In an N by N square grid, each cell is either empty (0) or blocked (1).

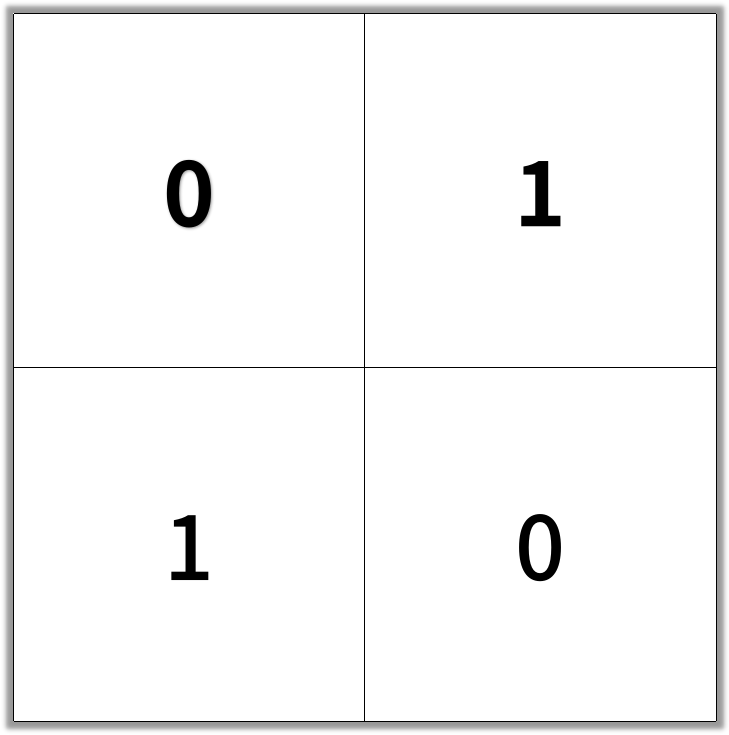
A *clear path from top-left to bottom-right* has length k if and only if it is composed of cells C\_1, C\_2, ..., C\_k such that:

* Adjacent cells C\_i and C\_{i+1} are connected 8-directionally (ie., they are different and share an edge or corner)
* C\_1 is at location (0, 0) (ie. has value grid[0][0])
* C\_k is at location (N-1, N-1) (ie. has value grid[N-1][N-1])
* If C\_i is located at (r, c), then grid[r][c] is empty (ie. grid[r][c] == 0).

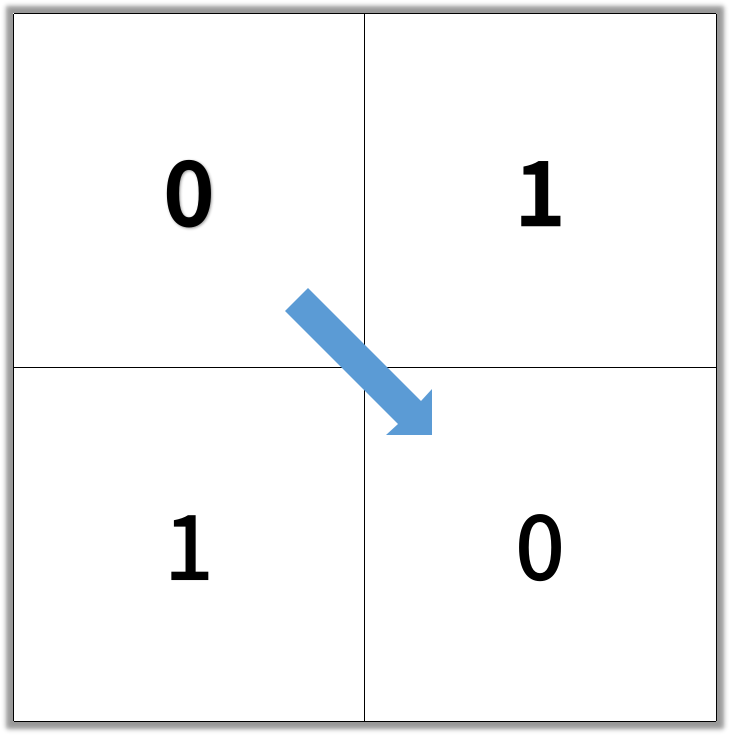
Return the length of the shortest such clear path from top-left to bottom-right.  If such a path does not exist, return -1.

**Example 1:**

**Input:** [[0,1],[1,0]]

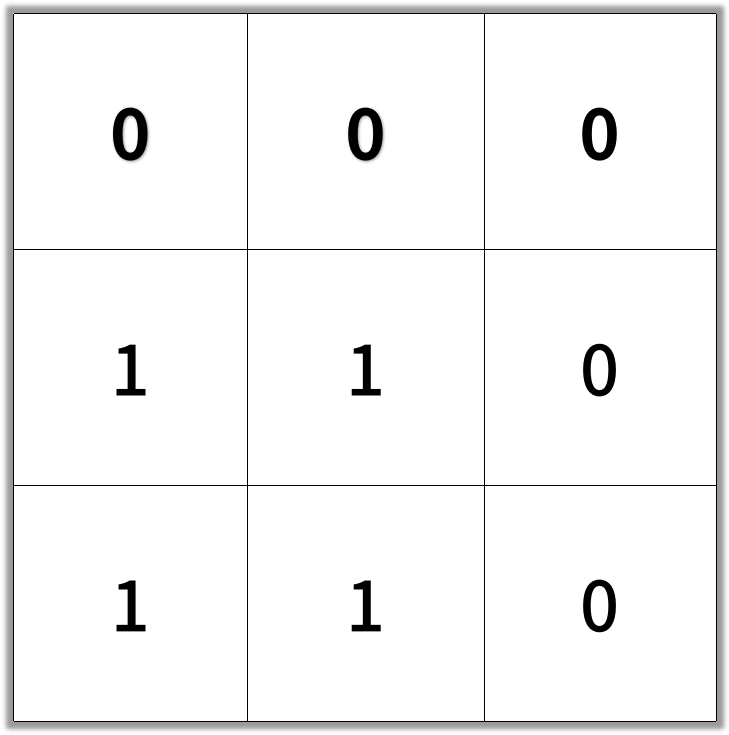


**Output:** 2

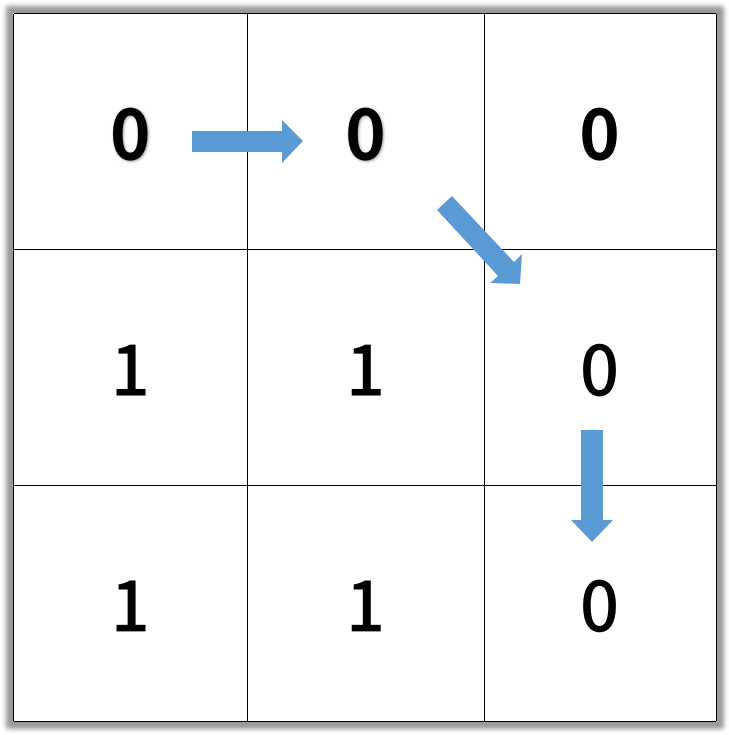


**Example 2:**

**Input:** [[0,0,0],[1,1,0],[1,1,0]]



**Output:** 4



**Note:**

1. 1 <= grid.length == grid[0].length <= 100
2. grid[r][c] is 0 or 1