

# Assignment #8: 田忌赛马来了

Updated 1021 GMT+8 Nov 12, 2024

2024 fall, Compiled by <mark>汤伟杰，信息管理系</mark>

说明：

- 1) 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。
- 2) 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。
- 3) 如果不能在截止前提交作业，请写明原因。

## 1. 题目

### 12558: 岛屿周长

matics, <http://cs101.openjudge.cn/practice/12558/>

思路：

用的 lake counting 和挖矿两道题的递归写法，写的很繁琐，思路是统计 1 的个数，乘 4，减去每一个 1 旁边 1 的个数（重叠的边），就是周长。

代码：

```
import sys
sys.setrecursionlimit(30000)

def f(i,j,a,b):
    s[i][j]=2
    minus_cnt,land_cnt=0,0
    dx=[0,-1,1,0]
    dy=[-1,0,0,1]
    for d in range(4):
        x=i+dx[d]
        y=j+dy[d]
        if 0<=x<=n-1 and 0<=y<=m-1:
            #额外查看四周有没有重叠边，记录下来
            if s[x][y]==1 or s[x][y]==2:
                minus_cnt+=1
            if s[x][y]==1:
                f(x,y,a,b)
                land_cnt+=1

    a.append(minus_cnt)
    b.append(land_cnt)
```

```

n,m=map(int,input().split())
s=[[int(i) for i in input().split()] for _ in range(n)]
a,b,result=[],[],0
for i in range(n):
    for j in range(m):
        if s[i][j]==1:
            f(i,j,a,b)
            #如果没有陆地就保持初始值0，否则加初始的1个
            if b:
                b.append(1)
            result+=4*sum(b)-sum(a)
            a,b=[],[]
print(result)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#### #47109634提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

import sys
sys.setrecursionlimit(30000)

def f(i,j,a,b):
    s[i][j]=2
    minus_cnt,land_cnt=0,0
    dx=[0,-1,1,0]
    dy=[-1,0,0,1]
    for d in range(4):
        x=i+dx[d]
        y=j+dy[d]
        if 0<=x<=n-1 and 0<=y<=m-1:
            #额外查看四周有没有重叠边，记录下来
            if s[x][y]==1 or s[x][y]==2:
                minus_cnt+=1
            if s[x][y]==1:
                f(x,y,a,b)
                land_cnt+=1

    a.append(minus_cnt)
    b.append(land_cnt)

n,m=map(int,input().split())
s=[[int(i) for i in input().split()] for _ in range(n)]
a,b,result=[],[],0
for i in range(n):
    for j in range(m):
        if s[i][j]==1:
            f(i,j,a,b)
            #如果没有陆地就保持初始值0，否则加初始的1个
            if b:
                b.append(1)
            result+=4*sum(b)-sum(a)
            a,b=[],[]
print(result)

```

基本信息

#: 47109634  
 题目: 12558  
 提交人: 24n2400016635  
 内存: 4012kB  
 时间: 28ms  
 语言: Python3  
 提交时间: 2024-11-12 11:04:19

## LeetCode54.螺旋矩阵

matrice, <https://leetcode.cn/problems/spiral-matrix/>

与 OJ 这个题目一样的 18106: 螺旋矩阵, <http://cs101.openjudge.cn/practice/18106>

思路:

题解的利用 有没有撞南墙 来用方向向量改变方向很巧妙, 自己写的递归很啰嗦而且调试了很长时间。

代码:

```
'''
螺旋矩阵 http://cs101.openjudge.cn/practice/18106
设置方向向量
'''
n=int(input())
protect=[[1000]*(n+2)]
mx=protect+[[1000]+[0]*n+[1000] for _ in range(n)]+protect

direction=[[0,1],[1,0],[0,-1],[-1,0]]

x,y,k=1,1,0
dx,dy=direction[0]

for j in range(1,n**2+1):
    mx[x][y]=j
    if mx[x+dx][y+dy]:
        k+=1
        dx,dy=direction[k%4]
    x+=dx
    y+=dy
for i in range(1,n+1):
    print(' '.join(map(str,mx[i][1:-1])))
#####
#####
def matrix(n,result,k,start):
    if n==1 or (n%2==1 and k+1==n):
        result[n//2][n//2]=n**2
        return result
    if n==2 or (n%2==0 and k+2==n):
        center=n//2
        d=[(-1,-1,3),(-1,0,2),(0,-1,0),(0,0,1)]
        for x,y,c in d:
            result[center+x][center+y]=n**2-c
        return result

    for i in range(k-1,n-k):
        result[k-1][i]=start+i-(k-2 if k>2 else 0)

    for j in range(k-1,n-k):
        result[j][n-k]=result[k-1][n-k-1]+j-(k-2)

    for i in range(k-1,n-k):
        result[n-k][n-i-1]=result[n-k-1][n-k]+i-(k-2)

    for j in range(k-1,n-k):
        result[n-j-1][k-1]=result[n-k][k]+j-(k-2)

    start=result[k][k-1]
```

```

        k+=1
        matrix(n,result,k,start)
        return result

n=int(input())
result=[[0]*n for _ in range(n)]
k,start=1,1
ans=matrix(n,result,k,start)
for i in ans:
    print(*i)

```

代码运行截图 ==（至少包含有"Accepted"）==



#47125378提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

def matrix(n,result,k,start):
    if n==1 or (n & 1 and k+1==n):
        result[n//2][n//2]=n**2
        return result
    if n==2 or ((not n & 1) and k+2==n):
        center=n//2
        d=[(-1,-1,3),(-1,0,2),(0,-1,0),(0,0,1)]
        for x,y,c in d:
            result[center+x][center+y]=n**2-c
        return result

    for i in range(k-1,n-k):
        result[k-1][i]=start+i-(k-2 if k>2 else 0)

    for j in range(k-1,n-k):
        result[j][n-k]=result[k-1][n-k-1]+j-(k-2)

    for i in range(k-1,n-k):
        result[n-k][n-i-1]=result[n-k-1][n-k]+i-(k-2)

    for j in range(k-1,n-k):
        result[n-j-1][k-1]=result[n-k][k]+j-(k-2)

    start=result[k][k-1]
    k+=1
    matrix(n,result,k,start)
    return result

n=int(input())
result=[[0]*n for _ in range(n)]
k,start=1,1
ans=matrix(n,result,k,start)
for i in ans:
    print(*i)

```

基本信息

#: 47125378  
 题目: 18106  
 提交人: 24n2400016635  
 内存: 3676kB  
 时间: 27ms  
 语言: Python3  
 提交时间: 2024-11-13 00:44:19

04133:垃圾炸弹

matrices, <http://cs101.openjudge.cn/practice/04133/>

思路:

将每一个垃圾的可被清除范围都填上价值，然后就只要找矩阵里面数值最大的就是可清理价值最多的，因为有些垃圾的可清楚范围会有重叠，重叠了价值就高，炸弹放在重叠的地方就好了。里面用到了 range 中用 min 和 max，以及加保护圈。

代码：

```
d=int(input())
n=int(input())
point=[];s=[[0]*1025 for _ in range(1025)]
for _ in range(n):
    x,y,z=map(int,input().split())
    point.append((x,y))
    for i in range(max(-x,-d),min(d+1,1024-x+1)):
        for j in range(max(-y,-d),min(d+1,1024-y+1)):
            s[x+i][y+j]+=z
m=0
for i in s:
    m=max(m,max(i))
cnt=0
for i in s:
    cnt+=i.count(m)
print(cnt,m)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#46617859提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
d=int(input())
n=int(input())
point=[];s=[[0]*1025 for _ in range(1025)]
for _ in range(n):
    x,y,z=map(int,input().split())
    point.append((x,y))
    for i in range(max(-x,-d),min(d+1,1024-x+1)):
        for j in range(max(-y,-d),min(d+1,1024-y+1)):
            s[x+i][y+j]+=z
m=0
for i in s:
    m=max(m,max(i))
cnt=0
for i in s:
    cnt+=i.count(m)
print(cnt,m)
```

基本信息

#: 46617859  
题目: 04133  
提交人: 24n2400016635  
内存: 11892kB  
时间: 61ms  
语言: Python3  
提交时间: 2024-10-20 16:23:13

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## LeetCode376.摆动序列

greedy, dp, <https://leetcode.cn/problems/wiggle-subsequence/>

与 OJ 这个题目一样的，26976:摆动序列, <http://cs101.openjudge.cn/routine/26976/>

思路：

第 3 个自己想出来的 dp 代码，，两个列表表示上升的数字和下降的数字，如果当前数字上升就取下降数字的最大值+1，相反同理。不过看了群里的代码突然发现最大值就是当前列表的最后一个数字，不需要取 max 了。

代码:

```
n=int(input())
s=list(map(int,input().split()))
dp=[[0]*n for _ in range(2)]
dp[0][0]=dp[1][0]=1
for i in range(1,n):
    if s[i]>s[i-1]:
        dp[0][i]=max(dp[1][j] for j in range(0,i))+1
        dp[1][i]=dp[1][i-1]
    elif s[i]<s[i-1]:
        dp[1][i]=max(dp[0][j] for j in range(0,i))+1
        dp[0][i]=dp[0][i-1]
    else:
        dp[1][i]=dp[1][i-1]
        dp[0][i]=dp[0][i-1]
print(max(dp[0][-1],dp[1][-1]))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47127686提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: [Accepted](#)

源代码

```
n=int(input())
s=list(map(int,input().split()))
dp=[[0]*n for _ in range(2)]
dp[0][0]=dp[1][0]=1
for i in range(1,n):
    if s[i]>s[i-1]:
        dp[0][i]=max(dp[1][j] for j in range(0,i))+1
        dp[1][i]=dp[1][i-1]
    elif s[i]<s[i-1]:
        dp[1][i]=max(dp[0][j] for j in range(0,i))+1
        dp[0][i]=dp[0][i-1]
    else:
        dp[1][i]=dp[1][i-1]
        dp[0][i]=dp[0][i-1]
print(max(dp[0][-1],dp[1][-1]))
```

基本信息

#: [47127686](#)  
题目: [26976](#)  
提交人: [24n2400016635](#)  
内存: 3704kB  
时间: 64ms  
语言: [Python3](#)  
提交时间: 2024-11-13 10:38:31

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## CF455A: Boredom

dp, 1500, <https://codeforces.com/contest/455/problem/A>

思路:

和篮球训练那道题很像，篮球是二维（两个列表），对于第一个列表的每个  $dp[1][i]$  位置，可选可不选，选就是  $dp[1][i]=dp[2][i-1]+height[i]$ ，不选就是  $dp[1][i]=dp[1][i-1]$ （状态没变，继承）；这道题也是，对于每个数字，选了就是  $dp[i]=dp[i-2]+s[i]*cnt$ ，不选就是  $dp[i]=dp[i-1]$ （继承）。这是因为  $dp[i]$  是由  $dp[i-1]$  和  $dp[i-2]$  选择与否决定的，并且这种递推公式也恰好符合题目的要求，选的时候自动避开了  $dp[i-1]$ 。好像有贪心的影子？

刚开始想的是  $dp[i]$  由  $dp[i-1]$  和  $dp[i+1]$  决定，但  $dp[i+1]$  首先要有  $dp[i]$  才能填充，卡了半天，，，看了 tutorial 才想明白可以换一个角度，并且递推公式显性地证明了本身的正确性。

代码:

```
from collections import defaultdict

n=int(input())
a=list(map(int,input().split()))
l=max(a)
d=defaultdict(int)
for i in a:
    d[i]+=1
dp=[0]*(l+1)
dp[0]=0
dp[1]=1*d[1]
for i in range(2,l+1):
    dp[i]=max(dp[i-2]+i*d[i],dp[i-1])
print(dp[-1])
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
291301995	Practice: twj_ink	<a href="#">455A</a> - 35	Python 3	Accepted	234 ms	15504 KB	2024-11-13 13:27:53	2024-11-13 13:27:53	★	<button>Compare</button>

→ Source Copy

```
from collections import defaultdict

n=int(input())
a=list(map(int,input().split()))
l=max(a)
d=defaultdict(int)
for i in a:
    d[i]+=1
dp=[0]*(l+1)
dp[0]=0
dp[1]=1*d[1]
for i in range(2,l+1):
    dp[i]=max(dp[i-2]+i*d[i],dp[i-1])
print(dp[-1])
```

## 02287: Tian Ji -- The Horse Racing

greedy, dfs <http://cs101.openjudge.cn/practice/02287>

思路:

自己写的贪心思路不对,看了题解写的。不能一直都用田忌的慢马对冲 king 的快马,因为这会和田忌的快马对冲 king 的慢马抵消,没收益。题解先判断慢马能不能赢,不能赢再比较快马能不能赢,还不能就只能慢对快了,想不到这个方法。。

代码:

```
while True:
    n=int(input())
    if n==0:
        break
    a=sorted(list(map(int,input().split())))
    b=sorted(list(map(int,input().split())))
    ai,aj,bi,bj,win,lose=0,n-1,0,n-1,0,0

    while ai<=aj and ai<n and aj>=0 and bi<n and bj>=0:
        if a[ai]>b[bj]:
            win+=1
```

```
        ai+=1
        bi+=1
    elif a[aj]>b[bj]:
        win+=1
        aj-=1
        bj-=1
    else:
        if a[ai]<b[bj]:
            lose+=1
            ai+=1
            bj-=1

    #         print(win,lose)
print(200*(win-lose))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47151706提交状态

[查看](#) [提交](#) [统计](#)

状态: Accepted

源代码

```
while True:
    n=int(input())
    if n==0:
        break
    a=sorted(list(map(int,input().split())))
    b=sorted(list(map(int,input().split())))
    ai,aj,bi,bj,win,lose=0,n-1,0,n-1,0,0

    while ai<=aj and ai<n and aj>=0 and bi<n and bj>=0:
        if a[ai]>b[bi]:
            win+=1
            ai+=1
            bi+=1
        elif a[aj]>b[bj]:
            win+=1
            aj-=1
            bj-=1
        else:
            if a[ai]<b[bj]:
                lose+=1
                ai+=1
                bj-=1

    #         print(win,lose)
    print(200*(win-lose))
```

基本信息

#: 47151706  
题目: 02287  
提交人: 24n2400016635  
内存: 26900kB  
时间: 161ms  
语言: PyPy3  
提交时间: 2024-11-14 11:36:51

## 2. 学习总结和收获

<mark>如果作业题目简单，有否额外练习题目，比如：OJ“计概 2024fall 每日选做”、CF、LeetCode、洛谷等网站题目。</mark>

田忌赛马、螺旋矩阵简单思路好难想到，岛屿周长自己能写但是不是最优解（直接加保护圈+找 1\*4 减去周围的 1），摆动序列和 boredom 对于 dp 的理解有帮助，关键找当前的 dp[i]到底是与之前的哪些状态有关，根据题目条件也可以考虑二维 dp 方便理解。每日选做还在补齐之前的题！