

整数转罗马数字

罗马数字包含以下七种字符：I，V，X，L，C，D和M。

字符	数值
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

例如，罗马数字 2 写做 II，即为两个并列的 1。12 写做 XII，即为 X + II。27 写做 XXVII，即为 XX + V + II。

通常情况下，罗马数字中小的数字在大的数字的右边。但也存在特例，例如 4 不写做 IIII，而是 IV。数字 1 在数字 5 的左边，所表示的数等于大数 5 减小数 1 得到的数值 4。同样地，数字 9 表示为 IX。这个特殊的规则只适用于以下六种情况：

- I 可以放在 V (5) 和 X (10) 的左边，来表示 4 和 9。
- X 可以放在 L (50) 和 C (100) 的左边，来表示 40 和 90。
- C 可以放在 D (500) 和 M (1000) 的左边，来表示 400 和 900。

给定一个整数，将其转为罗马数字。输入确保在 1 到 3999 的范围内。

示例 1:

输入: 3

输出: "III"

示例 2:

输入: 4

输出: "IV"

示例 3:

输入: 9

输出: "IX"

示例 4:

输入: 58

输出: "LVIII"

解释: L = 50, V = 5, III = 3.

示例 5:

输入: 1994

输出: "MCMXCIV"

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

暴力匹配 硬解码

```
class Solution {
public:
    string intToRoman(int num) {
        vector<string> thousand = { "", "M", "MM", "MMM" };
        vector<string> hundred = { "", "C", "CC", "CCC", "CD", "D", "DC", "DC
C", "DCCC", "CM" };
        vector<string> decad = { "", "X", "XX", "XXX", "XL", "L", "LX", "LXX",
"LXXX", "XC" };
        vector<string> unit = { "", "I", "II", "III", "IV", "V", "VI", "VII",
"VIII", "IX" };

        string out = thousand[num / 1000];
        out += hundred[(num % 1000) / 100];
        out += decad[(num % 1000 % 100) / 10];
        out += unit[num % 1000 % 100 % 10];
        return out;
    }
};
```

贪心算法

```
class Solution {
public:
    string intToRoman(int num) {
        string res;
        vector<int> values={1000,900,500,400,100,90,50,40,10,9,5,4,1};
        vector<string> symbol={"M","CM","D","CD","C","XC","L","XL","X","IX",
"V","IV","I"};
```

```
    for(int i=0;i<values.size()&&num>0;i++)
    {
        while(num>=values[i])
        {
            res.append(symbol[i]);
            num-=values[i];
        }
    }
    return res;
}
};
```