

# Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Compiled by <mark>胡新璞, 工学院</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>

代码:

s = 0

def dfs(a,b):

directions = [[1,1],[1,0],[1,-1],[0,1],[0,-1],[-1,1],[-1,0],[-1,-1]]

global s

if a < 0 or b < 0 or a >= len(matrix) or b >= len(matrix[0]) or matrix[a][b] == ".":

return

matrix[a][b] = "W"

s += 1

for i in range(len(directions)):

dfs(a + directions[i][0], b + directions[i][1])

cases = int(input())

for \_ in range(cases):

max\_s = 0

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

matrix = [ "." \* (m + 2) for i in range(n + 2)]

for \_ in range(1,n + 1):

matrix[\_][1:-1] = input()

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

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

if matrix[i][j] == "W":

s = 0

dfs(i,j)

if max\_s < s:

max\_s = s

print(max\_s)

代码运行截图 <mark> (至少包含有"Accepted") </mark>

状态: Accepted

源代码

```
s = 0
def dfs(a,b):
    directions = [[1,1],[1,0],[1,-1],[0,1],[0,-1],[-1,1],[-1,0],[-1,-1]]
    global s
    if a < 0 or b < 0 or a >= len(matrix) or b >= len(matrix[0]) or matrix[a][b] == ".":
        return
    matrix[a][b] = "W"
    s += 1
    for i in range(len(directions)):
        dfs(a + directions[i][0], b + directions[i][1])
```

基本信息

#: 47397759  
题目: 18160  
提交人: 2400011037  
内存: 3768kB  
时间: 223ms  
语言: Python3  
提交时间: 2024-11-25 23:06:51

### 19930: 寻宝

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

代码:

```
from collections import deque

def bfs(x,y):
    directions = [[0, 1], [1, 0], [-1, 0], [0, -1]]
    q = deque([(0,(x,y))])
    in_queue = {(x,y)}
    while q:
        step,(x,y) = q.popleft()
        if matrix[x][y] == 1:
            return step
        for i in range(len(directions)):
            nx = x + directions[i][0]
            ny = y + directions[i][1]
            if matrix[nx][ny] != 2 and (nx,ny) not in in_queue:
                in_queue.add((nx,ny))
                q.append((step + 1,(nx,ny)))
    return "NO"

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

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
from collections import deque

def bfs(x,y):
    directions = [[0, 1], [1, 0], [-1, 0], [0, -1]]
    q = deque([(0,(x,y))])
    in_queue = {(x,y)}
    while q:
        step,(x,y) = q.popleft()
        if matrix[x][y] == 1:
            return step
```

基本信息

#: 47404249  
题目: 19930  
提交人: 2400011037  
内存: 3696kB  
时间: 29ms  
语言: Python3  
提交时间: 2024-11-26 12:56:59

### 04123: 马走日

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

代码:

```
ans = 0
```

```
def dfs(x,y,cnt):
```

```
    directions = [[1, 2], [2, 1], [1, -2], [2, -1], [-1, 2], [-2, 1], [-1, -2], [-2, -1]]
```

```
    global ans
```

```
    if cnt == n * m:
```

```
        ans += 1
```

```
    return
```

```
    for i in range(len(directions)):
```

```
        nx = x + directions[i][0]
```

```
        ny = y + directions[i][1]
```

```
        if 0 <= nx < n and 0 <= ny < m:
```

```
            if matrix[nx][ny] != 1:
```

```
                matrix[nx][ny] = 1
```

```
                dfs(nx, ny, cnt + 1)
```

```
                matrix[nx][ny] = 0
```

```
cases = int(input())
```

```
for _ in range(cases):
```

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

```
    matrix = [[0] * m for _ in range(n)]
```

```
    ans = 0
```

```
    matrix[x][y] = 1
```

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

```
    print(ans)
```

代码运行截图 <mark> (至少包含有"Accepted") </mark>

状态: Accepted

源代码

```
ans = 0
def dfs(x,y,cnt):
    directions = [[1, 2], [2, 1], [1, -2], [2, -1], [-1, 2], [-2, 1], [-1, -2], [-2, -1]]
    global ans
    if cnt == n * m:
        ans += 1
    return
```

基本信息

#: 47404458

题目: 04123

提交人: 2400011037

内存: 3672kB

时间: 4467ms

语言: Python3

提交时间: 2024-11-26 13:23:33

### sy316: 矩阵最大权值路径 参照了讲义 sy315 的题解写

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

代码:

```
maxValue = float("-inf")
def dfs(x, y, nowValue):
    directions = [[0, 1], [1, 0], [-1, 0], [0, -1]]
    global maxValue, maxValue_path
    if x == n and y == m:
        if nowValue > maxValue:
            maxValue = nowValue
            maxValue_path = nowValue_path[:]
        return
    for i in range(len(directions)):
        nx = x + directions[i][0]
        ny = y + directions[i][1]
        if matrix[nx][ny] != 9999:
            tmp = matrix[x][y]
            matrix[x][y] = 9999
            nextValue = nowValue + matrix[nx][ny]
            nowValue_path.append((nx, ny))
            dfs(nx, ny, nextValue)
            matrix[x][y] = tmp
            nowValue_path.pop()
n, m = map(int, input().split())
maxValue_path = []
nowValue_path = [(1, 1)]
matrix = [[9999] * (m + 2) for i in range(n + 2)]
for _ in range(1, n + 1):
    matrix[_][1:-1] = map(int, input().split())
dfs(1, 1, matrix[1][1])
for i in range(len(maxValue_path)):
    print(maxValue_path[i][0], maxValue_path[i][1])
```

代码运行截图 <mark> (至少包含有"Accepted") </mark>



```
27 | matrix[_][1:-1] = map(int, input().split())
28 | dfs(1, 1, matrix[1][1])
29 | for i in range(len(maxValue_path)):
30 |     print(maxValue_path[i][0], maxValue_path[i][1])
31 |
32 |
```

测试输入 提交结果 历史提交

完美通过

100% 数据通过测试

[查看题解](#)

### LeetCode62.不同路径

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

(偷个懒 (不是)) (但是用组合数确实秒了)

代码:

```
from math import factorial
class Solution(object):
    def uniquePaths(self, m, n):
        return factorial(m + n - 2) / (factorial(m - 1) * factorial(n - 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>

代码:

```
from math import sqrt
def dfs(x):
    if x == len(A):
        return True
    n = 0
    for i in range(x, len(A)):
        n = n * 10 + A[i]
        if n != 0 and sqrt(n) % 1 == 0:
            if dfs(i + 1):
                return True
    return False
A = list(map(int, str(int(input()))))
if dfs(0):
    print("Yes")
else:
    print("No")
```

代码运行截图 <mark> (至少包含有"Accepted") </mark>



完美通过

[查看题解](#)

100% 数据通过测试

## 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>

每日选做继续跟进中，虽然目前能一遍过的题大多是之前落下的简单题，绝大多数人过得那种，难度中档往上的要么超时要么 WA，感觉对于自己的代码的优化还有很多的提升空间。这周借助作业题解和讲义基本熟练了 dfs 和 bfs 的写法，自己动手的时候在回溯这些地方还容易犯错导致出错，打算多刷几个题来巩固一下。