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

1. Create a Roman to Integer Mapping:
    - Create a dictionary roman that associates each Roman numeral character with its corresponding integer value. For instance, {"I": 1, "V": 5, "X": 10, "L": 50, "C": 100, "D": 500, "M": 1000}.
2. Initialize a Result Variable:
    - Initialize a variable result to store the total integer value. Set it to 0 initially.

3. Iterate Through the Roman Numeral String:

    - Use a loop to iterate through the characters of the input Roman numeral string s, from left to right.

4. Check for Special Cases:

    - Inside the loop, check if the current Roman numeral character is smaller than the next character (i.e., roman[s[i]] < roman[s[i+1]]).If this condition is met, it indicates a subtractive case (e.g., "IV" for 4 or "IX" for 9). In such cases:
        - Subtract the value of the current character from the result.
        - Move to the next character by incrementing i (i.e., i += 2) since two characters represent a subtractive value.
    - If the condition is not met, it means the current character is not in a subtractive case, so:
        - Add the value of the current character to the result.
        - Move to the next character by incrementing i by 1 (i.e., i += 1).

5. Return the Result:

    - After processing the entire Roman numeral string, the result variable holds the total integer value. Return this result as the final answer.

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