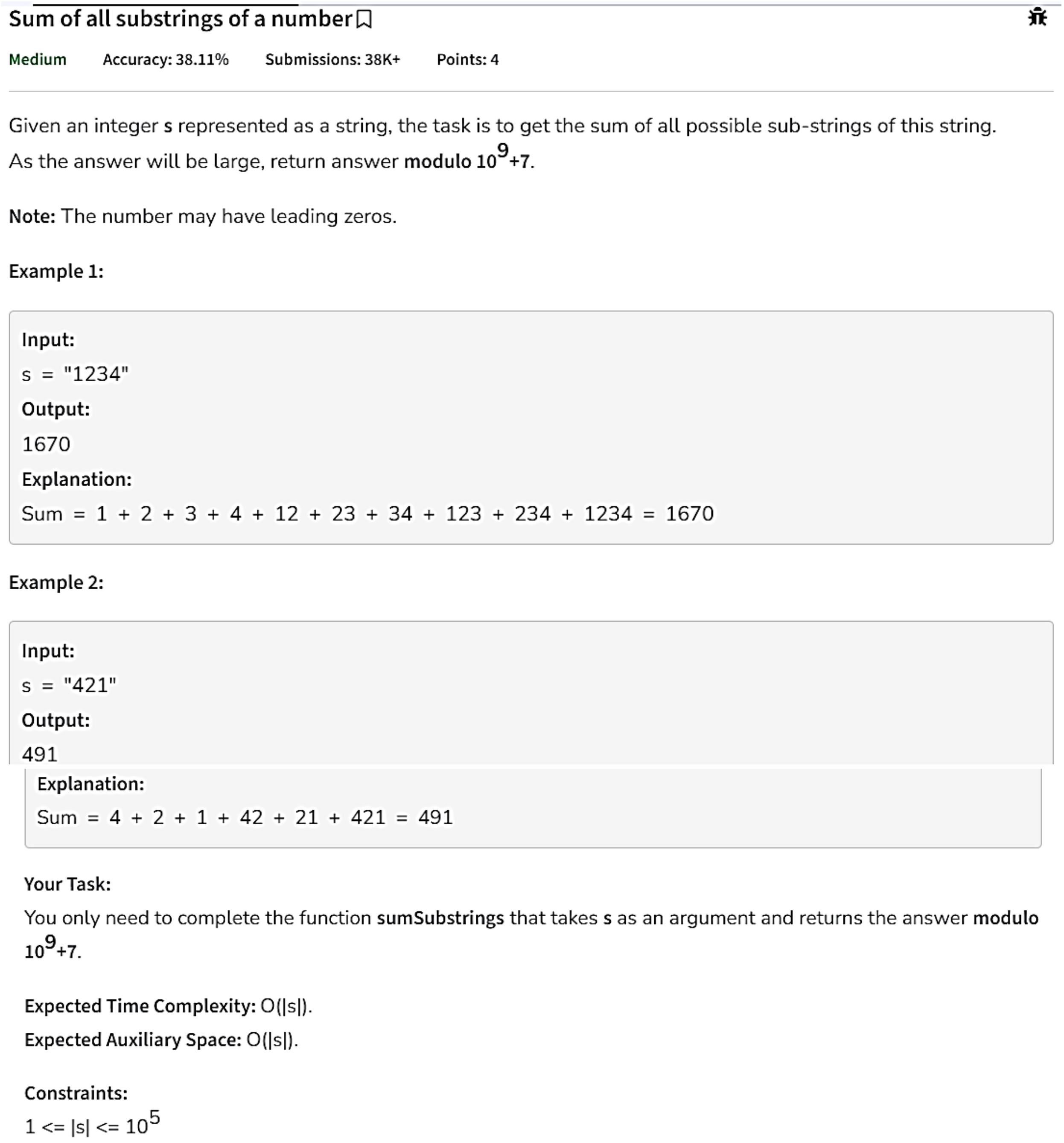
PROBLEM



Sum of all substrings of a number

Medium **Accuracy: 38.11%** **Submissions: 38K+** **Points: 4**

Given an integer **s** represented as a string, the task is to get the sum of all possible sub-strings of this string.
As the answer will be large, return answer **modulo 10^9+7**.

Note: The number may have leading zeros.

Example 1:

Input:
s = "1234"
Output:
1670
Explanation:
Sum = 1 + 2 + 3 + 4 + 12 + 23 + 34 + 123 + 234 + 1234 = 1670

Example 2:

Input:
s = "421"
Output:
491
Explanation:
Sum = 4 + 2 + 1 + 42 + 21 + 421 = 491

Your Task:
You only need to complete the function **sumSubstrings** that takes **s** as an argument and returns the answer **modulo 10^9+7**.

Expected Time Complexity: O(|s|).
Expected Auxiliary Space: O(|s|).

Constraints:
$1 <= |s| <= 10^5$

CODE

```
#User function Template for python3
class Solution:
    #Function to find sum of all possible substrings of the given string.
    def sumSubstrings(self,s):
        n=len(s)
        prev=int(s[0])
        res=prev
        for i in range(1,n):
            temp=(int(s[i]))*(i+1)+prev*10
            res=(res+temp)%1000000007
            prev=temp%1000000007
        return res
```