

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://typora.io.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 $m \% n$ 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"）

OpenJudge 题目ID, 标题, 描述 2300013289 信箱 账号

CS101 / 2024spring每日选做

题目 排名 状态 提问

#43975531提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```

# 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
        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)

```

基本信息

#: 43975531
 题目: 27653
 提交人: 2300013289
 内存: 3548kB
 时间: 20ms
 语言: Python3
 提交时间: 2024-02-23 23:37:52

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

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）

```

#原题
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

```

```

print(f"{count:.1f}")

#背包
n,w=(int(x) for x in input().split())#n是糖果箱数，w是最大载重

p=[]
dp=[[int(0) for i in range(w+1)] for u in range(n+1)]
for i in range(n):

    uu=[int(x) for x in input().split()]
    p.append(uu)
    uu=[]
p=[0]+p
for t in range(1,n+1):#纵向
    for i in range(1,w+1):#横向
        if i>=p[t][1]:
            dp[t][i]=max(dp[t-1][i],dp[t-1][i-p[t][1]]+p[t][0])
print(f"{dp[-1][-1]:.1f}")

```

代码运行截图（至少包含有"Accepted"）

OpenJudge 题目ID, 标题, 描述 2300013289 信箱 账号

CS101 / 题库 题目 排名 状态 提问

#44006356提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```

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
print(f"{count:.1f}")

```

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

基本信息

#: 44006356
 题目: 04110
 提交人: 2300013289
 内存: 3540kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-02-28 16:58:56

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

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:

```

```

pp[ji]=[xi]#这里的ji与列表不同，表示key的索引，此处的value值设置成list
else:
    pp[ji].append(xi)

```

newpp=sorted(pp)#默认key值升序,sorted会以list形式返回,且只返回时刻组成的list,与values无关

```

for i in newpp:
    if m>=len(pp[i]):
        b-=sum(pp[i])
    else:
        pp[i].sort(reverse=True)
        b-=sum(pp[i][0:m])
    if b <=0:
        print(i)
        break
if b>0:
    print("alive")

```

代码运行截图（AC代码截图，至少包含有"Accepted"）

OpenJudge 题目ID, 标题, 描述 2300013289 信箱 账号

CS101 / 题库 题目 排名 状态 提问

#44012443提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```

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:
            pp[ji]=[xi]#这里的ji与列表不同，表示key的索引，此处的value值设置成list
        else:
            pp[ji].append(xi)

    newpp=sorted(pp)#默认key值升序,sorted会以list形式返回,且只返回时刻组成的list

    for i in newpp:
        if m>=len(pp[i]):
            b-=sum(pp[i])
        else:
            pp[i].sort(reverse=True)
            b-=sum(pp[i][0:m])
        if b <=0:
            print(i)
            break
    if b>0:
        print("alive")

```

基本信息

- #: 44012443
- 题目: 18182
- 提交人: 2300013289
- 内存: 3764kB
- 时间: 70ms
- 语言: Python3
- 提交时间: 2024-02-29 13:21:45

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

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

```

```

        for t in range(p**2,1000001,p):
            if t<1000001:
                w[t]=False
n=int(input())
a=[int(x) for x in input().split()]
for i in range(len(a)):
    if a[i]**0.5==int(a[i]**0.5) and w[int(a[i]**0.5)] :
        print("YES")
    else:
        print("NO")

```

代码运行截图（AC代码截图，至少包含有"Accepted"）

The screenshot shows the Codeforces website interface. At the top, there's a navigation bar with links like HOME, TOP, CATALOG, CONTESTS, GYM, etc. Below that, a table lists submissions for problem 230B. The submission with ID 248709129 by user 'max12233' is highlighted with a blue background and shows a verdict of 'Accepted'. Below the table, the 'Source' tab is selected, displaying the Python code that solved the problem. The code defines a function to check if a number is a perfect square and then iterates through the input array to find the longest subarray where the sum of elements is a perfect square.

2. 1364A. XXXXX brute force/data structures/number theory/two pointers, 1200, <https://codeforces.com/problemset/problem/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=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:
        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
        else:
            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()))

```

代码运行截图（AC代码截图，至少包含有"Accepted"）



max12233 | Logout

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
249074069	Practice: max12233	1364A - 15	Python 3	Accepted	233 ms	18104 KB	2024-03-01 14:36:36	2024-03-01 14:36:36	☆	Compare

→ Source
 Copy

```

t=int(input())
for i in range(t):
    n,map(int,input().split())
    arr=list(map(int,input().split()))
    d=sum(arr)
    for u in range(n):
        if d%u:
            right=n-u
            break
        else:
            d-=arr[u]
            if u==n-1 and d%u==0:
                right=0
    for p in range(n-1,-1,-1):
        if d%u:
            left=p+1
            break
        else:
            u-=arr[p]
            if p==0 and d%u==0:
                left=0
    if left==0 and right==0:
        print(-1)
    else:
        print(max(left,right))

```

[Click to see test details](#)

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： $\text{int}(n^{**0.5})$ (这边是整型) $== n^{**0.5}$ (这边是浮点型)

如： $n=4 \rightarrow 2==2.0$, $n=3 \rightarrow 1.732...$

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}")

```

代码运行截图（AC代码截图，至少包含有"Accepted"）

OpenJudge

题目ID, 标题, 描述

2300013289 信箱 账号

CS101 / 题库

题目 排名 状态 提问

#44027712提交状态

查看 提交 统计 撤回

状态: Accepted

源代码

```
#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[
if len(grade)==0 or valid==[]:
    print('0')
else:
    a=sum(valid)/len(grade)
    print(f'{a:.2f}')
```

基本信息

#: 44027712
题目: 18176
提交人: 2300013289
内存: 4268kB
时间: 65ms
语言: Python3
提交时间: 2024-03-01 20:03:04

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

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