FACULTEIT INGENIEURSWETENSCHAPPEN
EN ARCHITECTUUR

Algorithms and datastructures
Solving sudoku puzzles with backtracking

April 3, 2020

1 Introduction

In this lab session, we will implement the backtracking algorithm to solve sudoku puzzles. The objective of a sudoku puzzle is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 subgrids that compose the grid contain all of the digits from 1 to 9.

There are 6,670,903,752,021,072,936,960 possible combinations in a 9x9 sudoku. Even if a large part of the sudoku is already provided, there are still too many possibilities for a brute force solution. Backtracking provides us with an efficient optimization. Backtracking is a form of depth first search where we build solutions iteratively and try to abort as soon as possible if we detect that the current partial solution will not result in a valid final solution.

2 Assignment

- The given `Sudoku` class contains the code to read a sudoku from a file and to print the contents.
- Implement the `isSolved()` method that checks whether the sudoku is completely solved.
- Implement backtracking in the `solve()` method.
- You can check the functionality on the puzzle provided in `1.txt`.
- The `sudokus` folder contains 50 different puzzles. Solve them all and calculate the sum of the three digit number in the upper left corners of all 50 puzzles. This sum should be 24702 if you solved all puzzles correctly (see https://projecteuler.net/index.php?section=problems&id=96).