class Solution:

def romanToInt(self, s: str) -> int:

R = {'I':1,'V':5,'X':10,'L':50,'C':100,'D':500,'M':1000}

rlist = []

StoR = lambda e : R[e]

result = 0

if 1 <= len(s) <= 15 and set(s).issubset(['I','V','X','L','C','D','M']):

slist = list(s)

for e in slist:

rlist.append(StoR(e))

for i in range(len(rlist)):

if i == len(rlist)-1:

result = result + rlist[i]

else:

if rlist[i]>=rlist[i+1]:

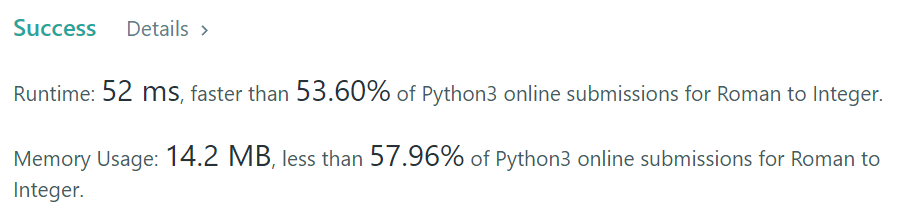
result = result + rlist[i]

else:

result = result - rlist[i]

if result<=3999 and result>=1:

return result



Others:

class Solution:

def romanToInt(self, s: str) -> int:

result\_number = 0

prevous\_number = 0

mapping = {'I':1,

'V':5,

'X':10,

'L':50,

'C':100,

'D':500,

'M':1000}

for symbol in s[::-1]:

if mapping[symbol] >= prevous\_number:

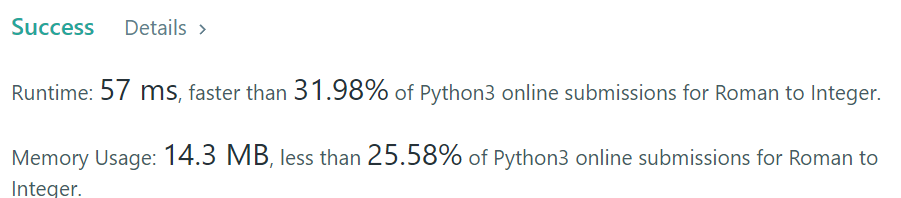
result\_number += mapping[symbol]

else:

result\_number -= mapping[symbol]

prevous\_number = mapping[symbol]

return result\_number



My 2 version:

class Solution:

def romanToInt(self, s: str) -> int:

R = {'I':1,'V':5,'X':10,'L':50,'C':100,'D':500,'M':1000}

rlist = []

StoR = lambda e : R[e]

result = 0

previous\_number = 0

if 1 <= len(s) <= 15 and set(s).issubset(['I','V','X','L','C','D','M']):

slist = list(s)

for e in slist:

rlist.append(StoR(e))

for e in rlist[::-1]:

if e >= previous\_number:

result += e

else:

result -= e

previous\_number = e

if result<=3999 and result>=1:

return result

