

Assignment #7: 矩阵、队列、贪心

Updated 1315 GMT+8 Oct 21, 2025

2025 fall, Complied by 基础医学院 郑文萱

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: 这个题和岛屿周长很像, 比较不同的是那次我判断了边界, 但这次选择在周围加保护圈, 因为判断确实有点麻烦, 还有课上学到的把周围的八个方向储存起来以便查找。

代码

```
def check(mtrx,x,y):  
    dx=[-1,-1,-1,0,0,1,1,1]  
    dy=[-1,0,1,-1,1,-1,0,1]  
    count=0  
    for i in range(8):  
        nx=x+dx[i]  
        ny=y+dy[i]  
        count+=mtrx[nx][ny]  
    if mtrx[x][y]==0 and count==3:  
        return 1  
    elif mtrx[x][y] and (count<2 or count>3):  
        return 0  
    return mtrx[x][y]  
n,m=map(int,input().split())  
mtrx=[ ]
```

```

mtrx.append([0]*(m+2))
for _ in range(n):
    mtrx.append([0]+list(map(int,input().split()))+[0])
mtrx.append([0]*(m+2))
Cell=[[0]*m for y in range(n)]
for i in range(n):
    for j in range(m):
        Cell[i][j]=check(mtrx,i+1,j+1)
for row in Cell:
    print(*row)

```

代码运行截图 (至少包含有"Accepted")

#50482275 提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

def check(mtrx,x,y):
    dx=[-1,-1,-1,0,0,1,1,1]
    dy=[-1,0,1,-1,1,-1,0,1]
    count=0
    for i in range(8):
        nx=x+dx[i]
        ny=y+dy[i]
        count+=mtrx[nx][ny]
    if mtrx[x][y]==0 and count==3:
        return 1
    elif mtrx[x][y] and (count<2 or count>3):
        return 0
    return mtrx[x][y]
n,m=map(int,input().split())
mtrx=[]
mtrx.append([0]*(m+2))
for _ in range(n):
    mtrx.append([0]+list(map(int,input().split()))+[0])
mtrx.append([0]*(m+2))
Cell=[[0]*m for y in range(n)]
for i in range(n):
    for j in range(m):
        Cell[i][j]=check(mtrx,i+1,j+1)
for row in Cell:
    print(*row)

```

基本信息

#: 50482275
 题目: M12560
 提交人: 25n2510305212(zx)
 内存: 3976kB
 时间: 39ms
 语言: Python3
 提交时间: 2025-10-21 16:12:36

M04133:垃圾炸弹

matrices, <http://cs101.openjudge.cn/pctbook/M04133/>

思路: 本来觉得像贪心题感觉很复杂, 不过既然标了矩阵没标贪心(), 就用矩阵的方法可以把广场上所有点的数据全储存下来, 就是很怕超内存, 还好并没有。这个思路稍微有点像雷达安装, 从垃圾出发, 找出来一个垃圾周围的炸弹可覆盖点然后加上垃圾数量, 取所有格子里面能炸到的最大数量以及这样的点的个数, 是一种反向思维。

代码

```

d=int(input())
n=int(input())
square=[[0]*1025 for _ 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
count=0
update=0
for i in range(1025):
    for j in range(1025):
        if square[i][j]>update:
            update=int(square[i][j])
            count=1
        elif square[i][j]==update:
            count+=1
print(count,update)

```

代码运行截图 (至少包含有"Accepted")

#50483941提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

d=int(input())
n=int(input())
square=[[0]*1025 for _ 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
count=0
update=0
for i in range(1025):
    for j in range(1025):
        if square[i][j]>update:
            update=int(square[i][j])
            count=1
        elif square[i][j]==update:
            count+=1
print(count,update)

```

基本信息

#: 50483941
 题目: M04133
 提交人: 25n2510305212(zx)
 内存: 12176kB
 时间: 234ms
 语言: Python3
 提交时间: 2025-10-21 16:55:06

M02746: 约瑟夫问题

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

思路: 这个高中做过类似的数学题, 不过用代码实现就不太用思考那么多了, 问了DS竟然还可以用递推公式算, 神奇神奇。以及用到了课上学习的queue的方法, 感觉非常方便, 而且很简单。

代码

```

def find(n,m):
    monkeys=list(range(1,n+1))
    index=0
    while len(monkeys)>1:
        index=(index+m-1)%len(monkeys)
        monkeys.pop(index)
    return monkeys[0]
while True:
    n,m=map(int,input().split())
    if m==0 and n==0:

```

```

        break
print(find(n,m))

#用deque也写了一个，这个超方便
from collections import deque
def find(n,m):
    queue=deque(range(1,n+1))
    while len(queue)>1:
        for _ in range(m-1):
            queue.append(queue.popleft())
        queue.popleft()
    return queue[0]
while True:
    n,m=map(int,input().split())
    if m==0 and n==0:
        break
    print(find(n,m))

```

代码运行截图 (至少包含有"Accepted")

#50485264提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

def find(n,m):
    monkeys=list(range(1,n+1))
    index=0
    while len(monkeys)>1:
        index=(index+m-1)%len(monkeys)
        monkeys.pop(index)
    return monkeys[0]
while True:
    n,m=map(int,input().split())
    if m==0 and n==0:
        break
    print(find(n,m))

```

基本信息

#: 50485264
 题目: M02746
 提交人: 25n2510305212(zx)
 内存: 3588kB
 时间: 21ms
 语言: Python3
 提交时间: 2025-10-21 17:48:05

©2002-2022 POJ [京ICP备20010980号-1](#)

[English](#) [帮助](#) [关于](#)

#50485438提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

from collections import deque
def find(n, m):
    queue = deque(range(1, n + 1))
    while len(queue) > 1:
        for _ in range(m - 1):
            queue.append(queue.popleft())
        queue.popleft()
    return queue[0]
while True:
    n,m=map(int,input().split())
    if m==0 and n==0:
        break
    print(find(n,m))

```

基本信息

#: 50485438
 题目: M02746
 提交人: 25n2510305212(zx)
 内存: 3596kB
 时间: 25ms
 语言: Python3
 提交时间: 2025-10-21 17:58:28

©2002-2022 POJ [京ICP备20010980号-1](#)

[English](#) [帮助](#) [关于](#)

M26976:摆动序列

greedy, <http://cs101.openjudge.cn/pctbook/M26976/>

思路：这个题只需看单调性，单调性改变时，长度就加一（若上行就取最大的，下行就取最小的即可得到子序列），代码还是很简单的。

代码

```
n=int(input())
num=list(map(int,input().split()))
if n<2:
    print(n)
else:
    length=1
    pd=0
    for i in range(1,n):
        cd=num[i]-num[i-1]
        if (pd>=0 and cd<0) or (pd<=0 and cd>0):
            length+=1
            pd=cd
    print(length)
```

代码运行截图 (至少包含有"Accepted")

#50493602提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
n=int(input())
num=list(map(int,input().split()))
if n<2:
    print(n)
else:
    length=1
    pd=0
    for i in range(1,n):
        cd=num[i]-num[i-1]
        if (pd>=0 and cd<0) or (pd<=0 and cd>0):
            length+=1
            pd=cd
    print(length)
```

基本信息

#: 50493602
 题目: M26976
 提交人: 25n2510305212(zx)
 内存: 3676kB
 时间: 22ms
 语言: Python3
 提交时间: 2025-10-22 09:58:42

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

T26971:分发糖果

greedy, <http://cs101.openjudge.cn/pctbook/T26971/>

思路：由于需要同时满足左右两边的条件，所以可以用老师上课讲的，先从左到右遍历一次再从右到左遍历一次，取两次遍历每个孩子所需糖果更多的那个数字加起来就能满足条件。代码也很简单，难的是思路(这竟然是个T)。但是这个代码debug了好久因为一会索引错，一会忘更新计数，一定要更严谨啊。

代码

```

n=int(input())
num=list(map(int,input().split()))
candy=1
candies=[1]
if n==0:
    print(0)
for i in range(1,n):
    if num[i]>num[i-1]:
        candy+=1
        candies.append(candy)
    else:
        candy=1
        candies.append(candy)
candy=1
for i in reversed(range(0,n-1)):
    if num[i]>num[i+1]:
        candy+=1
        candies[i]=max(candy,candies[i])
    else:
        candy=1
        candies[i]=max(candy,candies[i])
print(sum(candies))

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

n=int(input())
num=list(map(int,input().split()))
candy=1
candies=[1]
if n==0:
    print(0)
for i in range(1,n):
    if num[i]>num[i-1]:
        candy+=1
        candies.append(candy)
    else:
        candy=1
        candies.append(candy)
candy=1
for i in reversed(range(0,n-1)):
    if num[i]>num[i+1]:
        candy+=1
        candies[i]=max(candy,candies[i])
    else:
        candy=1
        candies[i]=max(candy,candies[i])
print(sum(candies))

```

基本信息

#: 50493868
 题目: T26971
 提交人: 25n2510305212(zx)
 内存: 4912kB
 时间: 30ms
 语言: Python3
 提交时间: 2025-10-22 10:28:25

1868A. Fill in the Matrix

constructive algorithms, implementation, 1300, <https://codeforces.com/problemset/problem/1868/A>

思路：这题看起来好复杂，翻译一团乱于是丢给DS让翻译了一下。试了几个发美观度最大为m和n-1里更小的那个，只要构造出来符合的即可，参考了题解，前几行用循环移位，后几行就固定排列，比较方便。代码实现

的时候有一点实在过于蠢，就是换行符一直不对（我之前也没作用过），后来才发现一直打的是/n！纠正了一个眼瞎误区！

代码

```
M=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(M):
    solve()
```

代码运行截图 (至少包含有"Accepted")

345120549	Oct/22/2025 12:34 UTC+8	zwx0208	1868A - Fill in the Matrix	Python 3	Accepted	359 ms	400 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。这次作业感觉没有上次难，总用时大概快5小时。摆动序列和分糖果是一种新的贪心题型很有意思，约瑟夫问题学习了queue的方法，垃圾炸弹和生存游戏是经典的矩阵查找问题，Fill in the Matrix感觉是一种考察理解能力的题，学会了行间空格隔开，整行输出后再换行的方法。最近快期中了，期中结束我将立马继续刷题！