

In this assignment, you will be writing a program that can solve Sudoku puzzles, using the ILOG constraint programming library. Sudoku puzzles consist of a 9x9 grid with a few cells filled in with number 1-9. You must fill in the rest of the grid to satisfy the following property using the numbers 1-9. Every row and every column must contain all the numbers 1-9. Also the 9 3x3 boxes that are an obvious sub division of the 9x9 grid must contain all the numbers 1-9.

If you are unfamiliar with how the puzzle works there are sites that have examples of puzzles (some with solutions). Also Newspapers like the Brown Daily Herald often have the puzzle.

- <http://sudoku.com/>
- <http://www.sudokufun.com/>

Provided Environment

Our Sudoku puzzles are encoded in a space delimited list in a file, where the first number in the file is the size of the board. The support code currently reads in the board (as its first and only argument), runs the solve method, then prints the solved board.

Your Task

You must write a constraint program using ILOG solver to find correct solutions to the Sudoku puzzles. Even though the standard board is 9x9, your solver should be able to handle boards of different sizes (i.e. 4x4 and 16x16), filling them with the numbers 1-n where n is the size of the board.

Extra Credit

Generate your own Sudoku puzzles. Puzzles must have unique solutions. This is a harder problem, but can be a lot of fun.

Support Code

When you are ready to begin coding, copy the support code from `/course/cs149/asgn/sudoku/`.

You can compile by typing:

```
cslab0a sudoku% make
```

and run by typing:

```
cslab0a sudoku% ./sudoku
```

We have included functions to parse puzzle files, print boards, and test if a board is a valid solution to some Sudoku puzzle.

The only method you really need to work with is `solvePuzzle()`

Puzzle Files

A few puzzle files have been provided for you in `sudoku/puzzles`. The file format that we use (and that the parser we give you recognizes works like this. The first line of the file is the size of the puzzle (always 9 for us). There are then size numbers of lines each with size numbers in them. Zero represent blank squares (solved puzzles should not have blank squares). See one of the puzzle files for an example.

Documentation

2

ILOG Solver example files:

`file:///com/ilog/solver62.lintel/solver62/examples` ILOG Solver Documentation:

`file:///com/cplex/doc/html/en-US/Content/Optimization/Documentation/CPLEX/_pubskel/index.html`

Online ILOG reference:

`http://www.lkn.ei.tum.de/arbeiten/faq/man/ILOG/CONCERT/concert20/refsolver/`