SudokuAI

# Artificial Intelligence 3.4 - Informatica B  
# Tom Schoonbeek 2032257 & Thomas Buurstede 2122226  
# Avans Breda 2020-2021

Algorithm designed to solve the killer-variant of the Sudoku puzzle by using AI programming techniques.

How does it work?

1. The algorithm is fed a solution grid (9x9) containing only zero’s, and a **killer** grid (9x9) containing combinations of matching numbers called **cages**.
2. A **heuristic** is formed by analysing the complete killer grid and sorting the cages based on quantity of squares, from smallest to largest. This is done because the smallest squares have the least possible combinations. Solving these early on will decrease the amount of possible combinations drastically. Adding this heuristic leads to **Informed Search.**
3. Using the priority-based queue, the next blank square to be filled in will be determined. If no blank squares are present, the Sudoku is complete.
4. A number from one to nine is filled in the blank square, after successfully passing the list of constraints:
   1. Horizontal constraint: 1-9 may only appear once per row
   2. Vertical constraint: 1-9 may only appear once per column
   3. Square constraint: 1-9 may only appear once in the nine 3x3 grids on the board
   4. Killer constraint:
      1. All values in a cage must add up to the assigned cage value
      2. A number may only appear once per cage
5. To apply the killer constraint, a method is called utilizing **Depth-first Search**, optimally determining neighbouring killer cage matches for each blank square selected, without having to loop through unnecessary grid locations:
   1. Left-directional Search
   2. Right-directional Search
   3. Up-directional Search
   4. Down-directional Search
6. The continuous output is measured by prints in the console.
7. Once there are no more blank spaces, the solution is printed out in a visual 2D Sudoku grid.

*PS: A 2D continuous output is available by commenting in lines 55 and 56. Do not forget to comment out line 52 afterwards.*

**DISCLAIMER: Input constraints**

**Due to limitations, only Killer Sudoku’s containing NO square cages (2x2 or more) are accepted. Furthermore, cages containing the same cage value CANNOT be neighbours.**