**# Assignment #9: dfs, bfs, & dp**

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Complied by <mark>陶嘉瑞-物理学院</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：

使用dfs求出各个连通区域面积再取最大值。

代码：

```python

dir=[[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]]

area=0

def dfs(x,y,matrix):

    global area

    if matrix[x][y]=='.':

        return

    matrix[x][y]='.'

    area+=1

    for i in range(len(dir)):

        dfs(x+dir[i][0],y+dir[i][1],matrix)

k=int(input())

for \_ in range(k):

    n,m=map(int,input().split())

    matrix = [['.' for \_ in range(m+2)] for \_ in range(n+2)]

    for i in range(1,n+1):

        matrix[i][1:-1] = input()

    max\_area=0

    for i in range(1,n+1):

        for j in range(1,m+1):

            if matrix[i][j]=='W':

                area=0

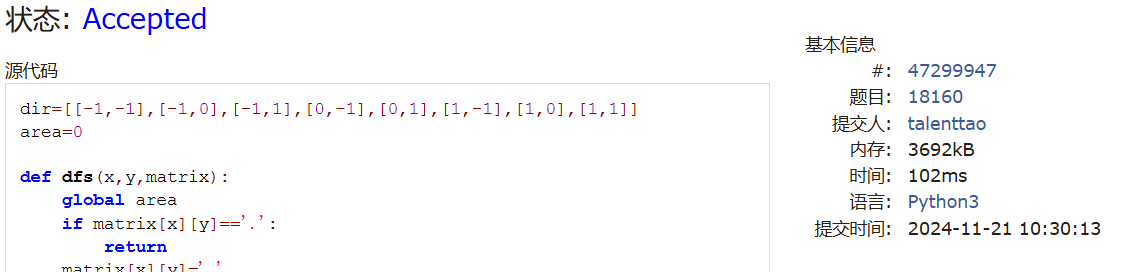
                dfs(i,j,matrix)

                max\_area=max(max\_area,area)

    print(max\_area)

```

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：

没想出来，参考答案的

代码：

```python

q = []

step = [[0, 1], [1, 0], [-1, 0], [0, -1]]

vis = [[0] \* 52 for \_ in range(52)]

g = []

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

for i in range(m):

    g.append([int(x) for x in 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 g[x][y] == 2):

        return False

    return True

q.append((0, 0))

head = 0

tail = 1

level = 0

while (head < tail):

    for k in range(head, tail):

        x, y = q[head]

        head += 1

        if (g[x][y] == 1):

            print(level)

            exit(0)

        for z in range(4):

            newx = x + step[z][0]

            newy = y + step[z][1]

            if (check(newx, newy)):

                vis[newx][newy] = 1

                q.append((newx, newy))

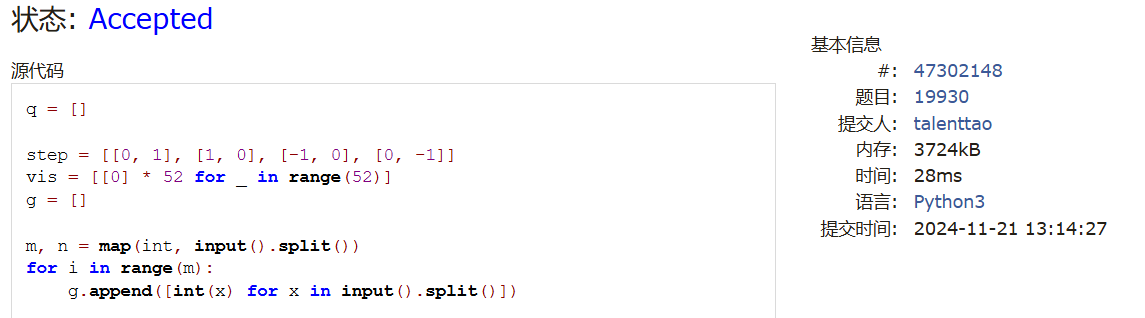
                tail += 1

    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：

利用递归即可

代码：

```python

k = 10

dirx = [-2,-1,1,2, 2, 1,-1,-2]

diry = [ 1, 2,2,1,-1,-2,-2,-1]

ans = 0

def dfs(num, x, y):

    if n\*m == num:

        global ans

        ans += 1

        return

    for r in range(8):

        s = x + dirx[r]

        t = y + diry[r]

        if matrix[s][t]==False and 0<=s<n and 0<=t<m :

            matrix[s][t]=True

            dfs(num+1, s, t)

            matrix[s][t] = False

for \_ in range(int(input())):

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

    matrix = [[False]\*k for \_ in range(k)]

    ans = 0

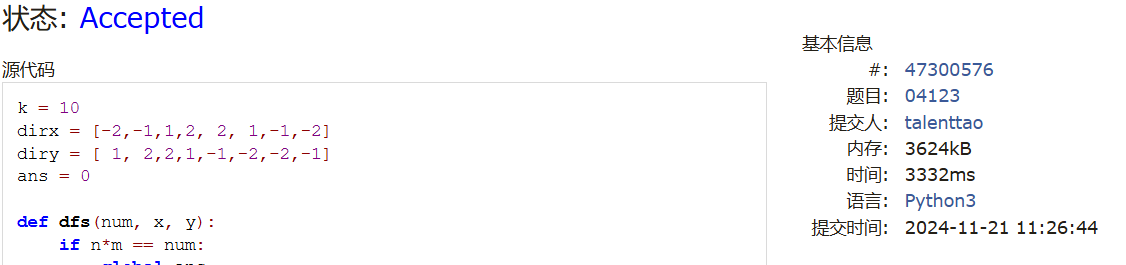
    matrix[x][y] = True

    dfs(1, x, y)

    print(ans)

```

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：

利用dfs求出每条可能的路径并取最大值

代码：

```python

def dfs(x,y,num):

    global max\_num,best\_path

    if x==n-1 and y==m-1:

        if num>max\_num:

            max\_num=num

            best\_path=path[:]

        return

    visited[x][y]=True

    for i in range(4):

        x1=x+dx[i]

        y1=y+dy[i]

        if 0<=x1<n and 0<=y1<m and (not visited[x1][y1]):

            num1=num+matrix[x1][y1]

            path.append((x1,y1))

            dfs(x1,y1,num1)

            path.pop()

    visited[x][y]=False

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

matrix=[]

for i in range(n):

    matrix.append(list(map(int,input().split())))

visited=[[False for i in range(m)] for j in range(n)]

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

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

max\_num=-float('inf')

best\_path=[]

path=[(0,0)]

dfs(0,0,matrix[0][0])

for x,y in best\_path:

    print(x+1,y+1)

```

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：

可用数学方法直接计算

代码：

```python

class Solution:

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

        array=[1]\*(n+m-1)

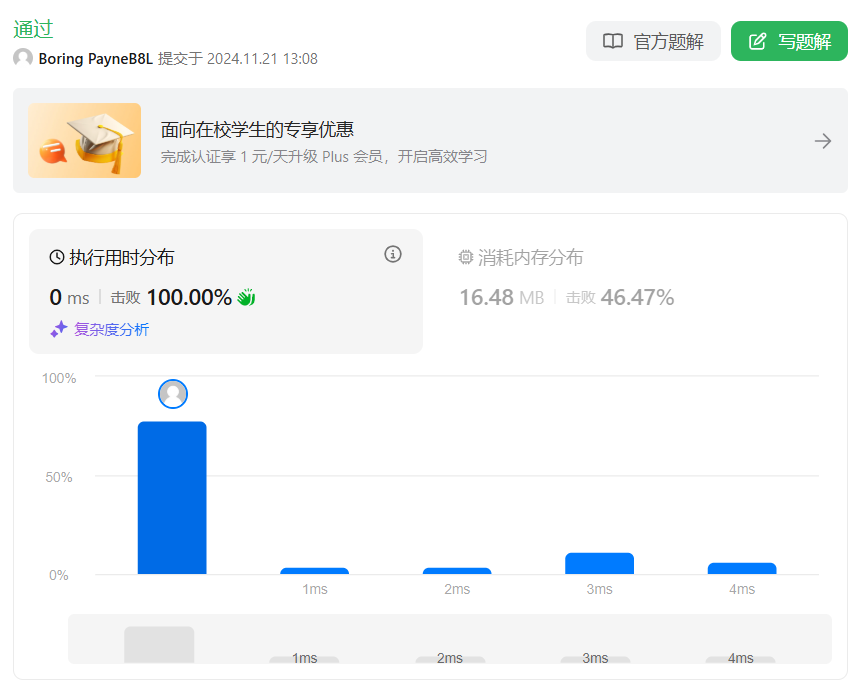
        for i in range(1,n+m-1):

            array[i]=i\*array[i-1]

        return (array[-1]//(array[n-1]\*array[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：

使用dfs暴力遍历

代码：

```python

squares = set()

i = 1

while i \* i <= 10\*\*9:

    squares.add(str(i \* i))

    i += 1

def dfs(num):

    global judge

    for i in range(1,len(num)+1):

        num1=num[:i]

        if num1 in squares:

            if num1==num:

                judge=True

            else:

                num2=num[i:]

                dfs(num2)

num=input()

judge=False

dfs(num)

print('Yes' if judge 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>

本周在补之前的练习，感觉dp和dfs以及掌握的较好，但bfs还不太会，作业中的bfs题目想了很久没想出来最后还是参考的答案，感觉还需要投入更多的精力。