

Assignment #6: 矩阵、贪心

Updated 1432 GMT+8 Oct 14, 2025

2025 fall, Compiled by 郑文萱 基础医学院

说明:

1. 解题与记录:

对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge, Codeforces, LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora <https://typoraio.cn> 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。

2. 提交安排: **提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的本人头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。

3. **延迟提交: **如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。

请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

M18211: 军备竞赛

greedy, two pointers, <http://cs101.openjudge.cn/pctbook/M18211>

思路：第一次做双指针，设置了左右各一个计数器。感觉这个题应该是很典的那种双指针，但一个要注意的点就是，只剩最后一件武器就不要卖，所以条件2里加上左小于右很有必要。

代码

```
initial=int(input())
weapons=sorted(map(int,input().split()))
left=0
right=len(weapons)-1
my_count=0
enemy_count=0
while left<=right:
    if initial>=weapons[left]:
        initial-=weapons[left]
        my_count+=1
        left+=1
    elif my_count>enemy_count and left<right:
        initial+=weapons[right]
        enemy_count+=1
        right-=1
```

```

else:
    break
print(my_count-enemy_count)

```

代码运行截图 (至少包含有"Accepted")

#50398116提交状态

查看

提交

统计

提问

状态: Accepted

源代码

```

initial=int(input())
weapons=sorted(map(int,input().split()))
left=0
right=len(weapons)-1
my_count=0
enemy_count=0
while left<=right:
    if initial>=weapons[left]:
        initial-=weapons[left]
        my_count+=1
        left+=1
    elif my_count>enemy_count and left<right:
        initial+=weapons[right]
        enemy_count+=1
        right-=1
    else:
        break
print(my_count-enemy_count)

```

基本信息

#: 50398116

题目: M18211

提交人: 25n2510305212(zx)

内存: 3616kB

时间: 20ms

语言: Python3

提交时间: 2025-10-16 17:21:50

M21554: 排队做实验

greedy, <http://cs101.openjudge.cn/pctbook/M21554/>

思路: 这个题思路很简单, 就是按时间从短到长排序, 输出之前的序号和时间总和的平均。但代码实现上出了点小问题: 在时间相同时index会返回相同的索引, 于是在请教D老师后引入了enumerate(), 学会了这个函数的用法, 还蛮好用的。

代码

```

n=int(input())
students=list(map(int,input().split()))
timelist=[]
Time=[]
sequence=[]
timing=0
for idx, time in enumerate(students, 1): # 从1开始编号
    timelist.append((time, idx))
timelist.sort()
for x in timelist:
    Time.append(x[0])
    sequence.append(x[1])
print(*sequence)
count=n
for item in Time:
    timing+=item*(count-1)

```

```
count-=1
print(f'{timing/n:.2f}')
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n=int(input())
students=list(map(int,input().split()))
timelist=[]
Time=[]
sequence=[]
timing=0
for idx, time in enumerate(students, 1): # 从1开始编号
    timelist.append((time, idx))
timelist.sort()
for x in timelist:
    Time.append(x[0])
    sequence.append(x[1])
print(*sequence)
count=n
for item in Time:
    timing+=item*(count-1)
    count-=1
print(f'{timing/n:.2f}')
```

基本信息

#: 50408073
 题目: M21554
 提交人: 25n2510305212(zx)
 内存: 3624kB
 时间: 21ms
 语言: Python3
 提交时间: 2025-10-17 09:45:27

E23555: 节省存储的矩阵乘法

implementation, matrices, <http://cs101.openjudge.cn/pctbook/E23555>

思路： 因为之前没有做过矩阵乘法的哪个题，光读题就读了二十分钟，最后理解了：当X的列号k等于Y的行号k时它们能配对贡献到Z[i,j]，最后输出所有的Z。除了理解题意，矩阵的代码实现也是一个大问题，这次本来准备用numpy，但矩阵太稀疏了所以最后并没有将他们还原，就手动输入大于零的值了。由于输出要求先行后列，所以用了sorted(Z.items())按照字典序排列。看了题解，发现他是还原成矩阵了，思路好简短好直观，虽然空间是 $O(n^2)$ ，时间是 $O(n^3)$ ，但是能完美通过。而且他用for循环根本不用按字典序排序就是按顺序的，我写了一大堆有点小丑了。

代码

```
import sys

data=sys.stdin.read().strip().split()
idx=0
n=int(data[idx])
idx+=1
m1 = int(data[idx])
idx += 1
m2 = int(data[idx])
idx += 1

X_triples = []
for _ in range(m1):
    r = int(data[idx])
    idx += 1
```

```
c = int(data[idx])
idx += 1
v = int(data[idx])
idx += 1
X_triples.append((r, c, v))

Y_by_row = {}
for _ in range(m2):
    r = int(data[idx])
    idx += 1
    c = int(data[idx])
    idx += 1
    v = int(data[idx])
    idx += 1
    if r not in Y_by_row:
        Y_by_row[r] = []
    Y_by_row[r].append((c, v))

Z = {}
for i, k, xval in X_triples:
    if k in Y_by_row:
        for j, yval in Y_by_row[k]:
            key = (i, j)
            Z[key] = Z.get(key, 0) + xval * yval

for (i, j), val in sorted(Z.items()):
    print(f"{i} {j} {val}")
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
import sys

data=sys.stdin.read().strip().split()
idx=0
n=int(data[idx])
idx+=1
m1 = int(data[idx])
idx += 1
m2 = int(data[idx])
idx += 1

X_triples = []
for _ in range(m1):
    r = int(data[idx])
    idx += 1
    c = int(data[idx])
    idx += 1
    v = int(data[idx])
    idx += 1
    X_triples.append((r, c, v))

Y_by_row = {}
for _ in range(m2):
    r = int(data[idx])
    idx += 1
    c = int(data[idx])
    idx += 1
    v = int(data[idx])
    idx += 1
    if r not in Y_by_row:
        Y_by_row[r] = []
    Y_by_row[r].append((c, v))

Z = {}
for i, k, xval in X_triples:
    if k in Y_by_row:
        for j, yval in Y_by_row[k]:
            key = (i, j)
            Z[key]=Z.get(key,0)+xval*yval
```

基本信息

#: 50414392

题目: E23555

提交人: 25n2510305212(zx)

内存: 3848kB

时间: 23ms

语言: Python3

提交时间: 2025-10-17 17:11:04

M12558: 岛屿周长

matics, <http://cs101.openjudge.cn/pctbook/M12558>

思路: 这个还好诶, 虽说标了dfs, 但不用图遍历也是可以而且很简洁的, 判断上下左右是0或是边界就可以

代码

```
m,n=map(int,input().split())
Map=[]
for _ in range(m):
    Map.append(list(map(int,input().split())))
count=0
for i in range(m):
    for j in range(n):
        if Map[i][j]==1:
            if i==0 or Map[i-1][j]==0:
                count+=1
```

```

        if j==0 or Map[i][j-1]==0:
            count+=1
        if i==m-1 or Map[i+1][j]==0:
            count+=1
        if j==n-1 or Map[i][j+1]==0:
            count+=1
    print(count)

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

m,n=map(int,input().split())
Map=[]
for _ in range(m):
    Map.append(list(map(int,input().split())))
count=0
for i in range(m):
    for j in range(n):
        if Map[i][j]==1:
            if i==0 or Map[i-1][j]==0:
                count+=1
            if j==0 or Map[i][j-1]==0:
                count+=1
            if i==m-1 or Map[i+1][j]==0:
                count+=1
            if j==n-1 or Map[i][j+1]==0:
                count+=1
print(count)

```

基本信息

#: 50416446
 题目: M12558
 提交人: 25n2510305212(zx)
 内存: 3608kB
 时间: 21ms
 语言: Python3
 提交时间: 2025-10-17 18:06:03

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

M01328: Radar Installation

greedy, <http://cs101.openjudge.cn/practice/01328/>

思路: 看了一会惊觉这是贪心题, 做着比较繁, 而且似乎最后有空行所以WA, 参考了题解在最后加上input()终于过了, 学到了 思路和校门外的树有点像, 都是比较区间

代码

```

import math
def solve(n,d,island):
    ranges=[]
    for x,y in island:
        if y>d:
            return -1
        length=math.sqrt(d**2-y**2)
        ranges.append((x-length,x+length))
    ranges.sort(key=lambda i:i[1])
    count=1
    current=ranges[0][1]
    for start,end in ranges[1:]:
        if current<start:
            count+=1

```

```

        current=end
    return count
Case=0
while True:
    try:
        n,d=map(int,input().split())
    except:
        break
    if n==0 and d==0:
        break
    Case+=1
    island=[]
    for _ in range(n):
        island.append(tuple(map(int,input().split())))
    result=solve(n,d,island)
    print(f'Case {Case}: {result}')
    input()

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

import math
def solve(n,d,island):
    ranges=[]
    for x,y in island:
        if y>d:
            return -1
        length=math.sqrt(d**2-y**2)
        ranges.append((x-length,x+length))
    ranges.sort(key=lambda i:i[1])
    count=1
    current=ranges[0][1]
    for start,end in ranges[1:]:
        if current<start:
            count+=1
            current=end
    return count
Case=0
while True:
    try:
        n,d=map(int,input().split())
    except:
        break
    if n==0 and d==0:
        break
    Case+=1
    island=[]
    for _ in range(n):
        island.append(tuple(map(int,input().split())))
    result=solve(n,d,island)
    print(f'Case {Case}: {result}')
    input()

```

基本信息

#: 50420740
 题目: 01328
 提交人: 25n2510305212(zx)
 内存: 3828kB
 时间: 47ms
 语言: Python3
 提交时间: 2025-10-17 21:05:08

545C. Woodcutters

dp, greedy, 1500, <https://codeforces.com/problemset/problem/545/C>

思路: 感谢第一个样例, 差点以为是开区间。这个题和雷达的题有点像, 但比那个简单好多, 做完雷达再做这个非常舒适。

代码

```

n=int(input())
trees=[]
for _ in range(n):
    trees.append(list(map(int, input().split())))
if n==0:
    print(0)
elif n==1:
    print(1)
else:
    count=2
    prev=trees[0][0]
    for i in range(1,n-1):
        x,h=trees[i]
        if x-h>prev:
            count+=1
            prev=x
        elif x+h<trees[i+1][0]:
            count+=1
            prev=x+h
        else:
            prev=x
    print(count)

```

代码运行截图 (至少包含有"Accepted")

344194410	Oct/17/2025 21:50 UTC+8	zwx0208	545C - Woodcutters	Python 3	Accepted	343 ms	21400 KB
-----------	-------------------------	---------	--------------------	----------	----------	--------	----------

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“计概2025fall每日选做”、CF、LeetCode、洛谷等网站题目。这周作业真是上强度了，做了两天，感觉岛屿周长、砍树和排队实验比较简单，然后矩阵和雷达都挺难，学到的很多，比如处理矩阵还有贪心的思路，对DFS、BFS做了一些了解，不过不是很熟练，下周会找相关的题来练的。另外，这周几乎是从头开始补每日选做，现在才补到9.16的，还是很有收获的，因为有些题型在作业里没有出现过。我看了24点的题解，发现没有跟我思路一样的，在此想补充一下我的方法，不用讨论正负，属于数学上过了能简化一点。

```

n=int(input())
def find(x):
    res=[]
    for l in x:
        res+= [i+[l] for i in res] #定义find(x)找到了四个数集合的所有子集
    return res
for _ in range(n):
    L=list(map(int,input().split()))
    if sum(L)<24 or sum(L)%2!=0: #先排除总合小于24以及总和为奇数的情况
        print('NO')
        continue
    subjects=find(L)

```



```
count=0
for r in subjects:
    if sum(r)==(sum(L)-24)//2:
        count+=1
if count>0:
    print('YES')
else:
    print('NO')
```

思路：设 $S=a+b+c+d$ （所有数的和）。当我们给某些数分配负号时，总和的变化是减去两倍的这些数的和。即 $S-2\times(\text{被选为负数的那些数的和})=24$ 。所以问题转化为：是否存在一个子集，其元素和为 t 。使得 $S-2t=24$ ，即 $t=(S-24)*1/2$ 。