# Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Complied by 洪千濠 工学院

**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：耗时40min，矩阵的行和列一直搞错，看了答案才发现www

代码：

```
dire=[[-1,0],[-1,-1],[0,-1],[-1,1],[1,-1],[1,1],[1,0],[0,1]]
area=0
def dfs(x,y):
    global area
    if matrix[x][y]=='.': return
    matrix[x][y]='.'
    area+=1
    for i in range(8):
        dfs(x+dire[i][0],y+dire[i][1])
for i in range(int(input())):
    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()
    final=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)
                final=max(final,area)
    print(final)
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
dire=[[-1,0],[-1,-1],[0,-1],[-1,1],[1,-1],[1,1],[1,0],[0,1]]
area=0
def dfs(x,y):
    global area
    if matrix[x][y]=='.': return
    matrix[x][y]='.'
    area+=1
    for i in range(8):
        dfs(x+dire[i][0],y+dire[i][1])
for i in range(int(input())):
    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()
    final=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)
                final=max(final,area)
    print(final)
```

基本信息

#: 473
题目: 181
提交人: 24n
内存: 378
时间: 103
语言: Pyth
提交时间: 202

## 19930: 寻宝

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

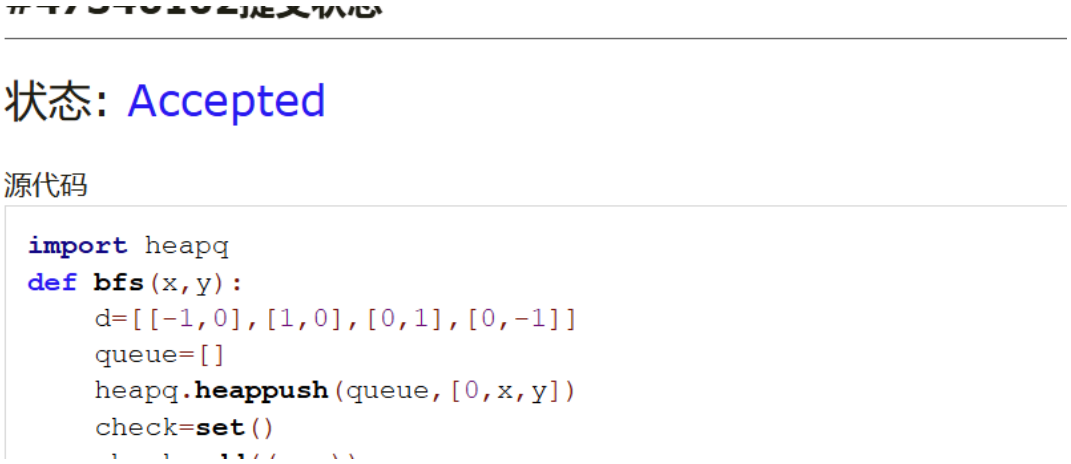
思路：耗时1h

代码：

```
import heapq
def bfs(x,y):
    d=[[-1,0],[1,0],[0,1],[0,-1]]
    queue=[]
    heapq.heappush(queue,[0,x,y])
    check=set()
    check.add((x,y))
    while queue:
        step,x,y=map(int,heapq.heappop(queue))
        if martix[x][y]==1:
            return step
        for i in range(4):
            dx,dy=x+d[i][0],y+d[i][1]
            if martix[dx][dy]!=2 and (dx,dy) not in check:
                heapq.heappush(queue,[step+1,dx,dy])
                check.add((dx,dy))
    return "NO"

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

```
martix=[[2]*(n+2)]+[[2]+list(map(int,input().split()))+[2] for i in range(m)]+
[[2]*(n+2)]
print(bfs(1,1))
```

代码运行截图 ==（至少包含有"Accepted"）==



状态: Accepted

源代码

```
import heapq
def bfs(x,y):
    d=[[-1,0],[1,0],[0,1],[0,-1]]
    queue=[]
    heapq.heappush(queue,[0,x,y])
    check=set()
```

## 04123: 马走日

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

思路：耗时1h

代码：

```
maxn = 10;
sx = [-2,-1,1,2, 2, 1,-1,-2]
sy = [ 1, 2,2,1,-1,-2,-2,-1]

ans = 0;

def Dfs(dep: int, x: int, y: int):
    #是否已经全部走完
    if n*m == dep:
        global ans
        ans += 1
        return

    #对于每个可以走的点
    for r in range(8):
        s = x + sx[r]
        t = y + sy[r]
        if chess[s][t]==False and 0<=s<n and 0<=t<m :
            chess[s][t]=True
            Dfs(dep+1, s, t)
            chess[s][t] = False; #回溯


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

```
    n,m,x,y = map(int, input().split())
    chess = [[False]*maxn for _ in range(maxn)]  #False表示没有走过
    ans = 0
    chess[x][y] = True
    Dfs(1, x, y)
    print(ans)
```

代码运行截图（至少包含有"Accepted"）

---

状态：Accepted

源代码

```
maxn = 10;
sx = [-2,-1,1,2, 2, 1,-1,-2]
sy = [ 1, 2,2,1,-1,-2,-2,-1]
```

## sy316: 矩阵最大权值路径

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

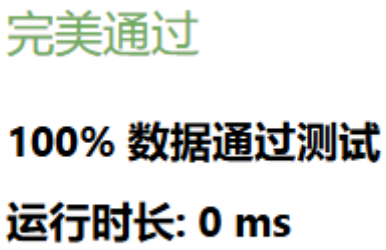
思路：耗时2h

代码：

```
def dfs(x, y, now_value):
    global max_value, opt_path
    if x == n - 1 and y == m - 1:
        if now_value > max_value:
            max_value = now_value
            opt_path = temp_path[:]
        return

    visited[x][y] = True

    for dx, dy in directions:
        next_x, next_y = x + dx, y + dy
        if 0 <= next_x < n and 0 <= next_y < m and not visited[next_x][next_y]:
            next_value = now_value + maze[next_x][next_y]
            temp_path.append((next_x, next_y))
            dfs(next_x, next_y, next_value)
            temp_path.pop()

    visited[x][y] = False

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

```
max_value = float('-inf')
opt_path = []
temp_path = [(0, 0)]
visited = [[False] * m for _ in range(n)]
directions = [(0, 1), (0, -1), (1, 0), (-1, 0)]

dfs(0, 0, maze[0][0])

for x, y in opt_path:
    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：耗时1.5h

代码：

```
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        import math
        def c(n, m):
            if n < m:
                return False
            else:
                return (math.factorial(n)) // (math.factorial(m)) //
(math.factorial(n - m))

        return c(m + n - 2, m - 1)
m,n=map(int,input().split())
answer=Solution().uniquePaths(m,n)
print(answer)
```

代码运行截图 （至少包含有"Accepted"）

通过

## sy358: 受到祝福的平方

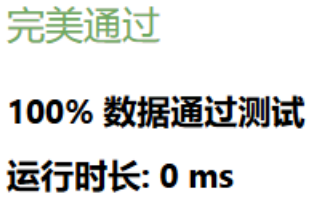
dfs, dp, https://sunnywhy.com/sfbj/8/3/539

思路：耗时30min

代码：

```
import math
def check(a,k):
    if k==len(a):
        return True
    now=0
    for i in range(k,len(a)):
        now=now*10+a[i]
        if math.sqrt(now)%1==0 and now!=0:
            if check(a,i+1):
                return True
    return False
a=[int(x) for x in input()]
if check(a,0):
    print('Yes')
else:
    print('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。

感觉题目做的越来越轻松了，从看题解中也发现许多很美妙的小细节，自己写代码也流畅很多了，不会磕磕绊绊，思路出来之后感觉蛮顺畅的，信心回来一些了，继续加油中