

Projectkeuze Programming Expert

Groepsleden: Bram Van Vleymen, Joran Claessens

Klas: 3AOND

Programmeertaal: Java

Link naar Github: <https://github.com/Raidok/Sudoku-solver>

Bij dit project wordt gebruik gemaakt van het backtrack algoritme. Hierbij moeten niet alle oplossingen bekeken worden. Bij dit algoritme zal het programma een aantal keuzes moeten maken. Wanneer het programma op een dood spoor terecht komt, zal het programma terug naar het begin gaan en andere keuzes proberen te zoeken die wel werken. Dus bij deze sudoku solver zal hij nagaan of hij deze waarde in een bepaalde cel kan gooien. Indien niet, dan zal hij verder kijken voor een andere waarde. Zo zal hij die lus blijven afgaan tot als hij een oplossing gevonden heeft. Hieronder is de algemene methode die deze sudoku solver gebruikt:

```
private boolean solve(int size, int count) {
    int row = count / size;
    int col = count % size;
    if (count++ == (size*size)) return true;

    if (cells.getCell(row, col) == 0) { // is cell preset or not?
        for (int val = 1; val <= size; val++) { // loop through valid values
            if (isLegal(row, col, val)) { // can this value be put in current cell
                cells.setCell(row, col, val); // sets cell's value
                if (solve(size, count)) return true;
            }
        } // end of loop through valid values
        cells.setCell(row, col, 0); // didn't find suitable value, reset cell
    } else { // next cell if this one is not empty
        if (solve(size, count)) return true;
    }
    return false;
}
```