

Assignment #A: 矩阵和动态规划

Updated 1406 GMT+8 Nov 14, 2023

2023 fall, Compiled by 刘思瑞 2100017810

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, 学号), 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、作业评论有md或者doc。
- 3) 如果不能在截止前提交作业, 请写明原因。

编程环境

操作系统: Windows 11 22H2 22621.2283

Python编程环境: Visual Studio (1.82.2); python 3.11.3

C/C++编程环境: 无

1. 必做题目

OJ12558: 岛屿周长

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

思路:

补齐一圈, 直接遍历

代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  n ,m =map(int,input().split())
5  matrix = [[0]*(m+2)]
6  for i in range(n):
7      matrix.append([0]+list(map(int,input().split()))+[0])
8  matrix.append([0]*(m+2))
9  c = 0
10 for i in range(1,n+1):
11     for j in range(1,m+1):
12         if matrix[i][j]:
13             c += 4-(matrix[i+1][j]+matrix[i][j+1]+matrix[i][j-1]+matrix[i-1][j])
```

代码运行截图

刘思瑞 2100017810

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
n,m=map(int,input().split())
matrix=[0]*(m+2)
for i in range(n):
    matrix.append([0]+list(map(int,input().split()))+[0])
matrix.append([0]*(m+2))
c=0
for i in range(1,n+1):
    for j in range(1,m+1):
        if matrix[i][j]:
            c+=4-(matrix[i+1][j]+matrix[i][j+1]+matrix[i][j-1]+matrix[i-1][j])
print(c)
```

OJ02760: 数字三角形

dp, <http://cs101.openjudge.cn/practice/02760/>

思路:

从下层遍历

代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  n=int(input())
5  matrix=[]
6  maxx=0
7  for i in range(n):
8      matrix.append(list(map(int,input().split())))
9  temp=matrix[-1]
10 for i in range(n-1,0,-1):
11     ttemp=[]
12     for j in range(i):
13         ttemp.append(max(matrix[i-1][j]+temp[j],matrix[i-1][j]+temp[j+1]))
14     temp=ttemp
15 print(temp[0])
```

代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
n = int(input())
matrix = []
maxx = 0
for i in range(n):
    matrix.append(list(map(int,input().split())))
temp = matrix[-1]
for i in range(n-1,0,-1):
    ttemp = []
    for j in range(i):
        ttemp.append(max(matrix[i-1][j]+temp[j],matrix[i-1][j]+temp[j+1]))
    temp = ttemp
print(temp[0])
```

OJ02773: 采药

dp, <http://cs101.openjudge.cn/practice/02773>

思路:

背包问题

代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  T,M = map(int,input().split())
5  li = []
6  value = 0
7  for i in range(M):
8      t,m = map(int,input().split())
9      li.append([t,m])
10 value = [[0]*(li[0][0]) + [li[0][1]]*(T+1-li[0][0])]
11 for i in range(len(li)-1):
12     value.append([0]*(T+1))
13 for i in range(1,len(li)):
14     for j in range(1,T+1):
15         if j >= li[i][0]:
16             value[i][j] = max(value[i-1][j],value[i-1][j-li[i][0]]+li[i][1])
17         else:
18             value[i][j] = value[i-1][j]
19 print(value[-1][-1])
```

代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
T,M = map(int,input().split())
li = []
value = 0
for i in range(M):
    t,m = map(int,input().split())
    li.append([t,m])
value = [[0]*(li[0][0]) + [li[0][1]]*(T+1-li[0][0])]
for i in range(len(li)-1):
    value.append([0]*(T+1))
for i in range(1,len(li)):
    for j in range(1,T+1):
        if j >= li[i][0]:
            value[i][j] = max(value[i-1][j],value[i-1][j-li[i][0]]+li[i][1])
        else:
            value[i][j] = value[i-1][j]
print(value[-1][-1])
```

OJ18106: 螺旋矩阵

matrices, <http://cs101.openjudge.cn/practice/18106/>

这个题目技巧性较强，可以看题解记住。

思路：

设定一个方向，注意每两次换向会改变长度

代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  def change(sign):
5      if sign == [0,-1]:
6          return [-1,0]
7      if sign == [-1,0]:
8          return [0,1]
9      if sign == [0,1]:
10         return [1,0]
11     if sign == [1,0]:
12         return [0,-1]
13 n = int(input())
14 m = []
15 for i in range(n):
16     m.append([0]*n)
17 for i in range(n):
18     m[0][i] = i+1
19 i = n
```

```
20 x = n-1
21 y = 0
22 sign = [0,-1]
23 for j in range(n-1,0,-1):
24     for k in range(j):
25         i+=1
26         x , y = x+sign[0],y-sign[1]
27         m[y][x] = i
28     sign = change(sign)
29     for k in range(j):
30         i+=1
31         x , y = x+sign[0],y-sign[1]
32         m[y][x] = i
33     sign = change(sign)
34 for i in m:
35     for j in i:
36         print(j,end=' ')
37     print('')
```

代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
def change(sign):
    if sign == [0,-1]:
        return [-1,0]
    if sign == [-1,0]:
        return [0,1]
    if sign == [0,1]:
        return [1,0]
    if sign == [1,0]:
        return [0,-1]
n = int(input())
m = []
for i in range(n):
    m.append([0]*n)
for i in range(n):
    m[0][i] = i+1
i = n
x = n-1
y = 0
sign = [0,-1]
for j in range(n-1,0,-1):
    for k in range(j):
        i+=1
        x , y = x+sign[0],y-sign[1]
        m[y][x] = i
    sign = change(sign)
    for k in range(j):
        i+=1
        x , y = x+sign[0],y-sign[1]
        m[y][x] = i
    sign = change(sign)
for i in m:
    for j in i:
        print(j,end=' ')
```

2. 选做题目

如果耗时太长，直接看解题思路，或者源码

CF189A: Cut Ribbon

brute force/dp, 1300, <https://codeforces.com/problemset/problem/189/A>

思路:

代码

```
1 #  
2
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

CF455A: Boredom

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

思路:

代码

```
1 #  
2
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

3. 学习总结和收获

期中周刚刚结束, 以前感觉dp的题目有思路但是不好动笔, 现在我发现最关键的是状态转移方程, 只要能从数学上得到方程, 就可以知道很关键的遍历顺序了。