

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>

OpenJudge 题目ID, 标题, 描述 2400093012 苏倩仪 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#48405711提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
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:
    # print(fz1+fz2, "/", fm2)
    m=gcd((fz1+fz2), fm2)
    print(int((fz1+fz2)/m), "/", int(fm2/m))
else:
    fm3 = fm1 * fm2
    fz3 = fz1 * fm2
    fz4 = fz2 * fm1
    # print(fz3+fz4, "/", fm3)
    m=gcd((fz3+fz4), fm3)
    print(f"{int((fz3+fz4)/m)}/{int(fm3/m)}")
```

基本信息

#: 48405711

题目: 27653

提交人: qy

内存: 3604kB

时间: 29ms

语言: Python3

提交时间: 2025-03-01 20:19:28

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

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

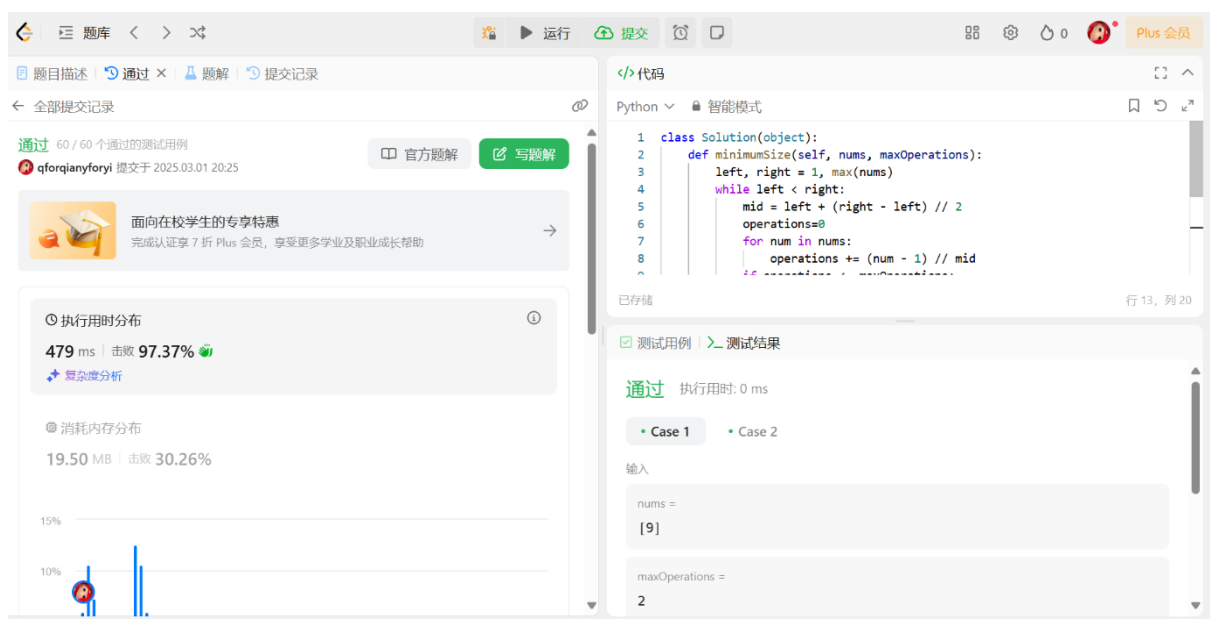
    else:

        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>

状态: **Accepted**

源代码

```
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)
```

基本信息

#: 48405876
题目: 04135
提交人: 2400093012 苏倩仪
内存: 7844kB
时间: 555ms
语言: Python3
提交时间: 2025-03-01 20:35:21

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>

#48406073提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
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])}")
```

基本信息

#: 48406073
 题目: 27300
 提交人: 2400093012 苏倩仪
 内存: 3676kB
 时间: 32ms
 语言: Python3
 提交时间: 2025-03-01 20:50:32

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。

