

# assignment7

## Assignment #7: 矩阵、队列、贪心

Updated 13:15 GMT+8 Oct 21, 2025

2025 fall, Complied by 顾桂榕 基础医学院



顾桂榕 医学预科办

### 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

## M12560: 生存游戏

matrices, <http://cs101.openjudge.cn/pctbook/M12560/> 

思路：

代码

```
def cell_live(arr,x,y):
    ans = 0
    if arr[x][y-1]==1:
        ans+=1
    if arr[x][y+1]==1:
        ans+=1
    if arr[x+1][y-1]==1:
        ans+=1
    if arr[x-1][y-1]==1:
        ans+=1
    if arr[x+1][y+1]==1:
        ans+=1
    if arr[x-1][y+1]==1:
        ans+=1
    if arr[x+1][y]==1:
        ans+=1
    if arr[x-1][y]==1:
        ans+=1
    return ans

n,m = map(int,input().split())
cell_matrix = [[0]*(m+2) for _ in range(n+2)]
updated_matrix = [[0]*m for _ in range(n)]
for i in range(1,n+1):
    line = list(map(int,input().split()))
    cell_matrix[i]=[0,*line,0]
for x in range(1,n+1):
    for y in range(1,m+1):
        if cell_matrix[x][y]==0:
            if cell_live(cell_matrix,x,y)==3:
                updated_matrix[x-1][y-1]=1
```

```

        if cell_matrix[x][y]==1:
            if cell_live(cell_matrix,x,y)==2 or
cell_live(cell_matrix,x,y)==3:
                updated_matrix[x-1][y-1]=1
            else:
                updated_matrix[x-1][y-1]=0
for k in range(n):
    print(*updated_matrix[k])

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

def cell_live(arr,x,y):
    ans = 0
    if arr[x][y-1]==1:
        ans+=1
    if arr[x][y+1]==1:
        ans+=1
    if arr[x+1][y-1]==1:
        ans+=1
    if arr[x-1][y-1]==1:
        ans+=1
    if arr[x+1][y+1]==1:
        ans+=1
    if arr[x-1][y+1]==1:
        ans+=1
    if arr[x+1][y]==1:
        ans+=1
    if arr[x-1][y]==1:
        ans+=1
    return ans

n,m = map(int,input().split())
cell_matrix = [[0]*(m+2) for _ in range(n+2)]
updated_matrix = [[0]*m for _ in range(n)]
for i in range(1,n+1):
    line = list(map(int,input().split()))
    cell_matrix[i]=[0,*line,0]
for x in range(1,n+1):
    for y in range(1,m+1):
        if cell_matrix[x][y]==0:
            if cell_live(cell_matrix,x,y)==3:
                updated_matrix[x-1][y-1]=1
        if cell_matrix[x][y]==1:
            if cell_live(cell_matrix,x,y)==2 or cell_live(cell_matrix,x,y)==3:
                updated_matrix[x-1][y-1]=1
            else:
                updated_matrix[x-1][y-1]=0
for k in range(n):
    print(*updated_matrix[k])

```

基本信息

#: 50507112  
 题目: M12560  
 提交人: R.  
 内存: 30656kB  
 时间: 222ms  
 语言: PyPy3  
 提交时间: 2025-10-22 19:58:39

## M04133:垃圾炸弹

matrices, <http://cs101.openjudge.cn/pctbook/Mo4133/>

思路：

每一个方格存储点开后的值是很巧妙的  
有点类似雷达安装  
这样就不会爆内存

代码

```
d = int(input())
n = int(input())
square = [[0]*1025 for i in range(1025)]
for _ in range(n):
    x,y,k = map(int,input().split())
    for i in range(max(x-d,0),min(x+d+1,1025)):
        for j in range(max(y-d,0),min(y+d+1,1025)):
            square[i][j]+=k
max_res = max(max(l) for l in square)
num = sum(l.count(max_res) for l in square)
print(num,max_res)
```

代码运行截图 (至少包含有"Accepted")

#50510444提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 50510444  
题目: M04133  
提交人: R.  
内存: 11860kB  
时间: 61ms  
语言: Python3  
提交时间: 2025-10-22 21:31:44

源代码

```
d = int(input())
n = int(input())
square = [[0]*1025 for i in range(1025)]
for _ in range(n):
    x,y,k = map(int,input().split())
    for i in range(max(x-d,0),min(x+d+1,1025)):
        for j in range(max(y-d,0),min(y+d+1,1025)):
            square[i][j]+=k
max_res = max(max(l) for l in square)
num = sum(l.count(max_res) for l in square)
print(num,max_res)
```

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

## M02746: 约瑟夫问题

implementation, queue, <http://cs101.openjudge.cn/pctbook/M02746/> 

思路:

代码

```
while True:
    n,m = map(int,input().split())
    if n==0 and m==0:
        break
    else:
        monkey = n*[1]
        interval = m
        i = 0
        while True:
            if sum(monkey)==1:
                print(monkey.index(1)+1)
                break
            if monkey[i]==1:
                interval-=1
            if interval==0:
                monkey[i]=0
                interval = m
            if i==n-1:
                i=monkey.index(1)
            else:
                i+=1
```

## 代码运行截图 (至少包含有"Accepted")

#50511114 提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
while True:
    n,m = map(int,input().split())
    if n==0 and m==0:
        break
    else:
        monkey = n*[1]
        interval = m
        i = 0
        while True:
            if sum(monkey)==1:
                print(monkey.index(1)+1)
                break
            if monkey[i]==1:
                interval-=1
            if interval==0:
                monkey[i]=0
                interval = m
            if i==n-1:
                i=monkey.index(1)
            else:
                i+=1
```

基本信息

#: 50511114  
题目: M02746  
提交人: R.  
内存: 3604kB  
时间: 256ms  
语言: Python3  
提交时间: 2025-10-22 22:10:46

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

## M26976:摆动序列

greedy, <http://cs1o1.openjudge.cn/pctbook/M26976/> ↗

思路:

代码

```
n = int(input())
num_list = list(map(int,input().split()))
ans = 1;direction = None
for i in range(n):
    if i==n-1:
        break
    if num_list[i]-num_list[i+1]==0:
        continue
    elif num_list[i]-num_list[i+1]>0:
        if direction==1:
            continue
        else:
            ans+=1
    else:
```

```

        direction=1
    elif num_list[i]-num_list[i+1]<0:
        if direction== -1:
            continue
        else:
            ans+=1
            direction=-1
print(ans)

```

代码运行截图 (至少包含有"Accepted")

#50515282提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

n = int(input())
num_list = list(map(int, input().split()))
ans = 1; direction = None
for i in range(n):
    if i==n-1:
        break
    if num_list[i]-num_list[i+1]==0:
        continue
    elif num_list[i]-num_list[i+1]>0:
        if direction==1:
            continue
        else:
            ans+=1
            direction=1
    elif num_list[i]-num_list[i+1]<0:
        if direction== -1:
            continue
        else:
            ans+=1
            direction=-1
print(ans)

```

基本信息

#: 50515282  
 题目: M26976  
 提交人: R.  
 内存: 3860kB  
 时间: 26ms  
 语言: Python3  
 提交时间: 2025-10-23 11:06:42

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## T26971:分发糖果

greedy, <http://cs1o1.openjudge.cn/pctbook/T26971/> 

思路:

参考了题解

对于第一个和最后一个，只要满足与单边邻居比较即可

先是从左到右（去除第一个），满足如果该元素rating高的话对应糖果数都比其左边的糖果数多

比如:

left: 1 1 2

再是从右到左（去除最后一个），满足如果该元素rating高的话对应糖果数比其右边的糖果数多

这个时候

right: 2 1 1

此时不再采取存储的操作，而是算一个right值就与之前存储的left列表中的值比较

res: 2 1 2

相加既得

max操作可以满足：如果该元素rating高的话对应糖果数都比其左边右边的糖果数都多而且是从右边递推过来的

代码

```
def candies_num(n, arr):
    left = [0]*n
    res = right = 0
    for i in range(0, n):
        if i>0 and rating[i]>rating[i-1]:
            left[i] = left[i-1]+1
        else:
            left[i]=1
    for j in range(n-1, -1, -1):
        if j<n-1 and rating[j]>rating[j+1]:
            right += 1
        else:
            right = 1
        res+=max(left[j], right)
    return res
n = int(input())
rating = list(map(int, input().split()))
print(candies_num(n, rating))
```

## 代码运行截图 (至少包含有"Accepted")

#50521166提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def candies_num(n,arr):
    left = [0]*n
    res = right = 0
    for i in range(0,n):
        if i>0 and rating[i]>rating[i-1]:
            left[i] = left[i-1]+1
        else:
            left[i]=1
    for j in range(n-1,-1,-1):
        if j<n-1 and rating[j]>rating[j+1]:
            right += 1
        else:
            right = 1
        res+=max(left[j],right)
    return res
n = int(input())
rating = list(map(int,input().split()))
print(candies_num(n,rating))
```

基本信息

#: 50521166  
题目: T26971  
提交人: R.  
内存: 4948kB  
时间: 29ms  
语言: Python3  
提交时间: 2025-10-23 16:52:18

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

## 1868A. Fill in the Matrix

constructive algorithms, implementation, 1300,

<https://codeforces.com/problemset/problem/1868/A> 

思路:

本来是想随机生成排列再筛出符合条件的 (如图1)

后来想到自己举例就好

n>max\_beauty想到了轮换

另一种没想到怎么做

唉想到了都会觉得很简单

$n=1 \rightarrow$  special ② 随机排列  
 $m=1 \rightarrow$  0  $0 \times n$  行

normal  $\quad m$

0, 1, 2  
permutation  $\times 3$

4x3			$m$
0	1	2	
0	0	1	
1	2	0	
2	0	2	
3	0	2	1

无缝 ✓

0, 1, 2, 3

$n=4 \quad 0, 2,$

1	2	0
1	2	0
2	0	1

$b \times b$

1~5	1	0	3	2	-
5	3	1	0	2	4
5	2	4	1	0	3
5	4	3	2	1	0
✓	✓	✓	✓	✓	✓

① ② 3 4 5  
 ↓ ↓ 一定没有  
 0 一定没有 0 一定要有  
 0 一定要有 0 一定要有

column

$n$  个 [ ].

permutation

$[x]_{\downarrow i} \rightarrow i \text{ not in } 0 \sim i-1$   
 某列

$[x]_0 \rightarrow 0 \text{ not in }$

0 1 2 3 4

1 2 3 4 0

2 3 4 0 1

3 4 0 1 2

0 1 2 3 4

0 1 2 3 4

0 1 2 3 4

代码

```
T = int(input())

def solve():
    n, m = map(int, input().split())
```

```

if m == 1:
    print(0)
elif n > m - 1:
    print(m)
else:
    print(n + 1)

for i in range(min(m - 1, n)):
    for j in range(m):
        print((j + i) % m, end=' ')
    print()

if n > m - 1:
    for i in range(m - 1, n):
        for j in range(m):
            print(j, end=' ')
    print()

for _ in range(T):
    solve()

```

### 代码运行截图 (至少包含有"Accepted")

General										
#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
345340853	Practice: Rose-Rong	<a href="#">1868A</a> - 34	PyPy 3-64	Accepted	217 ms	8484 KB	2025-10-23 15:11:29	2025-10-23 15:11:29		<a href="#">Compare</a>

  

Source										
<a href="#">Copy</a>										
<pre>T = int(input())  def solve():     n, m = map(int, input().split())     if m == 1:         print(0)     elif n &gt; m - 1:         print(m)     else:         print(n + 1)      for i in range(min(m - 1, n)):         for j in range(m):             print((j + i) % m, end=' ')         print()      if n &gt; m - 1:         for i in range(m - 1, n):             for j in range(m):                 print(j, end=' ')         print()  for _ in range(T):     solve()</pre>										

[Click](#) to see test details

## 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。

## #50390083提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
dictionary = {}
while True:
    line = input().strip()
    if not line:
        break
    english, foreign = line.split()
    dictionary[foreign]=english
    if not line:
        break
while True:
    try:
        s = str(input())
        if s in dictionary.keys():
            print(dictionary[s])
        else:
            print('eh')
    except EOFError:
        break
```

基本信息

#: 50390083  
题目: E02804  
提交人: R.  
内存: 22540kB  
时间: 602ms  
语言: Python3  
提交时间: 2025-10-16 10:47:54

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

line 若为非空字符串则会被判定为True；反之为False

strip是很严谨的操作，防止用户多输入一些空格之类的

By Rose-Rong, contest: Codeforces Round 165 (Div. 2), problem: (A) Fancy Fence, Accepted, #, [Copy](#)

```
n = int(input())
for i in range(n):
    angle = int(input())
    if 360%(180-angle)==0:
        print(' YES')
    else:
        print(' NO')
```



```
1 x,y,gap,[],[],set()
2 for i in range(8):
3     row = list(map(int,input().split()))
4     for j in range(8):
5         if row[j]!=0:
6             x.append(i)
7             y.append(j)
8             gap.add(i-j)
9 sorted_x = sorted(x)
10 sorted_y = sorted(y)
11 queen = False
12 if sorted_x==[0,1,2,3,4,5,6,7] and sorted_y==[0,
13     if len(gap)==8:
14         queen = True
15 print('YES' if queen else 'NO')
16
```

测试输入

提交结果

历史提交

查看题解

完美通过

100% 数据通过测试 [详情](#)

运行时长: 0 ms

状态: Accepted

源代码

```

test_cases = int(input())
for _ in range(test_cases):
    num = 3
    true_list = set()
    heavy_suspicion = {}
    light_suspicion = {}
    left1, right1, result1 = input().split()
    left2, right2, result2 = input().split()
    left3, right3, result3 = input().split()
    left=[left1, left2, left3]
    right=[right1, right2, right3]
    result=[result1, result2, result3]
    for i in range(num):
        if result[i]=='even':
            for coin in left[i]:
                true_list.add(coin)
            for coin in right[i]:
                true_list.add(coin)
    for j in range(num):
        if result[j] != 'even':
            for x in range(4):
                if left[j][x] not in true_list:
                    if result[j]=='up':
                        heavy_suspicion[left[j][x]]=heavy_suspicion.get
                    else:
                        light_suspicion[left[j][x]]=light_suspicion.get
                if right[j][x] not in true_list:
                    if result[j]=='down':
                        heavy_suspicion[right[j][x]]=heavy_suspicion.get
                    else:
                        light_suspicion[right[j][x]]=light_suspicion.get
    sorted_heavy_dict = sorted(heavy_suspicion.items(), key=lambda item
    sorted_light_dict = sorted(light_suspicion.items(), key=lambda item
    if heavy_suspicion and light_suspicion:
        if sorted_heavy_dict[0][1] > sorted_light_dict[0][1]:
            print(f'{sorted_heavy_dict[0][0]} is the counterfeit coin and it is he
        else:
            print(f'{sorted_light_dict[0][0]} is the counterfeit coin and it is li
    elif heavy_suspicion:
        print(f'{sorted_heavy_dict[0][0]} is the counterfeit coin and it is heavy.')
    elif light_suspicion:
        print(f'{sorted_light_dict[0][0]} is the counterfeit coin and it is light.')

```

## 基本信息

#: 50434194  
 题目: M02692  
 提交人: R.  
 内存: 3652kB  
 时间: 22ms  
 语言: Python3  
 提交时间: 2025-10-18 17:23:04

非常好的一道brute题，原来代码其实是存了一个字典并把一些参数存在字典里，后来发现嫌疑值计算后比较是最合理的

dict.get(o,某元素)+1

key = lambda item:item[1] (value值排序)

题解中取交集和补集的操作是很简洁的

补集其实是将在轻和重中都出现的默认为真币