Roman to Integer

Given a roman numeral, convert it to an integer.

Input is guaranteed to be within the range from 1 to 3999.

Solution 1

```
public class Solution {
    public int romanToInt(String s) {
        //: I (1) V (5) X (10) L (50) C (100) D (500) M (1000)
        // rules:位于大数的后面时就作为加数；位于大数的前面就作为减数
        //eg: III=3,IV=4,VI=6,XIX=19,XX=20,XLV=45,MCMXXC=1980
        //"DCXXI"

        if(s == null || s.length() == 0) return 0;
        int len = s.length();
        HashMap<Character,Integer> map = new HashMap<Character,Integer>();
        map.put('I',1);
        map.put('V',5);
        map.put('X',10);
        map.put('L',50);
        map.put('C',100);
        map.put('D',500);
        map.put('M',1000);
        int result = map.get(s.charAt(len -1));
        int pivot = result;
        for(int i = len -2; i>= 0;i--){
            int curr = map.get(s.charAt(i));
            if(curr >=  pivot){
                result += curr;
            }else{
                result -= curr;
            }
            pivot = curr;
        }
        return result;
    }
}
```

written by yidi original link here

Solution 2

count every Symbol and add its value to the sum, and minus the extra part of special cases.

```
public int romanToInt(String s) {
     int sum=0;
    if(s.indexOf("IV")!=-1){sum-=2;}
    if(s.indexOf("IX")!=-1){sum-=2;}
    if(s.indexOf("XL")!=-1){sum-=20;}
    if(s.indexOf("XC")!=-1){sum-=20;}
    if(s.indexOf("CD")!=-1){sum-=200;}
    if(s.indexOf("CM")!=-1){sum-=200;}

    char c[]=s.toCharArray();
    int count=0;

   for(;count<=s.length()-1;count++){
       if(c[count]=='M') sum+=1000;
       if(c[count]=='D') sum+=500;
       if(c[count]=='C') sum+=100;
       if(c[count]=='L') sum+=50;
       if(c[count]=='X') sum+=10;
       if(c[count]=='V') sum+=5;
       if(c[count]=='I') sum+=1;

   }

   return sum;

}
```

written by hongbin2 original link here

Solution 3

Problem is simpler to solve by working the string from back to front and using a map. Runtime speed is 88 ms.

```
int romanToInt(string s)
{
    unordered_map<char, int> T = { { 'I' , 1 },
                                   { 'V' , 5 },
                                   { 'X' , 10 },
                                   { 'L' , 50 },
                                   { 'C' , 100 },
                                   { 'D' , 500 },
                                   { 'M' , 1000 } };

   int sum = T[s.back()];
   for (int i = s.length() - 2; i >= 0; --i)
   {
       if (T[s[i]] < T[s[i + 1]])
       {
           sum -= T[s[i]];
       }
       else
       {
           sum += T[s[i]];
       }
   }

   return sum;
}
```

written by wsugrad77 original link here