

Assignment #7: 矩阵、队列、贪心

Updated 1315 GMT+8 Oct 21, 2025

2025 fall, Complied by 贺桢羽、心理与认知科学学院

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

M12560: 生存游戏

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

思路: 保护圈 (8分钟)

代码

```
#M12560
n,m=map(int,input().split())
li=[]
li.append([0]*(m+2))
for _ in range(n):
    li1=[0]
    li1.extend(list(map(int,input().split())))
    li1.append(0)
    li.append(li1)
li.append([0]*(m+2))
ans=[[0]*m for _ in range(n)]
for x in range(1,n+1):
    for y in range(1,m+1):
        check=li[x-1][y]+li[x-1][y-1]+li[x-1][y+1]+li[x][y-1]+li[x][y+1]+li[x+1][y-1]+li[x+1][y+1]
        [y]+li[x+1][y-1]+li[x+1][y+1]
```

```

if li[x][y]==1:
    if check<2 or check>3:
        ans[x-1][y-1]="0"
    else:
        ans[x-1][y-1]="1"
else:
    if check==3:
        ans[x-1][y-1]="1"
    else:
        ans[x-1][y-1]="0"

for i in ans:
    print(" ".join(i))

```

代码运行截图 (至少包含有"Accepted")

#50484447提交状态

查看 提交 统计 提问

状态: Accepted

基本信息	
#:	50484447
题目:	M12560
提交人:	25n2500013720
内存:	3868kB
时间:	32ms
语言:	Python3
提交时间:	2025-10-21 17:12:16

源代码

```

#M12560
n,m=map(int,input().split())
li=[]
li.append([0]*(m+2))
for _ in range(n):
    li1=[0]
    li1.extend(list(map(int,input().split())))
    li.append(li1)
li.append([0]*(m+2))
ans=[[0]*m for _ in range(n)]
for x in range(1,n+1):
    for y in range(1,m+1):
        check=li[x-1][y]+li[x-1][y-1]+li[x-1][y+1]+li[x][y-1]+li[x][y+1]
        if li[x][y]==1:
            if check<2 or check>3:
                ans[x-1][y-1]="0"
            else:
                ans[x-1][y-1]="1"
        else:
            if check==3:
                ans[x-1][y-1]="1"
            else:
                ans[x-1][y-1]="0"

for i in ans:
    print(" ".join(i))

```

©2002-2022 POI 京ICP备20010980号-1 English 帮助 关于

M04133:垃圾炸弹

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

思路: 一直超时, 甚至越优化时间越长, 于是学会了二维前缀和数组。

代码

```

#M04133
from collections import defaultdict

```

```

d=int(input())
n=int(input())
li=[]
for i in range(d):
    li.append([0]*(1025+2*d))
ma=[[0]*(1025+2*d) for _ in range(1025)]
li.extend(ma)
for i in range(d):
    li.append([0] * (1025 + 2 * d))
for _ in range(n):
    x,y,i=map(int,input().split())
    li[x+d][y+d]=i
prefix = [[0] * (1025+2*d+ 1) for _ in range(1025+2*d + 1)]
for i in range(1, 1025+2*d+ 1):
    for j in range(1, 1025+2*d + 1):
        prefix[i][j] = prefix[i-1][j] + prefix[i][j-1] - prefix[i-1][j-1] + li[i-1][j-1]
count=defaultdict(int)
for x in range(d, 1024 + d + 1):
    for y in range(d, 1024 + d + 1):
        x1, y1 = x - d, y - d
        x2, y2 = x + d, y + d
        total = prefix[x2 + 1][y2 + 1] - prefix[x1][y2 + 1] - prefix[x2 + 1][y1] + prefix[x1][y1]
        count[total] += 1
a=max(count.keys())
print(count[a],a)

```

代码运行截图 (至少包含有"Accepted")

#50486771提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

#M04133
from collections import defaultdict
d=int(input())
n=int(input())
li=[]
for i in range(d):
    li.append([0]*(1025+2*d))
ma=[[0]*(1025+2*d) for _ in range(1025)]
li.extend(ma)
for i in range(d):
    li.append([0] * (1025 + 2 * d))
for _ in range(n):
    x,y,i=map(int,input().split())
    li[x+d][y+d]=i
prefix = [[0] * (1025+2*d+ 1) for _ in range(1025+2*d + 1)]
for i in range(1, 1025+2*d+ 1):
    for j in range(1, 1025+2*d + 1):
        prefix[i][j] = prefix[i-1][j] + prefix[i][j-1] - prefix[i-1][j-1] + li[i-1][j-1]
count=defaultdict(int)
for x in range(d, 1024 + d + 1):
    for y in range(d, 1024 + d + 1):
        x1, y1 = x - d, y - d
        x2, y2 = x + d, y + d
        total = prefix[x2 + 1][y2 + 1] - prefix[x1][y2 + 1] - prefix[x2 + 1][y1] + prefix[x1][y1]
        count[total] += 1
a=max(count.keys())
print(count[a],a)

```

基本信息

#: 50486771
 题目: M04133
 提交人: 25n2500013720
 内存: 23512kB
 时间: 1507ms
 语言: Python3
 提交时间: 2025-10-21 19:16:37

M02746: 约瑟夫问题

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

思路：最开始想着用列表拼接，然后每次都从列表的第一个元素开始计算，后来发现这样会大大超时。最后选择使用了index来减少时间复杂度。

代码

```
#M02746
while 1:
    n,m=map(int,input().split())
    if n==0:
        break
    else:
        li=[i for i in range(1,n+1)]
        index=0
        while len(li)>1:
            index=(index+m-1)%len(li)
            li.pop(index)
        print(li[0])
```

代码运行截图 (至少包含有"Accepted")

#50491235提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
#M02746
while 1:
    n,m=map(int,input().split())
    if n==0:
        break
    else:
        li=[i for i in range(1,n+1)]
        index=0
        while len(li)>1:
            index=(index+m-1)%len(li)
            li.pop(index)
        print(li[0])
```

基本信息

#: 50491235
题目: M02746
提交人: 25n2500013720
内存: 3576kB
时间: 22ms
语言: Python3
提交时间: 2025-10-21 22:39:38

©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：一直RE，后来把边界情况全部处理了终于AC。大致思路就是把每个元素作为摆动序列的开头能得到的最大长度计算出来，然后挑选出最大值。但是感觉我的代码写得又臭又长。

代码

```
n=int(input())
li=list(map(int,input().split()))
li_=[]
if n==1:
    print(1)
if n==2:
    if li[0]==li[1]:
        print(1)
    else:
        print(2)
else:
    for i in range(1,n):
        li_.append(li[i]-li[i-1])
    ans_=[]
    for p in range(n-2):
        now=0
        ans_=1
        for i in range(p,n-1):
            if li_[i]*now>=0:
                now+=li_[i]
            if li_[i]*now<0:
                ans_+=1
                now=li_[i]
        if now==0:
            ans_.append(0)
        else:
            ans_.append(ans_)
    ans_.sort()
    print(ans_[-1]+1)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

n=int(input())
li=list(map(int,input().split()))
li_=[]
if n==1:
    print(1)
if n==2:
    if li[0]==li[1]:
        print(1)
    else:
        print(2)
else:
    for i in range(1,n):
        li_.append(li[i]-li[i-1])
    ans_=[]
    for p in range(n-2):
        now=0
        ans=1
        for i in range(p,n-1):
            if li_[i]*now>=0:
                now+=li_[i]
            if li_[i]*now<0:
                ans+=1
                now=li_[i]
        if now==0:
            ans_.append(0)
        else:
            ans_.append(ans)
    ans_.sort()
    print(ans_[-1]+1)

```

基本信息

#: 50496430
 题目: M26976
 提交人: 25n2500013720
 内存: 3684kB
 时间: 160ms
 语言: Python3
 提交时间: 2025-10-22 13:38:48

©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: 先从左往右扫描, 确保右边比左边高的孩子糖果数更多, 再接着从右往左扫描, 相同操作。

代码

```

n=int(input())
ratings=list(map(int,input().split()))
candies = [1] * n
for i in range(1, n):
    if ratings[i] > ratings[i - 1]:
        candies[i] = candies[i - 1] + 1
for i in range(n - 2, -1, -1):
    if ratings[i] > ratings[i + 1]:
        candies[i] = max(candies[i], candies[i + 1] + 1)
print(sum(candies))

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n=int(input())
ratings=list(map(int,input().split()))
candies = [1] * n
for i in range(1, n):
    if ratings[i] > ratings[i - 1]:
        candies[i] = candies[i - 1] + 1
for i in range(n - 2, -1, -1):
    if ratings[i] > ratings[i + 1]:
        candies[i] = max(candies[i], candies[i + 1] + 1)
print(sum(candies))
```

©2002-2022 POJ 京ICP备20010980号-1

基本信息

#: 50492291
 题目: T26971
 提交人: 25n2500013720
 内存: 4956kB
 时间: 28ms
 语言: Python3
 提交时间: 2025-10-22 00:35:51

English 帮助 关于

1868A. Fill in the Matrix

constructive algorithms, implementation, 1300, <https://codeforces.com/problemset/problem/186/8/A>

思路: 数学推理, 稍微在纸上演算一下就可以解决。 (20分钟)

代码

```
t=int(input())
for _ in range(t):
    n,m=map(int,input().split())
    if m==1:
        print(0)
        ma=[[0] for _ in range(n)]
    elif n>=m:
        print(m)
        ma=[list(range(m)) for _ in range(n-m+2)]
        for i in range(m-2):
            li=list(range(1+i,m))
            li.extend(list(range(0,i+1)))
            ma.append(li)
    elif n<m:
        print(n+1)
        ma=[list(range(m))]
        for i in range(n-1):
            li = list(range(1 + i, m))
            li.extend(list(range(0, i+1)))
            ma.append(li)
    for i in ma:
        i=list(map(str,i))
    print(" ".join(i))
```

代码运行截图 (至少包含有"Accepted")

```
t=int(input())
for _ in range(t):
    n, m=map(int, input().split())
    if m==1:
        print(0)
        ma=[[0] for _ in range(n)]
    elif n>=m:
        print(m)
        ma=[list(range(m)) for _ in range(n-m+2)]
        for i in range(m-2):
            li=list(range(1+i, m))
            li.extend(list(range(0, i+1)))
            ma.append(li)
    elif n<m:
        print(n+1)
        ma=[list(range(m))] * (n+1)
        for i in range(n-1):
            li = list(range(1 + i, m))
            li.extend(list(range(0, i+1)))
            ma.append(li)
    for i in ma:
        i=list(map(str, i))
    print(" ".join(i))
```

→Judgement Protocol

Test: #1, time: 62 ms., memory: 76 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

```
4
4 3
1 16
6 6
2 1
Output
```

```
3
0 1 2
0 1 2
0 1 2
1 2 0
```

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。

本次作业收获很大！

1. 强化了保护圈的掌握
2. 学习了二维前缀和数组，虽然有更简单的方法避免超时
3. 意识到greedy的恐怖了，真就是代码不是很难，但是需要很强的数学思维能力。

最近发现M题做不动了，开始做洛谷上的题目了。我发现洛谷真的挺好用，每个题都有很多详细的题解，虽然题解大多数都是cpp，但是其中的思想是一样的，大大拓宽了我对一道题的理解程度。最近在洛谷上刷dp题目，逐渐理解dp了，也学习了卡特兰数和其他常见递推方法。