Given a square matrix grid, return whether it is a valid solution to a game of 0h n0.

This is based on the game [0h n0](http://0hn0.com/), by Martin Kool, that has the following rules:

* Let's define a *visibility* of the cell as the number of cells till first 0 or the border of the grid in all 4directions;
* The grid is a solution to a game if for each cell containing number x > 0, its *visibility* is equal to x.

Given the grid, return whether it is the solution to the game or not.

**Example**  
For

grid = [[1, 2, 0],

[0, 1, 0],

[0, 0, 0]]

the output should be  
like0hn0(grid) = true.  
Let's look at all non-zero cells:

* Cell grid[0][0] contains number 1 and has *visibility* equal to 1 (only grid[0][1] is visible from this cell)
* Cell grid[0][1] contains number 2 and has *visibility* equal to 2 (grid[0][0] and grid[1][1]are visible from this cell)
* Cell grid[1][1] contains number 1 and has *visibility* equal to 1 (only grid[0][1] is visible from this cell)

**Input/Output**

* **[execution time limit] 3 seconds (cs)**
* **[input] array.array.integer grid**

A grid of the proposed solution.

*Guaranteed constraints:*  
3 ≤ grid.length ≤ 500,  
grid[i].length = grid.length,  
0 ≤ grid[i][j].

* **[output] boolean**

Return true if grid is the valid solution and false otherwise.

**[C#] Syntax Tips**

// Prints help message to the console

// Returns a string

**string** **helloWorld**(**string** name) {

Console.Write("This prints to the console when you Run Tests");

**return** "Hello, " + name;

}

<https://codefights.com/challenge/kT2y5mZdLZa9Nw8FX/solutions>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Program

{

static bool like0hn0(int[][] grid)

{

for (int i = 0; i < grid.Length; i++)

{

for (int j = 0; j < grid[i].Length; j++)

{

if (grid[i][j] != 0)

{

int k = i - 1;

int cont = 0;

while (k >= 0 && grid[k][j] != 0)

{

cont++;

k--;

}

k = i + 1;

while (k < grid.Length && grid[k][j] != 0)

{

cont++;

k++;

}

k = j - 1;

while (k >= 0 && grid[i][k] != 0)

{

cont++;

k--;

}

k = j + 1;

while (k < grid[i].Length && grid[i][k] != 0)

{

cont++;

k++;

}

if (cont != grid[i][j])

{

return false;

}

}

}

}

return true;

}

static void Main(string[] args)

{

int[][] grid = {new int[]{1, 2, 0},

new int[]{0, 1, 0},

new int[]{0, 0, 0}};

Console.WriteLine(like0hn0(grid));

Console.ReadLine();

}

}

}

---------------INTENTO DE HACERLO MAS EFICIENTE-------------------

https://codefights.com/challenge/kT2y5mZdLZa9Nw8FX/solutions/L6sDTWCFQYdF9JJt3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static int contarFila(int[][] grid) //, int fila)

{

int[][] nueva\_grilla = new int[grid[0].Length][];

for (int i = 0; i < grid.Length; i++)

{

nueva\_grilla[i] = new int[grid[i].Length];

}

//for (int fila = 0; fila < grid.Length; fila++)

//{

// // int[] f = new int[grid.Length];

// List<int> indCeros = new List<int>();

// for (int i = 0; i < grid.Length; i++)

// {

// if (grid[fila][i] == 0)

// {

// indCeros.Add(i);

// }

// }

// indCeros.Add(grid.Length);

// if (indCeros.Count > 0)

// {

// int x = 0;

// for (int i = 0; i < grid.Length; i++)

// {

// if (x ==0 )

// {

// if (indCeros[x] > 0)

// {

// nueva\_grilla[fila][i] = indCeros[x];

// x++;

// }

// }

// else if (x - 1 >= 0 && x < indCeros.Count)

// {

// if (grid[fila][i] != 0)

// {

// if (indCeros[x] - indCeros[x - 1] - 1 > 1)

// {

// nueva\_grilla[fila][i] = indCeros[x] - indCeros[x - 1] - 1;

// }

// }

// }

// else if (x < indCeros.Count && x == i)

// {

// x++;

// }

// }

// }

// else

// {

// for (int i = 0; i < grid.Length; i++)

// {

// nueva\_grilla[fila][i] = grid.Length;

// }

// }

//}

//for (int col = 0; col < grid.Length; col++)

{

int col = 1;

// int[] f = new int[grid.Length];

List<int> indCeros = new List<int>();

for (int i = 0; i < grid.Length; i++)

{

if (grid[i][col] == 0)

{

indCeros.Add(i);

}

}

//indCeros.Add(grid.Length);

if (indCeros.Count > 0)

{

int x = 0;

for (int i = 0; i < grid.Length; i++)

{

if (x == 0)

{

if (i != indCeros[x])

{

nueva\_grilla[i][col] = indCeros[x];

}

else

{

x++;

}

}

else if (x - 1 >= 0 && x < indCeros.Count)

{

if (grid[i][col] != 0)

{

if (indCeros[x] - indCeros[x - 1] - 1 > 1)

{

nueva\_grilla[i][col] += indCeros[x] - indCeros[x - 1] - 1;

}

}

}

else if (indCeros[x] == i)

{

x++;

}

}

}

else

{

for (int i = 0; i < grid.Length; i++)

{

nueva\_grilla[i][col] = grid.Length;

}

}

}

for (int i = 0; i < nueva\_grilla.Length; i++)

{

for (int j = 0; j < nueva\_grilla[i].Length; j++)

{

Console.Write(nueva\_grilla[i][j] + " ");

}

Console.WriteLine();

}

return 0;

}

//static bool like0hn0(int[][] grid)

//{

//}

static void Main(string[] args)

{

//int[][] grid =

//{

// //new int[] { 1,1,1,1,1},

// new int[] { 5,0,0,0,3},

// new int[] { 0,0,5,0,1},

// new int[] { 0,0,0,4,0},

// new int[] { 0,4,0,0,1},

// new int[] { 0,7,7,6,0}

//};

//int[][] grid =

//{

// //new int[] { 1,1,1,1,1},

// new int[] { 5,0,0,0,3},

// new int[] { 1,0,5,0,1},

// new int[] { 1,0,0,4,0},

// new int[] { 1,4,0,0,1},

// new int[] { 0,7,7,6,0}

// };

int[][] grid =

{

//new int[] { 1,1,1,1,1},

new int[] { 0,0,0,0,3},

new int[] { 1,0,5,0,1},

new int[] { 1,0,0,4,0},

new int[] { 1,4,0,0,1},

new int[] { 0,7,7,6,0}

};

contarFila(grid);

Console.ReadLine();

}

}

}

--------------------------------------------------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static int contarFila(int[][] grid) //, int fila)

{

int[][] nueva\_grilla = new int[grid[0].Length][];

for (int i = 0; i < grid.Length; i++)

{

nueva\_grilla[i] = new int[grid[i].Length];

}

//for (int fila = 0; fila < grid.Length; fila++)

//{

// // int[] f = new int[grid.Length];

// List<int> indCeros = new List<int>();

// for (int i = 0; i < grid.Length; i++)

// {

// if (grid[fila][i] == 0)

// {

// indCeros.Add(i);

// }

// }

// indCeros.Add(grid.Length);

// if (indCeros.Count > 0)

// {

// int x = 0;

// for (int i = 0; i < grid.Length; i++)

// {

// if (x ==0 )

// {

// if (indCeros[x] > 0)

// {

// nueva\_grilla[fila][i] = indCeros[x];

// x++;

// }

// }

// else if (x - 1 >= 0 && x < indCeros.Count)

// {

// if (grid[fila][i] != 0)

// {

// if (indCeros[x] - indCeros[x - 1] - 1 > 1)

// {

// nueva\_grilla[fila][i] = indCeros[x] - indCeros[x - 1] - 1;

// }

// }

// }

// else if (x < indCeros.Count && x == i)

// {

// x++;

// }

// }

// }

// else

// {

// for (int i = 0; i < grid.Length; i++)

// {

// nueva\_grilla[fila][i] = grid.Length;

// }

// }

//}

//for (int col = 0; col < grid.Length; col++)

//{

int col = 0;

// int[] f = new int[grid.Length];

List<int> indCeros = new List<int>();

for (int i = 0; i < grid.Length; i++)

{

if (grid[i][col] == 0)

{

indCeros.Add(i);

}

}

//indCeros.Add(grid.Length);

if (indCeros.Count > 0)

{

int x = 0;

for (int i = 0; i < grid.Length; i++)

{

if (x == 0)

{

if (i != indCeros[x])

{

nueva\_grilla[i][col] = indCeros[x];

}

else

{

x++;

}

}

else if (x - 1 > 0 && x < indCeros.Count)

{

if (grid[i][col] != 0)

{

if (indCeros[x] - indCeros[x - 1] - 1 > 1)

{

nueva\_grilla[i][col] += indCeros[x] - indCeros[x - 1] - 1;

}

}

}

else if (indCeros[x] == i)

{

x++;

}

}

}

else

{

for (int i = 0; i < grid.Length; i++)

{

nueva\_grilla[i][col] = grid.Length;

}

}

// }

for (int i = 0; i < nueva\_grilla.Length; i++)

{

for (int j = 0; j < nueva\_grilla[i].Length; j++)

{

Console.Write(nueva\_grilla[i][j] + " ");

}

Console.WriteLine();

}

return 0;

}

//static bool like0hn0(int[][] grid)

//{

//}

static void Main(string[] args)

{

//int[][] grid =

//{

// //new int[] { 1,1,1,1,1},

// new int[] { 5,0,0,0,3},

// new int[] { 0,0,5,0,1},

// new int[] { 0,0,0,4,0},

// new int[] { 0,4,0,0,1},

// new int[] { 0,7,7,6,0}

//};

int[][] grid =

{

//new int[] { 1,1,1,1,1},

new int[] { 5,0,0,0,3},

new int[] { 1,0,5,0,1},

new int[] { 1,0,0,4,0},

new int[] { 1,4,0,0,1},

new int[] { 0,7,7,6,0}

};

contarFila(grid);

Console.ReadLine();

}

}

}

----------------------------------------------------------------------------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static int contarFila(int[][] grid) //, int fila)

{

int[][] nueva\_grilla = new int[grid[0].Length][];

for(int i =0; i<grid.Length; i++)

{

nueva\_grilla[i] = new int[grid[i].Length];

}

for (int fila = 0; fila < grid.Length; fila++)

{

// int[] f = new int[grid.Length];

List<int> indCeros = new List<int>();

for (int i = 0; i < grid.Length; i++)

{

if (grid[fila][i] == 0)

{

indCeros.Add(i);

}

}

if (indCeros.Count > 0)

{

int x = 0;

for (int i = 0; i < grid.Length; i++)

{

if (x - 1 >= 0 && x < indCeros.Count)

{

if (grid[fila][i] != 0)

{

if (indCeros[x] - indCeros[x - 1] - 1 > 1)

{

nueva\_grilla[fila][i] = indCeros[x] - indCeros[x - 1] - 1;

}

}

}

else if (x < indCeros.Count && x == i)

{

x++;

}

}

}

else

{

for (int i = 0; i < grid.Length; i++)

{

nueva\_grilla[fila][i] = grid.Length;

}

}

}

for (int col = 0; col < grid.Length; col++)

{

// int[] f = new int[grid.Length];

List<int> indCeros = new List<int>();

for (int i = 0; i < grid.Length; i++)

{

if (grid[i][col] == 0)

{

indCeros.Add(i);

}

}

if (indCeros.Count > 0)

{

int x = 0;

for (int i = 0; i < grid.Length; i++)

{

if (x - 1 >= 0 && x < indCeros.Count)

{

if (grid[i][col] != 0)

{

if (indCeros[x] - indCeros[x - 1] - 1 > 1)

{

nueva\_grilla[i][col] += indCeros[x] - indCeros[x - 1] - 1;

}

}

}

else if (x < indCeros.Count && x == i)

{

x++;

}

}

}

else

{

for (int i = 0; i < grid.Length; i++)

{

nueva\_grilla[i][col] = grid.Length;

}

}

}

for(int i =0; i < nueva\_grilla.Length; i++)

{

for(int j =0; j< nueva\_grilla[i].Length; j++)

{

Console.Write(nueva\_grilla[i][j] + " ");

}

Console.WriteLine();

}

return 0;

}

//static bool like0hn0(int[][] grid)

//{

//}

static void Main(string[] args)

{

int[][] grid=

{

//new int[] { 1,1,1,1,1},

new int[] { 5,0,0,0,3},

new int[] { 0,0,5,0,1},

new int[] { 0,0,0,4,0},

new int[] { 0,4,0,0,1},

new int[] { 0,7,7,6,0}

};

contarFila(grid);

Console.ReadLine();

}

}

}