Assignment #2: 编程练习

Updated 0953 GMT+8 Feb 24, 2024

2024 spring, Complied by 胡登科、生科

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

（Microsoft Windows 11 家庭中文版、Python 3.7, PyCharm 2023.1.4）

1. 题目

27653: Fraction类

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

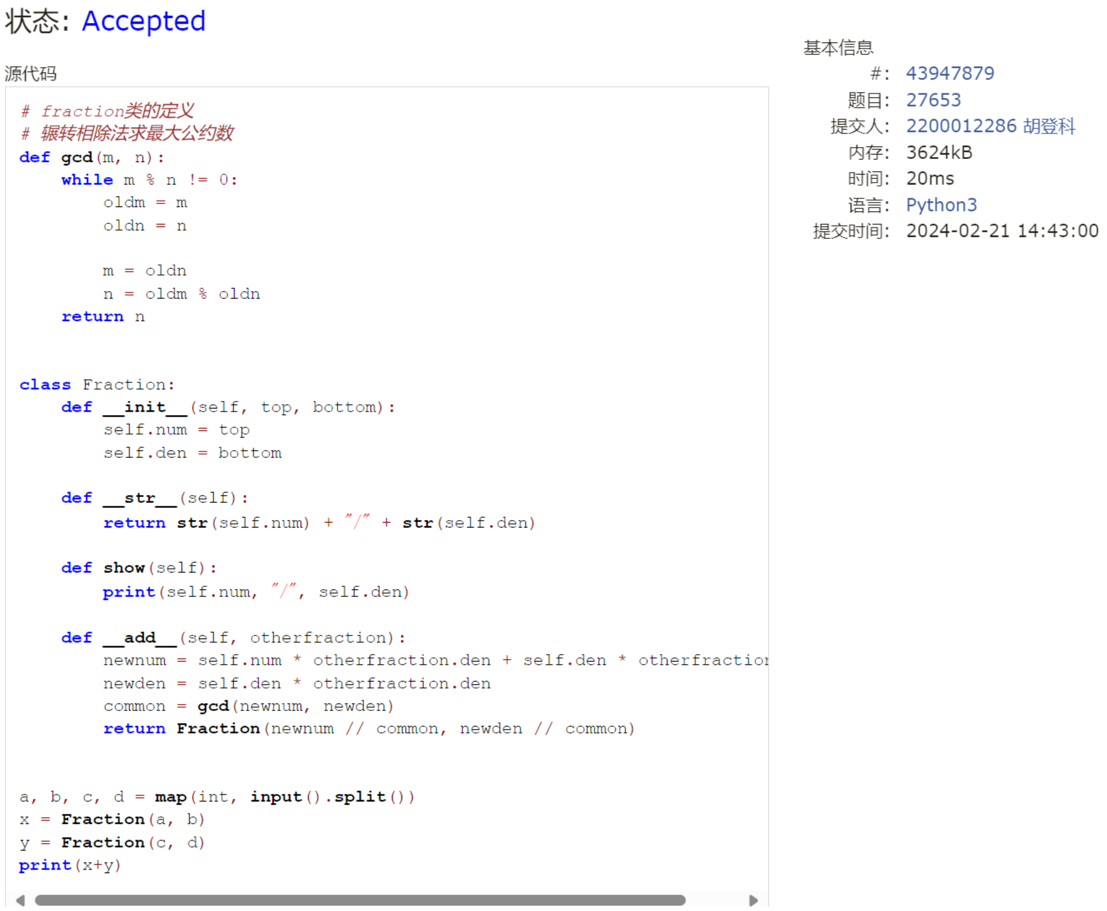
思路：根据上课讲的格式再练习一遍

用时：20min

代码

```
#
# fraction类的定义
# 辗转相除法求最大公约数
def gcd(m, n):
    while m % n != 0:
        oldm = m
        oldn = n

        m = oldn
        n = oldm % oldn
    return n


class Fraction:
    def __init__(self, top, bottom):
        self.num = top
        self.den = bottom

    def __str__(self):
        return str(self.num) + "/" + str(self.den)

    def show(self):
        print(self.num, "/", self.den)

    def __add__(self, otherfraction):
        newnum = self.num * otherfraction.den + self.den * otherfraction.num
        newden = self.den * otherfraction.den
        common = gcd(newnum, newden)
        return Fraction(newnum // common, newden // common)


a, b, c, d = map(int, input().split())
x = Fraction(a, b)
y = Fraction(c, d)
print(x+y)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
# fraction类的定义
# 辗转相除法求最大公约数
def gcd(m, n):
    while m % n != 0:
        oldm = m
        oldn = n

        m = oldn
        n = oldm % oldn
    return n


class Fraction:
    def __init__(self, top, bottom):
        self.num = top
        self.den = bottom

    def __str__(self):
        return str(self.num) + "/" + str(self.den)

    def show(self):
        print(self.num, "/", self.den)

    def __add__(self, otherfraction):
        newnum = self.num * otherfraction.den + self.den * otherfractio
        newden = self.den * otherfraction.den
        common = gcd(newnum, newden)
        return Fraction(newnum // common, newden // common)


a, b, c, d = map(int, input().split())
x = Fraction(a, b)
y = Fraction(c, d)
print(x+y)
```

基本信息

#:	43947879
题目:	27653
提交人:	2200012286 胡登科
内存:	3624kB
时间:	20ms
语言:	Python3
提交时间:	2024-02-21 14:43:00

04110: 圣诞老人的礼物-Santa Clau’s Gifts

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

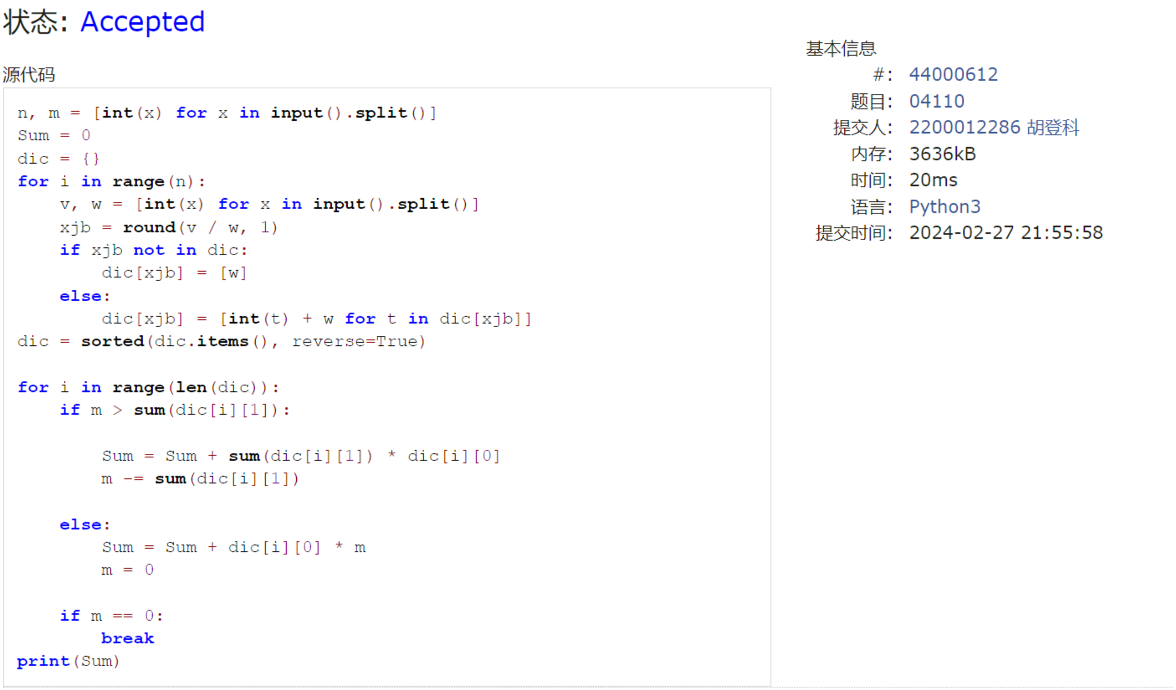
思路：性价比+打怪兽+眼神好

用时：不累的话，可能20min,实际上debug用了快半个小时。

代码

```
# n, m = [int(x) for x in input().split()]
Sum = 0
dic = {}
for i in range(n):
    v, w = [int(x) for x in input().split()]
    xjb = round(v / w, 1)
    if xjb not in dic:
        dic[xjb] = [w]
    else:
        dic[xjb] = [int(t) + w for t in dic[xjb]]
dic = sorted(dic.items(), reverse=True)

for i in range(len(dic)):
    if m > sum(dic[i][1]):

        Sum = Sum + sum(dic[i][1]) * dic[i][0]
        m -= sum(dic[i][1])
    else:
        Sum = Sum + dic[i][0] * m
        m = 0

    if m == 0:
        break
print(Sum)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
n, m = [int(x) for x in input().split()]
Sum = 0
dic = {}
for i in range(n):
    v, w = [int(x) for x in input().split()]
    xjb = round(v / w, 1)
    if xjb not in dic:
        dic[xjb] = [w]
    else:
        dic[xjb] = [int(t) + w for t in dic[xjb]]
dic = sorted(dic.items(), reverse=True)

for i in range(len(dic)):
    if m > sum(dic[i][1]):

        Sum = Sum + sum(dic[i][1]) * dic[i][0]
        m -= sum(dic[i][1])

    else:
        Sum = Sum + dic[i][0] * m
        m = 0

    if m == 0:
        break
print(Sum)
```

基本信息

#:	44000612
题目:	04110
提交人:	2200012286 胡登科
内存:	3636kB
时间:	20ms
语言:	Python3
提交时间:	2024-02-27 21:55:58

18182: 打怪兽

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

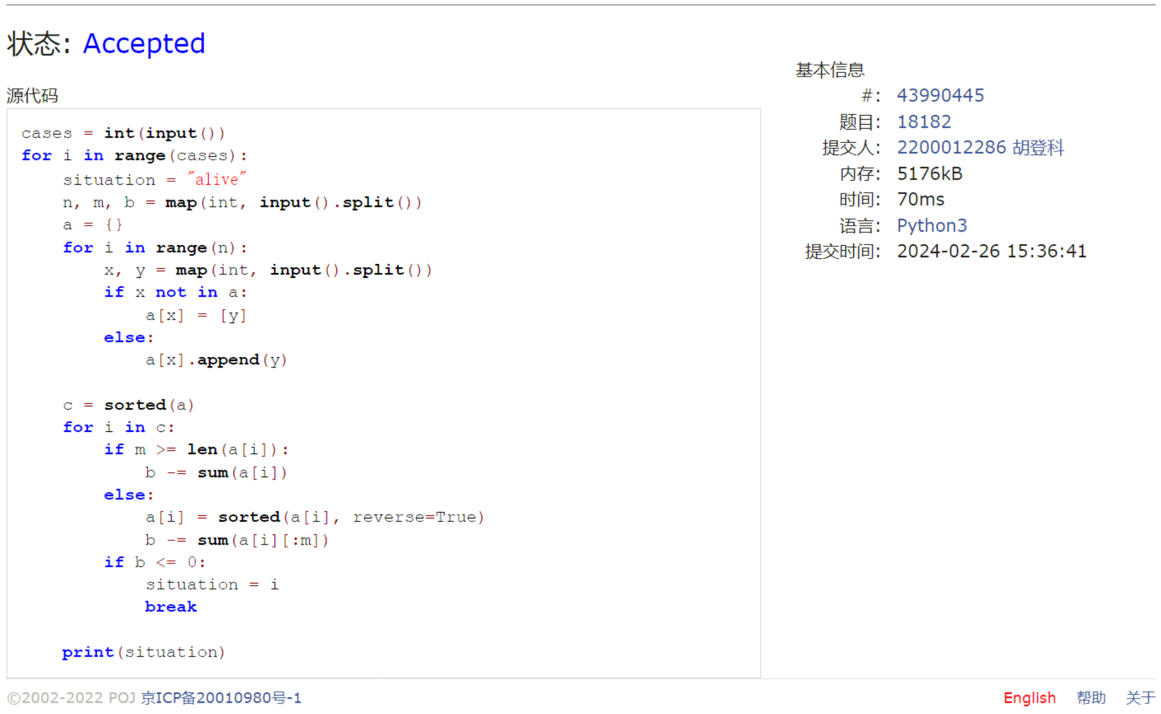
思路：较为灵巧地使用字典，将Value作为list来考虑问题。

用时：20min

代码

```
#
cases = int(input())
for i in range(cases):
    situation = "alive"
    n, m, b = map(int, input().split())
    a = {}
    for i in range(n):
        x, y = map(int, input().split())
        if x not in a:
            a[x] = [y]
        else:
            a[x].append(y)

    c = sorted(a)
    for i in c:
        if m >= len(a[i]):
            b -= sum(a[i])
        else:
            a[i] = sorted(a[i], reverse=True)
            b -= sum(a[i][:m])
        if b <= 0:
            situation = i
            break

    print(situation)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```
cases = int(input())
for i in range(cases):
    situation = "alive"
    n, m, b = map(int, input().split())
    a = {}
    for i in range(n):
        x, y = map(int, input().split())
        if x not in a:
            a[x] = [y]
        else:
            a[x].append(y)

    c = sorted(a)
    for i in c:
        if m >= len(a[i]):
            b -= sum(a[i])
        else:
            a[i] = sorted(a[i], reverse=True)
            b -= sum(a[i][:m])
        if b <= 0:
            situation = i
            break

    print(situation)
```

基本信息

#:	43990445
题目:	18182
提交人:	2200012286 胡登科
内存:	5176kB
时间:	70ms
语言:	Python3
提交时间:	2024-02-26 15:36:41

©2002-2022 POJ 京ICP备20010980号-1

230B. T-primes

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

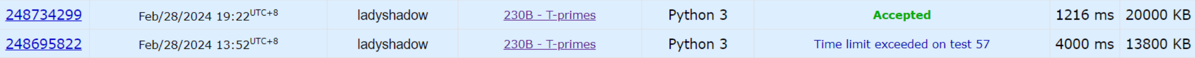
思路：感觉没怎么掌握这种算法的逻辑，用自己的算法来判断感觉时间复杂度太大了，总是超时。这个题和选课那题一样，需要用T-prime去思考问题。但是我还是没有理解这种算法的本质是啥。

用时：没在codeforces上跑成功。

代码

```python
#
# https://blog.dotcpp.com/a/69737
# https://blog.csdn.net/xuechen_gemgirl/article/details/79555123
def euler(r):
    prime = [0 for i in range(r+1)]
    common = []
    for i in range(2, r+1):
        if prime[i] == 0:
            common.append(i)
        for j in common:
            if i*j > r:
                break
            prime[i*j] = 1
            if i % j == 0:
                break
    return prime

s = euler(1000000)
#print(s)

input()
for i in map(int,input().split()):
    if i<4:
        print('NO')
        continue
    elif int(i**0.5)**2 != i:
        print('NO')
        continue
    if s[int(i**0.5)]==0:
        print('YES')
    else:
        print('NO')
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



248734299	Feb/28/2024 19:22^{UTC+8}	ladyshadow	230B - T-primes	Python 3	Accepted	1216 ms	20000 KB
248695822	Feb/28/2024 13:52^{UTC+8}	ladyshadow	230B - T-primes	Python 3	Time limit exceeded on test 57	4000 ms	13800 KB

1364A. XXXXX

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

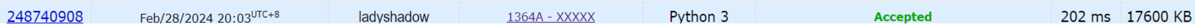
思路：读不懂题，orz

感觉英文题翻译过来也不咋地看得懂

代码

```python
# 
for _ in range(int(input())):
    a, b = map(int, input().split())
    s = -1
    A = list(map(lambda x: int(x) % b, input().split()))
    if sum(A) % b:
        print(a)
        continue
    for i in range(a//2+1):
        if A[i] or A[~i]:
            s = a-i-1
            break
    print(s)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



248740908 Feb/28/2024 20:03UTC+8 ladyshadow 1364A - XXXXX Python 3 Accepted 202 ms 17600 KB

18176: 2050年成绩计算

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

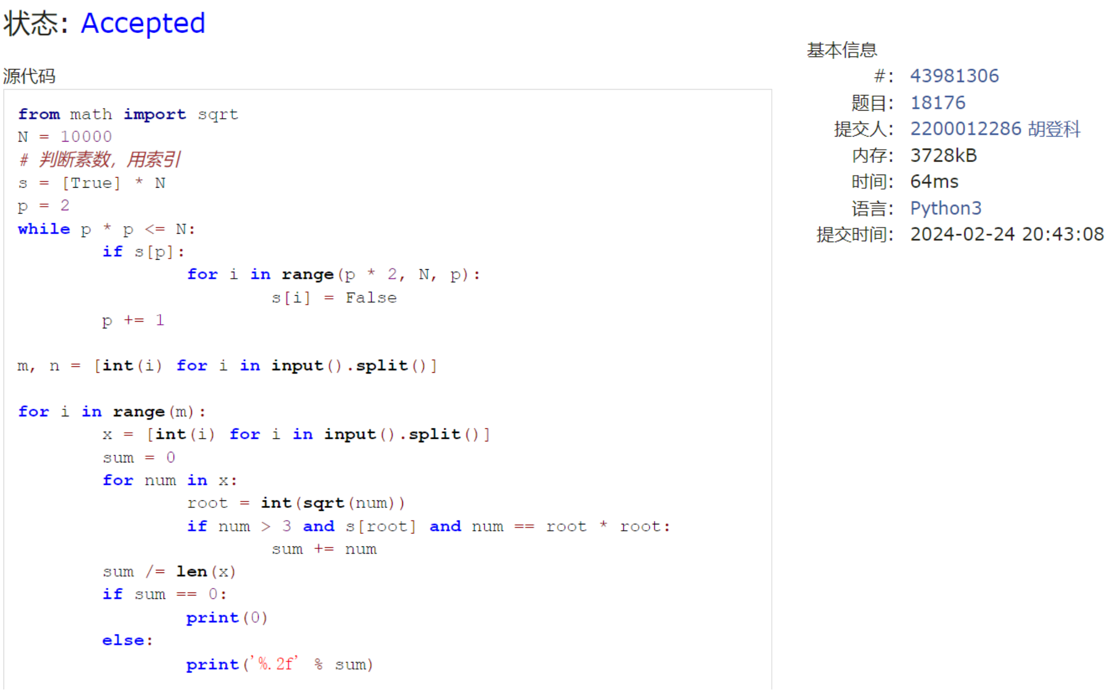
思路：这个题用素数判断函数会导致超时 这用了一种我不怎么理解的素数处理方法。

用时：没做出来

代码

```python
# 
from math import sqrt
N = 10000
# 判断素数，用索引
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"）



状态: Accepted

源代码

```
from math import sqrt
N = 10000
# 判断素数，用索引
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)
```

基本信息

#:	43981306
题目:	18176
提交人:	2200012286 胡登科
内存:	3728kB
时间:	64ms
语言:	Python3
提交时间:	2024-02-24 20:43:08

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、LeetCode、洛谷等网站题目。

感觉现在遇到一些瓶颈，就是比较不容理解一些算法的逻辑语言，然后时间也有点紧张，我尽力抓紧，一天一坤小时的时间去练习学习。加油，奥利给。我相信自己能学明白。