# 0286. 墙与门

ITCharge 大约 1 分钟

- 标签：广度优先搜索、数组、矩阵
- 难度：中等

## 题目链接

- 0286. 墙与门 - 力扣

## 题目大意

给定一个 `m * n` 的二维网络 `rooms` 。其中每个元素有三种初始值：

- `-1` 表示墙或者障碍物
- `0` 表示一扇门
- `INF` 表示为一个空的房间。这里用 $2^{31} = 2147483647$ 表示 `INF` 。通往门的距离总是小于 $2^{31}$。

要求：给每个空房间填上该房间到最近的门的距离，如果无法到达门，则填 `INF` 。

## 解题思路

从每个表示门开始，使用广度优先搜索去照门。因为广度优先搜索保证我们在搜索 `dist + 1` 距离的位置时，距离为 `dist` 的位置都已经搜索过了。所以每到达一个房间的时候一定是最短距离。

## 代码

```py
class Solution:
    def wallsAndGates(self, rooms: List[List[int]]) -> None:
        """
        Do not return anything, modify rooms in-place instead.
        """
        INF = 2147483647
```

```
        rows = len(rooms)
        if rows == 0:
            return
        cols = len(rooms[0])

        directions = {(1, 0), (-1, 0), (0, 1), (0, -1)}
        queue = []
        for i in range(rows):
            for j in range(cols):
                if rooms[i][j] == 0:
                    queue.append((i, j, 0))

        while queue:
            i, j, dist = queue.pop(0)
            for direction in directions:
                new_i = i + direction[0]
                new_j = j + direction[1]
                if 0 <= new_i < rows and 0 <= new_j < cols and rooms[new_i]
[new_j] == INF:
                    rooms[new_i][new_j] = dist + 1
                    queue.append((new_i, new_j, dist + 1))
```