

# Assignment #8: 田忌赛马来了

Updated 1021 GMT+8 Nov 12, 2024

2024 fall, Complied by 俞天麒 物理学院

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

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

思路:

对每一个岛屿方块，如果上下左右不是岛屿，那就分别+1.

代码:

```
n,m=map(int,input().split())
a=[[0]*(m+2) for i in range(n+2)]
for i in range(1,n+1):
    temp=list(map(int,input().split()))
    for j in range(1,m+1):
        a[i][j]=temp[j-1]
ans=0
for i in range(1,n+1):
    for j in range(1,m+1):
        if a[i][j]==1:
            if a[i-1][j]==0:
                ans+=1
            if a[i+1][j]==0:
                ans+=1
            if a[i][j-1]==0:
                ans+=1
            if a[i][j+1]==0:
                ans+=1
print(ans)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n,m=map(int,input().split())
a=[[0]*(m+2) for i in range(n+2)]
for i in range(1,n+1):
    temp=list(map(int,input().split()))
    for j in range(1,m+1):
        a[i][j]=temp[j-1]
ans=0
for i in range(1,n+1):
    for j in range(1,m+1):
        if a[i][j]==1:
            if a[i-1][j]==0:
                ans+=1
            if a[i+1][j]==0:
                ans+=1
            if a[i][j-1]==0:
                ans+=1
            if a[i][j+1]==0:
                ans+=1
print(ans)
```

©2002-2022 POJ 京ICP备20010980号-1

基本信息

#: 47109886  
题目: 12558  
提交人: 24n2400011425  
内存: 3680kB  
时间: 28ms  
语言: Python3  
提交时间: 2024-11-12 11:13:31

English 帮助 关于

## 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:

用双指针记录现在的格子，并用一个mark取4个值来标记移动方向，对应情况改变。

代码:

```
class Solution:
    def spiralOrder(self, matrix: List[List[int]]) -> List[int]:
        i,j=0,0
        ans=[]
        used=[[True]*len(matrix[0]) for i in range(len(matrix))]
        mark=1
        while True:
            mark1=True
            for k in range(len(matrix)):
                for l in range(len(matrix[0])):
                    if used[k][l]==True:
                        mark1=False
            if mark1:
                break
            ans.append(matrix[i][j])
            used[i][j]=False
            if mark==1:
                if j==len(matrix[0])-1:
                    mark=2
                    i+=1
                else:
                    if used[i][j+1]==False:
                        mark=2
                        i+=1
                    else:
```

```

        j+=1
    elif mark==2:
        if i==len(matrix)-1:
            mark=-1
            j-=1
    else:
        if used[i+1][j]==False:
            mark=-1
            j-=1
        else:
            i+=1
    elif mark==-1:
        if j==0:
            mark=-2
            i-=1
    else:
        if used[i][j-1]==False:
            mark=-2
            i-=1
        else:
            j-=1
    else:
        if used[i-1][j]==False:
            mark=1
            j+=1
        else:
            i-=1
return ans

```

代码运行截图 == (至少包含有"Accepted") ==

The screenshot shows the LeetCode platform interface. At the top, it says "通过" (Accepted) and "Practical PascalGNS 提交于 2024.11.12 13:34". Below this, there's a banner for "面向在校学生的专享优惠". On the left, there's a chart titled "执行用时分布" showing 0 ms with 100.00% completion. To its right is another chart titled "消耗内存分布" showing 16.49 MB with 40.73% completion. The main code editor window on the right contains the Python3 solution provided above. The code is well-formatted with color-coded syntax. Below the code editor, there's a "测试用例" (Test Case) section with "Case 1" and "Case 2" buttons, and a text input field containing the matrix [[1,2,3],[4,5,6],[7,8,9]]. At the bottom right of the editor window, there are "运行" (Run) and "提交" (Submit) buttons.

```

class Solution:
    def spiralOrder(self, matrix: List[List[int]]) -> List[int]:
        i,j=0,0
        ans=[]
        used=[[True]*len(matrix[0]) for i in range(len(matrix))]
        mark1=True
        while True:
            mark1=True
            for k in range(len(matrix)):
                for l in range(len(matrix[0])):
                    if used[k][l]==True:
                        mark1=False
            if mark1:
                break
            ans.append(matrix[i][j])
            used[i][j]=False
            if mark1:
                break
            if i==len(matrix)-1:
                j+=1
            elif j==len(matrix[0])-1:
                i+=1
            elif used[i+1][j]==False:
                j+=1
            elif used[i][j+1]==False:
                i+=1
            else:
                j-=1
            if i==0 and j==0:
                break

```

## 04133:垃圾炸弹

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

思路：

天才一般的思路，枚举垃圾而不是枚举每个炸弹格子

代码：

```
c=[[0]*1025 for i in range(1025)]
d=int(input())
n=int(input())
num,ans=1,0
for _ in range(n):
    x,y,m=map(int,input().split())
    for i in range(max(0,x-d),min(1025,x+d+1)):
        for j in range(max(0,y-d),min(1025,y+d+1)):
            c[i][j]+=m
for i in range(1025):
    for j in range(1025):
        if c[i][j]>ans:
            num=1
            ans=c[i][j]
        elif c[i][j]==ans:
            num+=1
print(num,ans)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
c=[[0]*1025 for i in range(1025)]
d=int(input())
n=int(input())
num,ans=1,0
for _ in range(n):
    x,y,m=map(int,input().split())
    for i in range(max(0,x-d),min(1025,x+d+1)):
        for j in range(max(0,y-d),min(1025,y+d+1)):
            c[i][j]+=m
for i in range(1025):
    for j in range(1025):
        if c[i][j]>ans:
            num=1
            ans=c[i][j]
        elif c[i][j]==ans:
            num+=1
print(num,ans)
```

基本信息

#: 46539802  
题目: 04133  
提交人: 24n2400011425  
内存: 11908kB  
时间: 240ms  
语言: Python3  
提交时间: 2024-10-17 10:06:27

©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：

其实只要枚举有几个拐点就行了，注意“鞍点”的处理，所以用一个up和down来标记之前数列的上升下降趋势。初值把这两个都设为True也很天才，这样无论是上升还是下降起手的数列都能处理

代码：

```
class Solution:
    def wiggleMaxLength(self, nums: List[int]) -> int:
        if len(nums)<=1:
            return len(nums)
        else:
            up,down=True,True
            count=1
            for i in range(1,len(nums)):
                if nums[i]>nums[i-1] and down:
                    down=False
                    up=True
                    count+=1
                if nums[i]<nums[i-1] and up:
                    down=True
                    up=False
                    count+=1
            return count
```

代码运行截图 (至少包含有"Accepted")

The screenshot shows a code editor interface with two main panes. The left pane displays the problem statement and submission details. The right pane shows the code, its execution statistics, and a test case.

**Code Editor Content:**

```
class Solution:
    def wiggleMaxLength(self, nums: List[int]) -> int:
        if len(nums)<=1:
            return len(nums)
        else:
            up,down=True,True
            count=1
            for i in range(1,len(nums)):
                if nums[i]>nums[i-1] and down:
                    down=False
                    up=True
                    count+=1
                if nums[i]<nums[i-1] and up:
                    down=True
                    up=False
                    count+=1
            return count
```

**Execution Statistics:**

- 执行用时分布: 1 ms | 击败 22.88%
- 消耗内存分布: 16.37 MB | 击败 79.85%

**Test Case:**

```
nums = [1,7,4,9,2,5]
```

## CF455A: Boredom

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

思路：

先统计一下每个数字有各几个，然后用dp，用两个数标记当前的数选还是不选时的最大值，按照转移方程书写。

代码：

```
n=int(input())
```

```

a=list(map(int,input().split()))
aco={}
nums=sorted(list(set(a)))
for num in a:
    if num in aco:
        aco[num]+=1
    else:
        aco[num]=1
#print(aco,nums)
ch,no=aco[nums[0]]*nums[0],0
for i in range(1,len(nums)):
    if nums[i]-nums[i-1]==1:
        ch,no=no+aco[nums[i]]*nums[i],max(no,ch)
    else:
        ch,no=max(no,ch)+aco[nums[i]]*nums[i],max(no,ch)
print(max(ch,no))

```

代码运行截图 (至少包含有"Accepted")

General										
#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
290312440	Practice: Chriniko	455A - 35	Python 3	Accepted	187 ms	13104 KB	2024-11-07 03:26:08	2024-11-07 03:26:08		<a href="#">Compare</a>

→ Source [Copy](#)

```

n=int(input())
a=list(map(int,input().split()))
aco={}
nums=sorted(list(set(a)))
for num in a:
    if num in aco:
        aco[num]+=1
    else:
        aco[num]=1
#print(aco,nums)
ch,no=aco[nums[0]]*nums[0],0
for i in range(1,len(nums)):
    if nums[i]-nums[i-1]==1:
        ch,no=no+aco[nums[i]]*nums[i],max(no,ch)
    else:
        ch,no=max(no,ch)+aco[nums[i]]*nums[i],max(no,ch)
print(max(ch,no))

```

[Click](#) to see test details

## 02287: Tian Ji -- The Horse Racing

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

思路：

真的难，用了好多的gpt。最开始是想到用递归，来判断自己的马从i往后和对面的马从j往后的最佳情况。但是超时，gpt告诉我可以用数组的dp来避免重复求解。然后写了一个二维数组版本，结果空间复杂度超了。然后gpt告诉我其实每一次dp只用了i+1的那个数据，所以可以只用一维数组。最后终于在空间复杂度O(n)，时间复杂度O(n^2)下完成了。回过头来审视，发现这其实就是我们认知事物的过程，而且这道题的优化过程好像和小偷背包有点像，相当于重新走了一边小偷背包的简化过程，如果转移方程能很好的判断出来，或许能一眼盯出最简单的办法。看来功力还是不够

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())))

```

```

pre=[0]*(n+1)
for i in range(n-1,-1,-1):
    for j in range(n):
        if a[i]<b[j]:
            pre[j]=-1+pre[j]
        elif a[i]>b[j]:
            pre[j]=1+pre[j+1]
        else:
            pre[j]=max(-1+pre[j],pre[j+1])
print(pre[0]*200)

```

代码运行截图 (至少包含有"Accepted")

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())))
    pre=[0]*(n+1)
    for i in range(n-1,-1,-1):
        for j in range(n):
            if a[i]<b[j]:
                pre[j]=-1+pre[j]
            elif a[i]>b[j]:
                pre[j]=1+pre[j+1]
            else:
                pre[j]=max(-1+pre[j],pre[j+1])
    print(pre[0]*200)

```

基本信息

#: 47112807  
 题目: 02287  
 提交人: 24n2400011425  
 内存: 3904kB  
 时间: 9745ms  
 语言: Python3  
 提交时间: 2024-11-12 14:10:15

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## 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2024fall每日选做”、CF、LeetCode、洛谷等网站题目。

从最后一题来看，自己的转移方程书写还是不是很熟练，一遇上不是那么模板的题就不会了。gpt是个很好的助手，他能在不直接告诉我答案的情况下一步一步引导我一步一步优化自己的代码，跟着走一遍的收获是很大的。