1. 题目

18160: 最大连通域面积

dfs similar, http://cs101.openjudge.cn/practice/18160

代码：

```
def dfs(matrix, row, col, visited):
    if row < 0 or row >= len(matrix) or col < 0 or col >= len(matrix[0]) \
            or matrix[row][col] != 'W' or visited[row][col]:
        return 0

    visited[row][col] = True
    size = 1

    for dr in [-1, 0, 1]:
        for dc in [-1, 0, 1]:
            size += dfs(matrix, row + dr, col + dc, visited)

    return size

def max_connected_area(matrix):
    max_area = 0
    visited = [[False for _ in range(len(matrix[0]))] for _ in range(len(matrix))]

    for row in range(len(matrix)):
        for col in range(len(matrix[0])):
            if matrix[row][col] == 'W' and not visited[row][col]:
                area = dfs(matrix, row, col, visited)
                max_area = max(max_area, area)

    return max_area

def main():
    T = int(input())
    for _ in range(T):
        N, M = map(int, input().split())
        matrix = [input().strip() for _ in range(N)]
        print(max_connected_area(matrix))

if __name__ == "__main__":
    main()
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
def dfs(matrix, row, col, visited):
    if row < 0 or row >= len(matrix) or col < 0 or col >= len(matrix[0])
            or matrix[row][col] != 'W' or visited[row][col]:
        return 0

    visited[row][col] = True
    size = 1
```

基本信息

#: 47415202
题目: 18160
提交人: misty
内存: 3776kB
时间: 134ms
语言: Python3
提交时间: 2024-11-26 20:55:56

19930: 寻宝

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

代码：

```
import heapqdef 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()
    check.add((x,y))
```

基本信息

#: 47415265
题目: 19930
提交人: misty
内存: 3716kB
时间: 37ms
语言: Python3
提交时间: 2024-11-26 20:58:24

04123: 马走日

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

代码：

```
T = int(input())
for i in range(T):
    list1 = list(map(int, input().split()))
```

```
    n = list1[0]
    m = list1[1]
    x = list1[2]
    y = list1[3]

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

    def dfs(x, y, lst, t):
        s = 0
        if t == n * m:
            return 1
        else:
            for v, w in direct:
                if (x + v, y + w) not in lst and x + v in range(n) and y + w in range(m):
                    lst.append((x + v, y + w))
                    s += dfs(x + v, y + w, lst, t + 1)
                    lst.pop()

        return s

    lst = [(x, y)]
    print(dfs(x, y, lst, 1))
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

源代码

```
T = int(input())

for i in range(T):
    list1 = list(map(int, input().split()))
    n = list1[0]
    m = list1[1]
```

基本信息

#: 47415427
题目: 04123
提交人: misty
内存: 3672kB
时间: 9879ms
语言: Python3
提交时间: 2024-11-26 21:04:56

sy316: 矩阵最大权值路径

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

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"）

状态: Accepted

源代码

```
T = int(input())

for i in range(T):
    list1 = list(map(int, input().split()))
    n = list1[0]
    m = list1[1]
    x = list1[2]
```

基本信息

#:	47415427
题目:	04123
提交人:	misty
内存:	3672kB
时间:	9879ms
语言:	Python3
提交时间:	2024-11-26 21:04:56

LeetCode62.不同路径

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

代码：

```
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        f = [1] * n
        for i in range(1, m):
            for j in range(1, n):
                f[j] += f[j - 1]
        return f[n - 1]
```

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

代码：

```
def is_blessed_id(A):
    squares = set()
    i = 1
    while i * i <= 10 ** 9:
        squares.add(i * i)
        i += 1

    digits = list(map(int, str(A)))

    def dfs(idx):
        if idx == len(digits):
            return True

        num = 0
        for i in range(idx, len(digits)):
            num = num * 10 + digits[i]
            if num in squares:
                if dfs(i + 1):
                    return True
        return False

    return "Yes" if dfs(0) else "No"


A = int(input())
print(is_blessed_id(A))
```

代码运行截图 （至少包含有"Accepted"）

2. 学习总结和收获

1.最近的讲义还没有过完，掌握的不好

2.dp 比之前会做了，感觉更容易有做题的思路

3.练习题 list 在慢慢赶进度