

Assignment #9: dfs, bfs & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Compiled by 任宇桐 物理学院

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:

按照提示，直接套模板即可，开始直接完成的，没有注意到当第一次扫到位置时就直接修改对应的值，导致直接出现WA。

代码:

```
t = int(input())
for _ in range(t):
    n, m = map(int, input().split())
    s = []
    for _ in range(n):
        str = input()
        temp = []
        for i in range(m):
            temp.append(str[i])
        s.append(temp[:])

    ans = 0
    def scope(x, y):
        if 0 <= x < n and 0 <= y < m:
            return True
        else:
            return False

    def dfs(x, y):
        global d
        d += 1
        s[x][y] = '.'
```

```

        for dx, dy in [(1, 1), (1, 0), (1, -1), (0, 1), (0, -1), (-1, 1), (-1, 0), (-1, -1)]:
            nx = x+dx
            ny = y+dy

            if scope(nx, ny):
                if s[nx][ny] == 'w':
                    dfs(nx, ny)
    for i in range(n):
        for j in range(m):
            if s[i][j] == 'w':
                d = 0
                dfs(i, j)
                ans = max(ans, d)

    print(ans)

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

t = int(input())
for _ in range(t):
    n, m = map(int, input().split())
    s = []
    for _ in range(n):
        str = input()
        temp = []
        for i in range(m):
            temp.append(str[i])
        s.append(temp[:])

    ans = 0
    def scope(x, y):
        if 0 <= x < n and 0 <= y < m:
            return True
        else:
            return False

    def dfs(x, y):
        global d
        d += 1
        s[x][y] = '.'
        for dx, dy in [(1, 1), (1, 0), (1, -1), (0, 1), (0, -1), (-1, 1), (-1, 0), (-1, -1)]:
            nx = x+dx
            ny = y+dy

```

基本信息

#: 47289807
 题目: 18160
 提交人: 24n2400011498
 内存: 3720kB
 时间: 117ms
 语言: Python3
 提交时间: 2024-11-20 18:34:24

19930: 寻宝

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

思路:

直接按照bfs模板套用, 参考了题目中的讨论才知道要考虑第一个位置就是宝藏的情形。

代码:

```

from collections import deque

def scope(x, y):
    if 0 <= x < m and 0 <= y < n:
        return True
    else:
        return False

```

```

def bfs():
    queue = deque([(0, 0)])
    inQueue = [[False] * n for _ in range(m)]
    inQueue[0][0] = True
    step = 0
    if s[0][0] == 1:
        return step
    while queue:
        l = len(queue)
        for _ in range(l):
            (x, y) = queue.popleft()
            for dx, dy in [(0,1), (0, -1), (1, 0), (-1, 0)]:
                nx = x+dx
                ny = y+dy
                if scope(nx, ny) and not inQueue[nx][ny]:
                    if s[nx][ny] == 1:
                        return step+1
                    elif s[nx][ny] == 0:
                        queue.append((nx, ny))
                        inQueue[nx][ny] = True
            step += 1
    return 'NO'

m, n = map(int, input().split())
s = []
for _ in range(m):
    s.append(list(map(int, input().split())))
print(bfs())

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

from collections import deque

def scope(x, y):
    if 0 <= x < m and 0 <= y < n:
        return True
    else:
        return False

def bfs():
    queue = deque([(0, 0)])
    inQueue = [[False] * n for _ in range(m)]
    inQueue[0][0] = True
    step = 0
    if s[0][0] == 1:
        return step
    while queue:
        l = len(queue)

```

基本信息

#: 47306918
 题目: 19930
 提交人: 24n2400011498
 内存: 3716kB
 时间: 28ms
 语言: Python3
 提交时间: 2024-11-21 16:02:19

04123: 马走日

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

思路:

按照模板写即可。一开始使用了all(visited)判断是否是全部访问过，后来发现由于是二重列表，这样写是不成立的，发现可以直接用走过一个位置加1直接判断。

代码：

```
t = int(input())
for _ in range(t):
    n, m, x, y = map(int, input().split())
    def scope(i, j):
        return 0<= i < n and 0<= j < m

    visited = [[False]*m for _ in range(n)]
    visited[x][y] = True
    cnt = 0
    def dfs(i, j, step):
        global cnt
        if step == n*m:
            cnt += 1
        for di, dj in [(1, 2), (-1, 2), (1, -2), (-1, -2), (2, 1), (2, -1), (-2, 1), (-2, -1)]:
            ni = i+di
            nj = j+dj
            if scope(ni, nj):
                if not visited[ni][nj]:
                    visited[ni][nj] = True
                    dfs(ni, nj, step+1)
                    visited[ni][nj] = False

    dfs(x, y, 1)
    print(cnt)
```

代码运行截图 (至少包含有"Accepted")

#47307888提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
t = int(input())
for _ in range(t):
    n, m, x, y = map(int, input().split())
    def scope(i, j):
        return 0<= i < n and 0<= j < m

    visited = [[False]*m for _ in range(n)]
    visited[x][y] = True
    cnt = 0
    def dfs(i, j, step):
        global cnt
        if step == n*m:
            cnt += 1
        for di, dj in [(1, 2), (-1, 2), (1, -2), (-1, -2), (2, 1), (2, -1), (-2, 1), (-2, -1)]:
            ni = i+di
            nj = j+dj
            if scope(ni, nj):
                if not visited[ni][nj]:
                    visited[ni][nj] = True
                    dfs(ni, nj, step+1)
                    visited[ni][nj] = False

    dfs(x, y, 1)
    print(cnt)
```

基本信息

#: 47307888

题目: 04123

提交人: 24n2400011498

内存: 3672kB

时间: 3614ms

语言: Python3

提交时间: 2024-11-21 16:23:24

sy316: 矩阵最大权值路径

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

思路:

套用模板即可。

代码:

```
n, m = map(int, input().split())
s = []
ans = -float('inf')
ansp = []

for _ in range(n):
    s.append(list(map(int, input().split())))

visited = [[False]*m for _ in range(n)]
visited[0][0] = True

def scope(x, y):
    if 0 <= x < n and 0 <= y < m:
        return True
    else:
        return False

def dfs(x, y, maxv, temp):
    global ans, ansp
    if x == n-1 and y == m-1:
        if maxv > ans:
            ans = maxv
            ansp = temp[:]

    for dx, dy in [(0, 1), (0, -1), (-1, 0), (1, 0)]:
        nx = x + dx
        ny = y + dy

        if scope(nx, ny):
            if not visited[nx][ny]:
                visited[nx][ny] = True
                temp.append([nx+1, ny+1])
                dfs(nx, ny, maxv+s[nx][ny], temp)
                temp.pop()
                visited[nx][ny] = False

dfs(0, 0, s[0][0], [[1,1]])
for i in ansp:
    print(*i)
```

代码运行截图 (至少包含有"Accepted")

矩阵最大权值路径

通过数 842 提交数 2266 难度 中等 显示标签 ☆

题目描述

现有一个 $n * m$ 大小的矩阵，矩阵中的每个元素表示该位置的权值。现需要从矩阵左上角出发到达右下角，每次移动只能向上下左右移动一格（不允许移动到曾经经过的位置）。假设左上角坐标是 $(1,1)$ ，行数增加的方向为 x 增长的方向，列数增加的方向为 y 增长的方向。求最后到达右下角时路径上所有位置的权值之和最大的路径。

输入描述

第一行两个整数 n, m ($2 \leq n \leq 5, 2 \leq m \leq 5$)，分别表示矩阵的行数和列数；
接下来 n 行，每行 m 个整数 ($-100 \leq \text{整数} \leq 100$)，表示矩阵每个位置的权值。

```
1 n, m = map(int, input().split())
2 s = []
3 ans = -float('inf')
4 ansp = []
5
6 for _ in range(n):
7     s.append(list(map(int, input().split())))
8
9 visited = [[False]*m for _ in range(n)]
10 visited[0][0] = True
11
12 def scope(x, y):
13     if 0 <= x < n and 0 <= y < m:
14         return True
15     else:
16         return False
17
```

测试输入 提交结果 历史提交

完美通过 查看题解
100% 数据通过测试
运行时长: 0 ms

LeetCode62.不同路径

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

思路：

似乎是一道小学数学题，所以思路很明确，dp也很好想。

代码：

```
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        dp = [[1]*n for _ in range(m)]
        for i in range(1,m):
            for j in range(1,n):
                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")

62. 不同路径 已解答

中等 相关标签 相关企业 Aa

一个机器人位于一个 $m * n$ 网格的左上角（起始点在下图中标记为“Start”）。
机器人每次只能向下或者向右移动一步。机器人试图达到网格的右下角（在下图中标记为“Finish”）。
问总共有多少条不同的路径？

示例 1:



全部提交记录

通过 Optimistic Yalo... 提交于 2024.11.16 14:28 官方题解 写题解

执行用时分布 0 ms | 击败 100.00%

消耗内存分布 16.52 MB | 击败 30.31%

sy358: 受到祝福的平方

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

思路:

按照步长分割, 用dfs, 注意步长一定要设够所有可能的值, 否则会出现考虑不全的情形。

代码:

```
import math
s = input()
flag = False

def dfs(x):
    global flag
    if x == len(s):
        flag = True

    for dx in range(1, len(s)+1):
        nx = x + dx
        if nx <= len(s):
            temp = s[x:nx]
            temp = int(temp)
            if math.sqrt(temp).is_integer() and temp > 0:
                dfs(nx)

dfs(0)
if flag:
    print('Yes')
else:
    print('No')
```

代码运行截图 (至少包含有"Accepted")

东宫 算法笔记 题库 比赛 语言入门教程 考研算法大题特训 New 本期速递

题目

受到祝福

通过数 19

题目描述

在小元...
ID在被...
数字都...
被这个...
的人多...

令ID为...
是一个...

比如A...
{81,9,4...

比如A...
{1,001}...
{1,001}...

完美通过

100% 数据通过测试

运行时长: 0 ms

语言: Python

```
1 import math
2 s = input()
3 flag = False
4
5 def dfs(x):
6     global flag
7     if x == len(s):
8         flag = True
9
10    for dx in range(1, len(s)+1):
11        nx = x + dx
12        if nx <= len(s):
13            temp = s[x:nx]
14            temp = int(temp)
15            if math.sqrt(temp).is_integer() and temp > 0:
16                dfs(nx)
17
18
19 dfs(0)
20 if flag:
```

历史提交

结果

完美通过

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。

虽然模板题目不少，但是完全搞定还是花了不少时间的，回顾了一下，发现是随着代码长度的增加，各种小错误变得明显，比如不同的变量赋值成相同的名字但是没发现，初始时第一个点的确定，步长的确定等等，发现还需要勤加练习。