

CSCI-1200 Computer Science II — Spring 2006

Lab 10 — Sudoku Sets

In this lab we will write a simple Sudoku puzzle solver using the `set` container class we saw in lecture last week. Sudoku puzzles are typically played on a 9x9 grid in which some of the cells have been initialized and many of the cells left blank. The instructions are simply to: “Fill in the grid so that every row, every column, and every 3x3 box contains the digits 1 through 9”. If you are unfamiliar with Sudoku puzzles you may read about them here:

<http://en.wikipedia.org/wiki/Sudoku>

Please download the following files which contain a partial implementation of a Sudoku puzzle solver, and then turn off your internet connection.

http://www.cs.rpi.edu/academics/courses/spring06/cs2/labs/10_sets/sudoku.h
http://www.cs.rpi.edu/academics/courses/spring06/cs2/labs/10_sets/sudoku.cpp
http://www.cs.rpi.edu/academics/courses/spring06/cs2/labs/10_sets/puzzles.txt

Checkpoint 1

First familiarize yourself with the code and how we will be representing partially solved Sudoku puzzles. The 2D grid is stored using nested vectors. The elements of the inner vector are sets of integers, which represent the possible values for that cell. We know we’re done when each cell has exactly one possible value. If in the course of solving the puzzle we end up with no values in a particular cell, we know the puzzle is impossible to solve. The program handles 4x4 puzzles with 2x2 blocks and 9x9 puzzles with 3x3 blocks (and even larger puzzles too!)

Finish the implementation of the `Sudoku` class constructor and the `Sudoku::IsSolved` member function. At this point you should be able to compile and test your program on the input file (using file re-direction). The provided code for this assignment will print out the contents of each cell in the puzzle. At this point, every number (1-4 on the small puzzles and 1-9 on the large puzzles) will be an possible value for each unspecified cell in the puzzle.

To complete this checkpoint: Show a TA your program and how it behaves on the sample puzzles.

Checkpoint 2

In the next step you will propagate information from the values in the grid which are known. For example if “7” is the only remaining choice for the cell (i, j) then we can remove “7” from all the other cells on the i th row and the j th column and the corresponding block. Follow the structure in the code provided and complete the `Sudoku::Propagate` member function. Test your program on the provided input.

To complete this checkpoint: Show a TA your completed program. Be prepared to discuss why it can only solve some of the puzzles. Think about how you would extend the program to solve some of the more difficult puzzles.