Assignment #2: 编程练习 Updated 0953 GMT+8 Feb 24,

2024 2024 spring, Complied by 吴至超,城市与环境学院

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 (请改为同学的操作系统、编程环境等) 操作系统: Windows11

Python编程环境: pycharm 2023.2.3

1. 题目 27653: Fraction类 http://cs101.openjudge.cn/2024sp routine/27653/

思路:了解如何设计类 (class)

- ①要先初始化__init __ (self, top, bottom) 【此处的top与bottom理解为设计时所使用的两个数字】,几个设计中的变量self.xxx
- ②然后设置它print出来的格式。用show (self) , 然后print ("..你想要的格式.")
- ③自动转化为字符串的函数 __ str __ (self, another) another此处指另一个fraction
- ④设计加法函数,也就是根据数学进行运算
- ⑤有关寻找最大公约数的函数:这种方法的有效性基于以下观察:如果 m 和 n 是整数,并且 m > n,那么 m 和 n 的最大公约数等于 n 和 m 的最大公约数。因此不断减小问题规模,直到n 可整除m%n (类似递归)

代码

```
# 寻找最大公约数
def gys(m, n):
   while m % n != 0:
       oldm = m
       m = n
       n = oldm % n
   return n
class Fraction:#创建一个对象
   def __init__(self,top,bottom):#初始化
       self.num =top #可以理解为这里的变量名称为num与den,并且假定( , )前一个数为
top,另一个数为bottom
       self.den = bottom
   def show(self):
       print(self.num,"/",self.den)
   def __str__(self):#好像会自动转化
       return str(self.num)+"/"+str(self.den)
   def __add__(self, another):#会赋给"+"
       newnum=self.num * another.den + self.den * another.num
```

```
newden=self.den * another.den
    gg=gys(newnum,newden)
    newnum=newnum//gg
    newden=newden//gg
    return Fraction(newnum,newden)

alist=[int(x) for x in input().split()]
myf1=Fraction(alist[0],alist[1])
myf2=Fraction(alist[2],alist[3])
print(myf1 + myf2)
```

代码运行截图 (至少包含有"Accepted")



2. 04110: 圣诞老人的礼物-Santa Clau's Gifts greedy/dp, http://cs101.openjudge.cn/practice/0411

0

思路:可拆分,按比例来装满重量即可。

代码(+进阶背包版,原因是先入为主了)

```
#原題
n,w=(int(x) for x in input().split())#n是糖果箱数, w是最大载重
a=[]
for i in range(n):
    p=[int(x) for x in input().split()]
    a.append(p)
    p=[]
a.sort(key=lambda x : (-x[0],x[1]))
count=0
for u in range(len(a)):
    if w>=a[u][1]:
        count+=a[u][0]
        w-=a[u][1]
    elif w>0 and w<a[u][1]:
        count+=(w/a[u][1])*a[u][0]
        break
```

代码运行截图 (至少包含有"Accepted")



3. 18182: 打怪兽 implementation/sortings/data structures, http://cs101.openjudge.cn/practice/1
8182/

思路: 使用了列表遍历时刻会超时。遂参考答案,改用字典,遍历索引代替遍历时刻。 代码

```
n1=int(input())
for i in range(n1):
    n,m,b=(int(x) for x in input().split())
    pp={}
    for o in range(n):
        ji,xi=(map(int,input().split()))
        if ji not in pp:
```



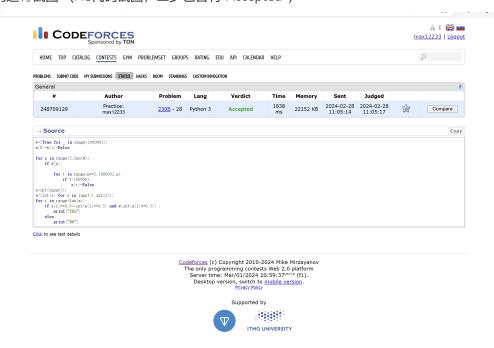
1. 230B. T-primes binary search/implementation/math/number theory, 1300, http://codeforces.com/problemset/problem/23 0/B

思路: 同2050成绩计算

代码

```
w=[True for _ in range(1000001)]
w[0]=w[1]=False

for p in range(2,len(w)):
    if w[p]:
```



2. 1364A. XXXXX brute force/data structures/number theory/two pointers, 1200, https://codeforces.com/problemset/proble m/1364/A

思路:参考了答案。可以用数学方法证明, subarray至少有一段会与原array重叠, 所以分别从左右两端起始看看, 哪个最长取哪个。不太好想到

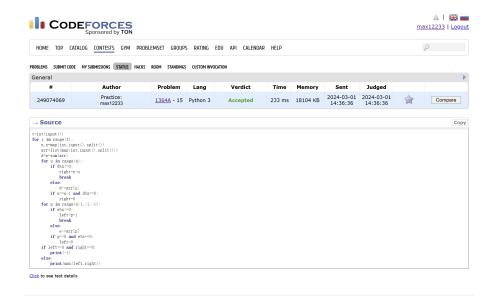
也可以用递归, 但是会re

代码

```
#由数学推导可知,这段子数组一定从某一头延续,只是不知道哪边更长

t=int(input())
for i in range(t):
    n,x=map(int,input().split())
    arr=list(map(int,input().split()))
    d=e=sum(arr)
    for u in range(n):
        if d%x!=0:
        right=n-u
        break
```

```
else:
            d-=arr[u]
        if u==n-1 and d%x==0:
            right=0
    for p in range(n-1,-1,-1):
        if e%x!=0:
            left=p+1
            break
        else:
            e-=arr[p]
        if p==0 and e%x==0:
            left=0
    if left==0 and right==0:
        print(-1)
    else:
        print(max(left,right))
递归版
import sys
sys.setrecursionlimit(10000)
from functools import lru_cache
@lru_cache(maxsize=10000000)
def queren(arr,i,j,x):
    if i<=j:</pre>
        if sum(arr[i:j])%x==0:
            return max(queren(arr,i+1,j,x), queren(arr,i,j-1,x))#减小规模
            return j-i#输出列表长度
    else:
        return -1#基本结束条件
t=int(input())
for i in range(t):
    n,x=map(int,input().split())
    arr=list(map(int,input().split()))
    a=queren(tuple(arr),0,n,x)#缓冲得用tuple
    print(a)
t=int(input())
for i in range(t):
    n,x=map(int,input().split())
    arr=list(map(int,input().split()))
```



1. 18176: 2050年成绩计算 http://cs101.openjudge.cn/practice/18176/

思路: 1)使用复杂度更小的筛法建立列表保存一定范围内的所有素数,当后续需要进行素数判断是直接判断其是否在表中即可。如果设计is_prime函数对(2,平方根+1)进行遍历,会超时,复杂度为O (sqrt (n)),而埃拉托斯特尼筛法的时间复杂度为O(N log log N),可以让测试样例从500ms左右变为100ms以内。

2) 如何判断一个数是否是平方数: int(n ** 0.5) (这边是整型) ==n **0.5(这边是浮点型) 如: n=4→2==2.0, n=3→1! =1.732... 代码

```
#埃拉托斯特尼筛法
#表示筛选[1,n]的素数
waiting=[True for x in range(10001)]
waiting[0]=False
waiting[1]=False
finished=[]
p=2
while p**2<=10000:
    if waiting[p]:
        for m in range(p**2,10000,p):
            waiting[m]=False
    p+=1
m,n=map(int,input().split())
for i in range(m):
    grade=[int(x) for x in input().split()]
    valid=[num for num in grade if num**0.5==int(num**0.5) and
waiting[int(num**0.5)] ]
    if len(grade)==0 or valid==[]:
        print("0")
    else:
        a=sum(valid)/len(grade)
        print(f"{a:.2f}")
```



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。 1 # 2

学习总结:

这次有两道题目使用我最初的想法无法ac,一是递归栈爆问题,二是list超时问题。

参考了答案,学习到用字典索引来减少遍历以及关于字典的一些语法,因为学计概时对dict基本就是一个置之不理的状态,这次发现它的大用途了,尤其是其value值可以是列表这一点。

以及计概时对递归理解不了一点,这次大胆尝试了,(尽管结合gpt修改了完善了一下),对递归的理解又进了一步,但是还不够,要继续加油!

在努力补每日选做