CAP-4630 Homework 2
Deadline: 2:05PM on 10/17 (Tuesday)
Submit code for programming exercise on Moodle and analytical exercises either at start of lecture or Moodle.

1. Exercise 4.1 from textbook. (10 pts)
2. Exercise 5.1 from textbook. (10 pts)
3. Exercise 5.6 from textbook. (10 pts)
4. Exercise 5.7 from textbook. (10 pts)
5. Exercise 5.8 from textbook. (10 pts)
6. Exercise 5.9 from textbook. (10 pts)
7. Exercise 5.14 from textbook. (10 pts)
8. Exercise 5.16 from textbook. (10 pts)
9. Exercise 6.1 from textbook. (10 pts)
10. Exercise 6.2 from textbook. (10 pts)
11. Exercise 6.5 from textbook. (10 pts)
12. Exercise 6.6 from textbook. (10 pts)
13. Exercise 6.7 from textbook. (10 pts)
14. Exercise 6.9 from textbook. (10 pts)
15. Exercise 6.11 from textbook. (10 pts)
16. Exercise 6.12 from textbook. (10 pts)
17. Quizzle (120 pts)
    Below are four instances of "quizzle," a logic puzzle. For each puzzle, you are provided with a backstory and clues, and you must piece together deductions in order to ultimately assign a unique value for each category. We encourage you to visit the site and attempt to solve some puzzles on your own, by going through the clues consecutively and eliminating the set of incompatible combinations of category values.

    You will solve each puzzle using two different approaches, and compare their performances. You are free to use any approach/combination of approaches you would like. We suggest that you use a CSP approach and integer linear programming approach (another good option could be logical inference). For these approaches, you can use built-in libraries from Python as the solvers. You would then need to create the models for the problems to be input to the solvers. This would involve describing the variables, the possible values for each variable, and the set of constraints. Note that if you use the two approaches we suggest, then you would just be creating the models, but not implementing solving algorithms yourself. We also encourage you to create your own solving algorithms for extra credit.

    For each problem instance answer the following questions:

    (a) Describe the two approaches you are using. What do the variables and values represent? What are the constraints? How many variables and constraints are there?

    (b) What are the resulting outputs of program? What are the running times of the two approaches? Which one is more efficient? Would the same be true for larger instances?

    For 20 points extra credit, you can implement a new different algorithm that runs faster than CSP and ILP approaches. You must describe the algorithm, and report the running times. Puzzle instances:

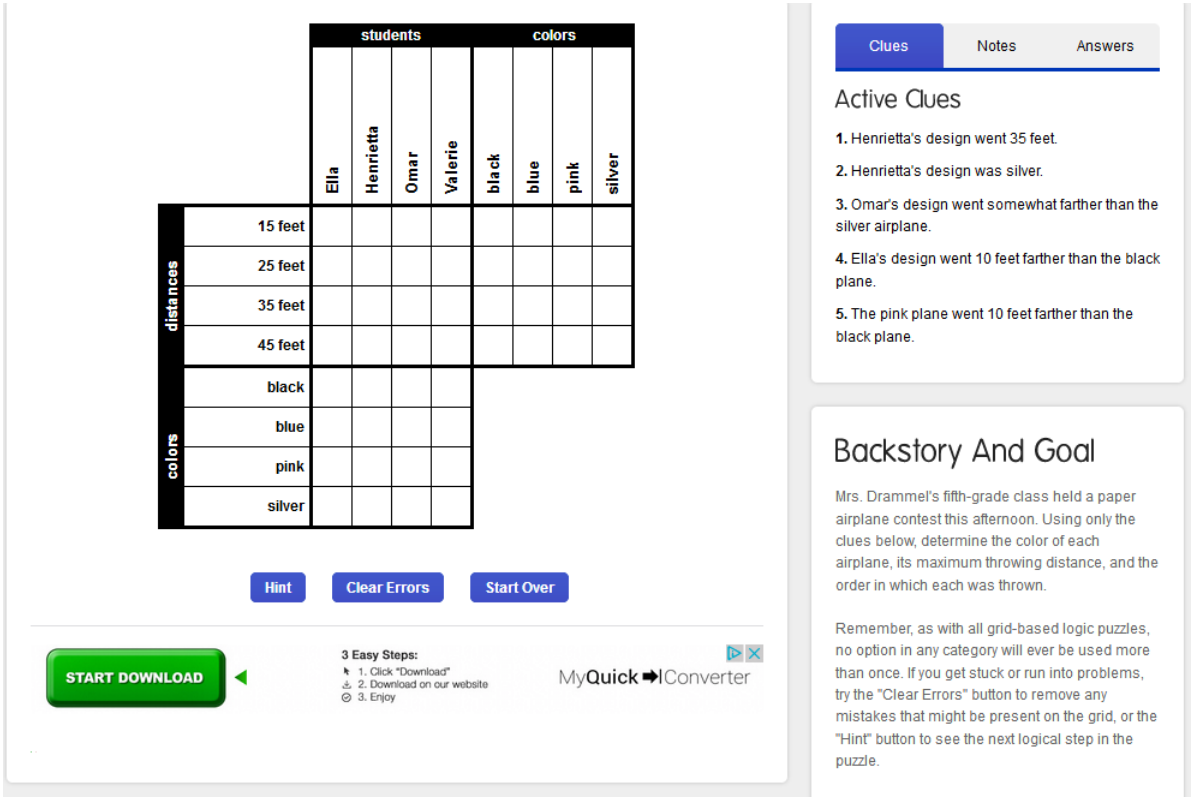

Figure 1: Quizzle 1 (60 pts).

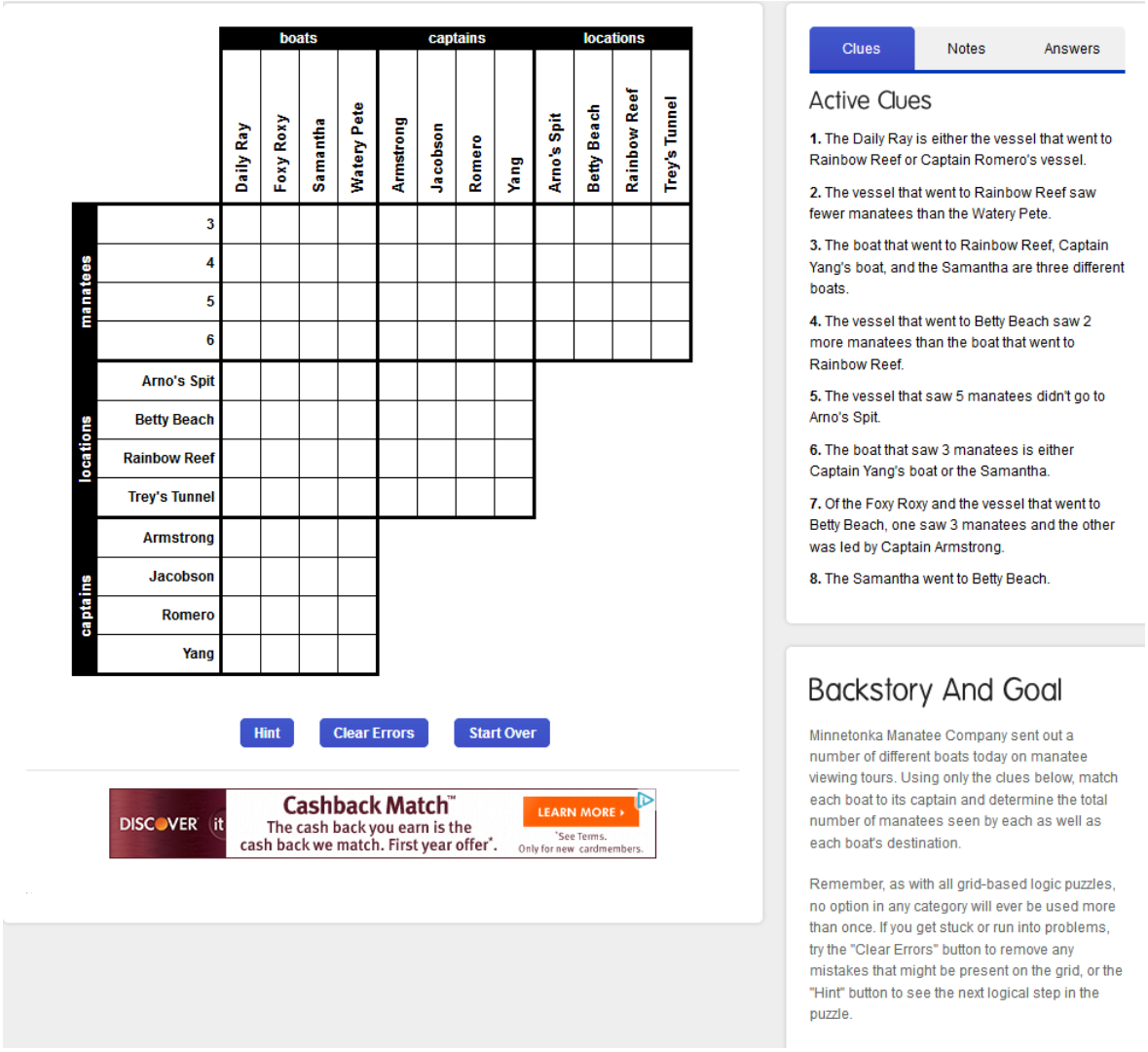

Figure 2: Quizzle 2 (60 pts).

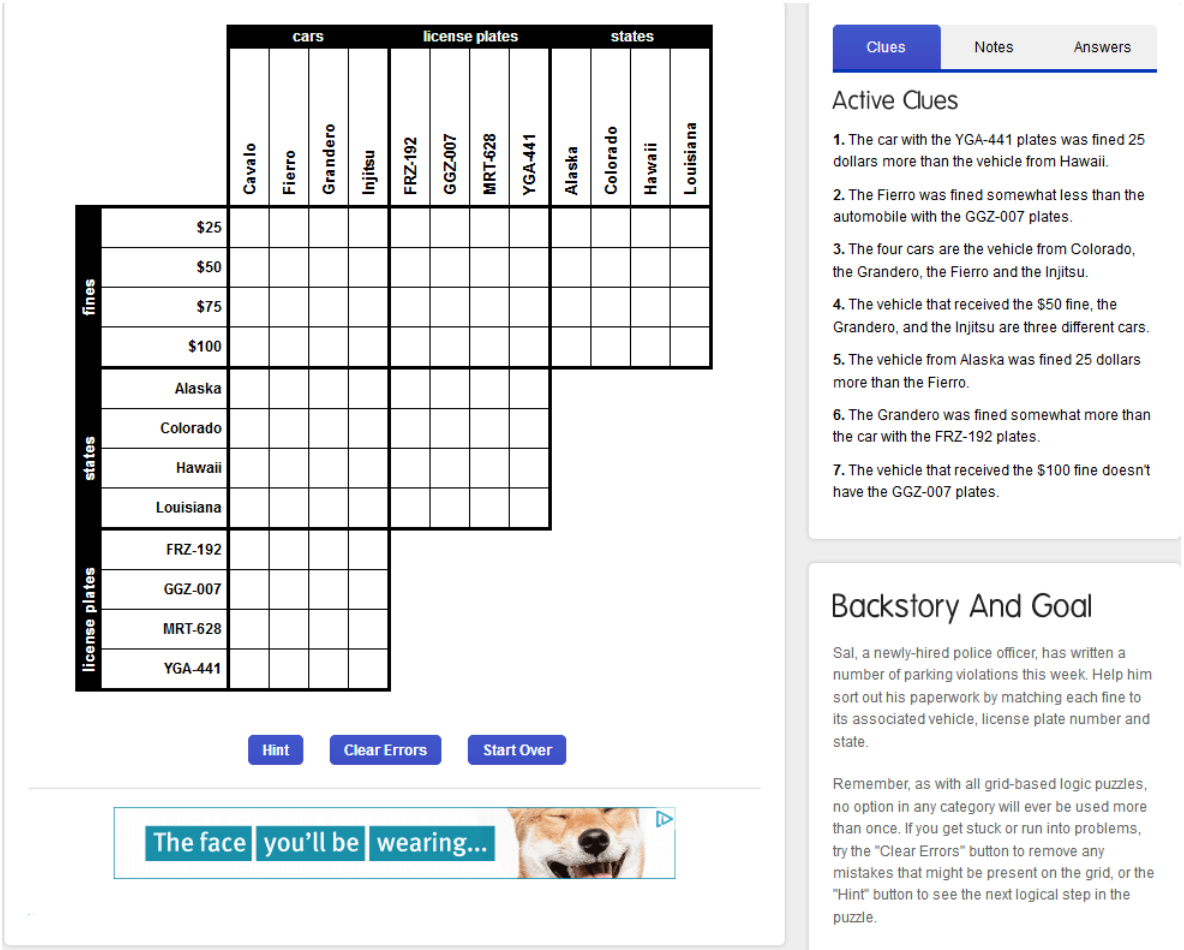

Figure 3: Quizzle 3 (10 pts of extra credit).

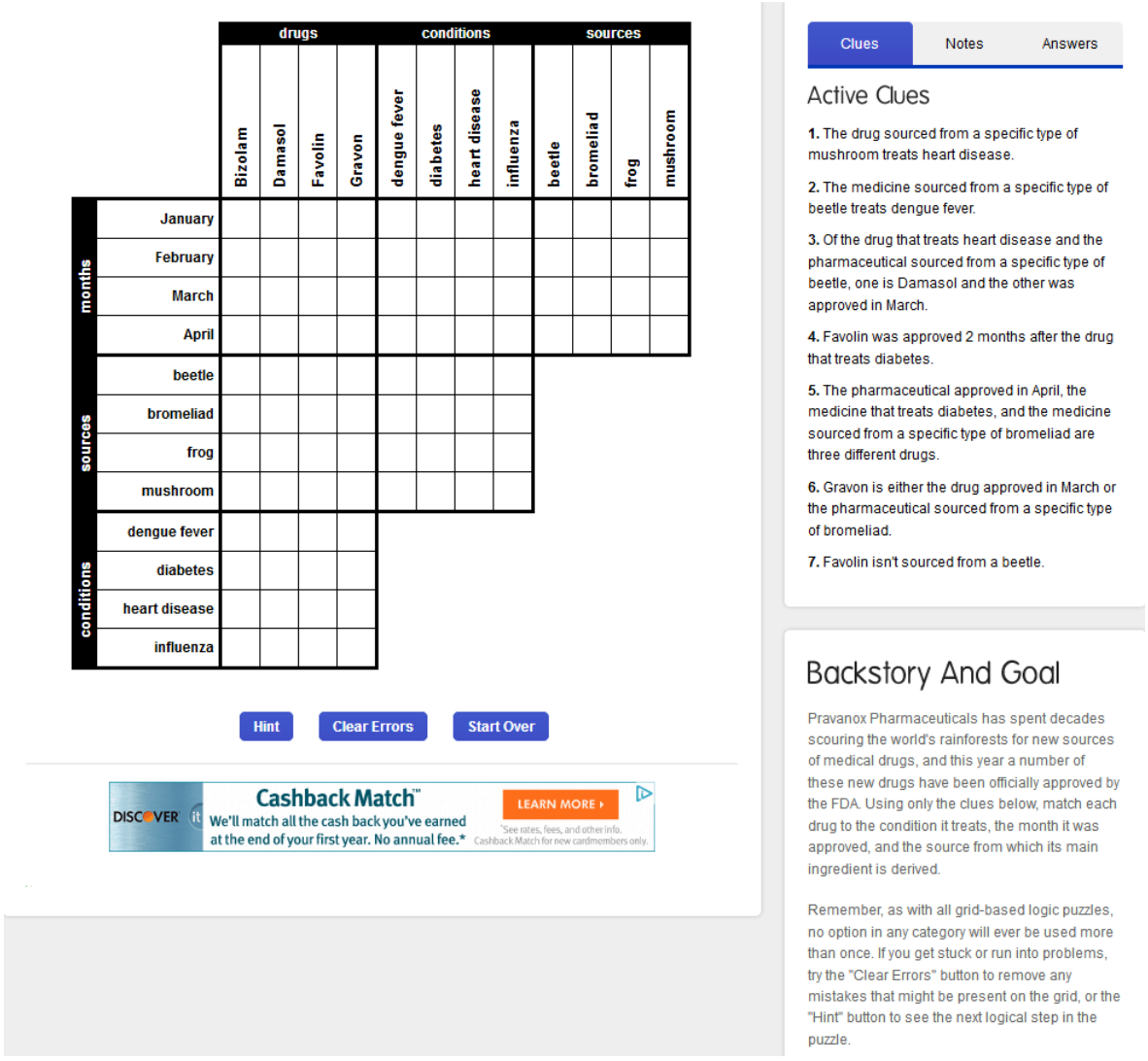

Figure 4: Quizzle 4 (20 pts of extra credit).