01 Matrix :: Level Order

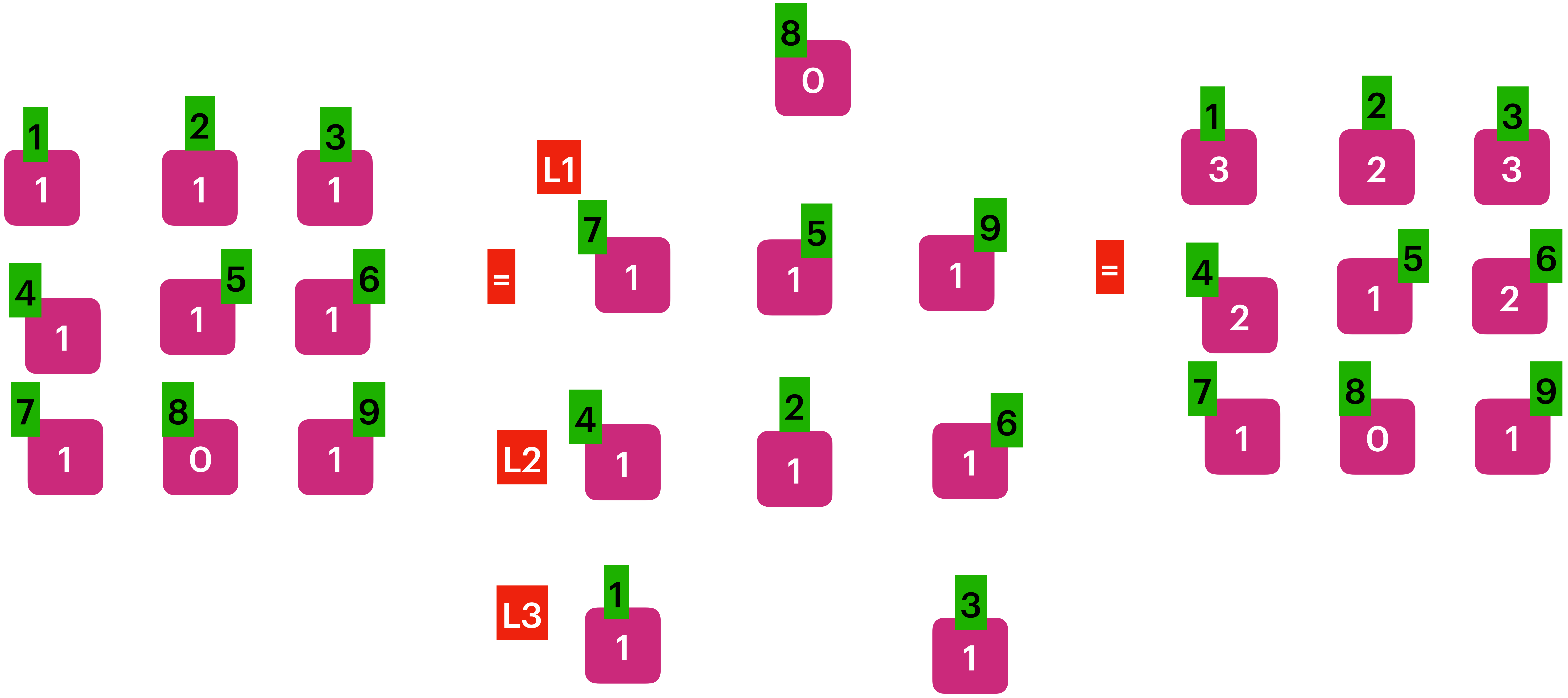
Given an m x n binary matrix mat, return the distance of the nearest 0 for each cell.
The distance between two adjacent cells is 1.

Input: mat = [
[0,0,0],
[0,1,0],
[0,0,0]
]
Output: [
[0,0,0],
[0,1,0],
[0,0,0]
]

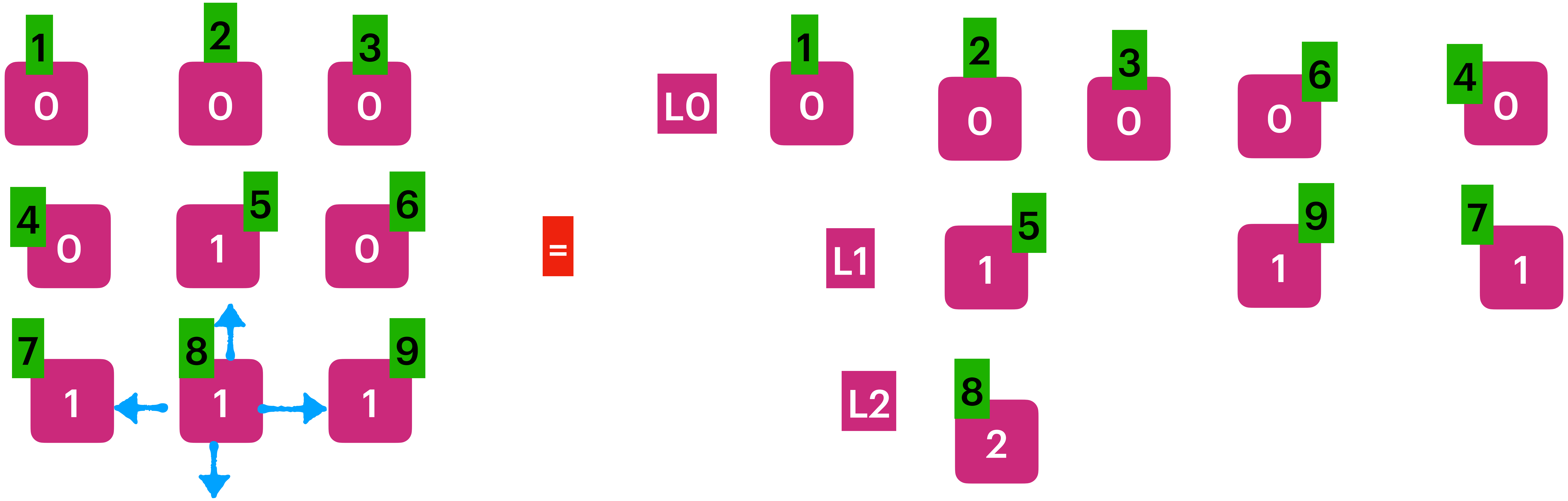
Input: mat = [
[0,0,0],
[0,1,0],
[1,1,1]
]
Output: [
[0,0,0],
[0,1,0],
[1,2,1]
]

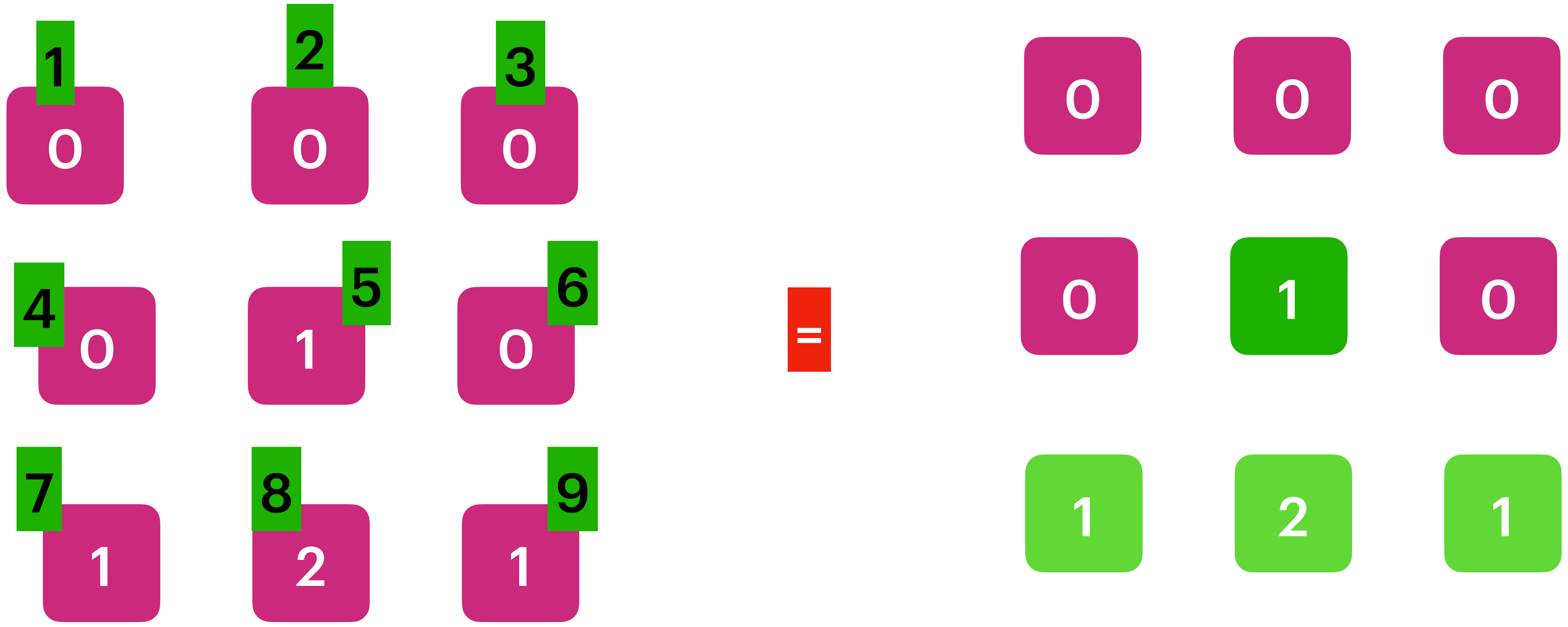
Constraints:

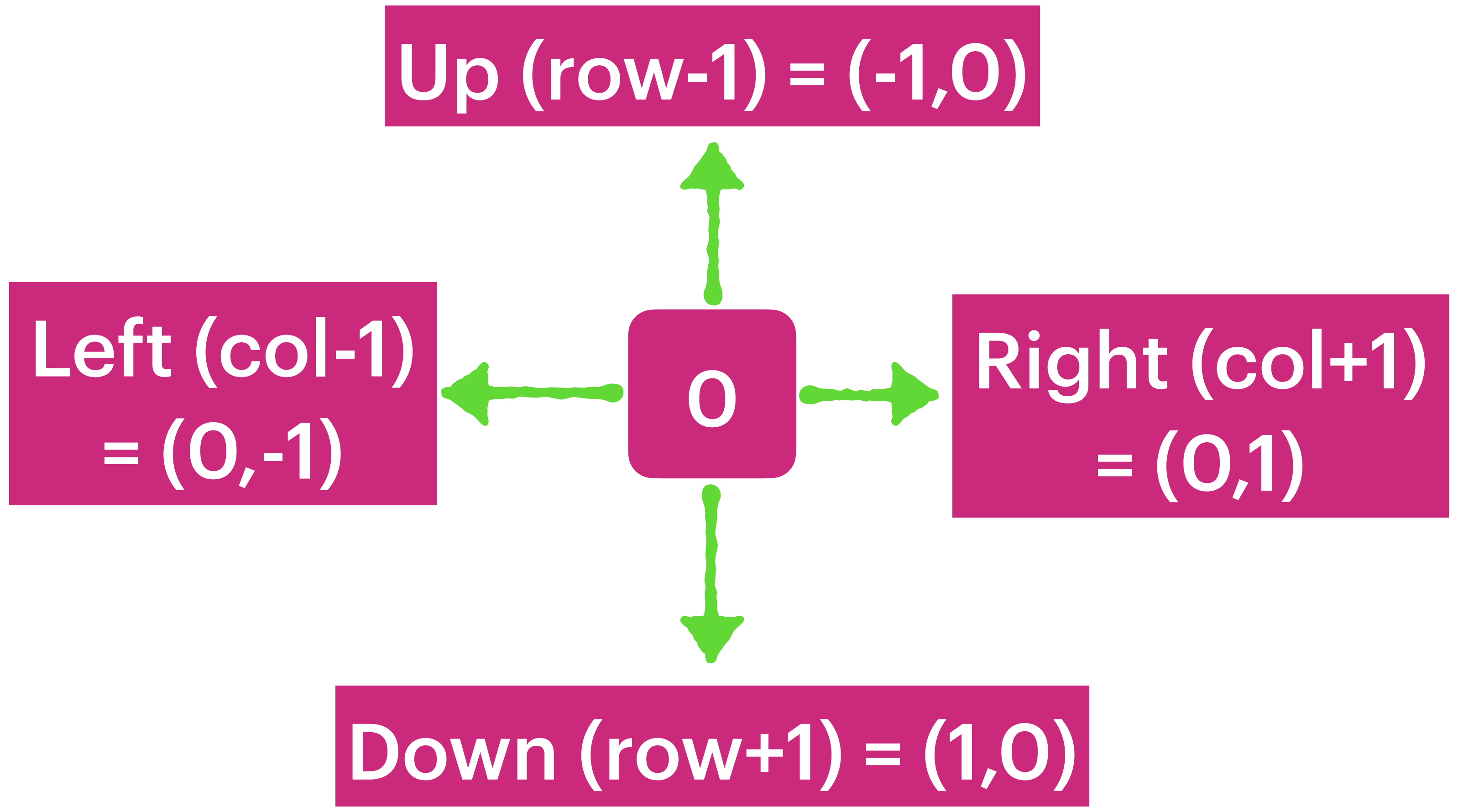
m == mat.length
n == mat[i].length
1 <= m, n <= 104
1 <= m * n <= 104
mat[i][j] is either 0 or 1.
There is at least one 0 in mat.



Output :
3 2 3
2 1 2
1 0 1


1 0 2 0 3 0
4 0 5 1 6 0
7 1 8 1 9 1
=
L0 1 0 2 0 3 0 6 0 4 0
L1 5 1 9 1 7 1
L2 8 2


1 0 2 0 3 0
4 0 5 1 6 0
7 1 8 2 9 1
=
0 0 0
0 1 0
1 2 1


Up (row-1) = (-1,0)
Left (col-1) = (0,-1)
0
Right (col+1) = (0,1)
Down (row+1) = (1,0)

Rotting Oranges

You are given an m x n grid where each cell can have one of three values:

0 representing an empty cell,
1 representing a fresh orange, or
2 representing a rotten orange.
Every minute, any fresh orange that is 4-directionally adjacent to a rotten orange becomes rotten.

Return the minimum number of minutes that must elapse until no cell has a fresh orange.
If this is impossible, return -1.

Input: grid = [[2,1,1],[1,1,0],[0,1,1]]
Output: 4

Input: grid = [[2,1,1],[0,1,1],[1,0,1]]
Output: -1
Explanation: The orange in the bottom left corner (row 2, column 0) is never rotten, because rotting only happens 4-directionally.

Input: grid = [[1]]
Output: -1
Explanation: Since the orange can never rotate.

Input: grid = [[0,0,0,0,0]]
Output: 0
Explanation: Since there are no oranges to rotate.

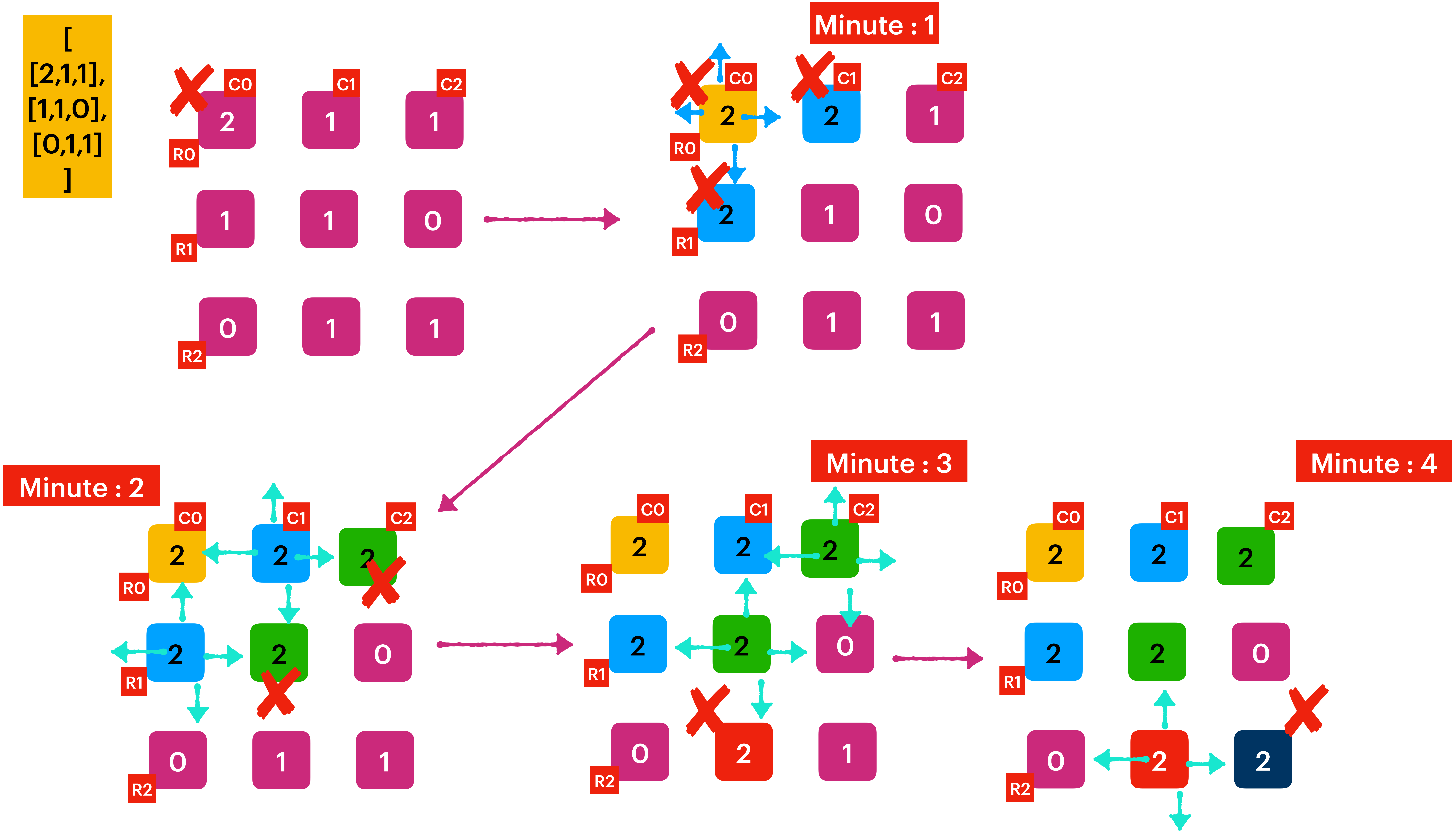
Input: grid = [[0]]
Output: 0
Explanation: Since there are already no fresh oranges at minute 0, the answer is just 0.

Input: grid = [[0,2]]
Output: 0
Explanation: Since there are already no fresh oranges at minute 0, the answer is just 0.

Constraints:

m == grid.length
n == grid[i].length
1 <= m, n <= 10
grid[i][j] is 0, 1, or 2.

[
[2,1,1],
[1,1,0],
[0,1,1]
]



	C0	C1	C2
R0	2	1	1
R1	1	1	0
R2	0	1	1

Minute : 1

	C0	C1	C2
R0	2	2	1
R1	2	1	0
R2	0	1	1

Minute : 2

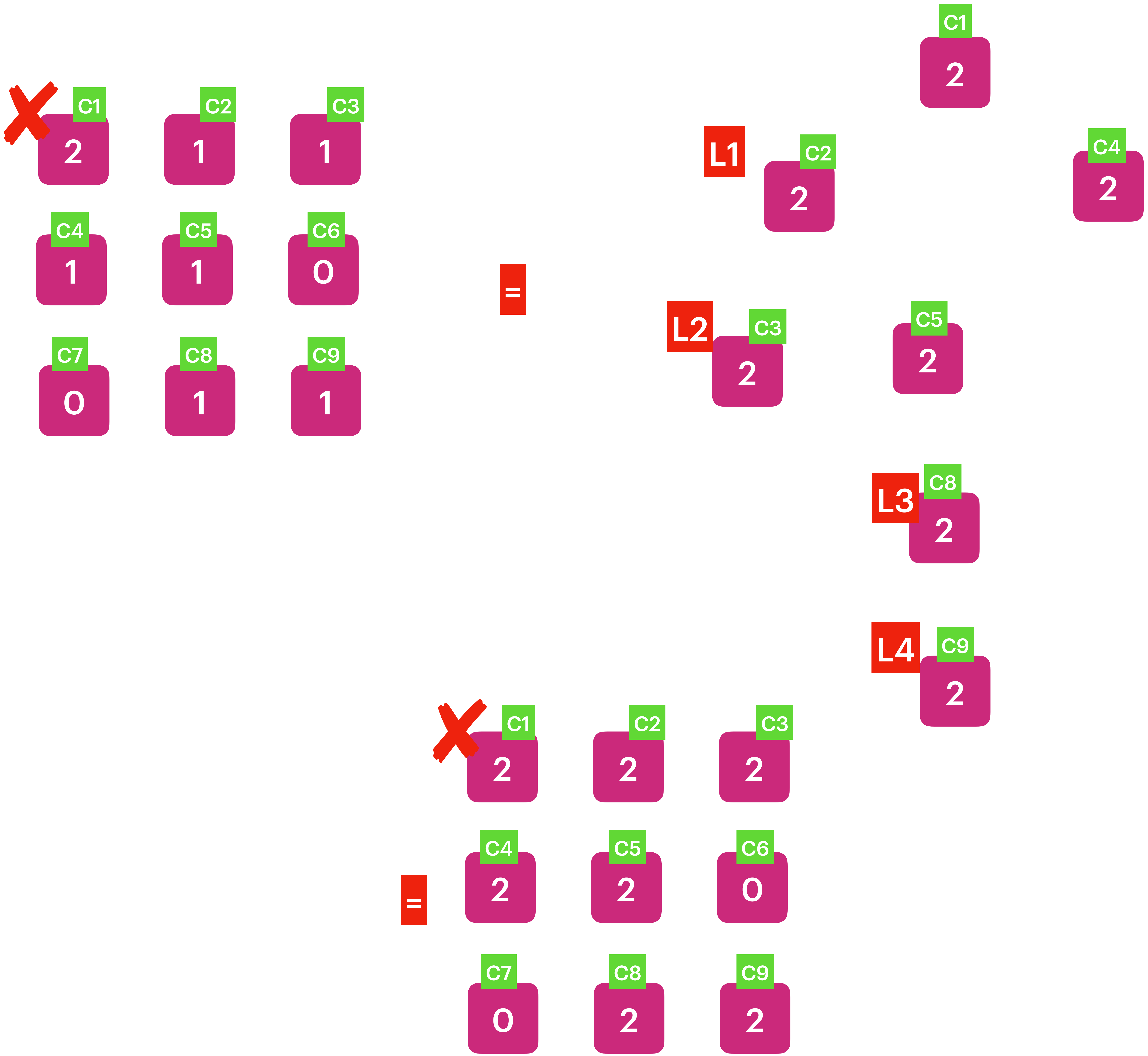
	C0	C1	C2
R0	2	2	2
R1	2	2	0
R2	0	1	1

Minute : 3

	C0	C1	C2
R0	2	2	2
R1	2	2	0
R2	0	2	1

Minute : 4

	C0	C1	C2
R0	2	2	2
R1	2	2	0
R2	0	2	2



✗ C1 2	C2 1	C3 1
C4 1	C5 1	C6 0
C7 0	C8 1	C9 1

=

C1 2

L1 C2 2

C4 2

L2 C3 2

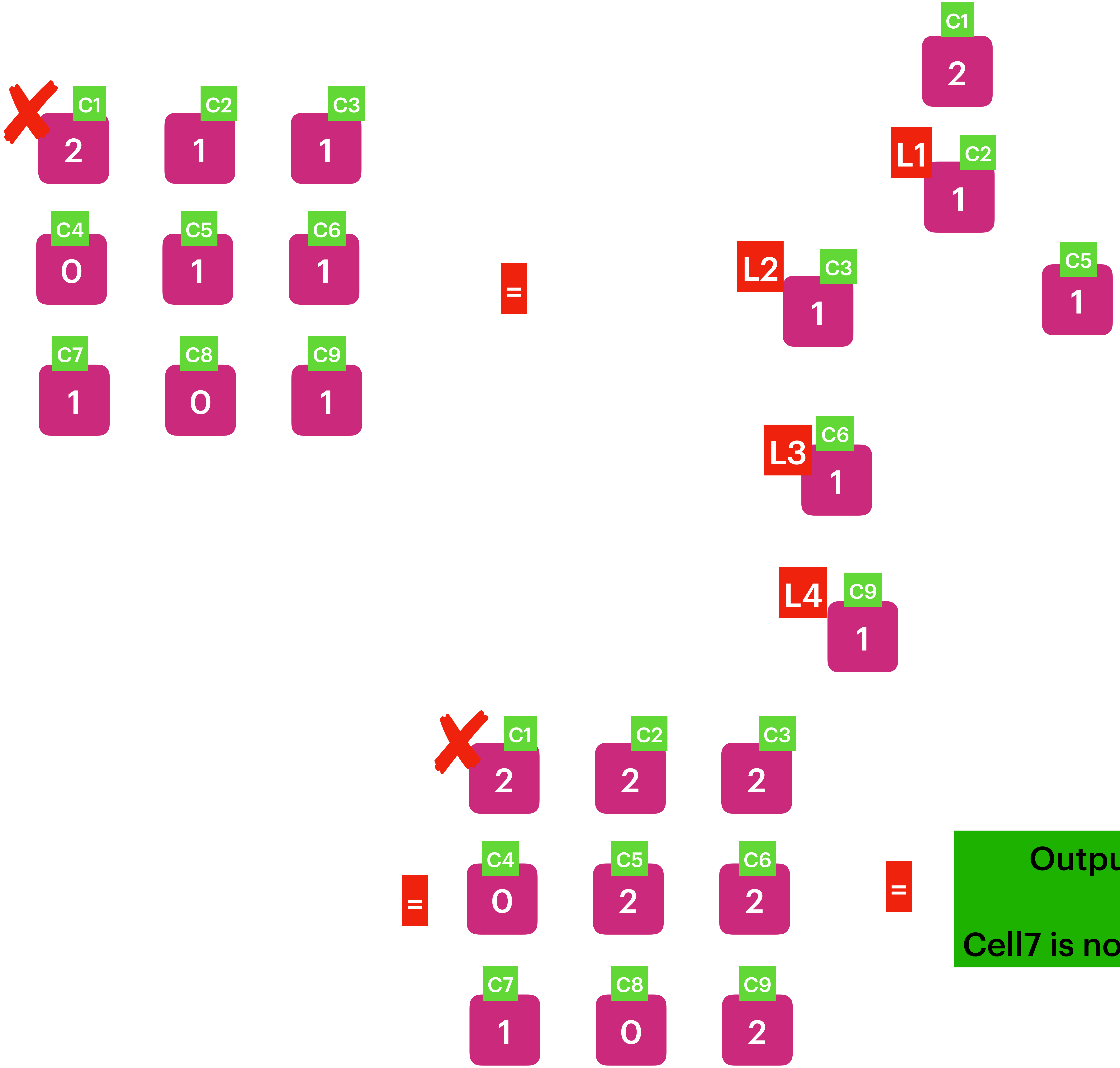
C5 2

L3 C8 2

L4 C9 2

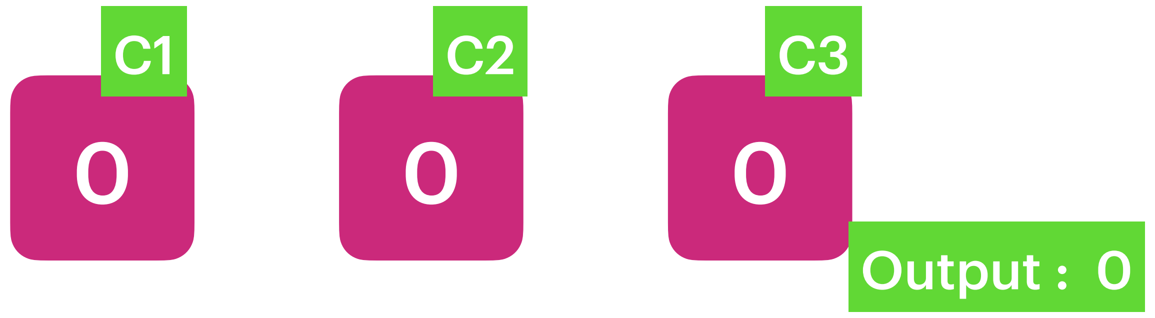
=

✗ C1 2	C2 2	C3 2
C4 2	C5 2	C6 0
C7 0	C8 2	C9 2



Output : -1

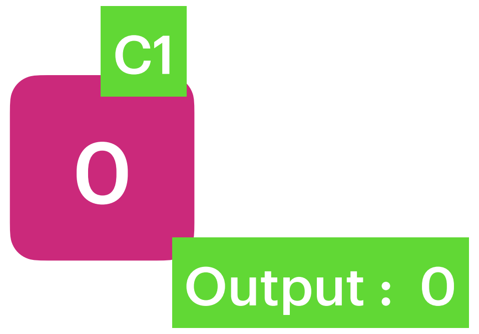
Cell7 is not rotated



C1 C2 C3

0 0 0

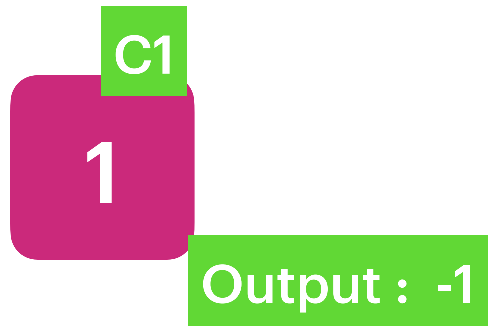
Output : 0



C1

0

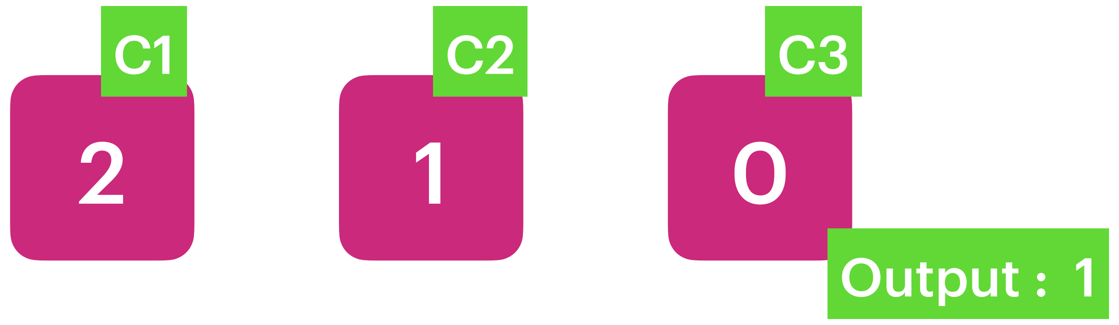
Output : 0



C1

1

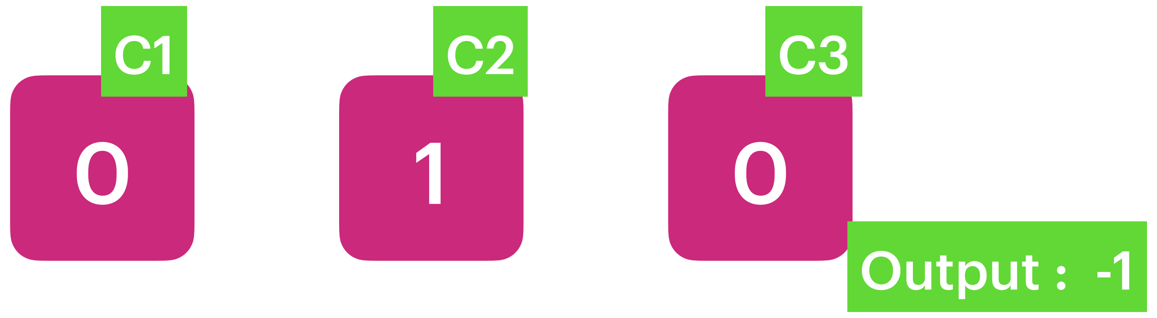
Output : -1



C1 C2 C3

2 1 0

Output : 1



C1 C2 C3

0 1 0

Output : -1