

Assignment #8: 田忌赛马来了

Updated 1021 GMT+8 Nov 12, 2024

2024 fall, Compiled by <mark>同学的姓名、院系</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：先找出所有为 1 的格子，再讨论格子的四条边（关键就是周长一定是在最外面的那部分，所以说只需要看与 0 的交界即可），注意还要讨论边界。（30min）

代码：

```
n, m = map(int, input().split())
map_list = []
for i in range(n):
    map_list.append(list(map(int, input().split())))
perimeter = 0
for i in range(n):
    for j in range(m):
        if map_list[i][j] == 1:
            if i == 0 or map_list[i - 1][j] == 0:
                perimeter += 1
```

```
        if i == n - 1 or map_list[i + 1][j] == 0:
            perimeter += 1

        if j == 0 or map_list[i][j - 1] == 0:
            perimeter += 1

        if j == m - 1 or map_list[i][j + 1] == 0:
            perimeter += 1

print(perimeter)
```

代码运行截图 [<mark>](#)（至少包含有"Accepted"）[</mark>](#)

#47251039提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: [Accepted](#)

源代码

```
n, m = map(int, input().split())
map_list = []
for i in range(n):
    map_list.append(list(map(int, input().split())))
perimeter = 0
for i in range(n):
    for j in range(m):
        if map_list[i][j] == 1:
            if i == 0 or map_list[i - 1][j] == 0:
                perimeter += 1
            if i == n - 1 or map_list[i + 1][j] == 0:
                perimeter += 1
            if j == 0 or map_list[i][j - 1] == 0:
                perimeter += 1
            if j == m - 1 or map_list[i][j + 1] == 0:
                perimeter += 1
print(perimeter)
```

基本信息

#: 47251039
题目: 12558
提交人: EuphoriaJ
内存: 3664kB
时间: 33ms
语言: Python3
提交时间: 2024-11-18 22:39:11

©2002-2022 POJ 京ICP备20010980号-1

[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: 先输入矩阵并把每个值都赋值为 0, 然后用 `directions` 来定义四个方向, 从 `num_to_fill=0` 开始向右填充, 而当超出这个方阵的范围时, 改变 `direction_index` 的值 (属实没想到), 再继续填充即可 (另一个没想到的地方是 `if` 条件句中不仅有对于边界情况的讨论, 还要讨论遇到之前已经填充过的数字, 这一点又遗漏了) (1h)

代码:

```
n = int(input())
result = [[0] * n for _ in range(n)]
directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]
direction_index = 0
current_row, current_col = 0, 0
num_to_fill = 1
while num_to_fill <= n ** 2:
```

```
    result[current_row][current_col] = num_to_fill
    num_to_fill += 1

    next_row = current_row +
directions[direction_index][0]

    next_col = current_col +
directions[direction_index][1]

    if (next_row < 0 or next_row >= n or next_col < 0 or
next_col >= n or

        result[next_row][next_col] != 0):

        direction_index = (direction_index + 1) % 4
```

```
    current_row += directions[direction_index][0]
    current_col += directions[direction_index][1]
```

```
for row in result:
    print(" ".join(map(str, row)))
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
n = int(input())

result = [[0] * n for _ in range(n)]
directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]
direction_index = 0
current_row, current_col = 0, 0
num_to_fill = 1

while num_to_fill <= n ** 2:
    result[current_row][current_col] = num_to_fill
    num_to_fill += 1
    next_row = current_row + directions[direction_index][0]
    next_col = current_col + directions[direction_index][1]
    if (next_row < 0 or next_row >= n or next_col < 0 or next_col >= n or
        result[next_row][next_col] != 0):
        direction_index = (direction_index + 1) % 4

    current_row += directions[direction_index][0]
    current_col += directions[direction_index][1]

for row in result:
    print(" ".join(map(str, row)))
```

基本信息

#: 47250427
题目: 18106
提交人: EuphoriaJ
内存: 3632kB
时间: 31ms
语言: Python3
提交时间: 2024-11-18 22:05:51

04133:垃圾炸弹

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

思路: 将每一组 x, y, i 输入列表中, 并讨论每一个 (x, y) 所能装载的垃圾数量, 并输入 `all_numbers` 列表中, 最后再输出 `all_numbers` 列表中的最大值即可(25min)

代码:

```
d=int(input())
n=int(input())
rubbish=[]
for _ in range(n):
    x,y,i=map(int,input().split())
    rubbish.append((x,y,i))
all_numbers=[]
for x1 in range(1025):
    for y1 in range(1025):
        r=0
        for x,y,i in rubbish:
            if abs(x-x1)<=d and abs(y-y1)<=d:
                r+=i
        all_numbers.append(r)
print(all_numbers.count(max(all_numbers)),max(all_numbers))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47259773提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
d=int(input())
n=int(input())
rubbish=[]
for _ in range(n):
    x,y,i=map(int,input().split())
    rubbish.append((x,y,i))
all_numbers=[]
for x1 in range(1025):
    for y1 in range(1025):
        r=0
        for x,y,i in rubbish:
            if abs(x-x1)<=d and abs(y-y1)<=d:
                r+=i
        all_numbers.append(r)
print(all_numbers.count(max(all_numbers)),max(all_numbers))
```

基本信息

#: 47259773
题目: 04133
提交人: EuphoriaJ
内存: 11952kB
时间: 1321ms
语言: Python3
提交时间: 2024-11-19 13:16:33

©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 0J 这个题目一样的，26976:摆动序列，

<http://cs101.openjudge.cn/routine/26976/>

思路：先命名两个数表 up 与 down 分别表示最后两个数的差值为正数（负数）的摆动序列的最长长度，然后写出 up(i) 的递推表达式，最后比较 up 与 down 中的最大数字并输出即可(没做出来，看的 ai)

代码:

```
n = int(input())  
nums = list(map(int, input().split()))  
up = [1] * n  
down = [1] * n
```

```
for i in range(1, n):  
    for j in range(i):  
        if nums[i] > nums[j]:  
            up[i] = max(up[i], down[j] + 1)  
        elif nums[i] < nums[j]:  
            down[i] = max(down[i], up[j] + 1)  
print(max(max(up), max(down)))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
n = int(input())
nums = list(map(int, input().split()))
up = [1] * n
down = [1] * n

for i in range(1, n):
    for j in range(i):
        if nums[i] > nums[j]:
            up[i] = max(up[i], down[j] + 1)
        elif nums[i] < nums[j]:
            down[i] = max(down[i], up[j] + 1)
print(max(max(up), max(down)))
```

基本信息

#: 47251120
题目: 26976
提交人: EuphoriaJ
内存: 3660kB
时间: 220ms
语言: Python3
提交时间: 2024-11-18 22:45:55

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 𠂇

CF455A: Boredom

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

思路：笨人先是创建的列表 dp 的方式来操作但超时了，这种方法与其类似，初始化两个变量 prev_0 和 prev_1 为 0 用于保存中间计算结果。然后不断更新 prev_0 与 prev_1 的值，最后输出（但感觉还是比 dp 难理解）（1.5h）

代码：

```
n=int(input())
a_list=list(map(int,input().split()))
b=max(a_list)
cnt=[0]*(b+1)
for item in a_list:
    cnt[item]+=1
prev_0=0
prev_1=0
for a in range(1,b+1):
    prev_0,prev_1=max(prev_0,prev_1),a*cnt[a]+prev_0
print(max(prev_0,prev_1))
```

代码运行截图 <mark>（至少包含有"Accepted"） </mark>

#	When	Who	Problem	Lang	Verdict	Time	Memory
292282762	Nov/19/2024 14:33 ^{UTC+8}	EuphoriaJ	A - Boredom	Python 3	Accepted	140 ms	13100 KB
292282190	Nov/19/2024 14:27 ^{UTC+8}	EuphoriaJ	A - Boredom	Python 3	Runtime error on test 7	62 ms	0 KB

02287: Tian Ji -- The Horse Racing

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

思路：借鉴了同学的思路，`bisect` 函数是真的妙啊，先把胜利的情况讨论出来，然后再讨论平局（最巧妙的！），即用第二轮列表来讨论平局情况即可（没做出来，参考的同学的方法）

代码：

```
from bisect import bisect_left

while True:

    n=int(input())

    if n==0:

        break

    tians=[int(x) for x in input().split()]

    kings=[int(x) for x in input().split()]

    tians.sort()

    kings.sort()

    ties=[]

    count=0

    for head in range(n):

        a=bisect_left(kings,tians[head])

        if a:

            count+=1
```

```
kings.pop(a-1)

else:

    ties.append(tians[head])

p2=0

for i in range(len(ties)):

    if ties[i]<kings[p2]:

        count-=1

    else:

        p2+=1

print(count*200)
```

代码运行截图 [<mark>](#)（至少包含有"Accepted"）[</mark>](#)

#47259550提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
from bisect import bisect_left
while True:
    n=int(input())
    if n==0:
        break
    tians=[int(x) for x in input().split()]
    kings=[int(x) for x in input().split()]
    tians.sort()
    kings.sort()
    ties=[]
    count=0
    for head in range(n):
        a=bisect_left(kings,tians[head])
        if a:
            count+=1
            kings.pop(a-1)
        else:
            ties.append(tians[head])
    p2=0
    for i in range(len(ties)):
        if ties[i]<kings[p2]:
            count-=1
```

基本信息

#: 47259550
题目: 02287
提交人: EuphoriaJ
内存: 3884kB
时间: 74ms
语言: Python3
提交时间: 2024-11-19 13:05:30

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>

感觉还是进步了很多，第一个和第三个题自己能写出来，螺旋矩阵卡在了设置方向上，boredom 超时了（比起之前全是 WA，感觉超时更让人崩溃，因为很难想出来更好的办法）

，感觉田忌赛马如果用 `bisect` 不是很难，但摆动数列那道题才是最难理解的，笨人脑子要想爆炸了 qwq，下周继续加油！