Charles Weng

112334040

CSE 526 - Final Project Design

ICLP 2020 Programing Contest

Part I. Problem and Plan

1. The goal of this project is to program solutions to the skyscraper puzzle (the simpler puzzle) and the first of the 5 puzzles in the ICLP programing contest, Aquarium (5). Originally, it was two problems from the ICLP contest, but I am not familiar with Clingo and will be using skyscrapers as a comfort puzzle. Added Masyu puzzle (the fourth ICLP puzzle).

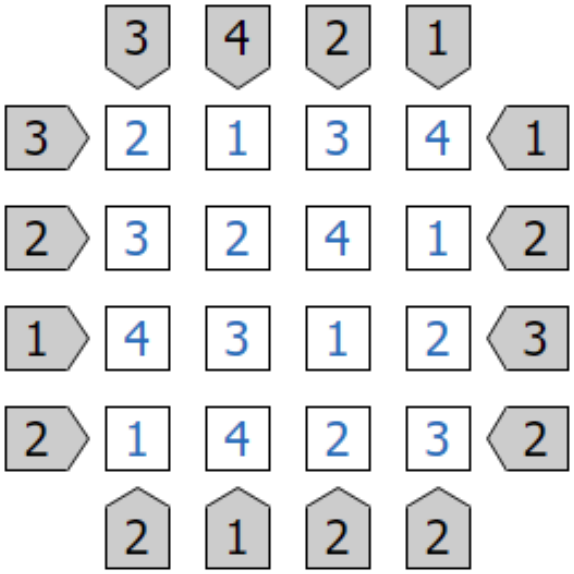
   The Skyscraper puzzle is a puzzle of determining the heights of a square of skyscrapers. Each row/column has a building of each height exactly once. You are given hints in the form of the number of skyscrapers you can see from each side of each row/column (you can see a skyscraper if there is no skyscraper of greater height closer to you). For harder puzzles some of these clues are omitted.
   The Aquarium puzzle is a puzzle of determining the water level of a group of tanks arranged to fit in a square. For each tank, the water tiles are level and follow downwards gravity (ie. you cannot have a water and air tile in the same

tank and same row or water above air in the same tank). Tanks should also be connected (you can travel orthogonally from any given tile in a tank to any other tile in that tank). You are given hints in the form of the number of water tiles in each row/column. Typically, every clue is provided.

The Maysu puzzle is a puzzle of determining a single cycle that passes through all the black and white circles in a given square. For white circles, you have to pass straight through and then turn at the next tile and/or the previous tile. For black tiles, you have to turn at, but pass straight through the two connected tiles. Naturally you can only travel orthogonally and cannot cross yourself.

a. Problem input and output:

   i. The skyscraper puzzle will take in a text file with 4 lines: the first line contains one number denoting the size of the board, the second line denotes the clues for the columns from the top in order, the third line denotes the clues for the columns from the bottom in order, the fourth line denotes the clues for the rows from the left in order, and the fifth line denotes the clues for the rows from the right in order. The output would be a text file that shows a board that follows the constraints. Empty clues should be denoted by _.



For example, for the solved puzzle to the left, the input and output would be:

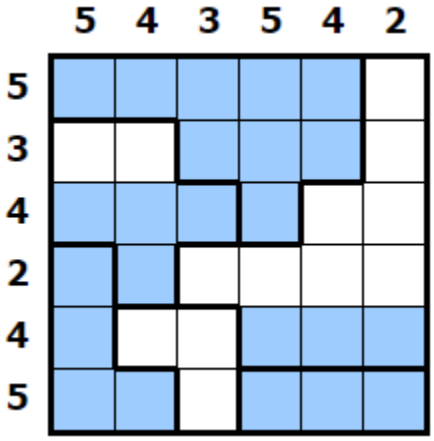
Input file:

```
4
3 4 2 1
2 1 2 2
3 2 1 2
1 2 3 2
```

Output file:

```
2 1 3 4
3 2 4 1
4 3 1 2
1 4 2 3
```

ii. The aquarium puzzle will take in a text file with the following: the first line denotes the size of the board, the second denotes the clues for the columns in order, the third line denotes the clues for the rows in order, and the rest denoting the tanks on the board. Empty clues should be denoted by _. The output will be a text file displaying the solution with * corresponding to water tiles and . corresponding to empty tiles.



For example, for the solved puzzle to the left, the input and output could be:

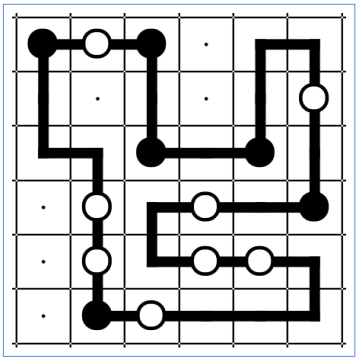
Input file:

```
6
5 4 3 5 4 2
5 3 4 2 4 5
1 1 1 1 1 2
3 3 1 1 1 2
3 3 3 1 2 2
4 3 2 2 2 2
4 5 5 2 2 2
4 4 5 6 6 6
```

Output file:

```
* * * * * .
. . * * * .
* * * * . .
* * . . . .
* . . * * *
* * . * * *
```

iii. The masyu puzzle will take in a text file with the following: the first line denotes the size of the board minus 1, the number of white circles, and the number of black circles; the next lines denote the coordinates (0-indexed, formatted as ‘x y’) of each white circle,



followed by the coordinates of the black circles. The output will be a text file with the following: the first line denotes the number of horizontal edges; the next lines denote the left tile's coordinate for each edge in lexicographical order; we repeat this for the vertical edges.

For example, for the solved puzzle to the left, the input and output could be:

Input file:	Output file:	(output cont.)
5 8 6	16	5 3
0 1	0 0	5 4
1 5	0 1	14
3 1	0 4	0 0
3 3	2 0	0 2
4 1	2 2	0 4
4 3	2 3	0 5
4 4	3 2	1 0
5 2	3 3	1 2
0 0	3 4	1 4
2 2	4 2	1 5
2 4	4 3	3 1
3 5	4 4	3 2
5 1	5 1	4 1
	5 2	4 5

b. The three puzzle solvers will be programmed using the Clingo module in Python. Python will be used as the framework of the solver. Ie. it will be used to read in the problem, timing, writing to file, etc.
c. Each puzzle has some immediate decorative applications to generate images that are not biased by human "randomness". More importantly, these puzzles can be used to compress data (NxN information -> 2N + k Information). Getting the numbers required for some NxN grid is

straightforward (even if you have to generate the extra constraints that some of the problems have like splitting the NxN grid into regions).

2. The contest problems link to an application where you can try out randomly generated puzzles for the three problems (3), however how they confirm the solution is not provided. They do have videos on how to tackle each type of puzzle, which can help with generating constraints. The website can also be used as a source of puzzles to test the solvers. Searching outside the ICLP pages, Skyscrapers has a publicly available solver in C++ (6) and Minizinc solution (1), and Aquarium has a publicly available solver in Perl (4). Masyu has a publicly available solver in Python (it uses backtracking), but this was not used.
3. Currently, I am not aware of any integrations with other software/projects. My timeline will be very rushed as most of the project is being done in the last week due to my horrible planning 😓😓.

Part II. Design

1. The project has 3 main components: the Python Solver program, the default input text files, and the default output text file.
2. The solution will rely on the external Answer Set Programing module Clingo. This adds on constraint functionalities to Python, which we will be using to solve the puzzles.
3. I will try to document the functions in the file itself~~, but I am open to using Sphinx to auto-generate documentation if it ends up being longer than expected~~.

4. Still not sure what this section is asking for either. It sounds like a summary of project efforts, for which this would be the first step. Also isn't this the design doc?
    a. March 22, 2021 - Initial project direction and design submitted
    b. April 26, 2021 - This document was revised and updated.
    c. May 2 2021 - Adjusted to fit implementation
    d. May 12 2021 - Added Masyu

Part V. References (APA6 Format)

1. Åhlander, M. (2020, May 08). Skyscraper puzzle with constraint module in python. Retrieved May 12, 2021, from https://stackoverflow.com/questions/61662722/skyscraper-puzzle-with-constraint-module-in-python
2. Alviano. (n.d.). Alviano/lpcp-contest-2020. Retrieved March 20, 2021, from https://github.com/alviano/lpcp-contest-2020/blob/master/README.md
3. Aquarium. (n.d.). Retrieved March 23, 2021, from https://www.puzzle-aquarium.com/
4. E. (2020, April 02). Aquarium - constraints. Retrieved April 25, 2021, from https://github.polettix.it/ETOOBUSY/2020/04/02/aquarium-search/
5. I. (n.d.). Lp/cp programming contest. Retrieved March 20, 2021, from https://iclp2020.unical.it/affiliated-events/lpcp-programming-contest
6. Pcooksey. (n.d.). Pcooksey/skyscraper-ps. Retrieved March 21, 2021, from https://github.com/pcooksey/skyscraper-ps