

# Assignment #2: 编程练习

---

Updated 0953 GMT+8 Feb 24, 2024

2024 spring, Compiled by ==祁轩宇、经济学院==

## 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：Windows 11, version 23H2

Python编程环境：VSCode 1.87.1

C/C++编程环境：

## 1. 题目

27653: Fraction类

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

思路：

## 代码

```
# 其实直接用math.gcd更好
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 __add__(self,otherself):
        newnum = self.num*otherself.den+self.den*otherself.num
        newden = self.den*otherself.den
        common=gcd(newnum,newden)
        return(Fraction(newnum//common,newden//common))

Input=list(map(int,input().split()))
x=Fraction(Input[0],Input[1])
y=Fraction(Input[2],Input[3])
print(x+y)

```

代码运行截图 ==（至少包含有"Accepted"）==

#43942590提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

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

Input=list(map(int,input().split()))
x=Fraction(Input[0],Input[1])
y=Fraction(Input[2],Input[3])
print(x+y)

```

基本信息

#: 43942590  
 题目: 27653  
 提交人: 2100015440  
 内存: 3620kB  
 时间: 20ms  
 语言: Python3  
 提交时间: 2024-02-20 21:06:02

04110: 圣诞老人的礼物-Santa Clau's Gifts

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

思路：

代码

```
n,w=map(int,input().split())
List=[]
for i in range(n):
    a,b=map(int,input().split())
    List.append((a/b,b))
List=sorted(List,key=lambda x:x[0],reverse=True)

Value=0
for i in range(n):
    Value+=List[i][0]*min(w,List[i][1])
    w-=List[i][1]
    if w<0:
        break
print(f'{Value:0.1f}')
```

代码运行截图 == （至少包含有"Accepted"） ==

#44016281提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
n,w=map(int,input().split())
List=[]
for i in range(n):
    a,b=map(int,input().split())
    List.append((a/b,b))
List=sorted(List,key=lambda x:x[0],reverse=True)

Value=0
for i in range(n):
    Value+=List[i][0]*min(w,List[i][1])
    w-=List[i][1]
    if w<0:
        break
print(f'{Value:0.1f}')
```

基本信息

#: 44016281  
题目: 04110  
提交人: 2100015440  
内存: 3644kB  
时间: 21ms  
语言: Python3  
提交时间: 2024-02-29 18:05:23

18182: 打怪兽

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

思路：

代码

```
nCases=int(input())
for i in range(nCases):
    n,m,b=map(int,input().split())
    skill={}
```

```
for j in range(n):
    t,x=input().split()
    t=int(t)
    x=float(x)
    if t in skill.keys():
        skill[t].append(x)
    else:
        skill[t]=[x]
for t in sorted(skill.keys()): # 注意对时刻进行排序
    List=sorted(skill[t],reverse=True)
    if len(List)>m:
        b-=sum(List[0:m])
    else:
        b-=sum(List)
    if b<=0:
        print(t)
        break
if b>0:
    print('alive')
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

#44019816提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
nCases=int(input())
for i in range(nCases):
    n,m,b=map(int,input().split())
    skill={}
    for j in range(n):
        t,x=input().split()
        t=int(t)
        x=float(x)
        if t in skill.keys():
            skill[t].append(x)
        else:
            skill[t]=[x]
    for t in sorted(skill.keys()): # 注意对时刻进行排序
        List=sorted(skill[t],reverse=True)
        if len(List)>m:
            b-=sum(List[0:m])
        else:
            b-=sum(List)
        if b<=0:
            print(t)
            break
    if b>0:
        print('alive')
```

基本信息

#: 44019816  
题目: 18182  
提交人: 2100015440  
内存: 3772kB  
时间: 72ms  
语言: Python3  
提交时间: 2024-02-29 22:01:17

230B. T-primes

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

思路 :

代码

```
import math

def t_primes(x,primes):
    if (math.sqrt(x)==int(math.sqrt(x))) & (primes[int(math.sqrt(x))]==1):
        return 'YES'
    else:
        return 'NO'

n=int(input())
l=list(map(int,input().split()))
limit=10**6

primes=[1]*(limit+1)
primes[0]=primes[1]=0
for i in range(2,int(math.sqrt(limit))+1):
    if primes[i]:
        for j in range(i**2,limit+1,i):
            primes[j]=0

for i in range(n):
    print(t_primes(l[i],primes))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"） ==

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
250327029	Practice: 2100015440	230B - 28	Python 3	Accepted	1060 ms	15992 KB	2024-03-09 12:38:33	2024-03-09 12:38:33		<div>Compare</div>

→ Source

Copy

```
import math

def t_primes(x,primes):
    if (math.sqrt(x)==int(math.sqrt(x))) & (primes[int(math.sqrt(x))]==1):
        return 'YES'
    else:
        return 'NO'

n=int(input())
l=list(map(int,input().split()))
limit=10**6

primes=[1]*(limit+1)
primes[0]=primes[1]=0
for i in range(2,int(math.sqrt(limit))+1):
    if primes[i]:
        for j in range(i**2,limit+1,i):
            primes[j]=0

for i in range(n):
    print(t_primes(l[i],primes))
```

[Click to see test details](#)

1364A. XXXXX

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

思路：

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：

### 代码

```
import math

# 使用Sieve of Eratosthenes算法生成素数表
def Generate_prime(limit):
    primes = [True] * (limit + 1)
    primes[0] = primes[1] = False
    for i in range(2, int(math.sqrt(limit)) + 1):
        if primes[i]:
            for j in range(i * i, limit + 1, i):
                primes[j] = False
    return primes

def T_prime(n, prime):
    if n**0.5 == int(n**0.5):
        return prime[int(n**0.5)]
    else:
        return 0

prime = Generate_prime(10**4)

m, n = map(int, input().split())

for i in range(m):
    score = list(map(int, input().split()))
    Score = 0
    for j in score:
        if T_prime(j, prime):
            Score += j
    if Score == 0:
        print(0)
    else:
        print(f'{Score/len(score):.2f}')
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

#43979906提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
import math

# 使用Sieve of Eratosthenes算法生成素数表
def Generate_prime(limit):
    primes = [True] * (limit + 1)
    primes[0] = primes[1] = False
    for i in range(2, int(math.sqrt(limit)) + 1):
        if primes[i]:
            for j in range(i * i, limit + 1, i):
                primes[j] = False
    return primes

def T_prime(n, prime):
    if n**0.5 == int(n**0.5):
        return prime[int(n**0.5)]
    else:
        return 0

prime = Generate_prime(10**4)

m, n = map(int, input().split())

for i in range(m):
    score = list(map(int, input().split()))
    Score = 0
    for j in score:
        if T_prime(j, prime):
            Score += j
    if Score == 0:
        print(0)
    else:
        print(f'{Score/len(score):.2f}')
```

基本信息  
#: 43979906  
题目: 18176  
提交人: 2100015440  
内存: 3868kB  
时间: 72ms  
语言: Python3  
提交时间: 2024-02-24 17:10:39

2. 学习总结和收获

前面3个题目还好，解题思路和语法上没有太大障碍。后面的题目对算法的时间复杂度有一定要求，花了一些时间把不同算法的复杂度想了一下还是挺有意思的。

- 1. 圣诞老人的礼物：用字典之后排序就可以，但是注意字典的键值可能会有覆盖的问题。
- 2. 打怪兽：一开始忽略了要对技能时间排序，导致WA了几次。
- 3. T\_Primes：先做了2050年成绩计算，之后做的这题，所以思路没有太大问题，不过重打一遍还是用了15-20分钟。
- 4. 2050年成绩计算：一开始TLE了几次，问了gpt之后用了新的素数表生成方法，确实时间复杂度低很多。