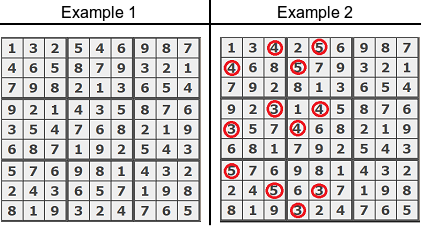
*Sudoku* is a number-placement puzzle. The objective is to fill a 9 × 9 grid with digits so that each column, each row, and each of the nine 3 × 3 sub-grids that compose the grid contains all of the digits from 1 to 9.

This algorithm should check if the given grid of numbers represents a correct solution to Sudoku.

**Example**

For the first example below, the output should be true. For the other grid, the output should be false: each of the nine 3 × 3 sub-grids should contain all of the digits from 1 to 9.



**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] array.array.integer grid**

A matrix representing 9 × 9 grid already filled with numbers.

* **[output] boolean**

true if the given grid represents a correct solution to Sudoku, false otherwise.

<https://codefights.com/arcade/code-arcade/waterfall-of-integration/tQgasP8b62JBeirMS>

static bool sudoku(int[][] grid)

{

//comprobar fila por fila

for (int i = 0; i < 9; i++)

{

HashSet<int> filaActual = new HashSet<int>();

HashSet<int> colActual = new HashSet<int>();

for (int j = 0; j < 9; j++)

{

if (filaActual.Contains(grid[i][j]))

{

return false;

}

filaActual.Add(grid[i][j]);

if (colActual.Contains(grid[j][i]))

{

return false;

}

colActual.Add(grid[j][i]);

}

}

for (int fila = 0; fila < 9; fila += 3)

{

for (int col = 0; col < 9; col += 3)

{

HashSet<int> cuadrante = new HashSet<int>();

for (int i = fila; i < fila + 3; i++)

{

for (int j = col; j < col + 3; j++)

{

if (cuadrante.Contains(grid[i][j]))

{

return false;

}

cuadrante.Add(grid[i][j]);

}

}

}

}

return true;

}