



#13



# Roman to Integer

Akhtamov Azimjon **2024299010**

2025-01-16

# Problem Definition (1)

- Source: **Leetcode**



- Title: **Merge Intervals**
- Difficulty: **easy**
- Type: **Hashmaps**



# Problem Definition (1)

## 13. Roman to Integer

Easy

Topics

Companies

Hint

Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

### Constraints:

- $1 \leq s.length \leq 15$
- $s$  contains only the characters ('I', 'V', 'X', 'L', 'C', 'D', 'M').
- It is **guaranteed** that  $s$  is a valid roman numeral in the range  $[1, 3999]$ .

### Example 1:

Input:  $s = \text{"III"}$

Output: 3

Explanation: III = 3.

### Example 2:

Input:  $s = \text{"LVIII"}$

Output: 58

Explanation: L = 50, V = 5, III = 3.

### Example 3:

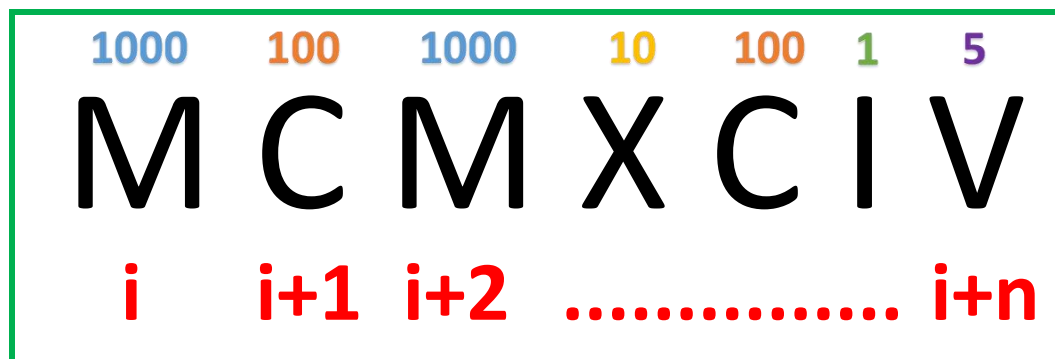
Input:  $s = \text{"MCMXCIV"}$

Output: 1994

Explanation: M = 1000, CM = 900, XC = 90 and IV = 4.



# Solution (1)



$n$  = Length of string (input)

Case 3



Input

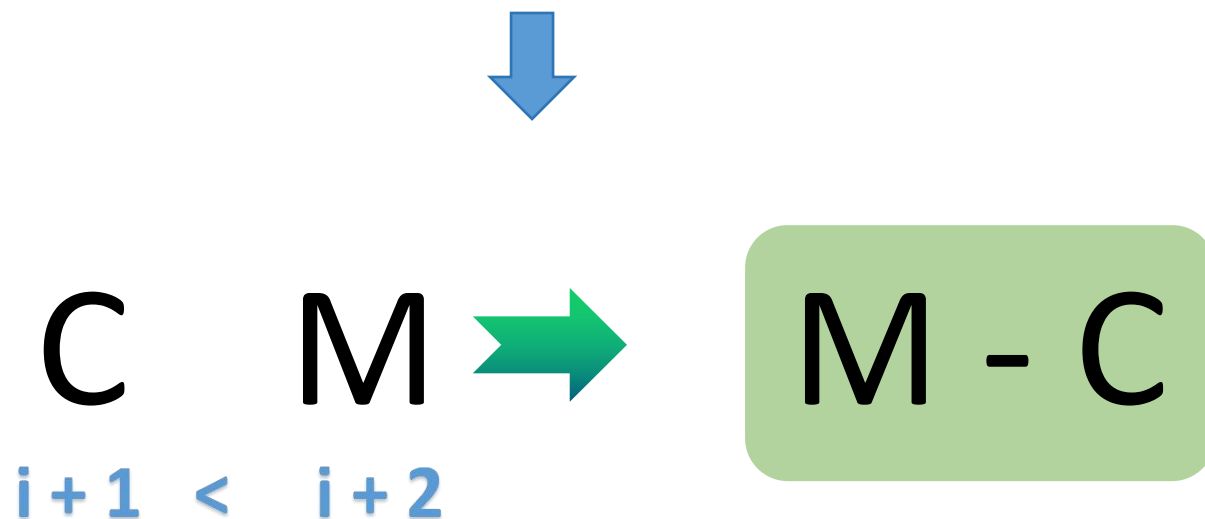
$s =$   
"MCMXCIV"

Output

1994

Expected

1994





# Solution (2)

Arrays & Strings > 3- Roman to Integer #13.py > ...

```
1  class Solution:
2      def romanToInt(self, s: str) -> int:
3
4          d = {"I":1, "V":5, "X":10, "L":50, "C":100, "D":500, "M":1000} #Hashmap for Roman Numbers
5
6          sum = 0 # to store
7          n = len(s) #length of input String
8          i = 0 #to track hashmaps' indexes
9
10         while i < n:
11             if i < n-1 and d[s[i]] < d[s[i+1]]: # Like example C < M ----> M - C
12                 sum+= d[s[i+1]] - d[s[i]]
13                 i+=2 # 2 strings as a one number
14             else:
15                 sum+=d[s[i]]
16                 i+=1
17         return sum # final
```



# Solution (3)

Accepted 3999 / 3999 testcases passed

Azimjon submitted at Jan 16, 2025 14:15

Editorial

Solution

Runtime

7 ms | Beats 43.14%

[Analyze Complexity](#)

Memory

17.68 MB | Beats 45.40%



Accepted Runtime: 0 ms

Case 1

Case 2

Case 3

Input

S =  
"MCMXCIV"

Output

1994

Expected

1994



# What I have learned

## ❖ Hashmaps:

- ✓ I learnt to check **HashMap's elements by its indexes**
- ✓ Understood logic of **Roman Numbers** for **CM, IV, IX, XL** cases

✓ I used **n operators(times)**, so -----> **time:  $O(n)$**

✓ I didn't use space, so -----> **space:  $O(1)$**



# Questions and Answers



Greetings