

第九周作业

18160:最大连通域面积(matrix,dfs)

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

```
def dfs(x,y):
    cnt = 1

    maze[x][y]='.'
    for dx,dy in directions:
        nx=x+dx
        ny=y+dy
        if 0<=nx<N and 0<=ny<M and maze[nx][ny]=='W':
            cnt += dfs(nx,ny)
    return cnt
```

```
T=int(input())
for _ in range(T):
    N,M=map(int,input().split())
    maze=[list(input()) for _ in range(N)]
    max_area=0

    for i in range(N):
        for j in range(M):
            if maze[i][j]=='W':
                max_area=max(max_area,dfs(i,j))

    print(max_area)
```

比赛	题目	结果	内存	时间	代码长度	语言	提交时间
 题库（包括计概、 数算题目）	18160: 最大连通域面积 (matrix,dfs)	Accepted	3696kB	95ms	552 B	Python3	1小时前

19930: 寻宝

```
from collections import deque
```

```
dx = [0, 0, 1, -1]
dy = [1, -1, 0, 0]
```

```
def bfs(x,y):
    q=deque()
```

```

q.append((x,y))
inq = set()
inq.add((x,y))
step=0
while q:
    for _ in range(len(q)):
        cur_x, cur_y = q.popleft()
        if maze[cur_x][cur_y]==1:
            return step
        for direction in range(4):
            nx = cur_x + dx[direction]
            ny = cur_y + dy[direction]
            if 1<=nx<=m and 1<=ny<=n and maze[nx][ny]!=2 and (nx,ny) not in inq:
                inq.add((nx,ny))
                q.append((nx,ny))
    step += 1

return 'NO'

```

```

m, n = map(int, input().split())
maze = [[-1] * (n + 2)] + [[-1] + list(map(int, input().split())) + [-1] for i in range(m)] + [[-1] * (n + 2)]

print(bfs(1,1))

```

这道题有一个疑问。。标黄色的那句，我本来写的是“`maze[nx][ny]==0`”，但就是一直不AC，无论输入什么输出的都是NO或者0，但我改成`!=2`就立马AC了，不懂为什么会这样....

比赛	题目	结果	内存	时间	代码长度	语言	提交时间
 题库 (包括计概、数算题目)	19930: 寻宝	Accepted	3712kB	29ms	796 B	Python3	3分钟前

04123: 马走日

```

directions=[(1,-2),(1,2),(-1,-2),(-1,2),(2,1),(2,-1),(-2,1),(-2,-1)]

```

```

cnt=0

```

```

def dfs(depth,x,y):
    maze[x][y] = True
    if depth==n*m:
        global cnt

```

```

        cnt+=1
    return

for dx,dy in directions:
    nx=x+dx
    ny=y+dy
    if 0<=nx<n and 0<=ny<m and maze[nx][ny]==False:
        maze[nx][ny]=True
        dfs(depth+1,nx,ny)
        maze[nx][ny]=False

T=int(input())
for _ in range(T):
    n,m,a,b=map(int,input().split())
    maze=[[False]*m for _ in range(n)]
    cnt=0
    dfs(1,a,b)
    print(cnt)

```

比赛	题目	结果	内存	时间	代码长度	语言	提交时间
 题库 (包括计概、数算题目)	04123: 马走日	Accepted	3600kB	2788ms	547 B	Python3	55分钟前

sy316: 矩阵最大权值路径

```

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

n,m=map(int,input().split())
maze=[list(map(int,input().split())) for _ in range(n)]

visited=[[False]*m for _ in range(n)]
max_sum=-float('inf')
max_path=[]

def dfs(x,y,cur_sum,cur_path):
    global max_sum
    global max_path

    if x==n-1 and y==m-1:
        if cur_sum>max_sum:
            max_sum=cur_sum
            max_path=cur_path[:]
        return

```

```

for dx,dy in directions:
    nx=x+dx
    ny=y+dy
    if 0<=nx<n and 0<=ny<m and visited[nx][ny]==False:
        visited[nx][ny]=True
        cur_path.append((nx,ny))
        cur_sum+=maze[nx][ny]

        dfs(nx,ny,cur_sum,cur_path)

        cur_sum-=maze[nx][ny]
        cur_path.pop()
        visited[nx][ny]=False

visited[0][0]=True
dfs(0,0,maze[0][0],[(0,0)])
for x,y in max_path:
    print(x+1,y+1)

```

完美通过

100% 数据通过测试

运行时长: 0 ms

Leetcode:

```

class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        directions = [(1, 0), (0, 1)]

        visited = [[False] * n for _ in range(m)]

        def dfs(x, y):
            if (x, y) == (m-1, n-1):
                return 1

            total_paths = 0
            for dx, dy in directions:

```

```

        nx, ny = x + dx, y + dy
        if 0 <= nx < m and 0 <= ny < n and not visited[nx][ny]:
            visited[nx][ny] = True

            total_paths += dfs(nx, ny)
            visited[nx][ny] = False

    return total_paths

visited[0][0] = True
return dfs(0, 0)

```

☑ 测试用例 | >_ 测试结果

通过 执行用时: 0 ms

• Case 1 • Case 2

Leetcode 的判定标准和 oj 和 cf 都不一样, 还是问了 chatgpt 才知道 leetcode 最后不需要 print 只需要 return 即可。还有一个在我的本地运行很流畅的代码在 leetcode 提交就出了问题, 不知道为什么。leetcode 说是 `cnt+=1` 那个语句有问题。以下是那个代码。

```

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

```

```

m,n=map(int,input().split())
visited=[[False]*n for _ in range(m)]

```

```

def dfs(x,y):
    global cnt
    if (x,y)==(m-1,n-1):
        cnt+=1
        return

    for dx,dy in directions:
        nx=x+dx
        ny=y+dy
        if 0<=nx<m and 0<=ny<n and visited[nx][ny]==False:
            visited[nx][ny]=True
            dfs(nx,ny)

        visited[nx][ny]=False

```

```
visited[0][0]=True
dfs(0,0)
print(cnt)
```

sy358: 受到祝福的平方

```
def lucky(digits):
    squares=set()
    i=1
    while i*i<=10**9:
        squares.add(i*i)
        i+=1
    n=len(digits)

    dp=[False]*(n+1)
    dp[0]=True

    for i in range(1,n+1):
        for j in range(i):
            if int(digits[j:i]) in squares and dp[j]==True:
                dp[i]=True
                break

    return 'Yes' if dp[n] else 'No'

A=input()
print(lucky(A))
```

测试输入

提交结果

历史提交

完美通过

100% 数据通过测试

运行时长: 0 ms

感谢 dfs 模板题让我久违地找回了会做题的感觉。。。希望期末考试有更多的模板题哈哈。