



[Click on the logo to find **Problem Statement**]

Intuition

The Roman numeral system uses specific combinations of letters to represent numbers. To convert a Roman numeral to an integer, we iterate through the given string from left to right. We keep track of the total integer value as we encounter each Roman numeral character.

The special cases to consider are when a smaller numeral appears before a larger numeral, which results in subtraction. For example, in "IV," the "I" (1) is subtracted from the "V" (5), making it 4. To handle these cases, we check if the current numeral is smaller than the next numeral. If it is, we subtract the current numeral's value from the total result.

For other cases where a smaller numeral appears after a larger numeral, such as "VI" (6), we simply add the values because the numerals are in descending order.

By iterating through the string and applying these rules, we accurately convert the Roman numeral to its corresponding integer value. The approach effectively handles all possible combinations of Roman numerals and produces the correct output.

Approach

Create a Roman to Integer Mapping: Create a dictionary `roman` that maps each Roman numeral character to its corresponding integer value. For example: `{"I": 1, "V": 5, "X": 10, "L": 50, "C": 100, "D": 500, "M": 1000}`.

Initialize Result Variable: Initialize a variable `result` to store the total integer value. Set it to 0 initially.

Iterate Through the Input String: Iterate through the input string `s` from left to right using a loop.

Check for Special Cases: Within the loop, check if the current Roman numeral is smaller than the next numeral. If it is, subtract the current numeral's value from the result. Example: When `s[i] = "I"` and `s[i+1] = "V"`, subtract 1 from the result because IV represents 4 in Roman numerals.

Add to Result: If the current Roman numeral is not smaller than the next numeral, add its value to the result. Example: When `s[i] = "V"` and `s[i+1] = "I"`, add 5 to the result because VI represents 6 in Roman numerals.

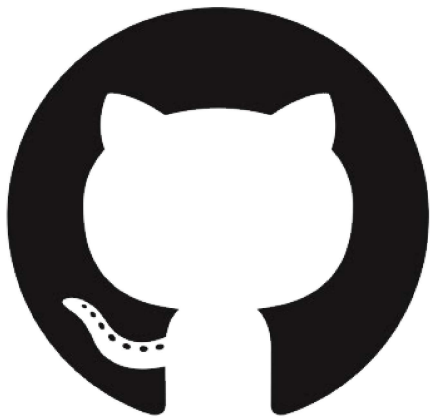
Return the Result: After iterating through the entire string, the result variable will hold the total integer value of the Roman numeral input. Return this result.

Complexity

- Time complexity: $O(n)$
- Space complexity: $O(1)$

Code

```
class Solution:
    def romanToInt(self, s: str) -> int:
        roman = {"I":1, "V":5, "X":10, "L":50, "C":100, "D":500, "M":1000}
        result = 0
        for i in range(len(s)):
            if i+1 < len(s) and roman[s[i]] < roman[s[i+1]]:
                result -= roman[s[i]]
            else:
                result += roman[s[i]]
        return result
```



GitHub