Problem 3286: Find a Safe Walk Through a Grid

Problem Information

Difficulty: Medium
Acceptance Rate: 31.90%
Paid Only: No
Tags: Array, Breadth-First Search, Graph, Heap (Priority Queue), Matrix, Shortest Path

Problem Description

You are given an `m x n` binary matrix `grid` and an integer `health`.

You start on the upper-left corner `(0, 0)` and would like to get to the lower-right corner `(m - 1, n - 1)`.

You can move up, down, left, or right from one cell to another adjacent cell as long as your health _remains_ **positive**.

Cells `(i, j)` with `grid[i][j] = 1` are considered **unsafe** and reduce your health by 1.

Return `true` if you can reach the final cell with a health value of 1 or more, and `false` otherwise.

Example 1:

Input: grid = [[0,1,0,0,0],[0,1,0,1,0],[0,0,0,1,0]], health = 1

Output: true

Explanation:

The final cell can be reached safely by walking along the gray cells below.

![](https://assets.leetcode.com/uploads/2024/08/04/3868_examples_1drawio.png)

Example 2:

Input: grid = [[0,1,1,0,0,0],[1,0,1,0,0,0],[0,1,1,1,0,1],[0,0,1,0,1,0]], health = 3

Output: false

Explanation:

A minimum of 4 health points is needed to reach the final cell safely.

![](https://assets.leetcode.com/uploads/2024/08/04/3868_examples_2drawio.png)

Example 3:

Input: grid = [[1,1,1],[1,0,1],[1,1,1]], health = 5

Output: true

Explanation:

The final cell can be reached safely by walking along the gray cells below.

![](https://assets.leetcode.com/uploads/2024/08/04/3868_examples_3drawio.png)

Any path that does not go through the cell `(1, 1)` is unsafe since your health will drop to 0 when reaching the final cell.

Constraints:

* `m == grid.length` * `n == grid[i].length` * `1 <= m, n <= 50` * `2 <= m * n` * `1 <= health <= m + n` * `grid[i][j]` is either 0 or 1.

Code Snippets

C++:

```
class Solution {
public:
```

```
bool findSafeWalk(vector<vector<int>>& grid, int health) {

}
};
```

Java:

```
class Solution {
public boolean findSafeWalk(List<List<Integer>> grid, int health) {

}
}
```

Python3:

```
class Solution:
def findSafeWalk(self, grid: List[List[int]], health: int) -> bool:
```