

Sudoku Generator & Solver

Erick González & Mario Contreras

<https://github.com/Nazul/sudokuSolver>

Sudoku (Number Place)

- A game based on Latin square
- First published in 1979 as “Number Place” game
- Gain popularity and started to be named as Sudoku (from Japanese for “single numbers”)

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

How Many (Valid) Sudokus Can Be Created?

- Typical Sudoku is a Sudoku of order 9
 - Each cell 1..9
 - 9x9 grid
 - 9 regions
- If we just calculate all combinations for 9x9 grid using 1..9 values, we have 9^{81} grids!
- But it is a Latin-square after all, so we need to remove some of them, then we just have 5,524,751,496,156,892,842,531,225,600 Latin squares of order 9.

How Many (Valid) Sudokus Can Be Created? (Cont.)

- Some grids are considered equivalents
 - Swap one row/column with another
- If we remove those grids that are considered equivalents, then we have 5,472,730,538 grids
 - 5.4 billion (total world population is 7.3 billion)

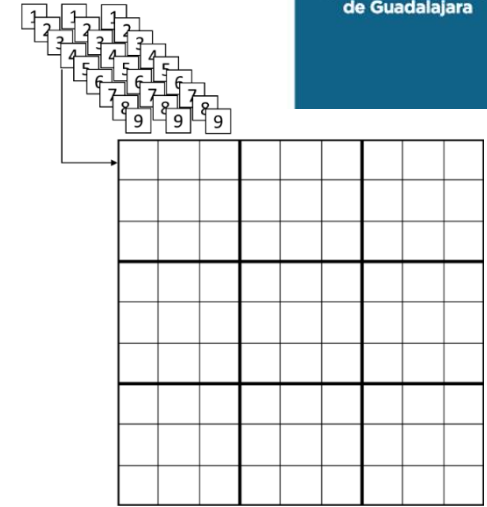
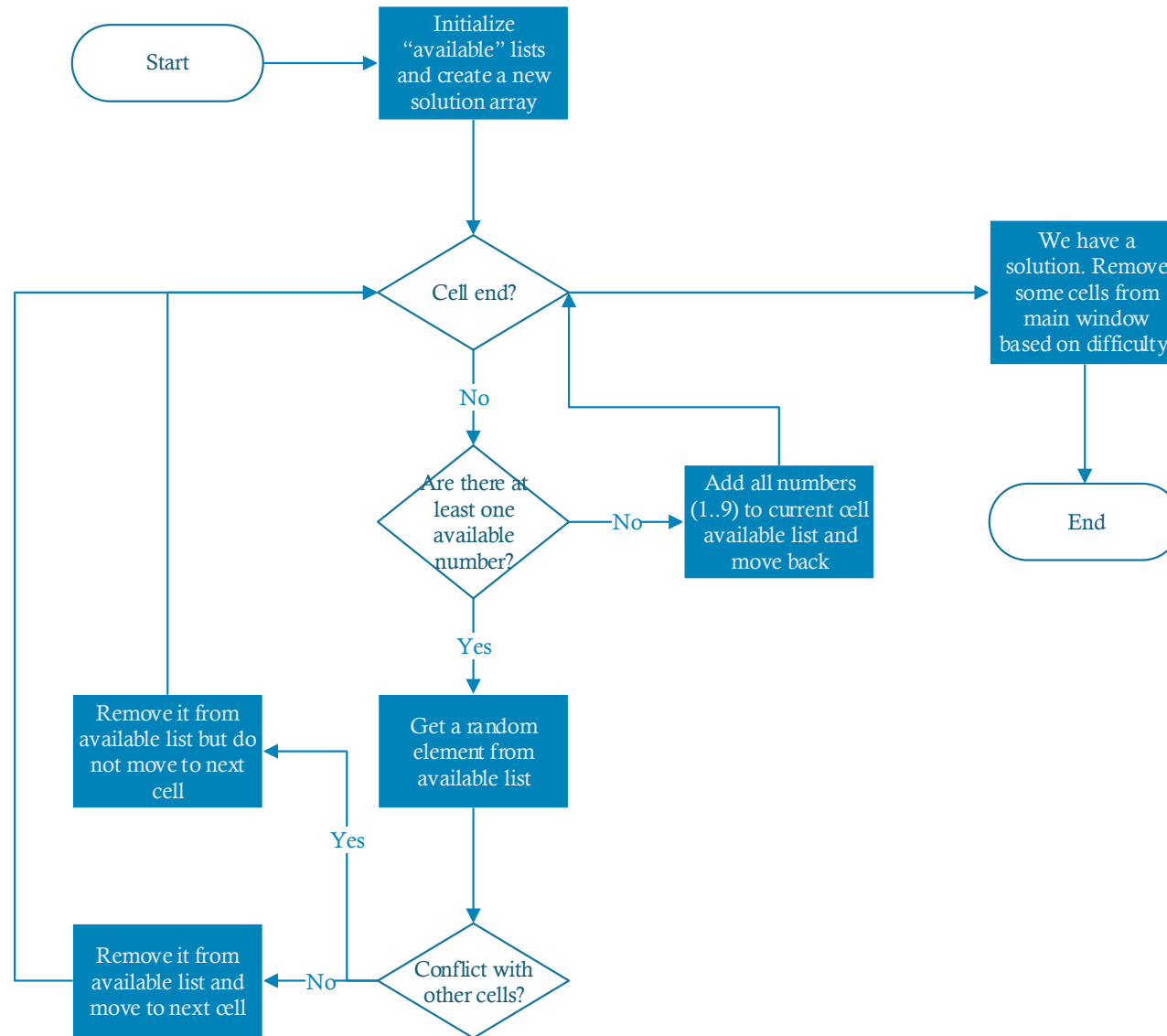
Building a Sudoku Game Application

- A new grid must be generated which is a solution
- From the solution, we remove some of the cells leaving clues only
- We can solve the game:
 - Using the previously calculated the solution
 - Or to calculate the values without using the solution

New Game / New Grid

- Create an array of lists called available
- available size is 81, which is the number of cells
- Each element starts with a list that contains 1..9
- For each cell, we randomly pick one of the available numbers
- Check if there is no conflict with the numbers in the solution (same row, same column, same region)
 - If there is no conflict, store the value in solution and move to the next cell
 - If there is a conflict, remove the value from the available list
- If we ran out of available values, we clear the current cell and move to the previous cell

New Game / New Grid (Cont.)



Solving a Sudoku Using Backtracking

- We implemented a variant that includes a letter at each cell
- Letters are S, M and L, and indicate a relation between cells
 - $S < M < L$
- For this application, each 3 horizontal cells in a region are labeled with this rule

S 4	M 5	L 6
S 3	L 8	M 7
S 1	M 2	L 9

- We check for all cells that are not clues and create a list for each one
- List will be populated with all the possible numbers based on the letters restriction
 - Worst cases:
 - S: 1..7
 - M: 2..8
 - L: 3..9

[illegible]

Solving a Sudoku Using Backtracking (Cont.)

- For each list we take the first element and review if there is no conflict
- If there is conflict, we remove it from the list and add it to an error list
- If we checked all lists without finding errors, we have a solution
- If we have errors, we move back and select a different element

ITESO



Universidad Jesuita
de Guadalajara

Demo

ITESO



Universidad Jesuita
de Guadalajara

Q&A