**## 1. 题目**

**### 27653: Fraction类**

http://cs101.openjudge.cn/practice/27653/

思路：定义一个函数gcd用于找出最大公约数，并用分子和分母调用函数gcd然后赋值给m，之后就可以直接将分子与分母用m化为最简，然后先考虑分母相同的情况，直接进行分子相加并化为最简分数，然后考虑分母不同的情况，先进性通分再将调整后的分子相加，然后在输出最简分数。

代码：

fz1, fm1, fz2, fm2 = **map**(int,**input**().**split**())

**def** **gcd**(a,b):

**while** b != 0:

a,b=b,a%b

**return** a

**if** fm1 == fm2:

m=**gcd**((fz1+fz2),fm2)

**print**(**int**((fz1+fz2)/m),"/",**int**(fm2/m))

**else**:

fm3 = fm1 \* fm2

fz3 = fz1 \* fm2

fz4 = fz2 \* fm1

m=**gcd**((fz3+fz4),fm3)

**print**(f"{int((fz3+fz4)/m)}/{int(fm3/m)}")

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

  
大约用时：30分钟

**### 1760.袋子里最少数目的球**

 https://leetcode.cn/problems/minimum-limit-of-balls-in-a-bag/

思路：使用了二分查找，将范围定义为1至max（nums）表示最小和最大可能值，然后计算中间值mid，之后遍历nums，计算最每个mid所需的分割次数，如果<=maxOperations说明mid可行并尝试更小的范围，否则尝试更大的值，直至开销最小化。

代码：

class Solution(object):

    def minimumSize(self, nums, maxOperations):

        left, right = 1, max(nums)

        while left < right:

            mid = left + (right - left) // 2

            operations=0

            for num in nums:

                operations += (num - 1) // mid

            if operations <= maxOperations:

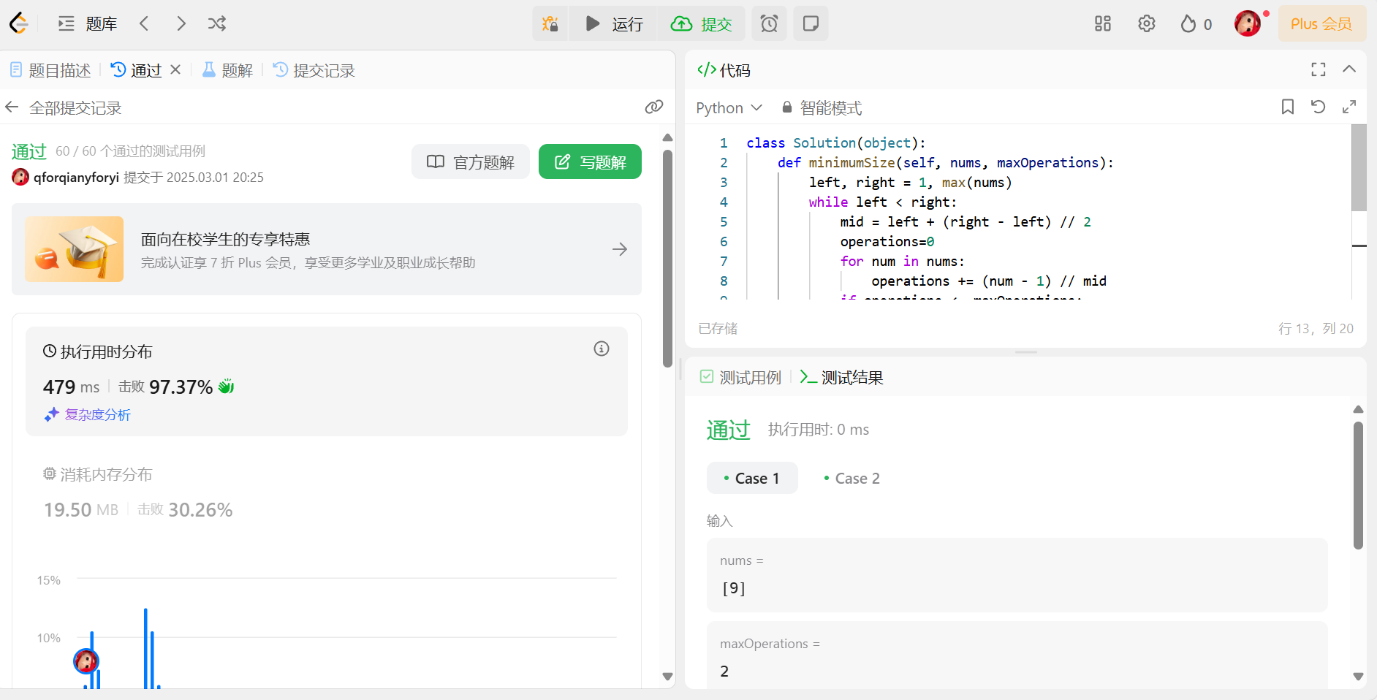
                right = mid

            else:

                left = mid + 1

        return left

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

  
  
大约用时：45分钟

**### 04135: 月度开销**

http://cs101.openjudge.cn/practice/04135

思路：同样使用了二分查找，范围在expenses中的最大值至expenses的总和并计算中间值，之后遍历expenses累加开销直至>最大开销并计算周期，如果<M说明mid值可行，并缩小范围，否则尝试更大的值，直至找到最小的最大开销。

代码：

N, M = **map**(int, **input**().**split**())

expenses = [**int**(**input**()) **for** \_ **in** **range**(N)]

left = **max**(expenses)

right = **sum**(expenses)

result = right

**while** left <= right:

current\_sum=0

periods=1

mid = left + (right - left) // 2

**for** expense **in** expenses:

**if** current\_sum + expense > mid:

periods += 1

current\_sum = expense

**else**:

current\_sum += expense

**if** periods <= M:

result = mid

right = mid - 1

**else**:

left = mid + 1

**print**(result)

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

  
大约用时：50分钟

**### 27300: 模型整理**

http://cs101.openjudge.cn/practice/27300/

思路：是用字典储存不在字典中的keys，value用列表储存，如果已有当前的keys则往该keys中添加对应的value，之后定义一个convert函数将M和B等单位转换成数值，之后进行排序后用f-string格式输出。

代码：

N=**int**(**input**())

dict={}

**for** \_ **in** **range**(N):

models,Gates=**input**().**split**("-",1)

**if** models **not** **in** dict:

dict[models]=[]

dict[models].**append**(Gates)

**def** **convert**(num):

**if** num[-1] == 'M':

**return** **float**(num[:-1]) \* 1e6

**elif** num[-1] == 'B':

**return** **float**(num[:-1]) \* 1e9

**else**:

**return** **float**(num)

**for** i **in** **sorted**(dict):

dict[i].**sort**(key=convert)

**print**(f"{i}: {', '.join(dict[i])}")

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

  
大约用时：30分钟

**## 2. 学习总结和个人收获**

<mark>如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算2025spring每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。</mark>

我认为作业的第一题和第四题相对较简单，我用第一题回顾了之前学过的gcd，并用第四题再次复习了用sorted来对字母进行排序、数值转换和f-string的格式，其中第二题和第三天我认为较难，因为之前没做过关于二分查找的题目，所以在这两题上花费了较多的时间，但也收获不少。  
此外，因为感觉自己做起每日选做的题目还比较吃力，所以我也找了leetcode100题来做，里面帮我复习了各种解题思路，我认为非常有用，但对于选做的大模型作业，我认为自己的能力还尚未能理解，因此先选择不做，但在这个学期我会尽量对这个课题进行初步的尝试。