1:][::-1]) * -1
else:
res = int(str(x)[::-1])
if res > 2 ** 31 - 1 or res < -2 ** 31:
return 0
return res
Saved
Ln 12, Col 19
Testcase
Test Result
Accepted
Runtime: 51 ms
Case 1
Case 2
Case 3
Input
x =
123
Output
321
Expected
321
Free ChatGPT Extension
Open
12.9K
347


Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
8. String to Integer (atoi)
Medium
Topics
Companies
Implement the myAtoi(string s) function, which converts a string to a 32-bit signed integer.
The algorithm for myAtoi(string s) is as follows:
1. Whitespace: Ignore any leading whitespace (" ").
2. Signedness: Determine the sign by checking if the next character is '-' or '+', assuming positivity is neither present.
3. Conversion: Read the integer by skipping leading zeros until a non-digit character is encountered or the end of the string is reached. If no digits were read, then the result is 0.
4. Rounding: If the integer is out of the 32-bit signed integer range [-2^31, 2^31 - 1], then round the integer to remain in the range. Specifically, integers less than -2^31 should be rounded to -2^31, and integers greater than 2^31 - 1 should be rounded to 2^31 - 1.
Return the integer as the final result.
Example 1:
Input: s = "42"
Output: 42
Explanation:
The underlined characters are what is read in and the caret is the current reader position.
Step 1: "42" (no characters read because there is no leading whitespace)
Step 2: "42" (no characters read because there is neither a '-' nor '+')
4.4K
477
Code
Python3
Auto
13 break
14 if res == '' or res in '-+':
15 return 0
16 else:
17 if int(res) < -(2**31):
18 return -(2**31)
19 elif int(res) > (2**31 - 1):
20 return (2**31 - 1)
21 else:
22 return int(res)
Saved
Ln 22, Col 32
Testcase
Test Result
Accepted
Runtime: 41 ms
Case 1
Case 2
Case 3
Case 4
Case 5
Input
s =
"42"
Output
42
Expected
42


Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
9. Palindrome Number
Easy
Topics
Companies
Hint
Given an integer x, return true if x is a palindrome, and false otherwise.
Example 1:
Input: x = 121
Output: true
Explanation: 121 reads as 121 from left to right and from right to left.
Example 2:
Input: x = -121
Output: false
Explanation: From left to right, it reads -121. From right to left, it becomes 121-. Therefore it is not a palindrome.
Example 3:
Input: x = 10
Output: false
Explanation: Reads 01 from right to left. Therefore it is not a palindrome.
Constraints:
-2^31 <= x <= 2^31 - 1
Follow up: Could you solve it without converting the integer to a string?
12.5K
370
Code
Python3
Auto
return False
reverse = 0
xcopy = x
while x > 0:
reverse = (reverse * 10) + (x % 10)
x //= 10
return reverse == xcopy
Saved
Ln 13, Col 32
Testcase
Test Result
Accepted
Runtime: 35 ms
Case 1
Case 2
Case 3
Input
x =
121
Output
true
Expected
true


Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
10. Regular Expression Matching
Hard
Topics
Companies
Given an input string s and a pattern p, implement regular expression matching with support for '.' and '*' where:
'.' Matches any single character.
'*' Matches zero or more of the preceding element.
The matching should cover the entire input string (not partial).
Example 1:
Input: s = "aa", p = "a"
Output: false
Explanation: "a" does not match the entire string "aa".
Example 2:
Input: s = "aa", p = "a*"
Output: true
Explanation: '*' means zero or more of the preceding element, 'a'. Therefore, by repeating 'a' once, it becomes "aa".
Example 3:
Input: s = "ab", p = ".*"
Output: true
Explanation: ".*" means "zero or more (*) of any character (.)".
Constraints:
12K
306
Code
Python3
Auto
class Solution:
def isMatch(self, s: str, p: str) -> bool:
i, j = len(s) - 1, len(p) - 1
return self.backtrack({}, s, p, i, j)
def backtrack(self, cache, s, p, i, j):
key = (i, j)
if key in cache:
return cache[key]
if i == -1 and j == -1:
cache[key] = True
return True
if i != -1 and j == -1:
Saved
Ln 51, Col 26
Testcase
Test Result
Accepted
Runtime: 33 ms
Case 1
Case 2
Case 3
Input
s =
"aa"
p =
"a"
Output
false

1. Two Sum

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:

```
Input: nums = [3,3], target = 6
Output: [0,1]
```

56.5K | 848

Python3 | Auto

```python
class Solution:
    def twoSum(self, nums: List[int], target: int) -> List[int]:
        a=[]
        for i in range(len(nums)):
            for j in range(i+1,len(nums)):
                if (nums[i]+nums[j]==target):
                    a.append(i)
                    a.append(j)
                    break
        return a
```

Saved | Ln 10, Col 17

Testcase | Test Result

Accepted Runtime: 50 ms

• Case 1 • Case 2 • Case 3

Input

nums =
[2,7,11,15]

target =
9

Output