CSCI5240 - Combinatorial Search and Optimization with Constraints
Fall 2015

Teacher: Prof. Jimmy Lee

Assignment 1

Due: 5 October, 2015 (Monday)

1. *Graph Coloring Problem*

   The *Graph Coloring Problem* is one of the most important fundamental problems in computer science. Given a graph and a set of available colors, we aim at finding a color assignment for all vertices such that every vertex is assigned one color and adjacent vertices are assigned different colors. Figure 1 shows a small instance with 7 vertices, 8 edges and 3 given colors: *red*, *green*, and *blue*.

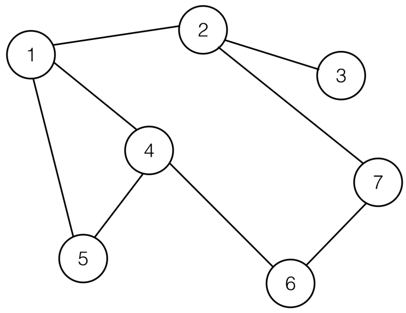


Figure 1: A Graph Coloring Problem instance

   (a) Give one CSP model to model the problem in Figure 1, using one variable for each vertex in the graph.

   (b) Give one CSP model to model the problem in Figure 1, using one variable for each color. (Hint: variables can take *sets* as values.)

   (c) Show how a solution to the first model can be transformed to a solution to the second model, and vice versa.

2. *Killer Sudoku*

   Sudoku is a puzzle that requires a player to complete a partially filled 9×9 grids with numbers 1 to 9 such that each row, each column and each of the nine 3×3 boxes (shown in Figure 2 with thick border lines) contain different numbers.

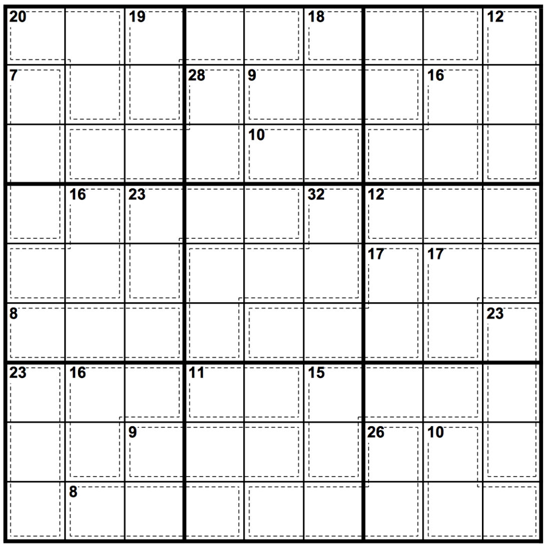
   Now we encounter a variant of the traditional Sudoku problem. The grids are now all empty but associated with a set of *cages*. A cage is a set of contiguous cells (surrounded by dashed lines) and a total. In addition to the Sudoku requirements, numbers in the cells of a cage must also add up to the given total. Such a problem is called *Killer Sudoku Problem*.

   Give one possible CSP model for the given Killer Sudoku Problem in Figure 2. Implement the model in Gecode 4.4.0 and find a solution.

3. *Imperfect Squared Square*

   The *square placement problem* (also called the *squared square problem*) is to pack a set of squares with given integer sizes into a bigger square (usually called the *master square*) in such a way that:

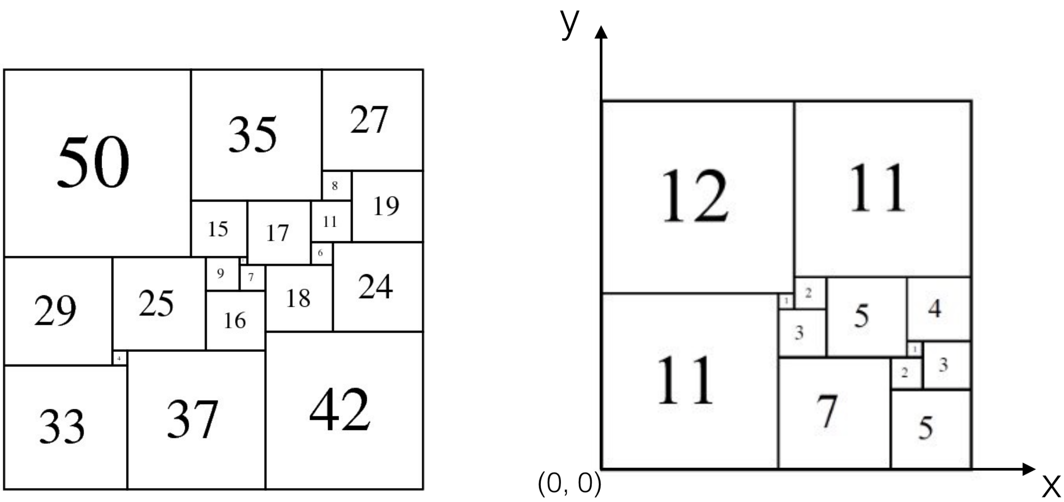
   - No squares overlap with each other;
   - All square borders are parallel to the border of the master square;
   - The sum of the square surfaces is equal to the surface of the master square.



Copyright (c) 2015, killersudokuonline.com

Figure 2: A Killer Sudoku Problem instance

If all smaller squares have different sizes, the master square is called a *perfect squared square (PSS)* and otherwise an *imperfect squared square (ISS)*. The *Order* of a squared square is defined as the number of small squares it contains. Figure 3(a) shows a PSS of order 21 while Figure 3(b) shows an ISS of order 13. Even when we know the sizes of smaller squares that constitute a master square, it is not trivial to know how to pack them.



(a) A PSS of order 21

(b) An ISS of order 13

Figure 3: Examples of squared squares

(a) One natural modelling of the placement problem is as follows: take the bottom left vertex of the master square as origin and set up a rectangular coordinate system, as already shown in Figure 3(b). For each smaller square, we use two variables x and y, which together form the coordinates of the bottom left vertex of the square, to specify its position within the master square. Give a CSP model of the placement problem in Figure 3(b) using these variables.

(b) Implement the model in Gecode 4.4.0 **using** *and* **without using** the **nooverlap** constraint provided, and find a solution where two squares of size 11 must be located at the bottom left corner and top right corner of the master square, in both implementations.