

## 1. 题目

18160: 最大连通域面积

dfs similar, <http://cs101.openjudge.cn/practice/18160>

思路:

代码:

```
k=int(input())
def dfs(x,y,l):
    if l[x][y]=='.':
        return 0
    else:
        l[x][y]='.'
        return
1+dfs(x-1,y-1,l)+dfs(x-1,y,l)+dfs(x-1,y+1,l)+dfs(x,y-1,l)+dfs(x,y+1,l)+dfs(x+1,y-1,l)+dfs(
x+1,y,l)+dfs(x+1,y+1,l)

for i in range(k):
    n,m=map(int,input().split())
    l=[]
    l.append(['.']*(2+m))
    for j in range(n):
        a=input()
        la=['.']+a+['.']
        l.append(la)
    l.append(['.']*(2+m))
    ma=0
    for p in range(1,n+1):
        for q in range(1,m+1):
            if l[p][q]=='W':
                ml=dfs(p,q,l)
                if ml>ma:
                    ma=ml
    print(ma)
```

代码运行截图 (至少包含有"Accepted")



19930: 寻宝

bfs, <http://cs101.openjudge.cn/practice/19930>

思路：

代码：

```
n,m=map(int,input().split())
```

$$I = []$$

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

```
l.append(list(map(int,input().split())))
```

```
trace=[[0,0,0]]
```

```
step=[[1,0],[-1,0],[0,1],[0,-1]]
```

```
while len(trace)>0:
```

```
if l[0][0]==1:
```

```
print(0)
```

break

```
for i in range(4):
```

```
x=trace[0][0]+step[i][0]
```

```
y=trace[0][1]+step[i][1]
```

```
s=trace[0][2]+1
```

if  $0 \leq x < n$  and  $0 \leq y < m$  and  $l[x][y] \neq 2$ :

```
if l[x][y]==1:
```

```
print(s)
```

break

```
else:
```

```
trace.append([x,y,s])
```

$$l[x][y]=2$$

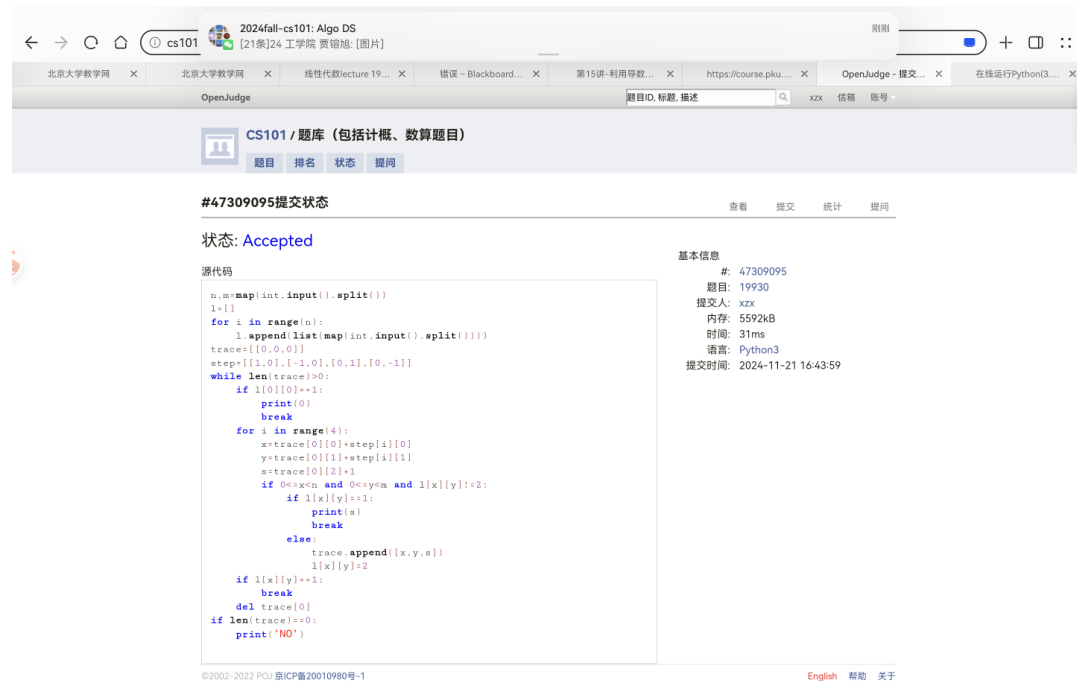
```
if l[x][y]==1:
```

```

        break
    del trace[0]
if len(trace)==0:
    print('NO')

```

代码运行截图 ==（至少包含有"Accepted"）==



04123: 马走日

dfs, <http://cs101.openjudge.cn/practice/04123>

思路:

代码:

```

k=int(input())
ct=0
def fps(x,y,l,s):
    global step,su,ct
    if s==su:
        ct+=1
    for i in range(8):
        nx=x+step[i][0]
        ny=step[i][1]+y
        ns=s+1
        if 0<=nx<n and 0<=ny<m and l[nx][ny]==0:
            l[nx][ny]=1
            fps(nx,ny,l,ns)
            l[nx][ny]=0

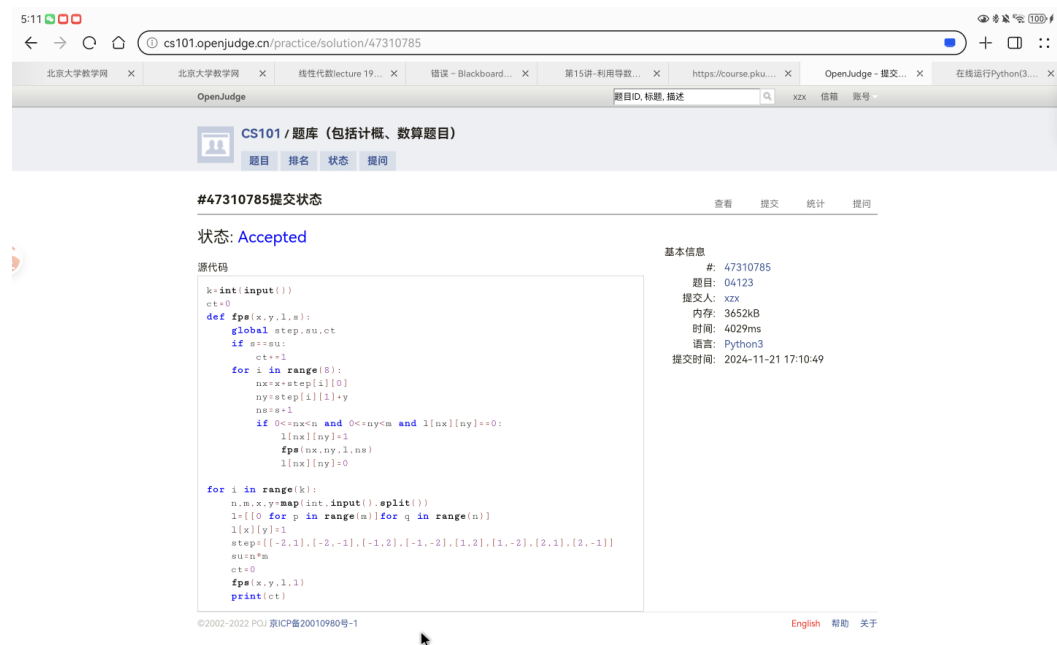
```

```

for i in range(k):
    n,m,x,y=map(int,input().split())
    l=[[0 for p in range(m)]for q in range(n)]
    l[x][y]=1
    step=[[-2,1],[-2,-1],[-1,2],[-1,-2],[1,2],[1,-2],[2,1],[2,-1]]
    su=n*m
    ct=0
    fps(x,y,l,1)
    print(ct)

```

代码运行截图 （至少包含有"Accepted"）



sy316: 矩阵最大权值路径

dfs, <https://sunnywhy.com/sfbj/8/1/316>

思路:

代码:

```
n,m=map(int,input().split())
```

```
l=[]
```

```
trace=[[0 for i in range(m)]for j in range(n)]
```

```
trace[0][0]=1
```

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

```

l.append(list(map(int,input().split()))))

step=[[1,0],[-1,0],[0,1],[0,-1]]

ms=0

mt=[]

for i in range(n-1):

    ms+=l[i][0]

    mt.append([i+1,1])

for j in range(m):

    mt.append([n,j+1])

ms+=sum(l[n-1])


def fps(x,y,s,t):

    global ms,step,mt

    if x==n-1 and y==m-1 and s>ms:

        ms=s

        mt=t

    for i in range(4):

        nx=x+step[i][0]

        ny=y+step[i][1]

        if 0<=nx<n and 0<=ny<m and trace[nx][ny]==0:

            ns=s+l[nx][ny]

            trace[nx][ny]=1

```

```
nt=t+[[nx+1,ny+1]]
```

```
fps(nx,ny,ns,nt)
```

```
trace[nx][ny]=0
```

```
fps(0,0,l[0][0],[[1,1]])
```

```
for k in mt:
```

```
print(k[0],k[1])
```

代码运行截图（至少包含有"Accepted"）



LeetCode62.不同路径

dp, <https://leetcode.cn/problems/unique-paths/>

思路：

代码：

```
class Solution(object):
```

```
def uniquePaths(self, m, n):
```

```
l = [[0 for i in range(n)] for j in range(m)]
```

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

$l[0][i] = 1$

for j in range(m):

$l[j][0] = 1$

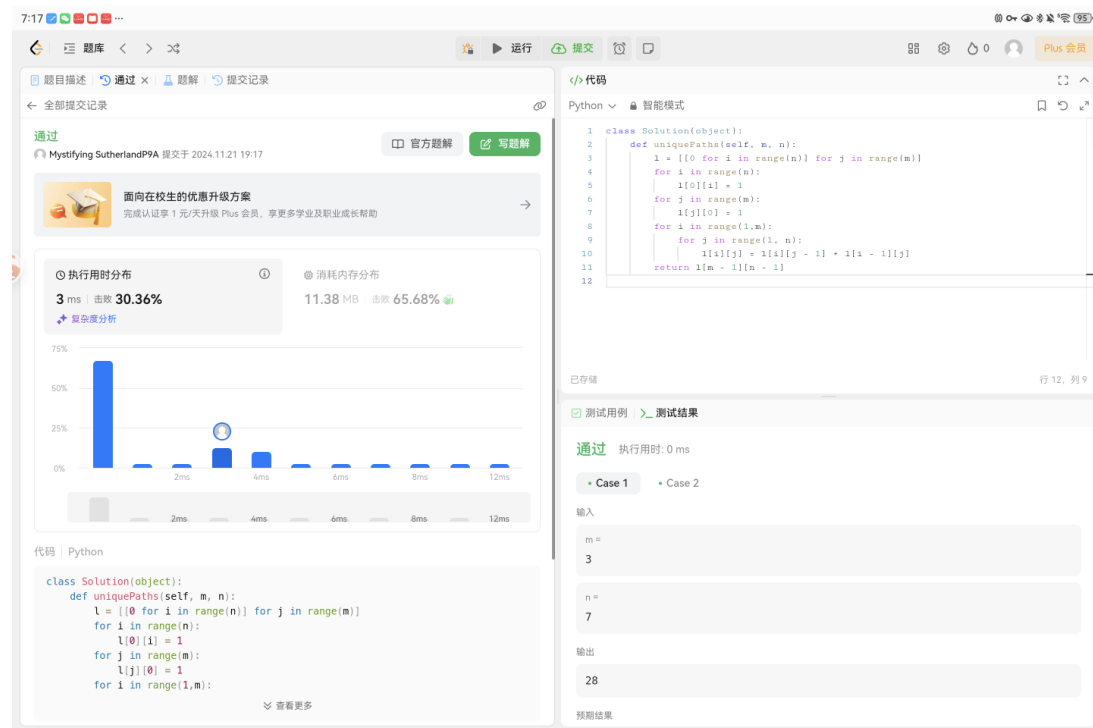
for i in range(1,m):

for j in range(1, n):

$l[i][j] = l[i][j - 1] + l[i - 1][j]$

return  $l[m - 1][n - 1]$

代码运行截图（至少包含有"Accepted"）



sy358: 受到祝福的平方

dfs, dp, <https://sunnywhy.com/sfbj/8/3/539>

思路:

代码:

a=input()

fl=False

def bless(a,p):

```

global fl

if p==len(a):

    fl=True

for i in range(p+1,len(a)+1):

    s=int(a[p:i])

    if int(s**0.5)==s**0.5 and s>0:

        bless(a,i)

bless(a,0)

if fl:

    print('Yes')

else:

    print('No')

```

代码运行截图（至少包含有"Accepted"）

The screenshot shows a coding competition interface. On the left, there's a sidebar with navigation links like '提高篇(2)', '综合练习精选', and '受到祝福的平方'. The main area displays the problem '受到祝福的平方' (Blessed Square) with its description, input/output, and constraints. The problem description explains that a number is 'blessed' if it can be split into two parts whose squares sum to the original number. Examples are given for A=8194, A=1001, and A=36. The input is a positive integer A, and the output is 'Yes' or 'No'. The submission area on the right shows the code from the previous block, and the results indicate a '完美通过' (Perfect Pass) with 100% data passed and 0ms runtime.

## 2. 学习总结和收获

本次作业题比较简单，因为 dfs 和 bfs 的题目思路很相似，基本是一题通，题题通，所以做起来比较快。



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。

日常跟进 OJ 每日选做