*# Assignment #8: 田忌赛马来了*

Updated 1021 GMT+8 Nov 12, 2024

2024 fall, Complied by <mark>付耀贤，信息管理系</mark>

*## 1. 题目*

*### 12558: 岛屿周⻓*

matices, http://cs101.openjudge.cn/practice/12558/

思路：

检测上下左右是否为0即可，注意边界值。

代码：

row,line=**map**(int,**input**().**split**())

mat=[**list**(**map**(int,**input**().**split**())) **for** i **in** **range**(row)]

c=0

**for** i **in** **range**(row):

**for** j **in** **range**(line):

**if** mat[i][j]==1:

**if** i==0 **or** mat[i-1][j]==0:

c+=1

**if** i==row-1 **or** mat[i+1][j]==0:

c+=1

**if** j==0 **or** mat[i][j-1]==0:

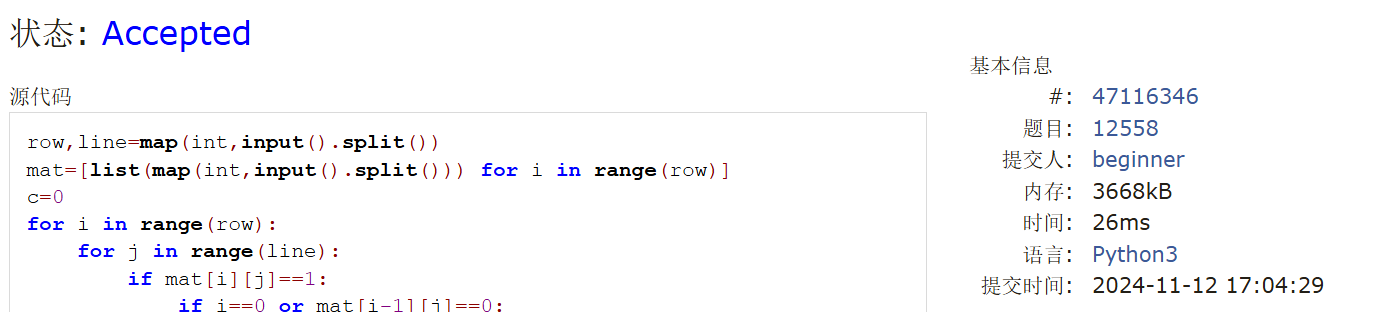
c+=1

**if** j==line-1 **or** mat[i][j+1]==0:

c+=1

**print**(c)

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



*### 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：

真的不会做，求助了AI，果然写不出来情有可原。

代码：

**def** **generate\_spiral\_matrix**(n):

matrix = [[0] \* n **for** \_ **in** **range**(n)]

*# 定义方向：右 -> 下 -> 左 -> 上*

directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]

current\_direction = 0 *# 初始化方向索引*

row, col = 0, 0 *# 初始化位置*

**for** num **in** **range**(1, n \* n + 1):

matrix[row][col] = num *# 填充当前位置*

*# 计算下一个位置*

next\_row = row + directions[current\_direction][0]

next\_col = col + directions[current\_direction][1]

*# 检查下一个位置是否越界或已被填充*

**if** (0 <= next\_row < n **and** 0 <= next\_col < n **and** matrix[next\_row][next\_col] == 0):

row, col = next\_row, next\_col *# 移动到下一个位置*

**else**:

*# 转向下一个方向*

current\_direction = (current\_direction + 1) % 4

row += directions[current\_direction][0]

col += directions[current\_direction][1]

**return** matrix

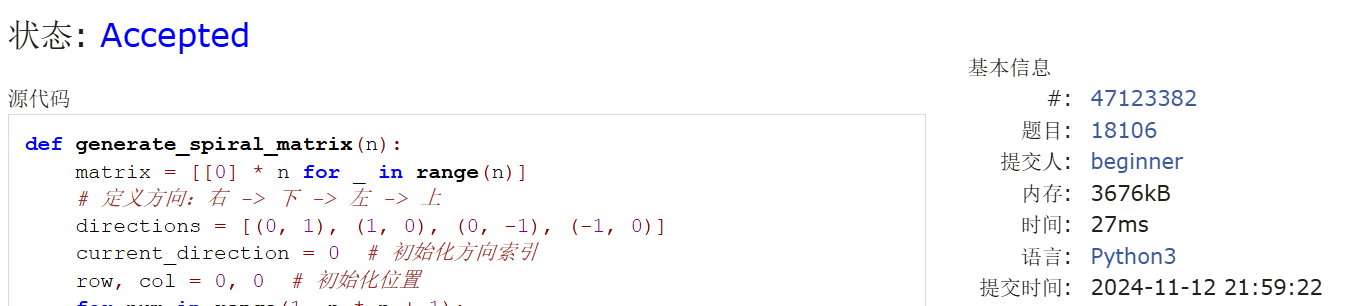
n = **int**(**input**())

spiral\_matrix = **generate\_spiral\_matrix**(n)

**for** row **in** spiral\_matrix:

**print**(" ".**join**(**map**(str, row)))

代码运行截图 ==（至少包含有"Accepted"）==



*### 04133:垃圾炸弹*

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

思路：

提交7次，想简化代码，直接从每个垃圾点出发，但发现这样不能求出所有的可能点，最后还是采用了耗时较长的写法，又看了看答案耗时较短的解法。

代码：

d=**int**(**input**())

n=**int**(**input**())

ga=[]

ma=0

pos=[]

**for** \_ **in** **range**(n):

x,y,g=**map**(int,**input**().**split**())

ga.**append**((x,y,g))

**for** x **in** **range**(0,1025):

**for** y **in** **range**(0,1025):

c=0

**for** x1,y1,g1 **in** ga:

**if** **abs**(x1 - x)<=d **and** **abs**(y1 - y) <= d:

c += g1

**if** c > ma:

ma = c

pos=[]

pos.**append**((x,y))

**elif** c == ma:

pos.**append**((x,y))

**print**(**len**(**set**(pos)),ma)

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

res = max\_point = 0

**for** i **in** **range**(0, 1025):

**for** j **in** **range**(0, 1025):

**if** square[i][j] > max\_point:

max\_point = square[i][j]

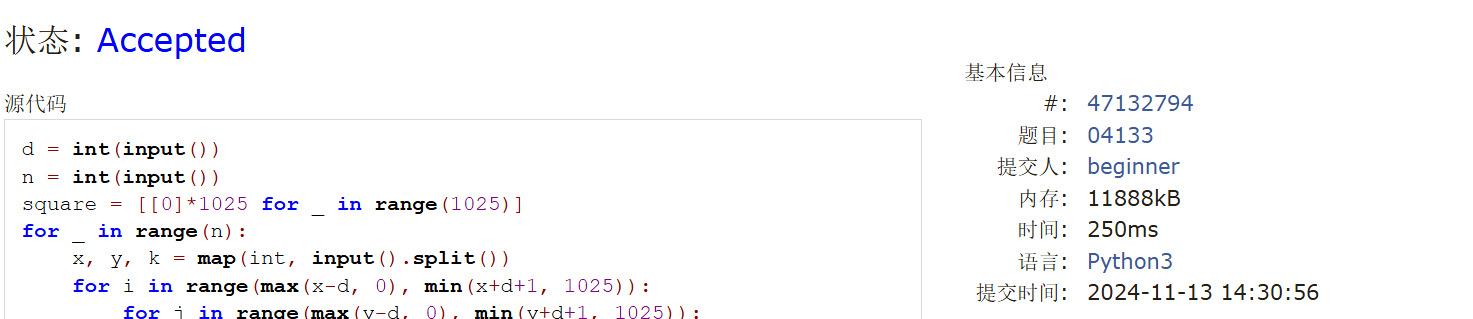
res = 1

**elif** square[i][j] == max\_point:

res += 1

**print**(res, max\_point)

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



*### 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：

最长上升子序列的升级版，思维难度高。

代码：

n=**int**(**input**())

nums=**list**(**map**(int,**input**().**split**()))

**def** **wiggleLength**(nums):

n = **len**(nums)

**if** n < 2:

**return** n

up = 1

down = 1

**for** i **in** **range**(1, n):

**if** nums[i] > nums[i - 1]:

up = down + 1

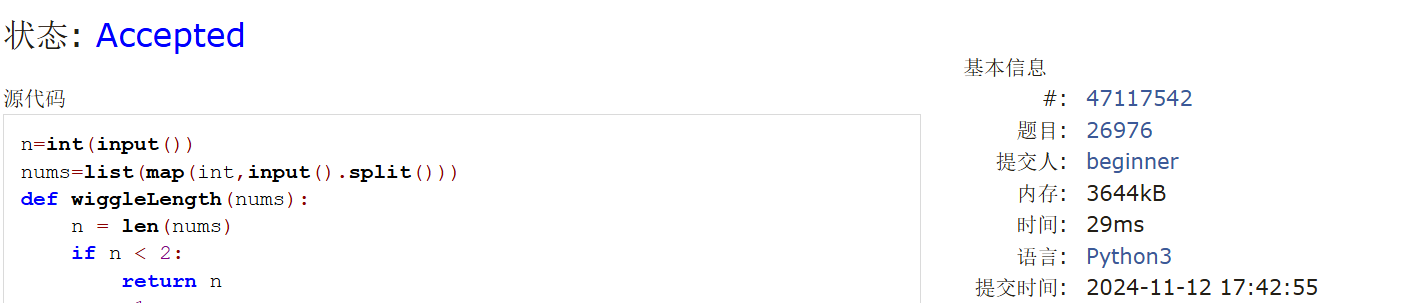
**elif** nums[i] < nums[i - 1]:

down = up + 1

**return** **max**(up, down)

**print**(**wiggleLength**(nums))

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



*### CF455A: Boredom*

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

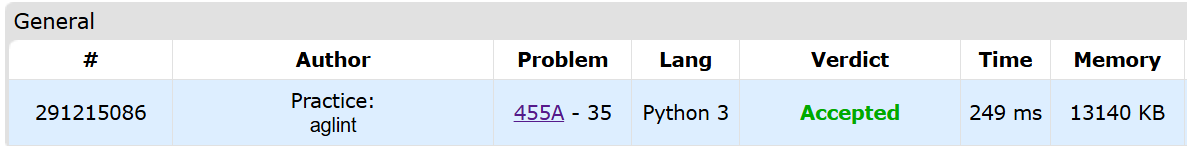
思路：

DP数组，比往常的DP要绕点弯，但最后还是想出来了嘿嘿。

代码：

from collections import Counter  
n=int(input())  
num=sorted(list(map(int,input().split())))  
count = Counter(num)  
n1=n+1  
m=100001  
dp=[0]\*m  
dp[0]=0  
dp[1]=count[1]  
for i in range(2,m):  
 dp[i]=max(dp[i-1],dp[i-2]+i\*count[i])  
print(dp[m-1])

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



*### 02287: Tian Ji -- The Horse Racing*

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

思路：

按照自己的思路一直WA，后来发现思路还是有漏洞。。看了题解的答案，好牛。

代码：

**def** **main**():

**while** True:

n = **int**(**input**())

**if** n == 0:

**break**

a = **list**(**map**(int, **input**().**split**()))

b = **list**(**map**(int, **input**().**split**()))

a.**sort**()

b.**sort**()

ans = -200 \* n

**for** j **in** **range**(n):

k = 0

**for** i **in** **range**(n):

**if** a[i] > b[(i + j) % n]:

k += 200

**elif** a[i] < b[(i + j) % n]:

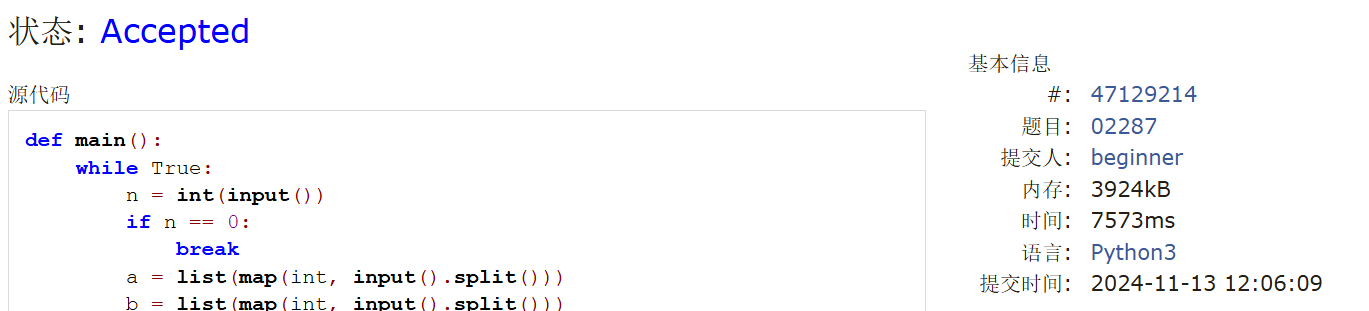
k -= 200

ans = **max**(ans, k)

**print**(ans)

**main**()

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



*## 2. 学习总结和收获*

还是觉得好难。不过这周DP已经可以自己写出来了，只是花的时间有点长。这几周体会到随着难度增大，进步的速度变慢很多，毕竟硬骨头不好啃，希望用时间慢慢磨吧。