# Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Complied by <mark>同学的姓名、院系</mark>

\*\*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：关键点在于不回溯，想到这点之后就比较简单了，一开始把now\_S放进了函数参数里面导致会随着dfs往回返的时候变小，测试数据一直过不去，最后索性把它弄出来了；还有做的时候最后那里写成了range(1,n)和range(1,m)，又导致wa，看了眼答案才发现这错了，心太粗了

代码：

```python

T=int(input())

for i in range(T):

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

QP=[]

QP.append(['.']\*(m+2))

for j in range(n):

QP.append(['.']+list(input())+['.'])

QP.append(['.']\*(m+2))

max\_S=0

def dfs(QP,x,y):

global max\_S,now\_S

if now\_S>max\_S:

max\_S=now\_S

QP[x][y]='.'

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

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

for i in range(8):

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

if QP[nx][ny]=='W':

now\_S+=1

dfs(QP,nx,ny)

for k in range(1,n+1):

for l in range(1,m+1):

if QP[k][l]=='W':

now\_S=1

dfs(QP,k,l)

print(max\_S)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

### 19930: 寻宝

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

思路：确实是非常模板化的题，把pdf里的例题搞懂了做起来毫不费力。。。等等这是个bfs的题？？？

代码：

```python

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

min\_step=float('inf')

def dfs(maze,x,y,step):

global min\_step

if maze[x][y]==1:

min\_step=min(min\_step,step)

return

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

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

for i in range(4):

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

if maze[nx][ny]!=2:

maze[x][y]=2

dfs(maze,nx,ny,step+1)

maze[x][y]=0

maze=[]

maze.append([2]\*(n+2))

for i in range(m):

maze.append([2]+list(map(int,input().split()))+[2])

maze.append([2]\*(n+2))

dfs(maze,1,1,0)

print('NO' if min\_step==float('inf') else min\_step)

```

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：也是个模板题

代码：

```python

#pylint:skip-file

def dfs(QP, x, y, step):

global path

if step == n \* m - 1:

path += 1

return

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

for i in range(8):

dx, dy = dir[i]

nx, ny = x + dx, y + dy

if QP[nx][ny] != 1:

QP[x][y] = 1

dfs(QP, nx, ny, step + 1)

QP[x][y] = 0

T=int(input())

for i in range(T):

n,m,x,y=map(int,input().split())

path=0

QP=[]

QP.append([1]\*(m+4))

QP.append([1]\*(m+4))

for k in range(n):

QP.append([1]\*2+[0]\*m+[1]\*2)

QP.append([1]\*(m+4))

QP.append([1]\*(m+4))

dfs(QP,x+2,y+2,0)

print(path)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

### sy316: 矩阵最大权值路径

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

思路：这题有个有趣的地方是path[:]那里，如果不用拷贝的话best\_path会随着后续path改变，就会wa

代码：

```python

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

M=[]

M.append([float('-inf')]\*(m+2))

for i in range(n):

M.append([float('-inf')]+list(map(int,input().split()))+[float('-inf')])

M.append([float('-inf')]\*(m+2))

max\_value=float('-inf')

best\_path=[]

def dfs(M,x,y,now\_value,path):

global max\_value,best\_path

if x==n and y==m and now\_value>max\_value:

max\_value=now\_value

best\_path=path[:]

return

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

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

for i in range(4):

nx=x+dx[i]

ny=y+dy[i]

if M[nx][ny]!=float('-inf'):

a=M[x][y]

M[x][y]=float('-inf')

next\_value=now\_value+M[nx][ny]

path.append([nx,ny])

dfs(M,nx,ny,next\_value,path)

M[x][y]=a

path.pop()

dfs(M,1,1,M[1][1],[[1,1]])

for \_ in best\_path:

print(\*(\_),sep=' ')

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

### LeetCode62.不同路径

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

思路：正经按照dfs弄当然也弄得了，但是这题条件太特殊了，只能往右或者下走，相当于是往右n-1次和往下m-1次组合，纯纯高中排列组合题

代码：

```python

from math import comb

class Solution:

def uniquePaths(self, m: int, n: int) -> int:

return comb(m+n-2,m-1)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

### sy358: 受到祝福的平方

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

思路：一点一点把A切开看，小坑点是必须要是正整数的平方，所以0不行

代码：

```python

A=input()

bless=False

def dfs(A):

global bless

if int(A)\*\*(1/2)==int(int(A)\*\*(1/2)) and int(A)>0:

bless=True

return

n=len(A)

for i in range(1,n):

if int(A[:i])\*\*(1/2)==int(int(A[:i])\*\*(1/2)) and int(A[:i])>0:

dfs(A[i:])

dfs(A)

print('Yes' if bless else 'No')

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

## 2. 学习总结和收获

<mark>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。</mark>

这周作业挺善良的，稍微学一学记一下模板就都能做出来，要是考试题也能这么善良就好了（在想peach）