Philip A. Cobb* (phcobb@prodigy.net), Queensborough Community College, 222-05 56 Avenue, Bayside, NY 11364. A Computer Algorithm for Solving Sudoku.

This talk will present an algorithm to solve sudoku puzzles by computer. One array holds the numbers in the puzzle itself. A three-dimensional array \( p(i, j, k) \) is set to 0 or 1 according to whether the \((i, j)\) entry can or cannot be \(k\). If the number \(k\) appears in a row, column, or three-by-three box, then the other entries in the column, row, or box can’t be \(k\). More subtly, if \(k\) is known to be in a row of a box even though the exact location is not immediately decidable, we may deduce that \(k\) is not in any other position in that row.

We then scan each row column, and box, counting the possible locations for \(k\). If one, we place \(k\) there. If none, there is a contradiction. If no guess was made, there is no solution. Otherwise, the guess is wrong. Then we can fix one location in the \(s(i, j)\) array and count how many numbers are possible there.

If deductions were made the process is iterated if necessary. If no deductions were made in the previous cycle, we make a guess in the first blank cell. If further deductions lead to a contradiction, the guess and its consequences are erased and a new guess is made. If necessary another guess is made. If the guess leads to a solution, we may stop or make an another guess to search for other solutions, if any. (Received September 01, 2006)