Problem 286: Walls and Gates

Problem Information

Difficulty: Medium
Acceptance Rate: 63.43%
Paid Only: Yes
Tags: Array, Breadth-First Search, Matrix

Problem Description

You are given an `m x n` grid `rooms` initialized with these three possible values.

* `-1` A wall or an obstacle. * `0` A gate. * `INF` Infinity means an empty room. We use the value `231 - 1 = 2147483647` to represent `INF` as you may assume that the distance to a gate is less than `2147483647`.

Fill each empty room with the distance to _its nearest gate_. If it is impossible to reach a gate, it should be filled with `INF`.

Example 1:

![](https://assets.leetcode.com/uploads/2021/01/03/grid.jpg)

Input: rooms = [[2147483647,-1,0,2147483647],[2147483647,2147483647,2147483647,-1],[2147483647,-1,2147483647,-1],[0,-1,2147483647,2147483647]] **Output:** [[3,-1,0,1],[2,2,1,-1],[1,-1,2,-1],[0,-1,3,4]]

Example 2:

Input: rooms = [[-1]] **Output:** [[-1]]

Constraints:

* `m == rooms.length` * `n == rooms[i].length` * `1 <= m, n <= 250` * `rooms[i][j]` is `-1`, `0`, or `231 - 1`.

Code Snippets

C++:

```
class Solution {
public:
void wallsAndGates(vector<vector<int>>& rooms) {

}
};
```

Java:

```
class Solution {
public void wallsAndGates(int[][] rooms) {

}
}
```

Python3:

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