

# Assignment #2: 编程练习

---

Updated 0953 GMT+8 Feb 24, 2024

2024 spring, Compiled by 李佳霖, 心理与认知科学学院, 2000013713

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

操作系统: macOS Sonoma

Python编程环境: VSCode

## 1. 题目

---

### 27653: Fraction类

[http://cs101.openjudge.cn/2024sp\\_routine/27653/](http://cs101.openjudge.cn/2024sp_routine/27653/)

思路: 参考教材《Python数据结构与算法分析第2版》

## 代码

```
# find the greatest common divisor
def gcd(m, n):
    while m % n != 0:
        oldm = m
        oldn = n

        m = oldn
        n = oldm % oldn
    return n

class Fraction:

    def __init__(self, num, den):
        self.num = num
        self.den = den

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

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

    def __eq__(self, other):
        firstnum = self.num * other.den
        secondnum = other.num * self.den
        return firstnum == secondnum

numbers = input().split()
f1 = Fraction(int(numbers[0]), int(numbers[1]))
f2 = Fraction(int(numbers[2]), int(numbers[3]))
print(f1 + f2)
```

代码运行截图 (至少包含有"Accepted")

状态: **Accepted**

源代码

```
# find the greatest common divisor
def gcd(m, n):
    while m % n != 0:
        oldm = m
        oldn = n

        m = oldn
        n = oldm % oldn
    return n
```

基本信息

#: 43996356  
题目: 27653  
提交人: 李佳霖2000013713  
内存: 3656kB  
时间: 22ms  
语言: Python3  
提交时间: 2024-02-27 15:25:56

## 04110: 圣诞老人的礼物-Santa Clau's Gifts

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

思路：参考了cs101的答案，先将输入转换为单位价值并添加至列表中，之后从大到小排序，根据切片的性质选取前n项和。有点类似于背包问题的处理思路。

代码

```
#
n, m = map(int, input().split())
candies = []

for i in range(n):
    v, w = map(int, input().split())
    for j in range(w):
        candies.append(v/w)

candies.sort(reverse=True)
value = sum(candies[:m])

print("{:.1f}".format(value))
```

代码运行截图 (至少包含有"Accepted")

状态: **Accepted**

源代码

```
n, m = map(int, input().split())
candies = []

for i in range(n):
    v, w = map(int, input().split())
    for j in range(w):
        candies.append(v/w)

candies.sort(reverse=True)
value = sum(candies[:m])

print("{:.1f}".format(value))
```

基本信息

#: 43997057  
题目: 04110  
提交人: 李佳霖2000013713  
内存: 3600kB  
时间: 20ms  
语言: Python3  
提交时间: 2024-02-27 16:29:23

## 18182: 打怪兽

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

思路：这道题似乎按照题目要求写就可以，可能唯一需要注意的点就是当一个时刻的技能数量多于一个时刻可以释放的技能数量时，需要按照大小排序，以最大化减少怪兽的血量。

代码

```
#
nCase=int(input())
for i in range(nCase):
    s="alive"
    t={}
    n,m,b=map(int,input().split())
    for i in range(n):
        ti,xi=map(int,input().split())
        if ti in t:
            t[ti].append(xi)
        else:
            t[ti]=[xi]
    tt = sorted(t) # 将技能按照时刻升序排序
    for ti in tt:
        if m>=len(t[ti]):
            # 如果技能数量多于一个时刻可以释放的技能数量，那么就释放所有技能
            for xi in t[ti]:
                b-=xi
```

```

        if b<=0:
            s=ti
            break
    else:
        # 如果技能数量少于一个时刻可以释放的技能数量，那么就释放伤害最高的m个技能
        t[ti].sort(reverse=True)
        for i in range(m):
            b-=t[ti][i]
        if b<=0:
            s=ti
            break
print(s)

```

代码运行截图 (AC代码截图，至少包含有"Accepted")

#44050670提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

nCase=int(input())
for i in range(nCase):
    s="alive"
    t={}
    n,m,b=map(int,input().split())
    for i in range(n):
        ti,xi=map(int,input().split())
        if ti in t:
            t[ti].append(xi)
        else:
            t[ti]=[xi]
    tt = sorted(t) # 将技能按照时刻升序排序
    for ti in tt:
        if m>=len(t[ti]):
            # 如果技能数量多于一个时刻可以释放的技能数量，那么就释放所有技能
            for xi in t[ti]:
                b-=xi
            if b<=0:
                s=ti
                break
        else:
            # 如果技能数量少于一个时刻可以释放的技能数量，那么就释放伤害最高的m个技能
            t[ti].sort(reverse=True)
            for i in range(m):
                b-=t[ti][i]
            if b<=0:
                s=ti
                break
    print(s)

```

基本信息

#: 44050670  
 题目: 18182  
 提交人: 李佳霖2000013713  
 内存: 3772kB  
 时间: 69ms  
 语言: Python3  
 提交时间: 2024-03-03 14:04:28

## 230B. T-primes

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

思路：通过答案才了解到埃氏筛法（之前从来没有听说过），大概思想便是从2到n来迭代，依次删除所有2的倍数，接下来是3，再删掉所有3的倍数。依次类推，最后剩下的最小的数m便是素数，再删除所有m的倍数。结合这道题而言，T-prime的定义是素数的基础上再加上一个可以整除N的数。因此剩下的都是开方后是整数的数。但注意这些数中还有一些数不符合T-prime的定义，例如16（除可以被4整除外，还可以被2和8整除），因此埃氏筛法可以把这些数去除。

### 代码

```
#
n = 1000000
a = [1] * n # 用于判断是否是T-prime
s = set()

#directly add the square of prime into a set, then check if num_input is in set.
for i in range(2,n):
    if a[i]: # 如果a[i]为1
        s.add(i*i) #注意是从平方之后的数开始找，避免这些数被筛掉，但这之后的倍数一定都不是T-prime
        # 例如3的平方是9，9的平方是81，81已经可以被3整除
        for j in range(i*i,n,i): # 从i*i开始，步长为i，将a中的数字为0，即i的倍数都不是T-prime
            a[j] = 0

input()
for x in map(int,input().split()):
    print(["NO","YES"][x in s]) # 通过布尔索引判断x是否在s中
```

### 代码运行截图 (AC代码截图，至少包含有"Accepted")

By JiaLinLi0822, contest: Codeforces Round 142 (Div. 2), problem: (B) T-primes, **Accepted**, #, [Copy](#)

```
n = 1000000
a = [1] * n # 用于判断是否是T-prime
s = set()

#directly add the square of prime into a set, then check if num_input is in set.
for i in range(2,n):
    if a[i]: # 如果a[i]为1
        s.add(i*i)
        for j in range(i*i,n,i): # 从i*i开始，步长为i，将a中的数字为0，即i的倍数都不是T-prime
            a[j] = 0

input()
for x in map(int,input().split()):
    print(["NO","YES"][x in s]) # 通过布尔索引判断x是否在s中
```

→Judgement Protocol

## 1364A. XXXXX

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

思路：再尝试brute force结果tle后，参考了答案的双指针的方法，提示我们有的时候写的短但并不快。

```
n = int(input())
for i in range(n):
    num, div = map(int, input().split())
    arr = list(map(int, input().split()))
    counter = 0

    prefix_sum = [0] * (num + 1)
    for i in range(num):
        prefix_sum[i + 1] = prefix_sum[i] + arr[i] # 计算前缀和

    max_length = -1
    for i in range(num):
        for j in range(i + 1, num + 1):
            subarray_sum = prefix_sum[j] - prefix_sum[i]
            if subarray_sum % div != 0:
                max_length = max(max_length, j - i)

    print(max_length)
```

## 代码

```
#
def prefix_sum(nums):
    prefix = [] # 存储前缀和
    total = 0 # 当前总和
    for num in nums:
        total += num
        prefix.append(total)
    return prefix

def suffix_sum(nums):
    suffix = [] # 存储后缀和
    total = 0 # 当前总和

    # 将nums倒序并求和
```

```

# 首先将列表反转
reversed_nums = nums[::-1]
for num in reversed_nums:
    total += num
    suffix.append(total)
# 将结果反转回来
suffix.reverse()
return suffix

t = int(input()) # 输入测试案例数量
for _ in range(t):
    N, x = map(int, input().split()) # 输入数组长度和不喜欢的数字
    a = [int(i) for i in input().split()] # 输入数组元素
    aprefix_sum = prefix_sum(a) # 计算数组的前缀和
    asuffix_sum = suffix_sum(a) # 计算数组的后缀和

    left = 0 # 左指针
    right = N - 1 # 右指针
    if right == 0:
        if a[0] % x != 0: # 如果数组仅有一个元素且不能被 x 整除
            print(1) # 输出1
        else:
            print(-1) # 否则输出-1
        continue

    leftmax = 0 # 左侧最大长度
    rightmax = 0 # 右侧最大长度
    while left != right:
        total = asuffix_sum[left] # 计算左侧子数组的后缀和
        if total % x != 0: # 如果不能被 x 整除
            leftmax = right - left + 1 # 更新左侧最大长度
            break
        else:
            left += 1 # 否则左指针右移

    left = 0
    right = N - 1
    while left != right:
        total = aprefix_sum[right] # 计算右侧子数组的前缀和
        if total % x != 0: # 如果不能被 x 整除
            rightmax = right - left + 1 # 更新右侧最大长度
            break
        else:
            right -= 1 # 否则右指针左移

    if leftmax == 0 and rightmax == 0:
        print(-1) # 如果没有满足条件的子数组, 输出-1
    else:
        print(max(leftmax, rightmax)) # 输出最大长度

```



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		
249483170	Practice: JiaLinLi0822	<a href="#">1364A</a> - 15	Python 3	Accepted	373 ms	25352 KB	2024-03-03 18:22:32	2024-03-03 18:22:32		<a href="#">Compare</a>

## 18176: 2050年成绩计算

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

思路：借用上面T-prime题目的埃氏筛选，找出所有的T-prime，然后再算平均值即可。这里因为成绩的区间比上一条要小，所以可以适当减小n值从而增加运行效率

代码

```
#
n = 10000 # 不用那么多
a = [1] * n # 用于判断是否是T-prime
s = set()

#directly add the square of prime into a set, then check if num_input is in set.
for i in range(2,n):
    if a[i]: # 如果a[i]为1
        s.add(i*i)
        for j in range(i*i,n,i): # 从i*i开始, 步长为i, 将a中的数置为0, 即i的倍数都不是T-prime
            a[j] = 0

m, n = map(int, input().split())
for i in range(m):
    score = list(map(int, input().split()))
    for j in range(len(score)):
        if score[j] not in s:
            score[j] = 0

    res = sum(score)/len(score)
    print("%.2f" % float(res) if res != 0 else "0")
    # print("{:.2f}".format(res))
    # print(round(res, 2))

# print("%.2f" % round(res, 2))
```

```
# print("%.2f" % round(res, 2))
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

#44056720提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
n = 10000
a = [1] * n # 用于判断是否是T-prime
s = set()

#directly add the square of prime into a set, then check if num_input is prime
for i in range(2,n):
    if a[i]: # 如果a[i]为1
        s.add(i*i)
        for j in range(i*i,n,i): # 从i*i开始, 步长为i, 将a中的数置为0, 即i的倍数
            a[j] = 0

m, n = map(int, input().split())
for i in range(m):
    score = list(map(int, input().split()))
    for j in range(len(score)):
        if score[j] not in s:
            score[j] = 0

res = sum(score)/len(score)
print("%.2f" % float(res) if res != 0 else "0")
# print("{:.2f}".format(res))
# print(round(res, 2))
# print("%.2f" % round(res, 2))
```

基本信息

#: 44056720

题目: 18176

提交人: 李佳霖2000013713

内存: 3876kB

时间: 50ms

语言: Python3

提交时间: 2024-03-03 17:49:57

©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。

感觉这周题目的难度比上周明显增加了一些, 尽管之中有一些题目是在之前上闫老师计概课的时候就已经做过一遍, 但再做一次包括看参考答案的时候发现仍然有很多要学习和提高的地方, 比如这次作业的背包问题, 切片的性质, 埃氏筛法以及双指针。自己也在课下在leetcode上刷一些动态规划的问题, 期待能学有所成。