

89. 罗马数字(Roman numerals)

罗马数字表示必须要遵守三个基本的规则，这种表示才是有效的。尽管规则允许同一个数字用不同的方式来表示，但是总有一个最好的表示方法。例如，至少有六种表示数字十六的方法：IIIIIIIIIIII, VIIIIIIII, VVIIII, XIIII, VIVI, XVI，然而，根据规则只有 XIIII 和 XVI 这两种表示方法是有效的，而且后者更有效率，因为它使用了更少的字母。文本文件 roman.txt 包含一千个有效的罗马数字，但是它们的写法不一定是最优的，可以参考[这个网址](#)了解与这个问题相关的规则。如果把些数字都以其最优形式表示，求可以节约的字母数量。

注：你可以假设文件中的罗马数字中没有连续四个相同的字母。

分析：通过题目中网址给出的介绍，我们知道有两种可以缩减罗马数字表示的字母数的方法：一种方法是把“加法”缩减，比如把 XIIII 缩减为 XVI，但是题目特别说明文本文件中的罗马数字中没有连续四个相同的字母，所以可以知道这种缩减方法对于文件中的罗马数字是不适用的。另一种方法是“减法”缩减，比如把 IIII 缩减为 IV，这种缩减方法对文件中的罗马数字是适用的。

根据网址中的说明，减法缩减需要遵守三个原则：一是 I 必须在 V 和 X 之前，也就是说可以把 IIII 缩减成 IV，把 VIIII 缩减为 IX。二是 X 必须在 L 和 C 之前，则可以把 XXXX 缩减为 XL，把 LXXXX 缩减为 XC。三是 C 必须在 D 和 M 之前，则可以把 DCCCC 缩减为 CM，把 CCCC 缩减为 CD。总而言之，我们有六种对罗马数字进行减法缩减的方法，缩减的结果都是把原来的表示变成了两个字母的表示。

因此，为解决题目中的问题，我们可以应用上面六个缩减规则，把符合标准的表示都替换成两个字母。因为题目只要我们求节约的字母数，所以替换成什么字母都可以了，只要是两个字符就可以了。然后用替换前的总字符数量减去替换后的总字符数量就是节约的字符数量。在涉及到文本处理时，使用 PYTHON 自带的正则表达式模块 `re` 非常的方便，这里我们可以直接调用 `re` 模块中的 `sub` 函数将满足上面六个模式的罗马数字替换成两个星号，然后对比替换前后的总字符数量就可以了。代码如下：

```
# time cost = 565 μs ± 11.4 μs

import re

def main():
    with open('data/ep89.txt') as f:
        romans = f.read()
        new = re.sub("DCCCC|LXXXX|VIIII|CCCC|XXXX|IIII", '**', romans)
        savings = len(romans) - len(new)
        return savings
```