Assignment #2: 编程练习

Updated 0953 GMT+8 Feb 24, 2024

2024 spring, Complied by 刘子暄 环境科学与工程学院

说明：

1）The complete process to learn DSA from scratch can be broken into 4 parts:

- Learn about Time and Space complexities
- Learn the basics of individual Data Structures
- Learn the basics of Algorithms
- Practice Problems on DSA

2）请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora https://typoraio.cn ，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。

3）课程网站是Canvas平台, https://pku.instructure.com, 学校通知3月1日导入选课名单后启用。**作业写好后，保留在自己手中，待3月1日提交。**

提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、"作业评论"区有上传的md或者doc附件。

4）如果不能在截止前提交作业，请写明原因。

编程环境

操作系统：Windows 11

Python编程环境: PyCharm Community Edition 2023.3

1. 题目

27653: Fraction类

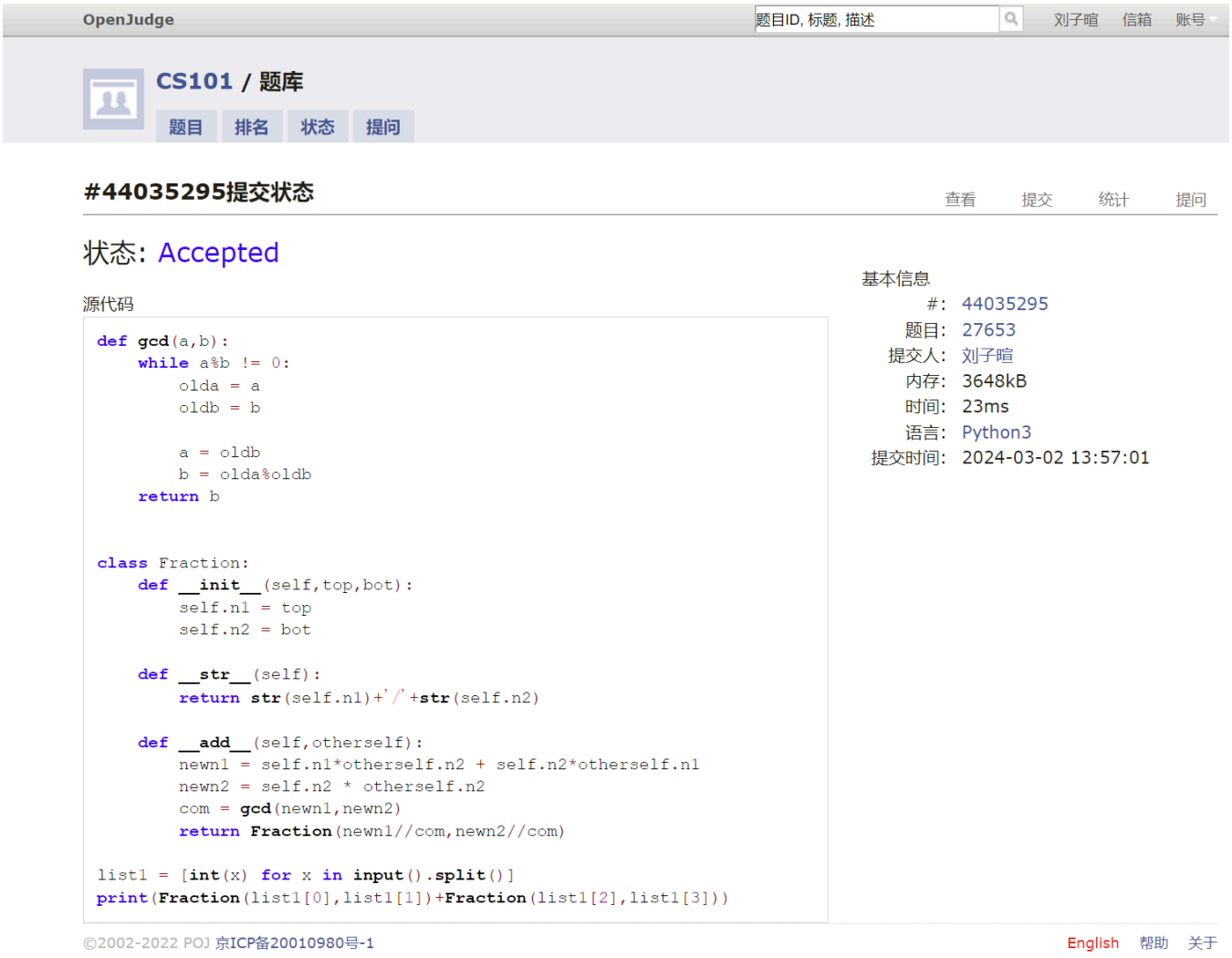
http://cs101.openjudge.cn/practice/27653/

思路：基本完全参考了闫老师的代码，但是我把这个也总结了

代码

```
def gcd(a,b):
    while a%b != 0:
        olda = a
        oldb = b

        a = oldb
        b = olda%oldb
    return b


class Fraction:
    def __init__(self,top,bot):
        self.n1 = top
        self.n2 = bot

    def __str__(self):
        return str(self.n1)+'/'+str(self.n2)

    def __add__(self,otherself):
        newn1 = self.n1*otherself.n2 + self.n2*otherself.n1
        newn2 = self.n2 * otherself.n2
        com = gcd(newn1,newn2)
        return Fraction(newn1//com,newn2//com)

list1 = [int(x) for x in input().split()]
print(Fraction(list1[0],list1[1])+Fraction(list1[2],list1[3]))
```

代码运行截图 （至少包含有"Accepted"）



04110: 圣诞老人的礼物-Santa Clau’s Gifts

greedy/dp, http://cs101.openjudge.cn/practice/04110

思路：方向对了，但是实现的时候没注意复杂度，所以wa了（悲伤），参考了标答
标答思路没有区别，但是把糖果每份都列出来而不是分堆计算，后面遍历很简单。以及倒序表同样简化遍历
（不知道为什么for遍历求和re了，反正记住求和直接sum就行）==【】’/

代码

```
#
```

代码运行截图 （至少包含有"Accepted"）

18182: 打怪兽

implementation/sortings/data structures, http://cs101.openjudge.cn/practice/18182/

思路：

代码

```
#
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

230B. T-primes

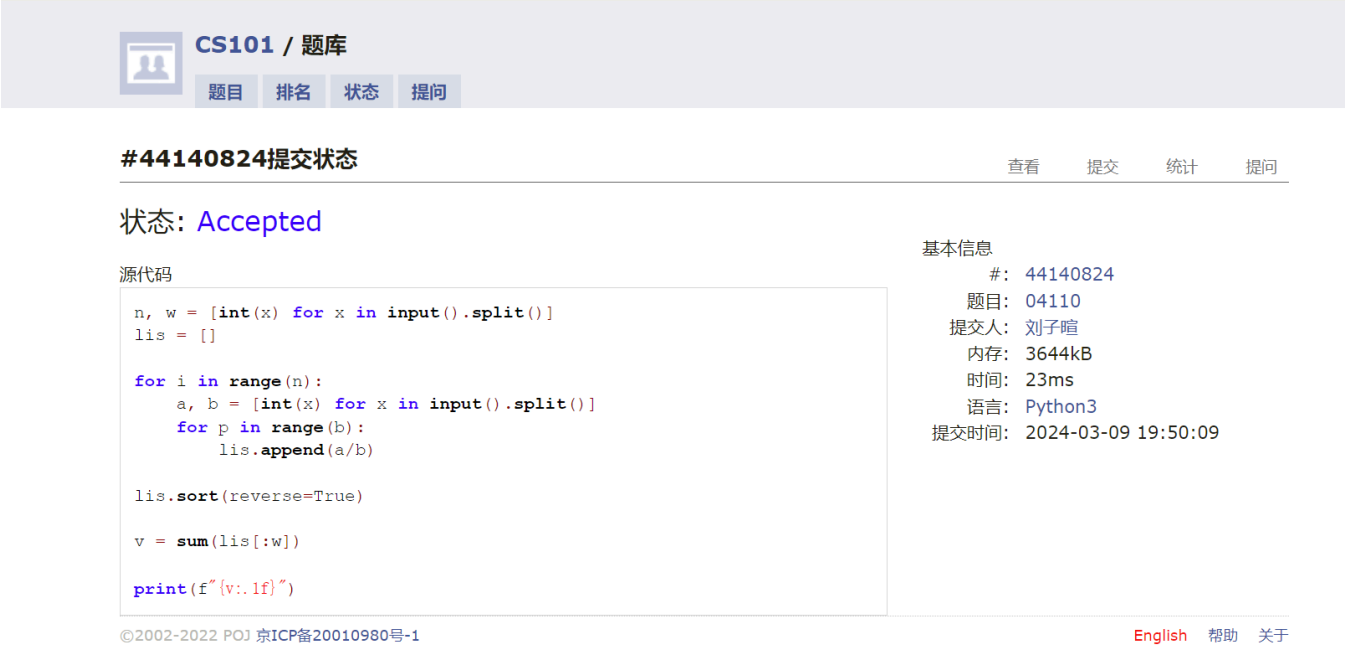
binary search/implementation/math/number theory, 1300, http://codeforces.com/problemset/problem/230/B

思路：

代码

```
n, w = [int(x) for x in input().split()]
lis = []

for i in range(n):
    a, b = [int(x) for x in input().split()]
    for p in range(b):
        lis.append(a/b)

lis.sort(reverse=True)

v = sum(lis[:w])

print(f"{v:.1f}")
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



1364A. XXXXX

brute force/data structures/number theory/two pointers, 1200, https://codeforces.com/problemset/problem/1364/A

思路：完全按照答案来的，思路分析放在下面了

代码

```
#
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

18176: 2050年成绩计算

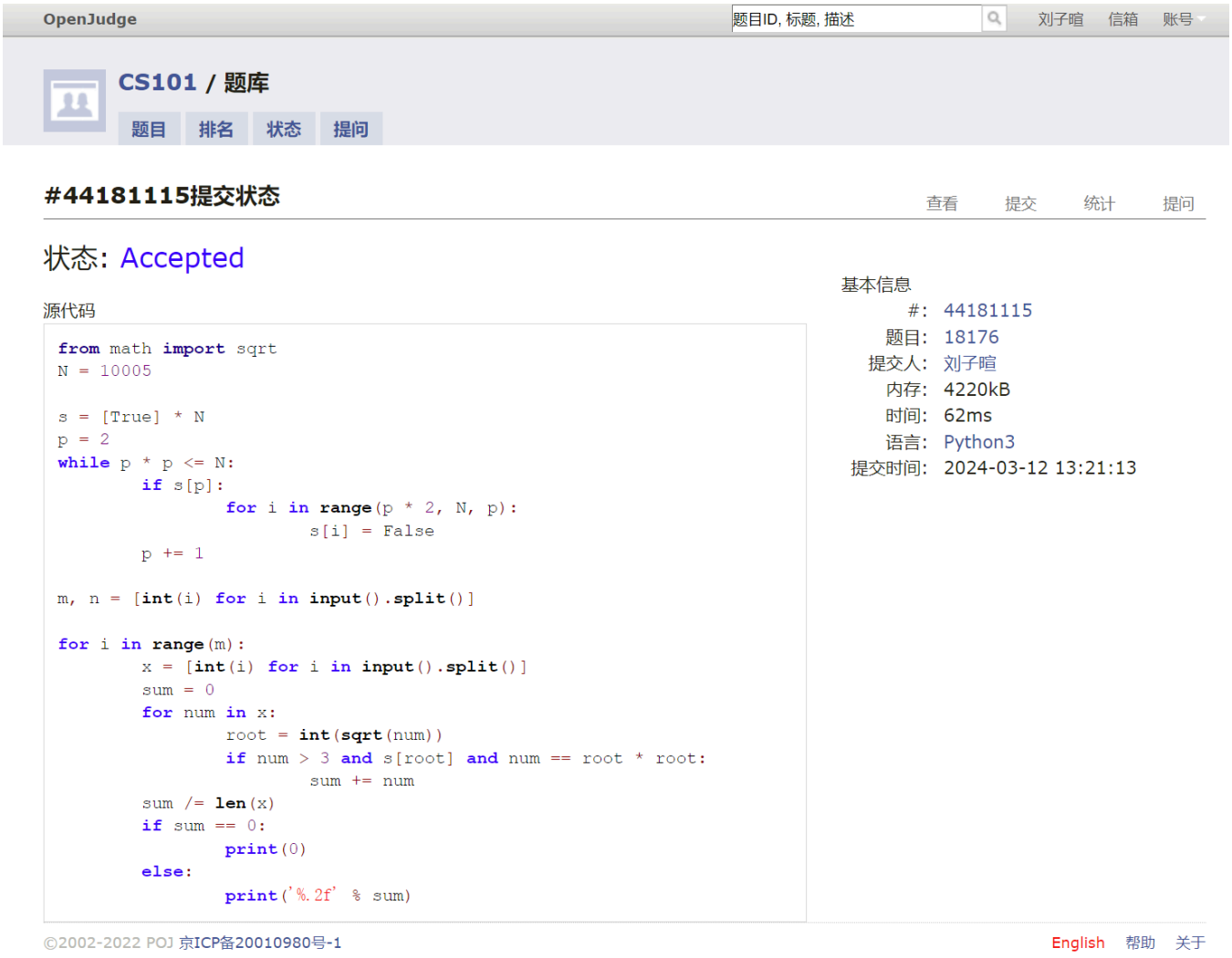
http://cs101.openjudge.cn/practice/18176/

思路：思路和标答是一样的，不知道为什么超时了，所以按照标答做了更改（感觉好像是没有做dp的问题，时间复杂度有点超了）

代码

```
from math import sqrt
N = 10005

s = [True] * N
p = 2
while p * p <= N:
        if s[p]:
                for i in range(p * 2, N, p):
                        s[i] = False
        p += 1

m, n = [int(i) for i in input().split()]

for i in range(m):
        x = [int(i) for i in input().split()]
        sum = 0
        for num in x:
                root = int(sqrt(num))
                if num > 3 and s[root] and num == root * root:
                        sum += num
        sum /= len(x)
        if sum == 0:
                print(0)
        else:
                print('%.2f' % sum)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“2024spring每日选做”、CF、LeetCde、洛谷等网站题目。

####圣诞老人题
for _ in range(n)
这样表明遍历的变量没有使用
map(int,---)
这是映射，表明将之后的数据集合中的所有数据做int处理

####XXXXX
着实对我是难题
第一段

```python
from itertools import accumulate
#accumulate函数：不循环求和 同时这是一个生成器 list会把每一次结果自动排成一个列表

def prefix_sum(nums):
    # prefix = []
    # total = 0
    # for num in nums:
    #     total += num
    #     prefix.append(total)
    # return prefix
    return list(accumulate(nums))

def suffix_sum(nums):
    # suffix = []
    # total = 0
    # # 首先将列表反转
    # reversed_nums = nums[::-1]
    # for num in reversed_nums:
    #     total += num
    #     suffix.append(total)
    # # 将结果反转回来
    # suffix.reverse()
    # return suffix
    return list(accumulate(reversed(nums)))[::-1]
#两个自定义函数：prefix 求正向和 suffix 求逆向和（reversed_nums = nums[::-1]有用的算法）
```

两个自定义函数：
prefix 求正向和
suffix 求逆向和
（reversed_nums = nums[::-1]有用的算法）

第二段

```
int(input())
for _ in range(t):
    N, x = map(int, input().split())
    a = [int(i) for i in input().split()]
    aprefix_sum = prefix_sum(a)
    asuffix_sum = suffix_sum(a)
    #对数据基本处理，结果是apre（正）和asuf（反）两个不同方向子序列的和的所有结果
    #以及元素数N和讨厌数x

    left = 0
    right = N - 1
    if right == 0:
        if a[0] % x != 0:
            print(1)
        else:
            print(-1)
        continue
    #特殊条件判定：当右指针到0时，如果左首项符合则长度为一，否则我负一

    leftmax = 0
    rightmax = 0
    while left != right:
        total = asuffix_sum[left]
        #指针的用途在这里体现，指明了位置
        if total % x != 0:
            leftmax = right - left + 1
            break
        else:
            left += 1

    left = 0
    right = N - 1
    while left != right:
        total = aprefix_sum[right]
        if total % x != 0:
            rightmax = right - left + 1
            break
        else:
            right -= 1
```

```
#上面两串表达的就是分别从右和从左遍历的过程，假如出现了不整除的情况就记录此时的left或right作为已知最长子序列

    if leftmax == 0 and rightmax == 0:
        print(-1)
    else:
        print(max(leftmax, rightmax))
```

这次作业的难度开始上升，其实主要还是我基础太差的问题，还是要多加练习