

Sudoku - A Programming Competition

Mike Spivey, February 2005

This competition is open to all current Oxford undergraduates, with cash prizes of £150 kindly funded by Data Connection Ltd. The competition is for a program (written in any reasonable programming language) that solves instance of the 'Sudoku' puzzle, using input and output formats that are described below. Entries must be handed to the receptionist at the Computing Laboratory by noon on Monday of the second week of Trinity Term, 2005.

Sudoku is a puzzle based on a 9×9 grid that is divided into nine smaller grids of 3×3 squares each. When the puzzle is solved, each small grid and each row and column of the whole puzzle should contain each digit from 1 to 9. Here is an example puzzle:

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

More puzzles may be found every day in *The Times* and the *Daily Telegraph*.

How to enter: Each entry to the competition must be a program that reads a Sudoku puzzle from standard input and outputs the solution to standard output, so that the program could be invoked with the command `program <input >output`. The input to the program is a text file of nine lines, each containing nine characters from the set `{'1', ..., '9', '.'}`, with `'.'` indicating a blank to be filled in. Thus the input for the example puzzle would be the nine lines shown at the right. If the puzzle has one or more solutions, then the output from the program should be one of the solutions, also represented by nine lines of nine digits each. If

there is no solution, then the program should produce no output. Entries will be judged on their speed and accuracy in solving Sudoku problems, but also on the cleverness and originality of the algorithms used.

We will accept as entries both programs written to run on a PC under Linux, and programs that run under Windows. Entries should be in the form of a DOS-format floppy disk or a CD-ROM, labelled with your name, college and e-mail address, and containing the following files:

- README, a 1-2 page description of the methods used in the program;
- the program itself in executable form;
- source code for the program;
- if possible, a Makefile that automates the process of building the program from the source code;
- up to six files named `puzz[1-6]` containing original puzzles in the format described above.

Extra points will be given to entries with puzzle files that some of the other programs do not succeed in solving.

If the supplied executable file does not work, we will attempt to rebuild your program from source code. We will do our best to fix any trivial problems, but we cannot guarantee to succeed. Unfortunately, we can't promise to install proprietary compilers in order to rebuild your program.

mike@comlab.ox.ac.uk