

1. Robot Rescuing from Spreading Flood

50 Points

0-indexed 2D integer array grid of size $m \times n$ is given. Each cell has one of three values:

0 represents plain land,

1 represents flood,

2 represents a wall that the robot and flood cannot pass through.

The robot will be always situated in the top-left cell, $(0, 0)$, and always want to rescue the human at the bottom-right cell, $(m - 1, n - 1)$. Every minute, the robot might plan to move to an adjacent land cell.

After every move, every flood cell will spread to all adjacent cells that are not walls.

Return the maximum number of minutes **using BFS** that the robot can stay in its initial position before moving while still safely rescuing the human. If this is impossible, return -1. If the robot can always reach the human regardless of the minutes stayed, return 1000000000.

Note that even if the flood spreads to cell where the human is situated immediately after the robot have reached it, it will be considered as rescuing the human.

A cell is adjacent to another cell if the former is directly north, east, south, or west of the latter (i.e., their sides are touching).

In the examples the blue represents flood cell and the red represent wall, the empty cells are plain land

Example 1.

robot						
						human

After 3 minutes

robot						
						human

After 1 minute

robot						
						human

After 9 minutes

						rescued

Explanation

Input: grid = $[[0,2,0,0,0,0,0],[0,0,0,2,2,1,0],[0,2,0,0,1,2,0],[0,0,2,2,2,0,2],[0,0,0,0,0,0,0]]$

Output: 3

Explanation: The figure above shows the scenario where the robot stayed in the initial position for 3 minutes.

The robot is still able to safely rescue the human.

Staying for more than 3 minutes will not allow the robot to safely rescue the human.

Example 2

robot			
			human

	Robot		
			human

Explanation

Input: grid = $[[0,0,0,0],[0,1,2,0],[0,2,0,0]]$

Output: -1

Explanation: The figure above shows the scenario where the robot immediately tries to rescue the human. Flood will spread to any cell the robot moves towards and it is impossible to safely rescue the human. Thus, -1 is returned.

Example 3

robot		
		human

Explanation

Input: grid = [[0,0,0],[2,2,0],[1,2,0]]

Output: 1000000000

Explanation: Notice that the flood is contained by walls and the robot will always be able to rescue the human. Thus, 1000000000 is returned.

Constraints

m and n (not more than 100),

grid[i][j] is either 0, 1, or 2,

grid[0][0] == grid[m - 1][n - 1] == 0.

Solve the above problem using BFS.

Your code should work on other test cases apart from the above three.

Comment your code.