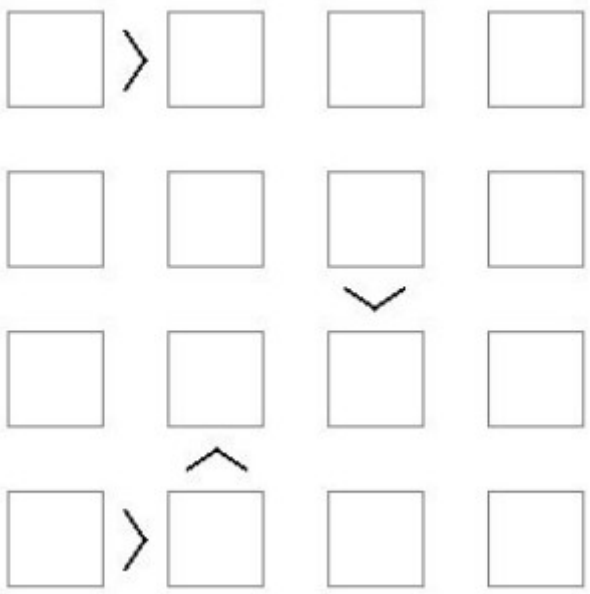
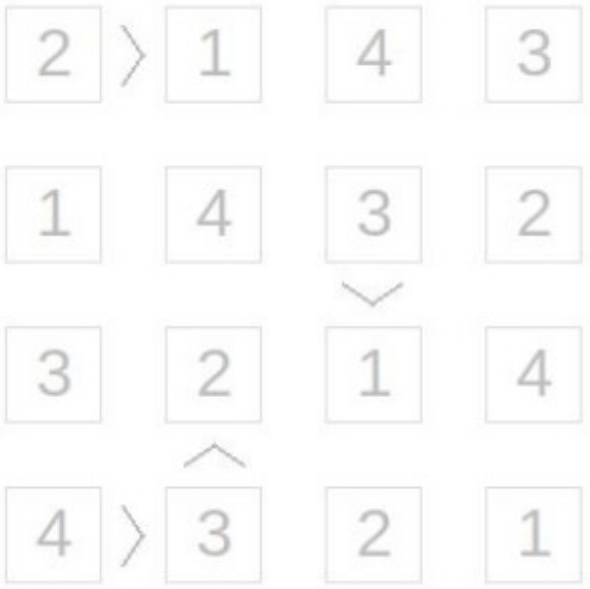
Overview:

You will write a solution validator for a board game similar to Sudoku. The game has an n by n board. Each row and column will be filled by numbers between 1 and n. Like Sudoku, each number can appear once and only once in each row and each column. But unlike Sudoku some adjacent cells have additional constraints between them. For example, in the board shown below cell (1, 1) should be greater than cell (1, 2), cell (2,3) should be greater than cell (3,3), and so on.

Example:



One valid solution:



Requirements :

You will be writing a program to validate a given solution of the puzzle. You need to come up with the data structures to represent the board, the constraints and the solution, as well as the algorithm to validate it.

For example here's the pseudo code:

```
boolean validate_solution(board, solution)
        return true if the solution is valid, false otherwise
```

For the purpose of creating a running program, you can hard code a 4x4 board with constraints, and valid/invalid solutions to validate. The constraints should be modeled as initial values in your data structure, as opposed to logic in your code (i.e. you cannot do things like if cell(1,1) > cell(1,2) then ...)