**难度: 简单**

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。

给定一个罗马数字，将其转换成整数。输入确保在 1 到 3999 的范围内。

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/roman-to-integer

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路:**

1). 直接使用选择结构switch就能解决了

|  |
| --- |
| public int romanToInt(String s) {  if (s.length() == 0) {  return 0;  }  int sum = 0;  for (int i = 0; i < s.length(); i++) {  switch(s.charAt(i)) {  case 'I' :  if (i + 1 < s.length() && (s.charAt(i+1) == 'V' || s.charAt(i+1) == 'X')) {  sum += s.charAt(i+1) == 'V' ? 4 : 9;  i++;  }else {  sum += 1;  }  break;  case 'V' :  sum += 5;  break;  case 'X' :  if (i+1 < s.length() && (s.charAt(i+1) == 'L' || s.charAt(i+1) == 'C')) {  sum += s.charAt(i+1) == 'L' ? 40 : 90;  i++;  }else {  sum += 10;  }  break;  case 'L' :  sum += 50;  break;  case 'C' :  if (i+1 < s.length() && (s.charAt(i+1) == 'D' || s.charAt(i+1) == 'M')) {  sum += s.charAt(i+1) == 'D' ? 400 : 900;  i++;  }else {  sum += 100;  }  break;  case 'D' :  sum += 500;  break;  case 'M' :  sum += 1000;  break;  }  }  return sum;  } |