

Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Compiled by 徐嘉期、地空

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:

代码:

```
Test = int(input())
dx = [0,0 ,1 ,1,1,-1,-1,-1]
dy = [1,-1,-1,0,1,-1, 0, 1]

def check(mat,x,y):
    if x in range(n) and y in range(m) and mat[x][y] == 'w':
        return True
    return False

for _ in range(Test):
    n,m = map(int,input().split())
    matrix = [[] for _ in range(n)]
    max1 = 0
    stack = [(0,0)]

    for i in range(n):
        matrix[i] = list(input())
        for j in range(m):
            if matrix[i][j] == 'w':
                stack.pop()
                stack.append((i,j))

    # print(stack)
    while True:
        # print(stack,)
```

```

curr = 0
while stack:
    x,y = stack.pop()
    if matrix[x][y] == 'W':
        curr += 1
        matrix[x][y] = '.'
        for i in range(8):
            nx,ny = x+dx[i],y+dy[i]
            if check(matrix,nx,ny):
                stack.append((nx,ny))
max1 = max(max1,curr)
# print(matrix,max)
stat = False
for i in range(n):
    if stat:break
    for j in range(m):
        if matrix[i][j] == 'W':
            stack.append((i,j))
            stat = True
            break
    if not stack:break
print(max1)

```

代码运行截图 (至少包含有"Accepted")

#47274238提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 47274238
 题目: 18160
 提交人: 24n2400012440徐嘉期
 内存: 3760kB
 时间: 489ms
 语言: Python3
 提交时间: 2024-11-20 00:02:37

源代码

```

Test = int(input())
dx = [0,0,1,1,1,-1,-1,-1]
dy = [1,-1,-1,0,1,-1,0,1]

def check(mat,x,y):
    if x in range(n) and y in range(m) and mat[x][y] == 'W':
        return True
    return False

for _ in range(Test):
    n,m = map(int,input().split())
    matrix = [[] for _ in range(n)]
    max1 = 0
    stack = [(0,0)]

    for i in range(n):
        matrix[i] = list(input())
        for j in range(m):
            if matrix[i][j] == 'W':
                stack.pop()
                stack.append((i,j))

    # print(stack)
    while True:
        # print(stack)
        curr = 0
        while stack:
            x,y = stack.pop()
            if matrix[x][y] == 'W':
                curr += 1
                matrix[x][y] = '.'
                for i in range(8):
                    nx,ny = x+dx[i],y+dy[i]
                    if check(matrix,nx,ny):
                        stack.append((nx,ny))
            max1 = max(max1,curr)
        # print(matrix,max1)

```

19930: 寻宝

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

思路:

代码:

```
from collections import deque
```

```

n,m = map(int,input().split())
matrix = [[] for i in range(n)]
visited = set()
dx=[0,0,1,-1]
dy=[-1,1,0,0]
def check(mat,x,y):
    if x in range(n) and y in range(m) and mat[x][y] != 2 and (x,y) not in visited:return True
    else:return False
for _ in range(n):
    matrix[_] = list(map(int,input().split()))
stack = deque([(0,0)])
step = -1
stat = True
while stack and stat:
    step += 1
    curr = stack.copy()
    # print(step)
    # print(curr)
    # print()
    while curr and stat:

        x,y = curr.pop()
        stack.popleft()
        visited.add((x,y))
        # print(x,y)
        # print(stack)
        # print(curr)
        if matrix[x][y] == 1:
            print(step)
            stat = False
            break
        for i in range(4):
            if check(matrix,x+dx[i],y+dy[i]):
                stack.append((x+dx[i],y+dy[i]))
        # print(x,y)
        # print(stack)
        # print(curr)
        # print()
    if stat:print("NO")

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

基本信息

#: 47274406
题目: 19930
提交人: 24n2400012440徐嘉期
内存: 3712kB
时间: 31ms
语言: Python3
提交时间: 2024-11-20 00:32:23

源代码

```
from collections import deque

n,m = map(int,input().split())
matrix = [[] for i in range(n)]
visited = set()
dx=[0,0,1,-1]
dy=[-1,1,0,0]
def check(mat,x,y):
    if x in range(n) and y in range(m) and mat[x][y] != 2 and (x,y) not in visited:
        return True
    else:
        return False
for _ in range(n):
    matrix[_] = list(map(int,input().split()))
stack = deque([(0,0)])
step = -1
stat = True
while stack and stat:
    step += 1
    curr = stack.copy()
    # print(step)
    # print(curr)
    # print()
    while curr and stat:
        x,y = curr.pop()
        stack.popleft()
        visited.add((x,y))
        # print(x,y)
        # print(stack)
        # print(curr)
        # print(stat)
        if matrix[x][y] == 1:
            print(step)
            stat = False
            break
        for i in range(4):
            if check(matrix,x+dx[i],y+dy[i]):
                stack.append((x+dx[i],y+dy[i]))
```

04123: 马走日

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

思路:

代码:

```
from collections import deque
import sys
sys.setrecursionlimit(1<<30)
count = 0
def check(mat,x,y):
    if x in range(n) and y in range(m) and mat[x][y] == 0:
        return True
    return False

def dfs(mat, x, y):
    global count
    if len(inqueue) == total:
        count += 1
        return
    for i in range(8):
        nx, ny = x+dx[i], y+dy[i]
        # print(nx,ny,count)
        if check(mat,nx,ny):
            mat[nx][ny] = 1
            inqueue.add((nx,ny))
            dfs(mat, nx, ny)
            mat[nx][ny] = 0
            inqueue.remove((nx,ny))

Test = int(input())

for _ in range(Test):
```

```

count = 0
n, m, x, y = map(int, input().split())
dx = [1, 1, -1, -1, 2, -2, 2, -2]
dy = [2, -2, 2, -2, 1, 1, -1, -1]
matrix = [[0]*m for _ in range(n)]
matrix[x][y] = 1
total = n*m
inqueue = set()
inqueue.add((x,y))
dfs(matrix, x, y)
print(count)

```

代码运行截图 (至少包含有"Accepted")

#47284240提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 47284240
 题目: 04123
 提交人: 24n2400012440徐嘉期
 内存: 3704kB
 时间: 8887ms
 语言: Python3
 提交时间: 2024-11-20 15:08:54

源代码

```

from collections import deque
import sys
sys.setrecursionlimit(1<<30)
count = 0
def check(mat,x,y):
    if x in range(n) and y in range(m) and mat[x][y] == 0:
        return True
    return False

def dfs(mat, x, y):
    global count
    if len(inqueue) == total:
        count += 1
        return
    for i in range(8):
        nx, ny = x+dx[i], y+dy[i]
        # print(nx,ny,count)
        if check(mat,nx,ny):
            mat[nx][ny] = 1
            inqueue.add((nx,ny))
            dfs(mat, nx, ny)
            mat[nx][ny] = 0
            inqueue.remove((nx,ny))

Test = int(input())

for _ in range(Test):
    count = 0
    n, m, x, y = map(int, input().split())
    dx = [1, 1, -1, -1, 2, -2, 2, -2]
    dy = [2, -2, 2, -2, 1, 1, -1, -1]
    matrix = [[0]*m for _ in range(n)]
    matrix[x][y] = 1
    total = n*m
    inqueue = set()
    inqueue.add((x,y))

```

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())
matrix = [[ ] for _ in range(n)]
visited = [[True]*m for _ in range(n)]
dx = [1, -1, 0, 0]
dy = [0, 0, 1, -1]
ans=[]

def check(mat, x, y):
    if x in range(n) and y in range(m) and mat[x][y]:
        return True
    return False

```

```
def dfs(mat, x, y, value, step):
    global max1
    global ans
    if x == n-1 and y == m-1:
        if value > max1:
            max1 = value
            ans = step.copy()
            # print(ans)
        return
    visited[x][y] = False
    for i in range(4):
        nx, ny = x+dx[i], y+dy[i]
        if check(visited, nx, ny):
            step.append((nx,ny))
            dfs(mat, nx, ny, value+mat[nx][ny], step)
            step.pop()
    visited[x][y] = True

max1 = -float("inf")
for i in range(n):
    matrix[i] = list(map(int, input().split()))
visited[0][0] = False
dfs(matrix, 0, 0, matrix[0][0], [(0,0)])
for i in ans:
    x,y = i
    print(x+1,y+1)
```

代码运行截图 (至少包含有"Accepted")

晴问

课程

训练营

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

提高篇 (2) —— 搜索专题

深度优先搜索 (DFS)

迷宫可行路径数

指定步数的迷宫问题

矩阵最大权值

矩阵最大权值路径

迷宫最大权值

题目

题解

矩阵最大权值路径

通过数 929 提交数 2477 难度 中等 显示标签 ☆

题目描述

现有一个 $n \times m$ 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。

输出描述

从左上角的坐标开始，输出若干行（每行两个整数，表示一个坐标），直到右下角的坐标。

数据保证权值之和最大的路径存在且唯一。

样例1

```

3 3
1 2 3
4 5 6
7 8 9
          
```

代码书写

ans=[]

def check(mat, x, y):

if x in range(n) and y in range(m) and mat[x][y]:

return True

return False

def dfs(mat, x, y, value, step):

global max1

global ans

if x == n-1 and y == m-1:

max1 = value

ans = step.copy()

print(ans)

return

visited[x][y] = False

for i in range(4):

nx, ny = x+dx[i], y+dy[i]

if check(visited, nx, ny):

step.append((nx,ny))

测试输入

历史提交

提交时间	结果	时长(ms)	语言
2024-11-20 16:26:14	完美通过	0	Python 查看

收起面板

运行

提交

LeetCode62.不同路径

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

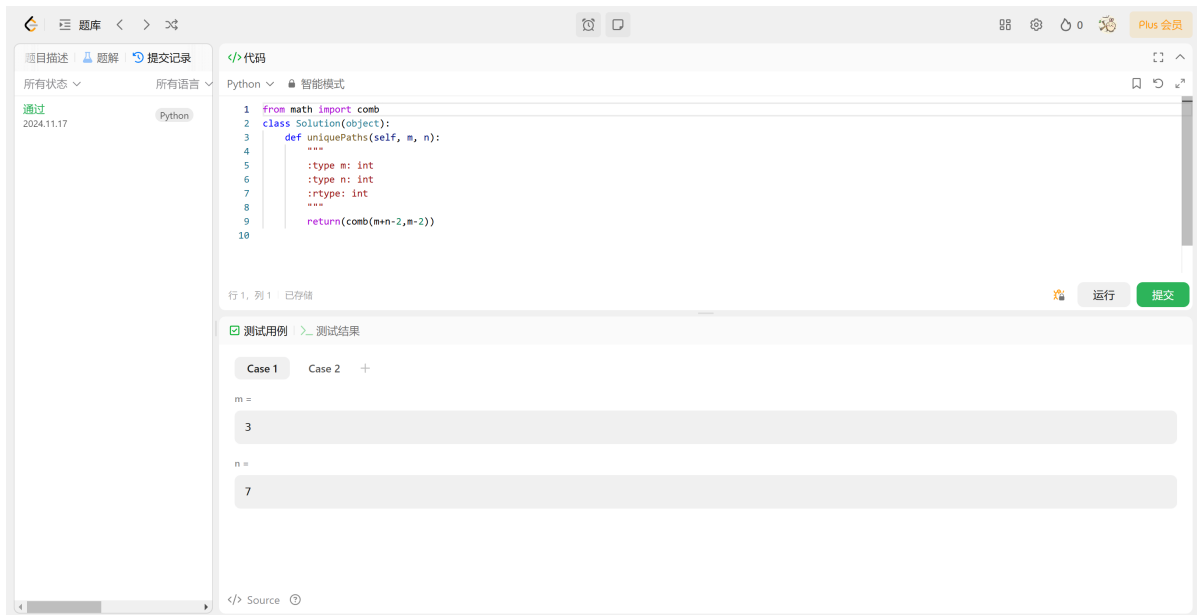
思路:

数学问题

代码:

```
from math import comb
class Solution(object):
    def uniquePaths(self, m, n):
        """
        :type m: int
        :type n: int
        :rtype: int
        """
        return(comb(m+n-2,m-2))
```

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:

```
from math import isqrt
def check(a):
    if a == 0: return False
```

```

        return isqrt(a)**2 == a
def dfs(s,step):
    global stat
    num = int(s)
    if stat:return
    if step == len(n)-1:
        if check (num):
            stat =True
        return
    else:
        if check(num):
            dfs(n[step+1],step+1)
        dfs(s+n[step+1],step+1)

n = input()
stat = False
dfs(n[0],0)
print("Yes" if stat else "No")

```

代码运行截图 (至少包含有"Accepted")

晴问 课程 训练营 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

提高篇 (2) ——搜索专题

综合练习精选

受到祝福的平方

题目 题解

受到祝福的平方

通过数 283 提交数 570 难度 中等 显示标签 ☆

题目描述

在小元的世界里，任何人出生后会世界分配一个随机ID，如果ID在被切割后，即ID满足按照从左至右顺序分割，且分割出来的数字都是某一个正整数的平方，分割时可以分为前导0，那么他就被这个世界祝福，最后获得快乐的数量和质量都比不满足这样的人多。

令ID为A，且A是一个正整数，取值范围为 $1 \leq A \leq 10^9$ ，问A是否是一个被受到祝福的ID。

比如A = 8194时，它是一个被受到祝福的ID，因为他可以被分割为{81,9,4} = {9²,3²,2²};

比如A = 1001时，它是一个被受到祝福的ID，因为他可以被分割为{1,001} = {1²,1²}, 或者{100,1} = {10²,1²}, 注意{1,00,1} = {1²,0²,1²}不是一个合法切割，因为分割出来的数字必须为正整数的平方;

比如A = 36时，36已经是一个平方数了，所以它同样满足条件;

比如A = 54，它不是一个被受到祝福的ID，因为他无法被切割为满足条件的集合。

输入描述

一个正整数A，无前导0。

代码书写 Python

```

1 from math import isqrt
2 def check(a):
3     if a == 0: return False
4     return isqrt(a)**2 == a
5 def dfs(s,step):
6     global stat
7     num = int(s)
8     if stat: return
9     if step == len(n)-1:
10         if check(num):
11             stat = True
12         return
13     else:
14         if check(num):
15             dfs(n[step+1],step+1)
16         dfs(s+n[step+1],step+1)
17
18 n = input()
19 stat = False
20 dfs(n[0],0)
21 print("Yes" if stat else "No")

```

测试输入 历史提交

提交时间	结果	时长(ms)	语言	
2024-11-20 16:55:23	完美通过	0	Python	查看
2024-11-20 16:48:27	编译错误	0	Python	查看

收起面板 运行 提交

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。

目前感觉良好，至少模板题都能打出来，还在跟进每日选做中