

# INTRODUCTION TO COMPUTER SCIENCE II

## PROJECT 3

### Sudoku!

## 1 The game of Sudoku

Sudoku is a popular game giving crossword puzzles a run for their money in newspapers.<sup>1</sup> It's a game well suited for computers because it is a matter of finding legal placements for numbers on the board; it is an example of a *constraint satisfaction problem*.

**The game:** Sudoku is typically played on a  $9 \times 9$  grid. Initially, a few of the cells of that grid contain a numbers, all integers between 1 and 9. The goal for the player is to fill in the remaining cells of the grid, also with integers between 1 and 9, given the following constraints:

- Each value (1 through 9) must appear exactly once in each row.
- Each value (1 through 9) must appear exactly once in each column.
- Each value (1 through 9) must appear exactly once in each  $3 \times 3$  *sub-grid*, where the  $9 \times 9$  board is divided into 9 such sub-grids.

A *valid* Sudoku game begins with initial numbers that, when combined with the constraints above, admit to **exactly one** complete solution. That is, the initial numbers cannot make it impossible to fill in the board legal, nor can they allow multiple solutions.

## 2 Your assignment

### 2.1 Copying files from my directory

You must obtain some files in order to get started on this project. Do whichever of the follow applies to you:

- **On remus/romulus:** Copy the files from my public directory:

```
cp ~sfkaplan/public/COSC-112/project-3/* .
```

- **On your own computer:** Click here to download a zip file of the files you need.

These files are:

- `Sudoku.java`: A static class that contains the `main()` method that will drive the program (`Sudoku.java`), as well as code to read in Sudoku *board files* (see below).

---

<sup>1</sup>When students stop knowing what *newspapers* are, I will know that I'm old and should retire.

- `SimpleStack.java`: The definition of the `SimpleStack<T>` interface.
- `easy.board`, `medium.board`, `evil.board`: The Sudoku board files, each of which contains a proper, initial puzzle. The zero values are placeholders for the empty positions on the board to be filled in.

## 2.2 The overall goal

You need to write:

1. **`Solver.java`**: Which contains, in the static `Solver` class, the single, public method, `solve()`. It solves the puzzle by attempting to search the space of possible value placements in the empty positions, **using a stack to track its progress and backtrack where needed**.
2. **A `SimpleStack` implementation**: Some container class that implements the `SimpleStack` interface. To be used by the solver.
3. **Any other supporting classes/objects**: You may want to create supporting object types, such as one that stores a coordinate pair that constitutes a position on the board.

When you have written these pieces, your program should execute like so:

```
(remus)~/project-3> java Sudoku medium.board
Initial board:
0 0 0 0 0 0 0 0 7
0 0 0 0 7 0 3 0 8
0 0 0 5 0 4 0 6 0
0 0 0 0 0 0 0 8 0
7 1 0 0 9 0 0 0 5
8 0 0 0 1 5 9 0 0
3 0 0 0 0 0 0 0 0
0 0 8 9 4 0 6 3 0
0 2 7 6 0 3 0 0 0
Solved board:
6 8 1 2 3 9 5 4 7
2 4 5 1 7 6 3 9 8
9 7 3 5 8 4 2 6 1
5 9 4 7 6 2 1 8 3
7 1 6 3 9 8 4 2 5
8 3 2 4 1 5 9 7 6
3 6 9 8 2 1 7 5 4
1 5 8 9 4 7 6 3 2
4 2 7 6 5 3 8 1 9
```

## 2.3 Testing your code

There are three Sudoku board files:

- `easy.board`—The easiest of the puzzles to solve. Start with this one.
- `medium.board`—Fewer initial values are provided than in `easy.board`, making the search a bit harder.
- `evil.board`—A minimal number of values are provided. This solution requires backtracking to work correctly. (That is, certain values will be placed, but their incorrectness will not be evident until much later in the search.)

## 3 How to submit your work

Use the CS submission systems to submit your work. Specifically, you will need to submit all of your source code files. You may use either of the following two methods, while connected to `remus` or `romulus`, to use the submission system:

- **Web-based:** Visit the submission system web page.
- **Command-line based:** Use the `~lamcgeoch/submit` command at your shell prompt.

**This assignment is due on Wednesday, Dec-14, 11:59 pm.**