# Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Complied by 李振硕、信息管理系

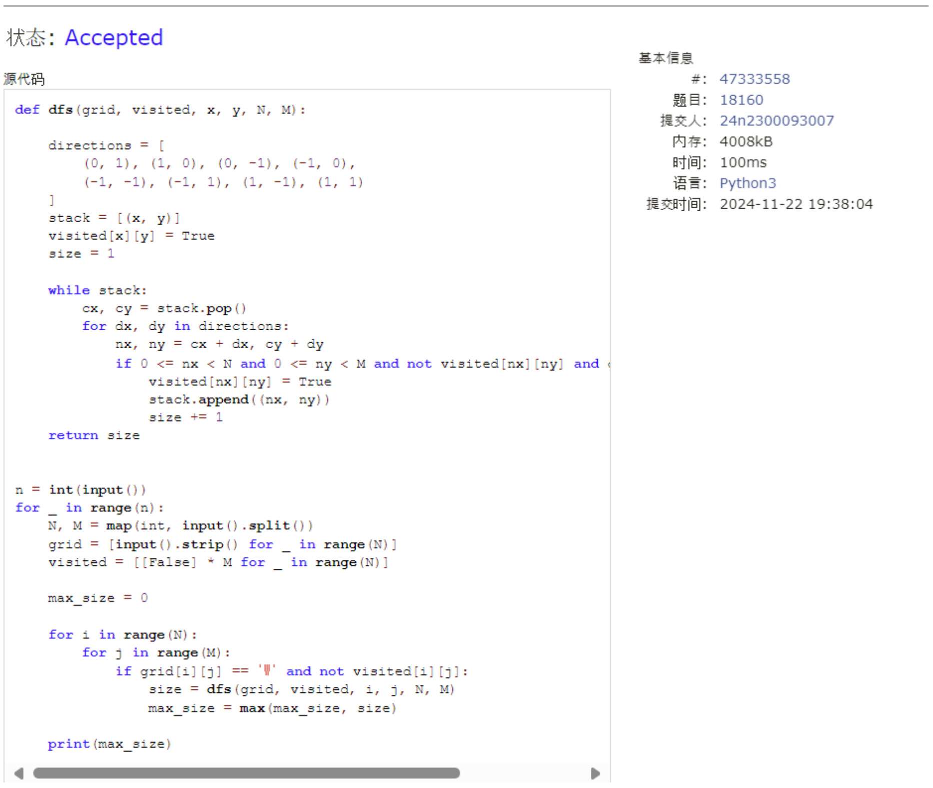
## 1. 题目

### 18160: 最大连通域面积

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

思路：

代码：



状态: Accepted

源代码

```python
def dfs(grid, visited, x, y, N, M):

    directions = [
        (0, 1), (1, 0), (0, -1), (-1, 0),
        (-1, -1), (-1, 1), (1, -1), (1, 1)
    ]
    stack = [(x, y)]
    visited[x][y] = True
    size = 1

    while stack:
        cx, cy = stack.pop()
        for dx, dy in directions:
            nx, ny = cx + dx, cy + dy
            if 0 <= nx < N and 0 <= ny < M and not visited[nx][ny] and
                visited[nx][ny] = True
                stack.append((nx, ny))
                size += 1
    return size


n = int(input())
for _ in range(n):
    N, M = map(int, input().split())
    grid = [input().strip() for _ in range(N)]
    visited = [[False] * M for _ in range(N)]

    max_size = 0

    for i in range(N):
        for j in range(M):
            if grid[i][j] == 'W' and not visited[i][j]:
                size = dfs(grid, visited, i, j, N, M)
                max_size = max(max_size, size)

    print(max_size)
```

基本信息

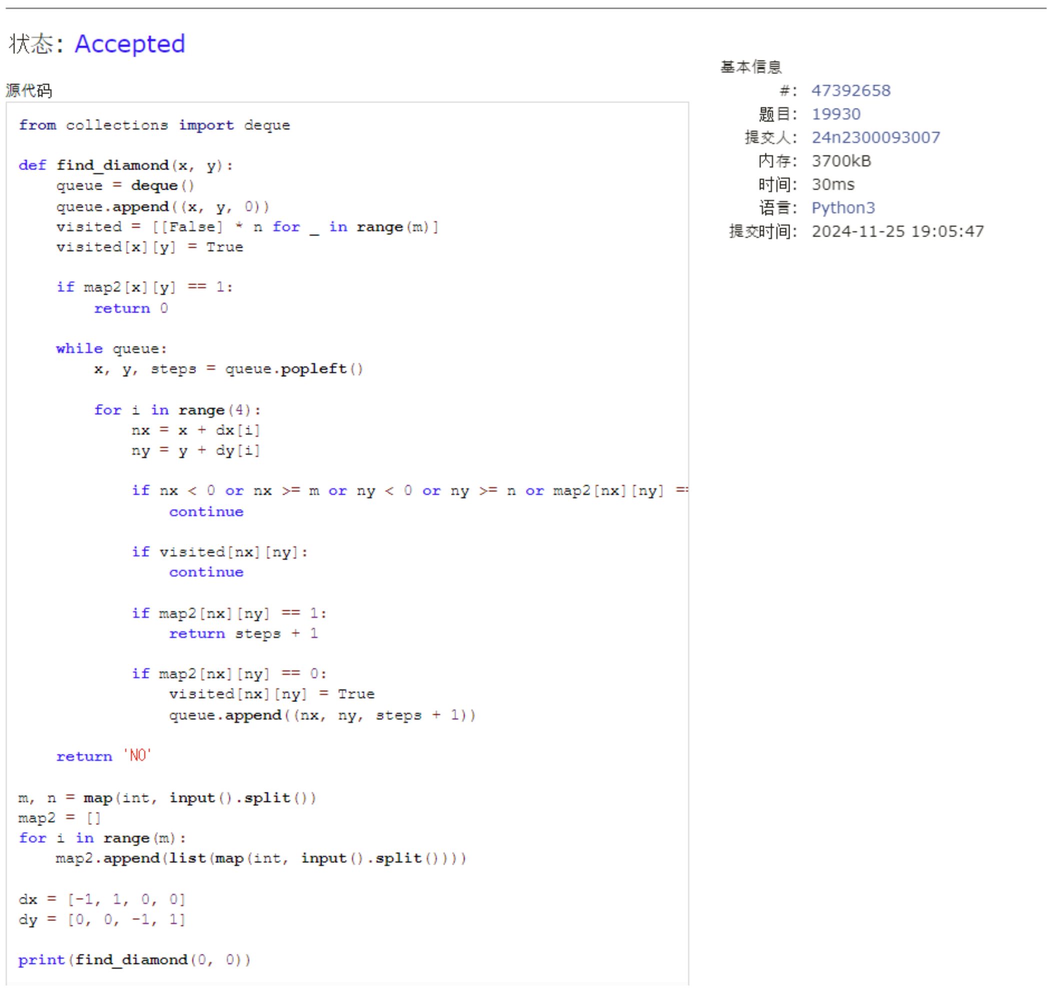
#:	47333558
题目:	18160
提交人:	24n2300093007
内存:	4008kB
时间:	100ms
语言:	Python3
提交时间:	2024-11-22 19:38:04

### 19930: 寻宝

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

思路：

代码：



状态: Accepted

源代码

```python
from collections import deque

def find_diamond(x, y):
    queue = deque()
    queue.append((x, y, 0))
    visited = [[False] * n for _ in range(m)]
    visited[x][y] = True

    if map2[x][y] == 1:
        return 0

    while queue:
        x, y, steps = queue.popleft()

        for i in range(4):
            nx = x + dx[i]
            ny = y + dy[i]

            if nx < 0 or nx >= m or ny < 0 or ny >= n or map2[nx][ny] =
                continue

            if visited[nx][ny]:
                continue

            if map2[nx][ny] == 1:
                return steps + 1

            if map2[nx][ny] == 0:
                visited[nx][ny] = True
                queue.append((nx, ny, steps + 1))

    return 'NO'

m, n = map(int, input().split())
map2 = []
for i in range(m):
    map2.append(list(map(int, input().split())))

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

print(find_diamond(0, 0))
```

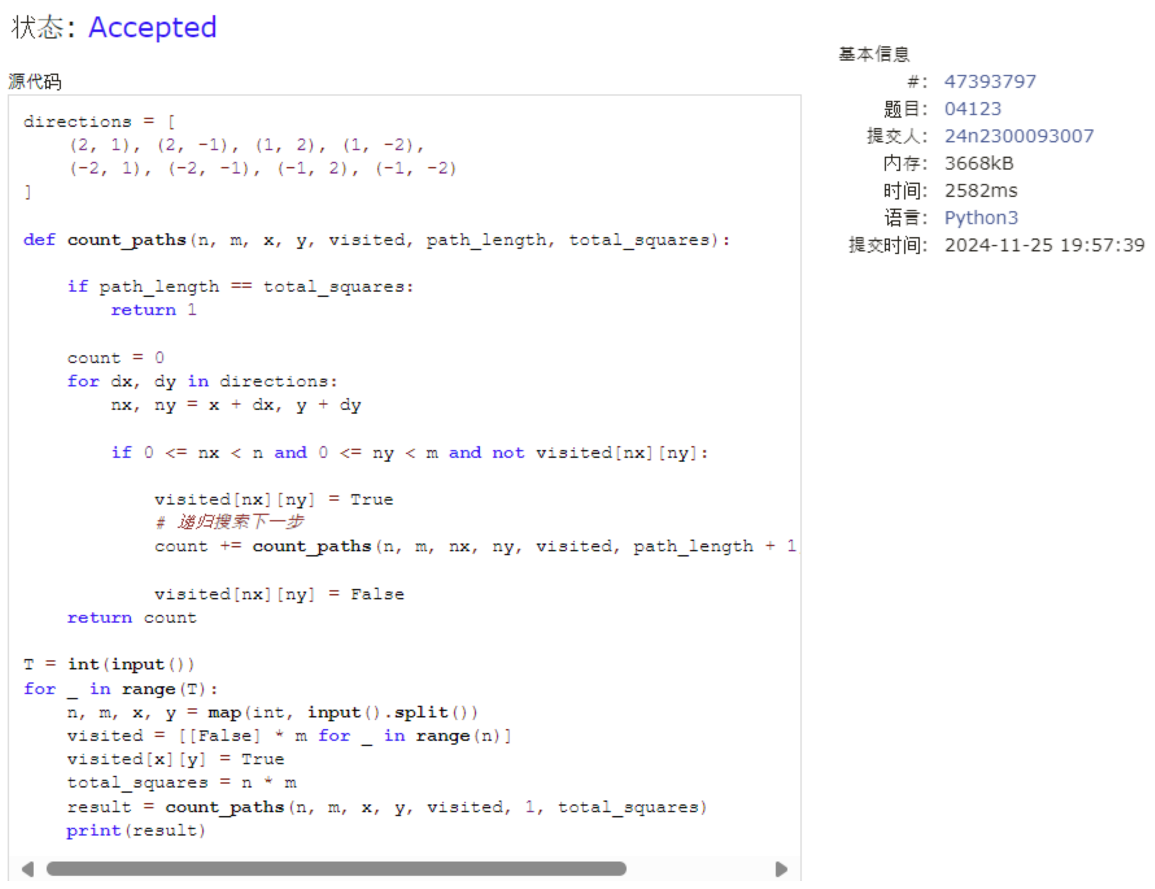
基本信息

#: 47392658
题目: 19930
提交人: 24n2300093007
内存: 3700kB
时间: 30ms
语言: Python3
提交时间: 2024-11-25 19:05:47

### 04123: 马走日

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

代码：



状态: Accepted

源代码

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

def count_paths(n, m, x, y, visited, path_length, total_squares):

    if path_length == total_squares:
        return 1

    count = 0
    for dx, dy in directions:
        nx, ny = x + dx, y + dy

        if 0 <= nx < n and 0 <= ny < m and not visited[nx][ny]:

            visited[nx][ny] = True
            # 递归搜索下一步
            count += count_paths(n, m, nx, ny, visited, path_length + 1

            visited[nx][ny] = False
    return count

T = int(input())
for _ in range(T):
    n, m, x, y = map(int, input().split())
    visited = [[False] * m for _ in range(n)]
    visited[x][y] = True
    total_squares = n * m
    result = count_paths(n, m, x, y, visited, 1, total_squares)
    print(result)
```

基本信息

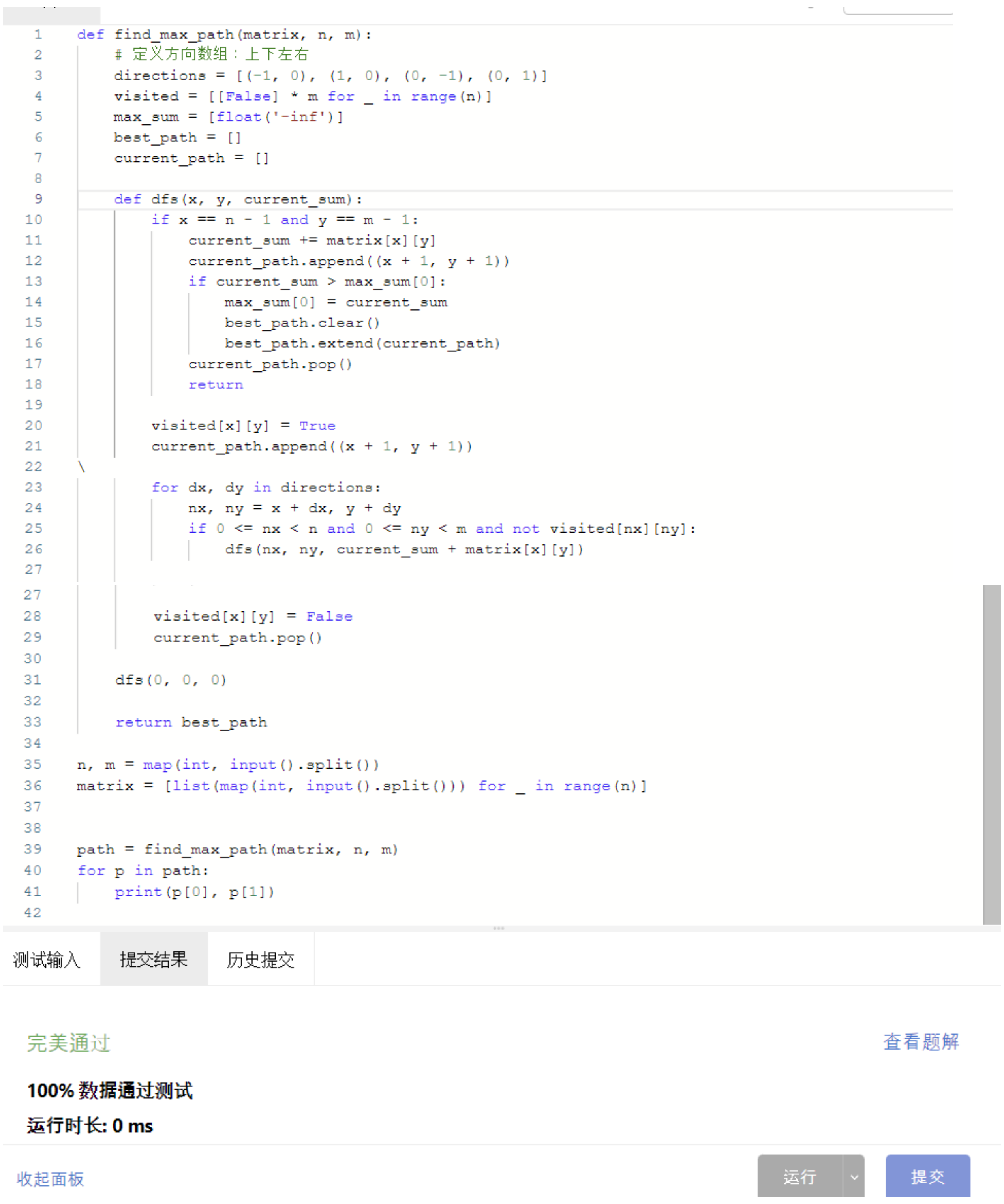
#: 47393797
题目: 04123
提交人: 24n2300093007
内存: 3668kB
时间: 2582ms
语言: Python3
提交时间: 2024-11-25 19:57:39

### sy316: 矩阵最大权值路径

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

思路：

代码：



```python
def find_max_path(matrix, n, m):
    # 定义方向数组：上下左右
    directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]
    visited = [[False] * m for _ in range(n)]
    max_sum = [float('-inf')]
    best_path = []
    current_path = []

    def dfs(x, y, current_sum):
        if x == n - 1 and y == m - 1:
            current_sum += matrix[x][y]
            current_path.append((x + 1, y + 1))
            if current_sum > max_sum[0]:
                max_sum[0] = current_sum
                best_path.clear()
                best_path.extend(current_path)
            current_path.pop()
            return

        visited[x][y] = True
        current_path.append((x + 1, y + 1))
\
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            if 0 <= nx < n and 0 <= ny < m and not visited[nx][ny]:
                dfs(nx, ny, current_sum + matrix[x][y])


        visited[x][y] = False
        current_path.pop()

    dfs(0, 0, 0)

    return best_path

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


path = find_max_path(matrix, n, m)
for p in path:
    print(p[0], p[1])
```

完美通过

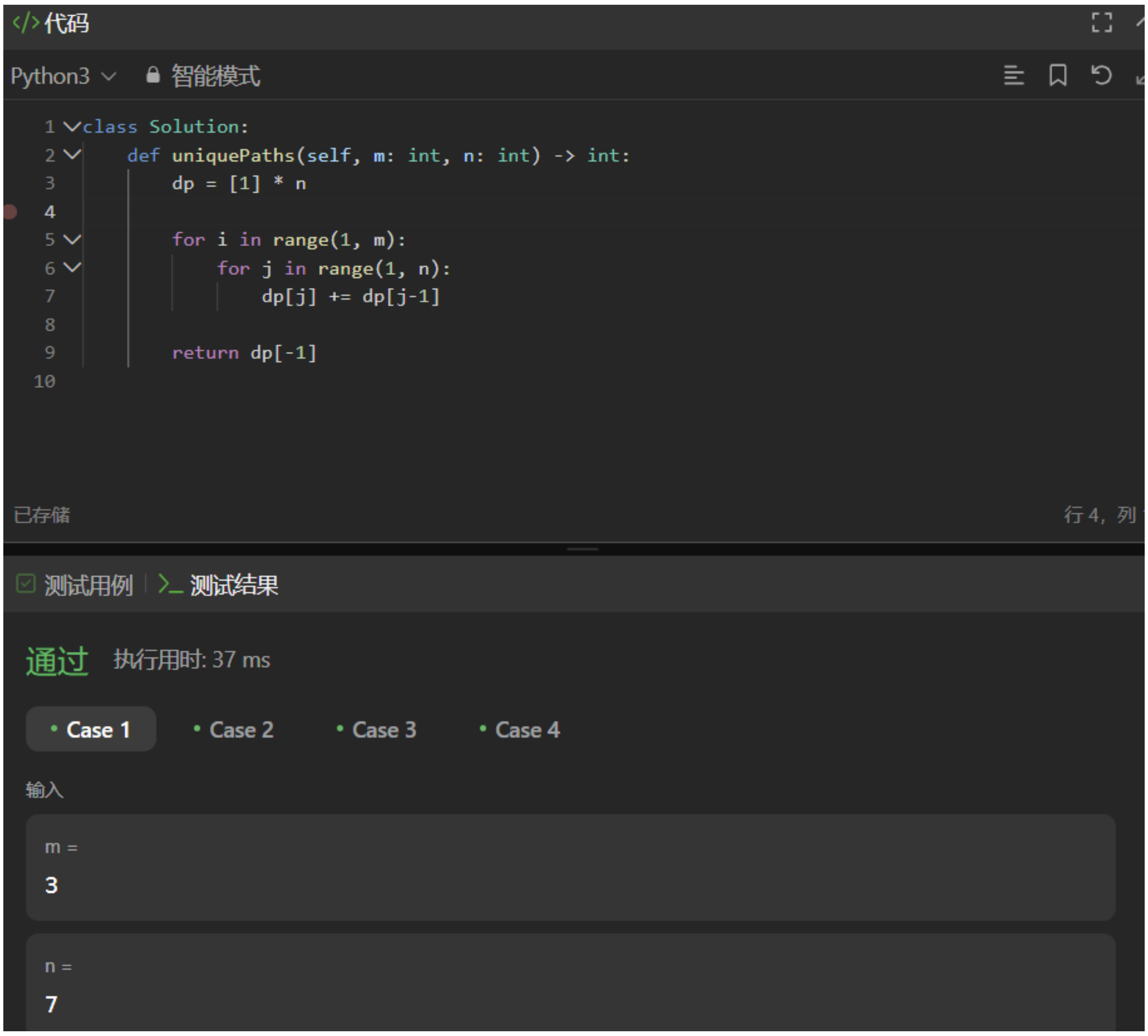
100% 数据通过测试

运行时长: 0 ms

### LeetCode62.不同路径

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

代码：



代码

Python3 智能模式

```python
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        dp = [1] * n

        for i in range(1, m):
            for j in range(1, n):
                dp[j] += dp[j-1]

        return dp[-1]

```

已存储 行 4，列

测试用例 测试结果

通过 执行用时: 37 ms

Case 1 Case 2 Case 3 Case 4

输入

m =
3

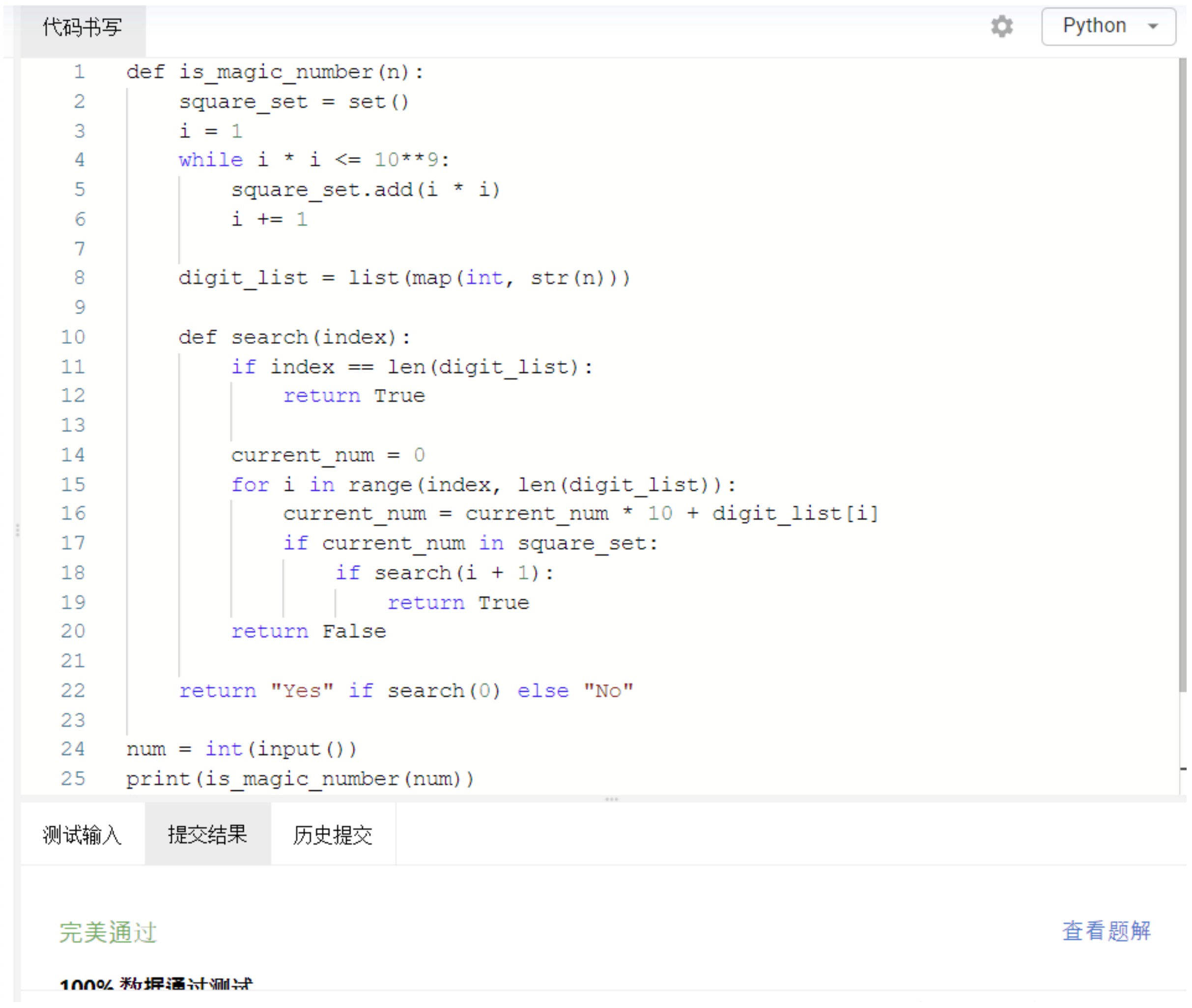
n =
7

### sy358: 受到祝福的平方

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

思路：

代码：



```python
def is_magic_number(n):
    square_set = set()
    i = 1
    while i * i <= 10**9:
        square_set.add(i * i)
        i += 1

    digit_list = list(map(int, str(n)))

    def search(index):
        if index == len(digit_list):
            return True

        current_num = 0
        for i in range(index, len(digit_list)):
            current_num = current_num * 10 + digit_list[i]
            if current_num in square_set:
                if search(i + 1):
                    return True
        return False

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

num = int(input())
print(is_magic_number(num))
```

完美通过

100% 数据通过测试

## 2. 学习总结和收获

通过写dp加dfs和bfs的问题，之前不会的dp问题现在就会写了。有了很大的进步!