Assignment 2: Constraint Satisfaction Problem

**Deadline: Monday, November 12, 11:59:59PM**

As on Assignment 1, you have the option of working in groups of up to three people. You are free to either stay with the same team or pick a new one.

CSP - Flow Free

Created by Litian Ma and Jialu Li based on <u>Flow_Free</u>

Introduction and rules

Flow Free is a puzzle game for iOS and Android released by Big Duck Games on June 2012. The goal for this game is to fill in colors on all empty grid cells to form "pipes" such that the colors are able to flow between two given sources. You will be given a puzzle with multiple color inputs. Each color has two sources on the grid, and you will need to draw a pipe of the same color to connect them. The pipes cannot intersect with each other, and a complete solution cannot contain any empty grid cell. For this assignment, each puzzle only has one unique solution. The following picture shows an example of the start state and goal state for one Flow Free instance.
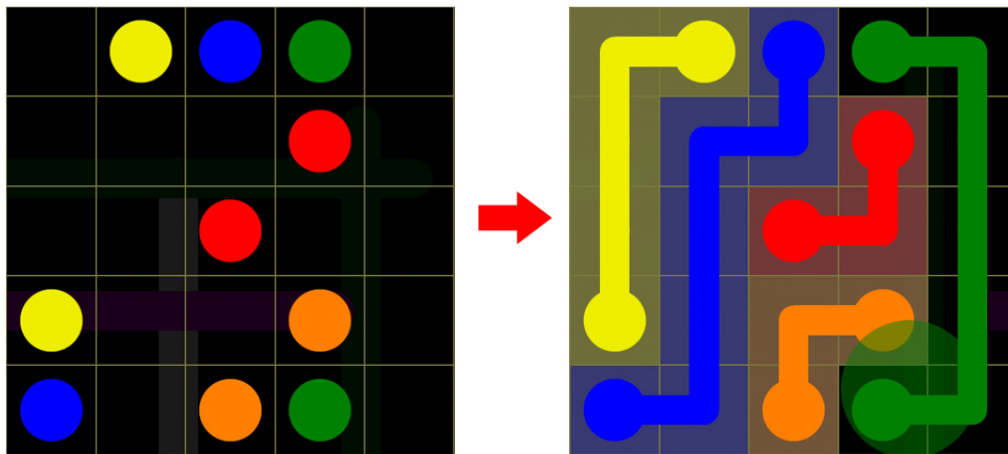
The other constraint for this assignment is that no zigzag pattern is allowed. This means that for each non-source cell, its four-connected neighborhood (consisting of cells above, below, to the left, and to the right, if they exist) should have exactly two cells filled with the same color. For each source cell, its neighborhood should have exactly one cell filled with the same color. The following picture shows an example of the disallowed "zigzag" pattern.
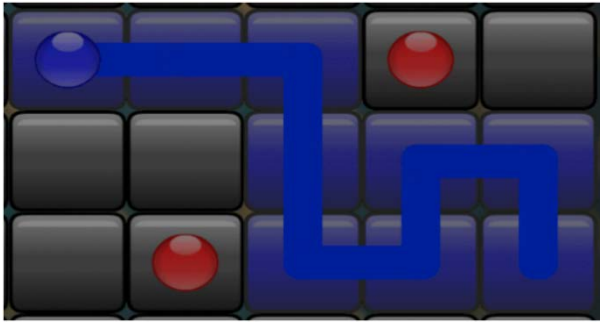
First, you need to formulate Flow Free as a CSP. In your report, give your definitions of variables, domains, and constraints. Next, with these in mind, implement backtracking search. You should create two versions of your implementation: the "dumb" one with random variable and value ordering (and no forward checking), and the "smart" one that includes any variable and value ordering heuristics that make sense for your formulation (which would probably include forward checking). In the report, describe your "smart" implementation and give a brief explanation of why you chose to use your particular combination of heuristics and inference techniques.

You need to run your solver on several input puzzles. You will use the input and output formats illustrated by the sample 5*5 puzzle and the corresponding solution.

For each of the inputs below, please include in your report:

A. Your solution, i.e., the filled game board that satisfies all the constraints like the example above.
B. For both the "dumb" and the "smart" implementation, the number of attempted assignments and the execution time of the algorithms (in seconds or milliseconds). If your "dumb" implementation takes too long on a certain input, feel free to cut it off and say so in your report (you will not lose points).

Inputs

You are required to solve the following 6 puzzles:

1. 7*7 puzzle
2. 8*8 puzzle
3. 9*9 puzzle
4. 10*10 puzzle
5. 12x12 puzzle
6. 14x14 puzzle

Report Checklist

Your report should briefly describe your implemented solution. Your description should focus on the most "interesting" aspects of your solution, i.e., any non-obvious implementation choices (including programming language chosen) and parameter settings, and what you have found to be especially important for getting good performance. Feel free to include pseudocode (DO NOT PUT CODE IN YOUR REPORT) or figures if they are needed to clarify your approach. Your report should be self-contained and it should (ideally) make it possible for us to understand your solution without having to run your source code.

Give your CSP formulation (variables, domains, and constraints) and briefly describe your backtracking search implementation. For each of the puzzles report the solution and provide the number of attempted variable assignments and execution time for your implementation for both the dumb and smart implementations.

It is required that you provide an analysis of your results in your report.

Statement of individual contribution:

- All group reports need to include a brief summary of which group member was responsible for which parts of the solution and submitted material. We reserve the right to contact group members individually to verify this information

Submission Instructions

By the submission deadline, **one designated person from the group** will need to upload the following to D2L/Brightspace:

1. A **report** in **PDF format**. Be sure to put the **names** of all the group members at the top of the report. The name of the report file should be **lastname_firstname_a2.pdf** (based on the name of the designated person).

2. Your **source code** compressed to a **single ZIP file**. The code should be well commented, and it should be easy to see the correspondence between what's in the code and what's in the report. You don't need to include executables or various supporting files (e.g., utility libraries) whose content is irrelevant to the assignment. If we find it necessary to run your code in order to evaluate your solution, we will get in touch with you. INCLUDE YOUR OUTPUT FILE IN THIS ZIP FILE.

   The name of the code archive should be **lastname_firstname_a2.zip**.

Multiple attempts will be allowed but only your last submission before the deadline will be graded. **We reserve the right to take off points for not following directions.**

Late policy: You must submit by Midnight of Nov 12 the full package (report and source code). No exceptions.