

# Assignment #2: 编程练习

---

Updated 0953 GMT+8 Feb 24, 2024

2024 spring, Compiled by 尹柚鑫 光华管理学院 2100015878

## 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：jupyter notebook

## 1. 题目

---

### 27653: Fraction类

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

耗时：10mins

思路：按照步骤进行计算就行

## 代码

```
#
frac=input().split(' ')
frac=[int(i) for i in frac]
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a
beishu=frac[1]*frac[3]
yinshu=gcd(frac[1],frac[3])
fenzi_1=frac[0]*frac[3]/yinshu
fenzi_2=frac[2]*frac[1]/yinshu
fenmu=beishu/yinshu
fenzi=fenzi_1+fenzi_2
new_yinshu=gcd(fenmu, fenzi)
new_fenmu=int(fenmu/new_yinshu)
new_fenzi=int(fenzi/new_yinshu)

print(f"{new_fenzi}/{new_fenmu}")
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
frac=input().split(' ')
frac=[int(i) for i in frac]
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a
beishu=frac[1]*frac[3]
yinshu=gcd(frac[1],frac[3])
fenzi_1=frac[0]*frac[3]/yinshu
fenzi_2=frac[2]*frac[1]/yinshu
fenmu=beishu/yinshu
fenzi=fenzi_1+fenzi_2
new_yinshu=gcd(fenmu, fenzi)
new_fenmu=int(fenmu/new_yinshu)
new_fenzi=int(fenzi/new_yinshu)

print(f"{new_fenzi}/{new_fenmu}")
```

基本信息

#: 43991220  
题目: 27653  
提交人: 尹柚鑫(2100015878)  
内存: 3640kB  
时间: 23ms  
语言: Python3  
提交时间: 2024-02-26 17:54:06

## 04110: 圣诞老人的礼物-Santa Clau's Gifts

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

耗时: 15mins

思路: 由于这个背包可以被取一部分, 直接按照性价比进行贪心算法

## 代码

```
#
n, max_weight = map(int, input().split())
candies = [list(map(int, input().split())) for _ in range(n)]
candies=sorted(candies,key=lambda x:x[0]/x[1],reverse=True)
```

```

number=0
weight_now=0
max_value=0
while number<n and weight_now<max_weight:
    if candies[number][1]<max_weight-weight_now:
        weight_now=weight_now+candies[number][1]
        max_value=max_value+candies[number][0]
        number=number+1
    else:
        max_value=max_value+(max_weight-weight_now)/candies[number]
        [1]*candies[number][0]
        break
print(f"{max_value:.1f}")

```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```

n, max_weight = map(int, input().split())
candies = [list(map(int, input().split())) for _ in range(n)]
candies=sorted(candies, key=lambda x:x[0]/x[1], reverse=True)
number=0
weight_now=0
max_value=0
while number<n and weight_now<max_weight:
    if candies[number][1]<max_weight-weight_now:
        weight_now=weight_now+candies[number][1]
        max_value=max_value+candies[number][0]
        number=number+1
    else:
        max_value=max_value+(max_weight-weight_now)/candies[number][1]*
        break
print(f"{max_value:.1f}")

```

基本信息

#: 43998237  
 题目: 04110  
 提交人: 尹柚鑫(2100015878)  
 内存: 7372kB  
 时间: 22ms  
 语言: Python3  
 提交时间: 2024-02-27 18:40:24

## 18182: 打怪兽

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

耗时: 30mins

思路: 自己写的代码非常复杂又报错, 这是其他同学的写法。使用带有默认值的字典存储列表, 排序后计算前两个, 非常巧妙

代码

```

#
from collections import defaultdict
for _ in range(int(input())):
    a=defaultdict(list)
    n,m,b=map(int,input().split())
    for _ in range(n):
        t,x=map(int,input().split())
        a[t].append(x)
    for t in sorted(a):
        b-=sum(sorted(a[t], reverse=True)[:m])

```

```

if b<=0:
    print(t)
    break
else:
    print('alive')

```

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

#44006653提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

from collections import defaultdict
for _ in range(int(input())):
    a=defaultdict(list)
    n,m,b=map(int,input().split())
    for _ in range(n):
        t,x=map(int,input().split())
        a[t].append(x)
    for t in sorted(a):
        b-=sum(sorted(a[t],reverse=True)[:m])
        if b<=0:
            print(t)
            break
    else:
        print('alive')

```

基本信息

#: 44006653  
 题目: 18182  
 提交人: 尹柚鑫(2100015878)  
 内存: 3752kB  
 时间: 71ms  
 语言: Python3  
 提交时间: 2024-02-28 17:22:02

## 230B. T-primes

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

耗时: 30mins

思路: 最为朴素的算素数的方法会超时, 这里是用的答案中给的筛法, 算法很巧妙, 在面对规模较大的问题时优越性

代码

```

#
a = [1]*(10**6)
a[0] = 0
for i in range(1,10**3,1):
    if a[i]==1:
        for j in range(2*i+1,10**6,i+1):
            a[j]=0

n = int(input())
l = [int(x) for x in input().split()]
for i in range(n):
    m = l[i]
    if m**0.5%1==0:
        r = int(m**0.5)
        if a[r-1]==1:
            print('YES')
        else:

```

```

        print('NO')
    else:
        print('NO')

```

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

#	When	Who	Problem	Lang	Verdict	Time	Memory
<a href="#">248843636</a>	Feb/29/2024 14:38UTC+8	YouxinYin	<a href="#">230B - T-primes</a>	Python 3	Accepted	904 ms	21800 KB

## 1364A. XXXXX

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

耗时: 30mins

思路: 和同余的问题有关, 可以考虑采用对称的双指针, 找到最大的长度, 非常适合这个题目

### 代码

```

#
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") ==

#	When	Who	Problem	Lang	Verdict	Time	Memory
<a href="#">248863795</a>	Feb/29/2024 17:39UTC+8	YouxinYin	<a href="#">1364A - XXXXX</a>	Python 3	Accepted	187 ms	17400 KB

## 18176: 2050年成绩计算

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

耗时: 15mins

思路: 和前面的素数判断类似, 使用筛法可以提高速度

代码

```
#
from math import sqrt
N = 10005

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") ==

#44016233提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
from math import sqrt
N = 10005

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

#: 44016233  
题目: 18176  
提交人: 尹柚鑫(2100015878)  
内存: 3728kB  
时间: 61ms  
语言: Python3  
提交时间: 2024-02-29 17:57:30

## 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。==

前两周有很多其他的事情，在数算上花的时间比较少，从这周开始，要多多做题，每日选做要补起来。

下面记录写这次作业遇到的很好的写法或者技巧：

1.辗转相除法计算最大公因数：

```
def gcd(a, b):  
    while b:  
        a, b = b, a % b  
    return a
```

这样写很简洁

2.按照自定的标准进行排序（匿名函数）

```
candies=sorted(candies,key=lambda x:x[0]/x[1],reverse=True)
```

3.注意使用这种有默认值的词典，并且可以设定是0，空字符串，空集合，空列表

```
from collections import defaultdict  
a=defaultdict(list)
```

`defaultdict` 是 Python 标准库中 `collections` 模块提供的一个类，它是字典（`dict`）的一个子类。`defaultdict` 允许你指定默认值的字典，这样在访问不存在的键时，会使用指定的默认值而不会引发 `KeyError`。

常见用法如下：

```
pythonCopy codefrom collections import defaultdict  
  
# 创建一个 defaultdict，指定默认值为 int 类型的 0  
my_dict = defaultdict(int)  
  
# 访问不存在的键，不会引发 KeyError，而是使用默认值 0  
value = my_dict['nonexistent_key']  
print(value) # 输出 0  
  
# 修改已有键的值  
my_dict['existing_key'] = 42  
print(my_dict['existing_key']) # 输出 42  
  
# 创建 defaultdict，指定默认值为 list 类型的空列表  
my_dict_list = defaultdict(list)  
  
# 访问不存在的键，不会引发 KeyError，而是使用默认值空列表  
value_list = my_dict_list['nonexistent_key']  
print(value_list) # 输出 []
```

```
# 添加元素到默认值是空列表的键
my_dict_list['existing_key'].append(1)
print(my_dict_list['existing_key']) # 输出 [1]
```

4.记得求素数的时候可以使用筛法，效率更高