BACK

Correct Nonogram

168,654

DESCRIPTION | MY SOLUTIONS | LEADERBOARD | COMMENTS | README

CODEWRITING SCORE: 300/300

A **nonogram** is also known as *Paint by Numbers* and *Japanese Crossword*. The aim in this puzzle is to color the grid into black and white squares. At the top of each column, and at the side of each row, there are sets of one or more *numbers* which describe the runs of black squares in that row/column in exact order. For example, if you see `10 1` along some column/row, this indicates that there will be a run of exactly ten black squares, followed by one or more white squares, followed by a single black square. The cells along the edges of the grid can also be white.

You are given a square **nonogram** of size `size`. Its grid is given as a square matrix `nonogramField` of size `(size + 1) / 2 + size`, where the first `(size + 1) / 2` cells of each row and and each column define the *numbers* for the corresponding row/column, and the rest `size × size` cells define the the grid itself.

Determine if the given **nonogram** has been solved correctly.

Note: here `/` means integer division.

Example

- For `size = 5` and

```
nonogramField = [["-", "-", "-", "-", "-", "-", "-", "-"],
                 ["-", "-", "-", "2", "2", "1", "-", "1"],
                 ["-", "-", "-", "2", "1", "1", "3", "3"],
                 ["-", "3", "1", "#", "#", "#", ".", "#"],
                 ["-", "-", "2", "#", "#", ".", ".", "."],
                 ["-", "-", "2", ".", ".", ".", "#", "#"],
                 ["-", "1", "2", "#", ".", ".", "#", "#"],
                 ["-", "-", "5", "#", "#", "#", "#", "#"]]
```

  the output should be `correctNonogram(size, nonogramField) = true`;

- For `size = 5` and

```
nonogramField = [["-", "-", "-", "-", "-", "-", "-", "-"],
                 ["-", "-", "-", "-", "-", "1", "-", "-"],
                 ["-", "-", "-", "3", "3", "2", "5", "5"],
                 ["-", "-", "3", ".", ".", ".", "#", "#"],
                 ["-", "2", "2", "#", "#", "#", "#", "#"],
                 ["-", "-", "5", "#", "#", "#", "#", "#"],
                 ["-", "-", "5", "#", "#", "#", "#", "#"],
                 ["-", "-", "2", ".", ".", ".", "#", "#"]]
```

  the output should be `correctNonogram(size, nonogramField) = false`.

  There are three mistakes in the **nonogram**:

  - In the `5th` (1-based) row there are *numbers* `["-", "2", "2"]`, so there should be two runs of `2` black squares separated by at least one white square. However, there is only one run of `5` black squares.
  - In the `6th` column there are *numbers* `["-", "1", "2"]`, so there should be a run of exactly `1` black square, followed by one or more white squares, followed by another `2` black squares. However, there is a single run of `3` black squares not separated by white ones.
  - Finally, in the `4th` row there are *numbers* `["-", "-", "3"]`, so there should be a single run of exactly `3` black squares. However, there is just a 2-square run in that row.

Input/Output

- **[execution time limit] 4 seconds (js)**
- **[input] integer size**

  A positive integer, the size of the grid.

  Guaranteed constraints: