Problem 2510: Check if There is a Path With Equal Number of 0's And 1's

Problem Information

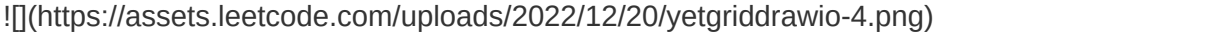
Difficulty: Medium
Acceptance Rate: 51.52%
Paid Only: Yes
Tags: Array, Dynamic Programming, Matrix

Problem Description

You are given a **0-indexed** `m x n` **binary** matrix `grid`. You can move from a cell `(row, col)` to any of the cells `(row + 1, col)` or `(row, col + 1)`.

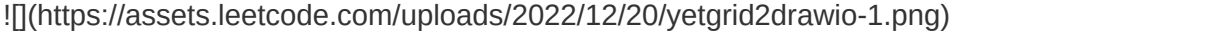
Return `true` _if there is a path from_`(0, 0)`_to_`(m - 1, n - 1)`_that visits an**equal** number of _`0` _' s and _`1` _' s_. Otherwise return `false`.

Example 1:



Input: grid = [[0,1,0,0],[0,1,0,0],[1,0,1,0]] **Output:** true **Explanation:** The path colored in blue in the above diagram is a valid path because we have 3 cells with a value of 1 and 3 with a value of 0. Since there is a valid path, we return true.

Example 2:



Input: grid = [[1,1,0],[0,0,1],[1,0,0]] **Output:** false **Explanation:** There is no path in this grid with an equal number of 0's and 1's.

Constraints:

* `m == grid.length` * `n == grid[i].length` * `2 <= m, n <= 100` * `grid[i][j]` is either `0` or `1`.

Code Snippets

C++:

```
class Solution {
public:
bool isThereAPath(vector<vector<int>>& grid) {

}
};
```

Java:

```
class Solution {
public boolean isThereAPath(int[][] grid) {

}
}
```

Python3:

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