Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Complied by 同学的姓名、院系

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

18160: 最大连通域面积

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

思路：

照例看到这种题先上保护圈，从某一个还未被访问的点开始，如果它的四周都不能走就停止，反之走到下一个点继续，并将上一个点标记为不可访问。

代码：

```
connect = 0
D = [[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]]
def dfs(x, y):
    global connect
    if maps[x][y] == '.':
        return
    maps[x][y] = '.'
    connect += 1
    for ele in D:
        dfs(x + ele[0], y + ele[1])

for _ in range(int(input())):
    m, n = map(int, input().split())
    maps = [['.']*(n + 2)]
    for i in range(m):
        maps.append(['.'] + list(input()) + ['.'])
    maps.append(['.']*(n + 2))
    count = 0
    for i in range(1, m + 1):
        for j in range(1, n + 1):
            if maps[i][j] == 'W':
                connect = 0
```

```
                dfs(i, j)
                count = max(count, connect)
    print(count)
```

代码运行截图 （至少包含有"Accepted"）

19930: 寻宝

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

思路：

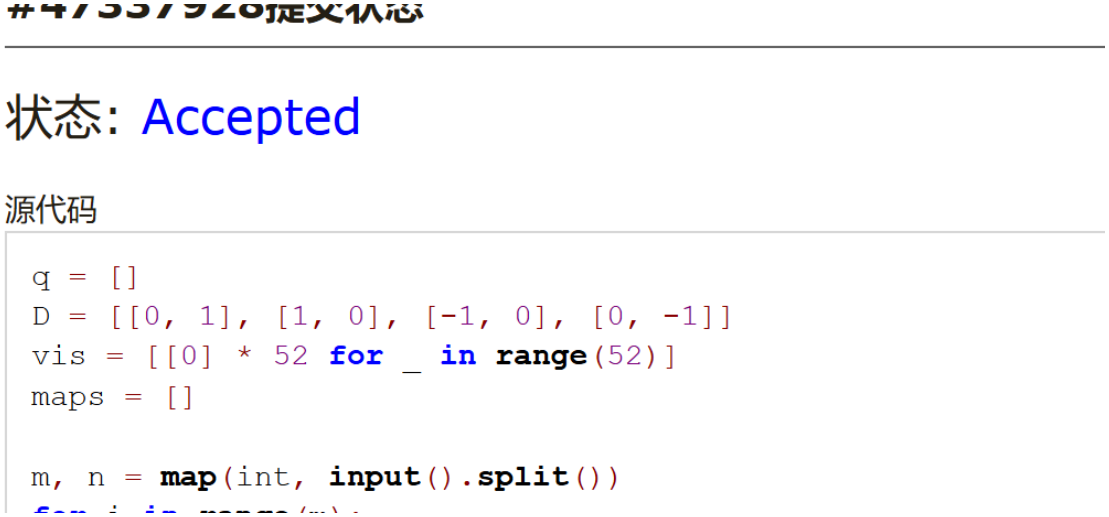
全程看着答案做的.....

代码：

```
q = []
D = [[0, 1], [1, 0], [-1, 0], [0, -1]]
vis = [[0] * 52 for _ in range(52)]
maps = []

m, n = map(int, input().split())
for i in range(m):
    maps.append(list(map(int, input().split())))

def check(x, y):
    if (x < 0 or y < 0 or x >= m or y >= n):
        return False
    if (vis[x][y] or maps[x][y] == 2):
        return False
    return True

q.append((0, 0))
begin, end = 0, 1
level = 0
while begin < end:
    for k in range(begin, end):
        x, y = q[begin]
        begin += 1
        if (maps[x][y] == 1):
```

```
            print(level)
            exit(0)
        for ele in D:
            next_x, next_y = x + ele[0], y + ele[1]
            if (check(next_x, next_y)):
                vis[next_x][next_y] = 1
                q.append((next_x, next_y))
                end += 1
    level += 1
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:

比较标准的dfs（个人理解），但还是卡了很久（实力太差导致的）

用到了回溯的cao'z

代码:

```
D = [[1, 2], [1, -2], [-1, 2], [-1, -2], [2, 1], [2, -1], [-2, 1], [-2, -1]]
ans = 0
def dfs(count, x, y):
    if count == m*n:
        global ans
        ans += 1
        return
    for ele in D:
        if maps[x + ele[0]][y + ele[1]] == 1:
            maps[x + ele[0]][y + ele[1]] = 0
            dfs(count + 1, x + ele[0], y + ele[1])
            maps[x + ele[0]][y + ele[1]] = 1

for t in range(int(input())):
```

```
    n, m , x, y = map(int, input().split())
    maps = [[0]*(m + 4)]*2 + [[0, 0] + [1 for _ in range(m)] + [0, 0] for i in
range(n)] + [[0]*(m + 4)]*2
    ans = 0
    maps[x + 2][y + 2] = 0
    dfs(1, x + 2, y + 2)
    print(ans)
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

源代码

```
D = [[1, 2], [1, -2], [-1, 2], [-1, -2], [2, 1], [2, -1], [-2, 1], [-
ans = 0
```

sy316: 矩阵最大权值路径

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

思路：

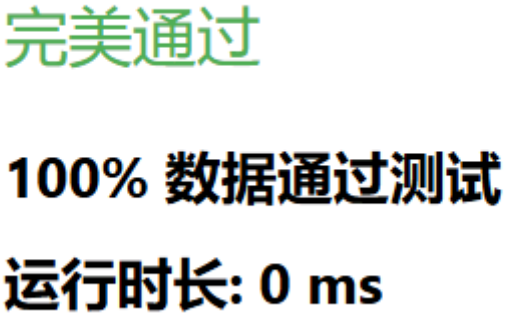
标准dfs, 可以就继续向下搜索，不行就回溯

代码：

```
D = [[1, 0], [0, 1], [-1, 0], [0, -1]]
def dfs(x, y, count):
    global max_v, ans
    if (x, y) == (n - 1, m - 1):
        if count > max_v:
            max_v, ans = count, temp_path[:]

    visited[x][y] = True
    for ele in D:
        next_x, next_y = x + ele[0], y + ele[1]
        if 0 <= next_x < n and 0 <= next_y < m and not visited[next_x][next_y]:
            next_v = count + maps[next_x][next_y]
            temp_path.append((next_x, next_y))
            dfs(next_x, next_y, next_v)
            temp_path.pop()

    visited[x][y] = False

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

max_v = float('-inf')
ans = []
temp_path = [(0, 0)]
visited = [[False]*m for _ in range(n)]
```

```
dfs(0, 0, maps[0][0])
for x, y in ans:
    print(x + 1, y + 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：

数学思路：直接

$$C_{m+n}^{2}=\frac{(m+n)(m+n-1)}{2}$$

代码思路：

小学奥数思路，角上的数相加

代码：

```
class Solution(object):
    def uniquePaths(self, m, n):
        """
        :type m: int
        :type n: int
        :rtype: int
        """
        dp = [[1]*m] + [[1] + [0]*(m - 1) for i in range(n - 1)]
        for i in range(1, n):
            for j in range(1, m):
                dp[i][j] = dp[i - 1][j] + dp[i][j - 1]
        return dp[-1][-1]
```

代码运行截图 （至少包含有"Accepted"）

所有状态	所有语言	执行用时	消耗内存	备注
通过 2024.11.19	Python	0 ms	11.3 MB	

sy358: 受到祝福的平方

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

思路：

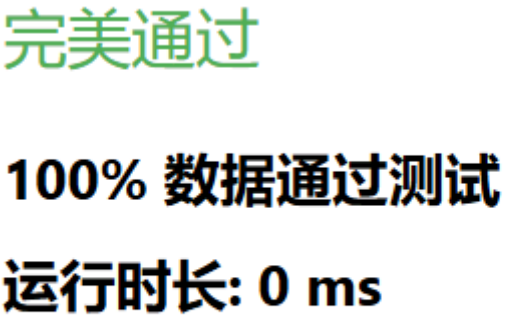
在这道题上犯了个低级错误 按照思维惯性没有加return……加上一开始没想到361 vs 36的特殊情况卡了一会

总的来说，先把字符串能按整体分割就按整体分割，然后递归处理后续未分割的字符串，如果之前分的太整了再调整

代码：

```
import math

def square(y):
    x = int(y)
    if x != 0 and math.sqrt(x) == int(math.sqrt(x)):
        return True

def div(x):
    if square(x):
        return True
    for i in range(1, len(x)):
        if square(x[: - i]):
            if div(x[len(x) - i:]):
                return True
            else:
                continue
print('Yes' if div(input()) else 'No')
```

代码运行截图 （至少包含有"Accepted"）



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。

dfs理解了，能自主写了

bfs没完全理解，这道作业完全写一行看一眼答案的程度，接下来要多练了

感受到老师挑选题目的用心，目前都还是基本的没变形的模板题目，对菜鸟还算友好（