Game of Life

According to Wikipedia's article: "The **Game of Life**, also known simply as **Life**, is a cellular automaton devised by the British mathematician John Horton Conway in 1970."

The board is made up of an m x n grid of cells, where each cell has an initial state: **live** (represented by a 1) or **dead** (represented by a 0). Each cell interacts with its eight neighbors (horizontal, vertical, diagonal) using the following four rules (taken from the above Wikipedia article):

1. Any live cell with fewer than two live neighbors dies as if caused by under-population.
2. Any live cell with two or three live neighbors lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by over-population.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

The next state of the board is determined by applying the above rules simultaneously to every cell in the current state of the m x n grid board. In this process, births and deaths occur **simultaneously**.

Given the current state of the board, **update** the board to reflect its next state.

Note that you do not need to return anything.

Example 1:

0	1	0
0	0	1
1	1	1
0	0	0

⇨

0	0	0
1	0	1
0	1	1
0	1	0

Input: board = [[0,1,0],[0,0,1],[1,1,1],[0,0,0]]

Output: [[0,0,0],[1,0,1],[0,1,1],[0,1,0]]

Example 2:

1	1
1	0

⇨

1	1
1	1

Input: board = [[1,1],[1,0]]

Output: [[1,1],[1,1]]

Constraints:

- m == board.length
- n == board[i].length
- 1 <= m, n <= 25
- board[i][j] is 0 or 1.