Assignment #2: 编程练习

Updated 0953 GMT+8 Feb 24, 2024

2024 spring, Complied by Xinjie Song, Phy

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 22H2

Python编程环境：PyCharm 2023.2 (Community Edition)

C/C++编程环境：g++ (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0

1. 题目

27653: Fraction类

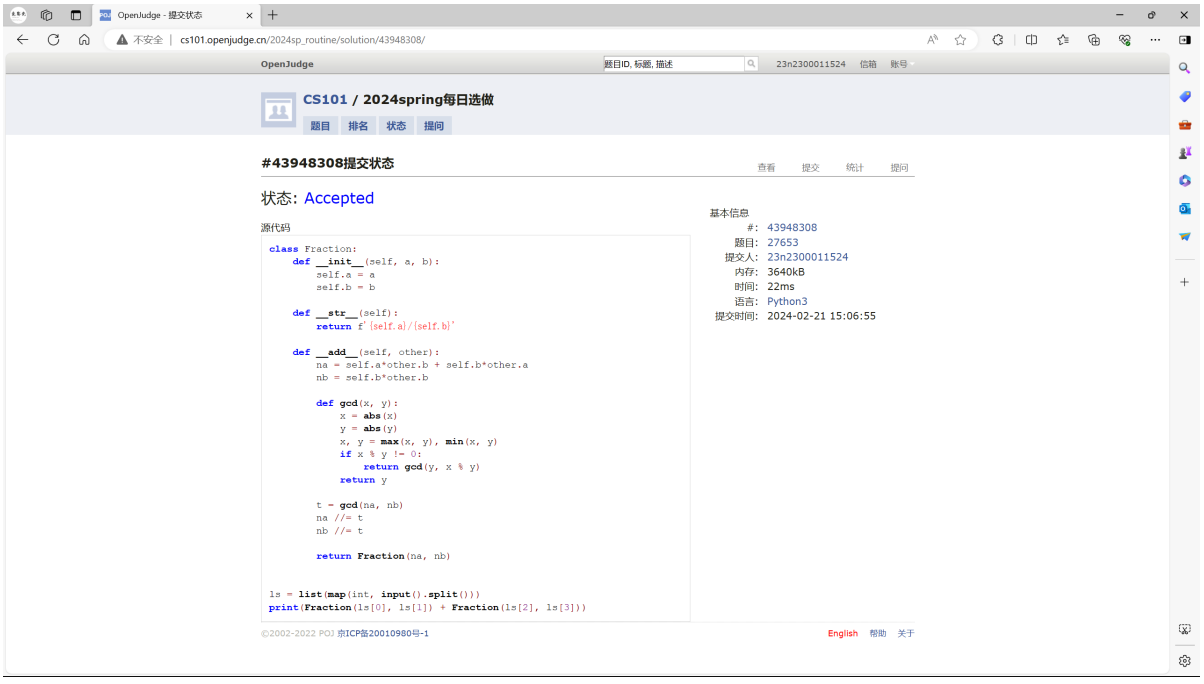
http://cs101.openjudge.cn/2024sp_routine/27653/

思路：常规思路

代码

```
class Fraction:
    def __init__(self, a, b):
        self.a = a
        self.b = b

    def __str__(self):
        return f'{self.a}/{self.b}'

    def __add__(self, other):
        na = self.a*other.b + self.b*other.a
        nb = self.b*other.b

        def gcd(x, y):
            x = abs(x)
            y = abs(y)
            x, y = max(x, y), min(x, y)
            if x % y != 0:
                return gcd(y, x % y)
            return y

        t = gcd(na, nb)
        na //= t
        nb //= t

        return Fraction(na, nb)


ls = list(map(int, input().split()))
print(Fraction(ls[0], ls[1]) + Fraction(ls[2], ls[3]))
```

代码运行截图



04110: 圣诞老人的礼物-Santa Clau’s Gifts

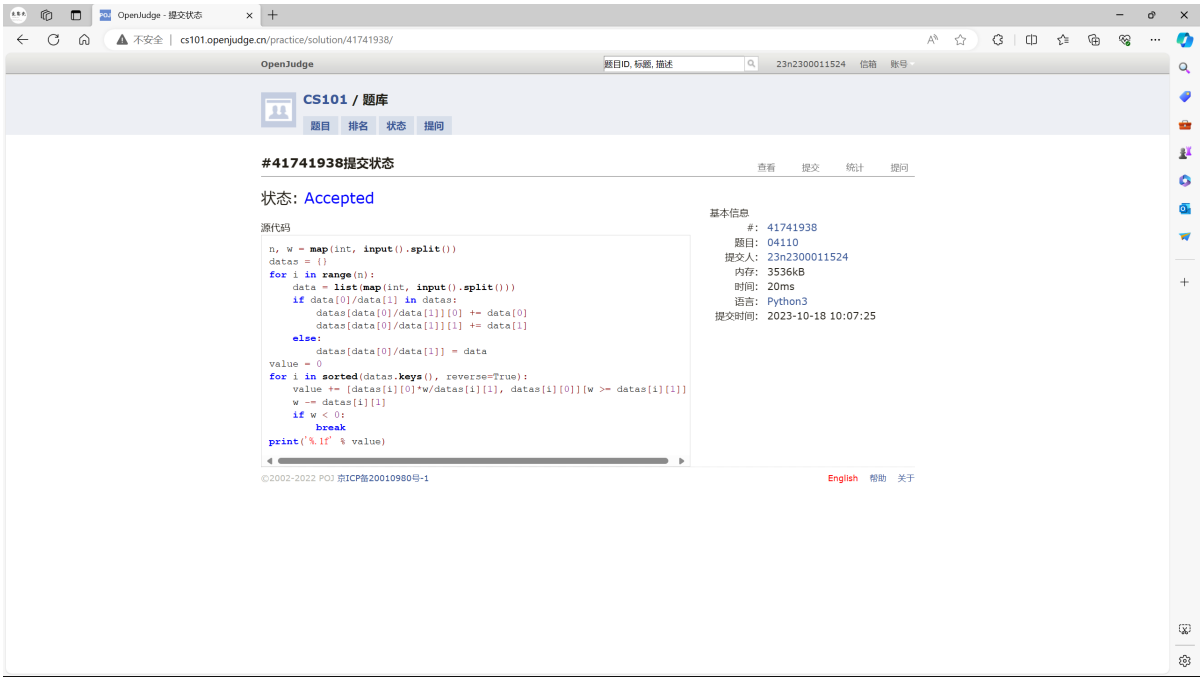
greedy/dp, http://cs101.openjudge.cn/practice/04110

思路：常规思路

代码

```
n, w = map(int, input().split())
datas = {}
for i in range(n):
    data = list(map(int, input().split()))
    if data[0]/data[1] in datas:
        datas[data[0]/data[1]][0] += data[0]
        datas[data[0]/data[1]][1] += data[1]
    else:
        datas[data[0]/data[1]] = data
value = 0
for i in sorted(datas.keys(), reverse=True):
    value += [datas[i][0]*w/datas[i][1], datas[i][0]][w >= datas[i][1]]
    w -= datas[i][1]
    if w < 0:
        break
print('%.1f' % value)
```

代码运行截图



18182: 打怪兽

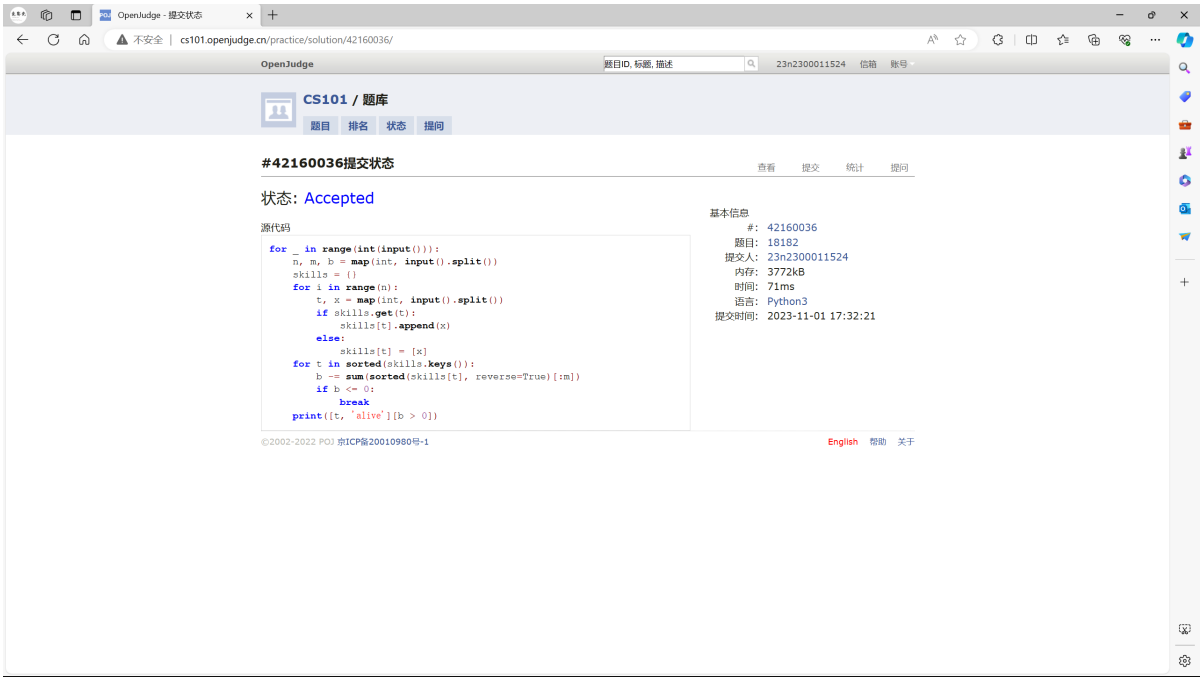
implementation/sortings/data structures, http://cs101.openjudge.cn/practice/18182/

思路：常规思路

代码

```
for _ in range(int(input())):
    n, m, b = map(int, input().split())
    skills = {}
    for i in range(n):
        t, x = map(int, input().split())
        if skills.get(t):
            skills[t].append(x)
        else:
            skills[t] = [x]
    for t in sorted(skills.keys()):
        b -= sum(sorted(skills[t], reverse=True)[:m])
        if b <= 0:
            break
    print([t, 'alive'][b > 0])xxxxxxxxxx for _ in range(int(input())):    n, m, b = map(int, input().split())    skills = {}    for i in range(n):        t, x = map(int, input().split())        if skills.get(t):            skills[t].append(x)        else:            skills[t] = [x]    for t in sorted(skills.keys()):        b -= sum(sorted(skills[t], reverse=True)[:m])        if b <= 0:            break    print([t, 'alive'][b > 0])#
```

代码运行截图



230B. T-primes

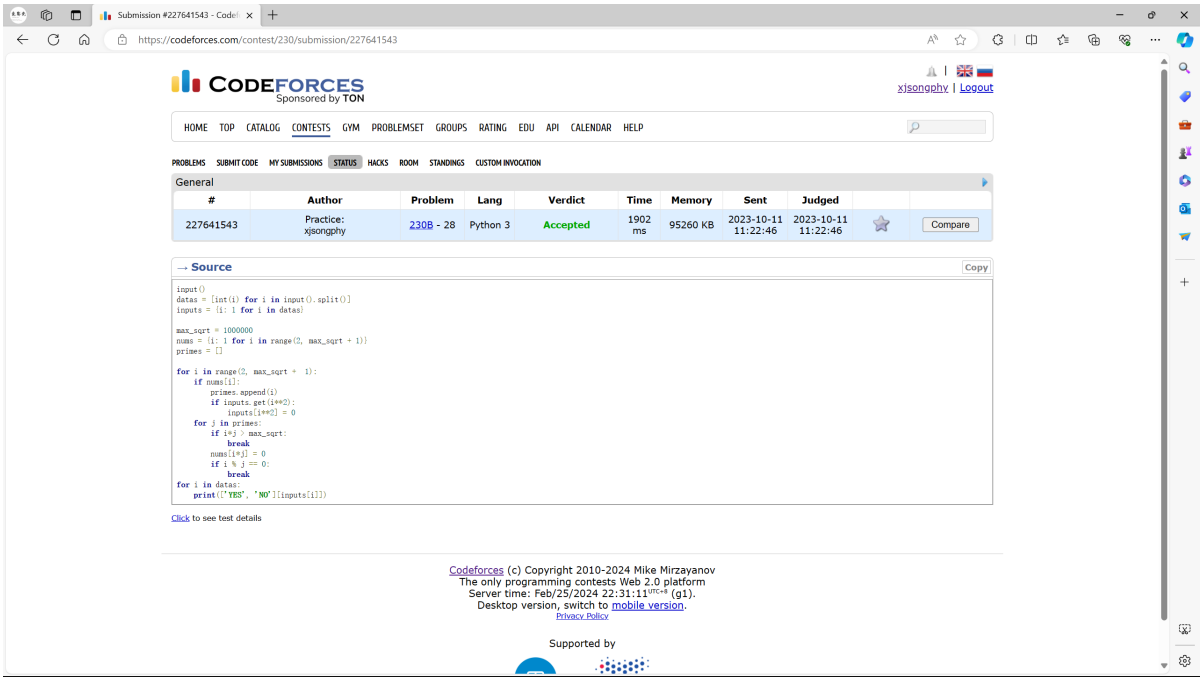
binary search/implementation/math/number theory, 1300, http://codeforces.com/problemset/problem/230/B

思路：常规思路

代码

```
input()
datas = [int(i) for i in input().split()]
inputs = {i: 1 for i in datas}

max_sqrt = 1000000
nums = {i: 1 for i in range(2, max_sqrt + 1)}
primes = []

for i in range(2, max_sqrt +  1):
    if nums[i]:
        primes.append(i)
        if inputs.get(i**2):
            inputs[i**2] = 0
    for j in primes:
        if i*j > max_sqrt:
            break
        nums[i*j] = 0
        if i % j == 0:
            break
for i in datas:
    print(['YES', 'NO'][inputs[i]])
```

代码运行截图



1364A. XXXXX

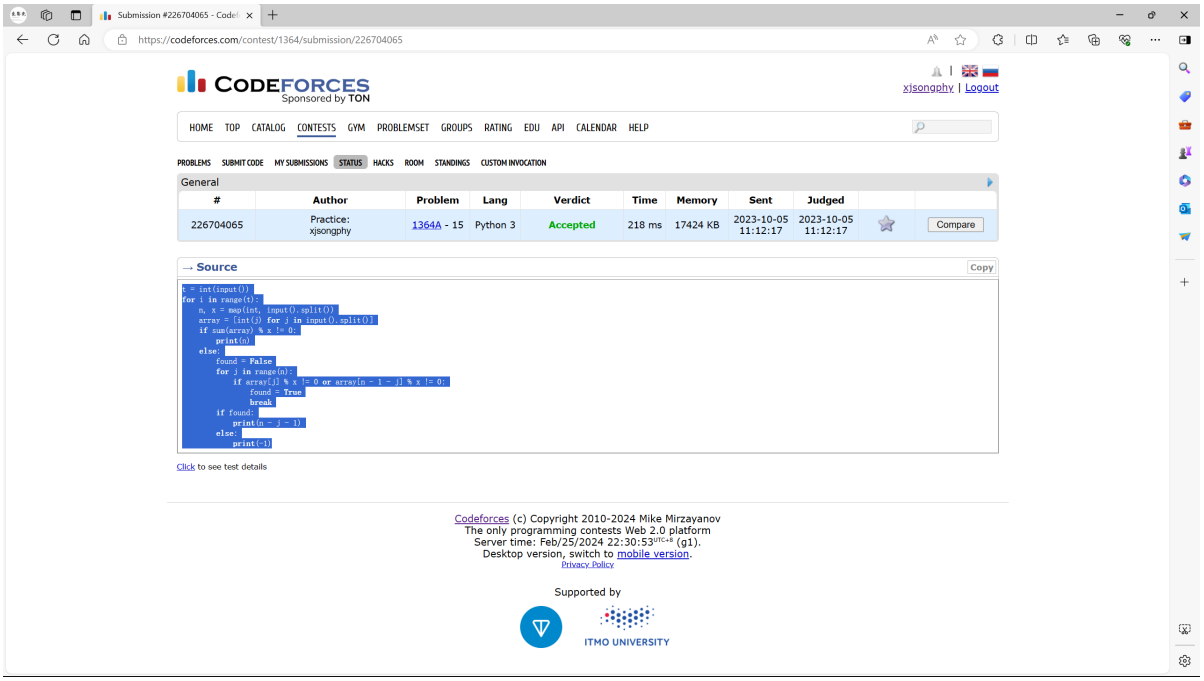
brute force/data structures/number theory/two pointers, 1200, https://codeforces.com/problemset/problem/1364/A

思路：常规思路

代码

```
t = int(input())
for i in range(t):
    n, x = map(int, input().split())
    array = [int(j) for j in input().split()]
    if sum(array) % x != 0:
        print(n)
    else:
        found = False
        for j in range(n):
            if array[j] % x != 0 or array[n - 1 - j] % x != 0:
                found = True
                break
        if found:
            print(n - j - 1)
        else:
            print(-1)
```

代码运行截图



18176: 2050年成绩计算

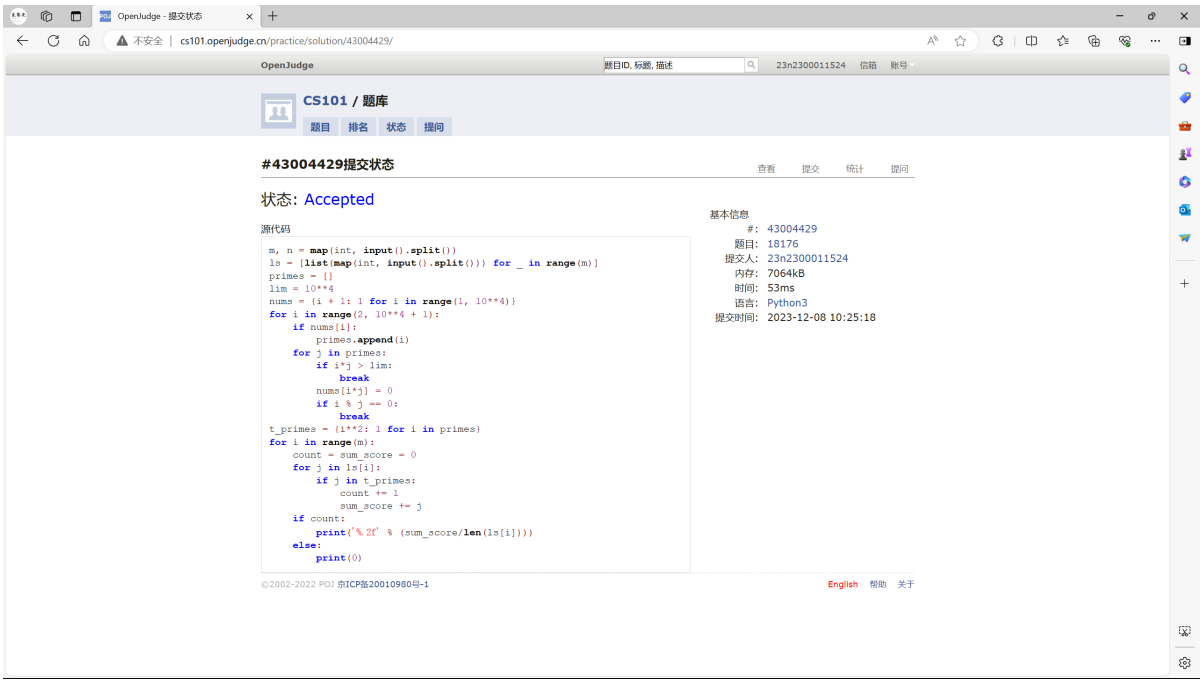
http://cs101.openjudge.cn/practice/18176/

思路：常规思路

代码

```
m, n = map(int, input().split())
ls = [list(map(int, input().split())) for _ in range(m)]
primes = []
lim = 10**4
nums = {i + 1: 1 for i in range(1, 10**4)}
for i in range(2, 10**4 + 1):
    if nums[i]:
        primes.append(i)
    for j in primes:
        if i*j > lim:
            break
        nums[i*j] = 0
        if i % j == 0:
            break
t_primes = {i**2: 1 for i in primes}
for i in range(m):
    count = sum_score = 0
    for j in ls[i]:
        if j in t_primes:
            count += 1
            sum_score += j
    if count:
        print('%.2f' % (sum_score/len(ls[i])))
    else:
        print(0)
```

代码运行截图



2. 学习总结和收获

确实简单，教材看了一半，看完以后开始做每日选做