CMSC 671 (Introduction to AI) – Fall 2017

Homework 3: CSPs, Games, and Probabilities (70 points)
Due: 10/23 at 11:59pm.
Turnin: Blackboard.

Please submit **Parts I–III** together as a **single PDF file** named *lastname*_hw3.pdf, with parts clearly marked and delineated, and Part IV as a zipfile named *lastname*_hw3.zip containing python code.

All files must start with your last name and have your full name(s) in the file, at/near the top.

You are ***encouraged*** to work on Parts I-III of this homework assignment in groups of up to three students. If you work in a group, only one person should submit a solution, with everyone's name clearly noted at the top of the the file (files should be named after the person who submits). If you work in a group, you must actually work ***as*** a group! ***Don't*** have each member work on one part of the assignment, then submit the collection of independent solutions.

Part IV must be done individually.

PART I. CONSTRAINT SATISFACTION EXERCISES (10 POINTS)

For these questions, you will be solving a map-coloring problem on the map shown here. Each of the regions A–E must be green, blue, or red, and no adjacent regions may be the same color.

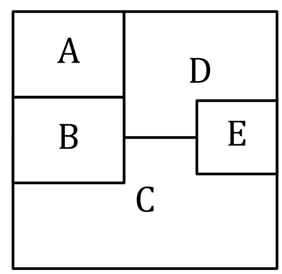

Figure 1. Color this map blue, green and red, with no adjacent areas sharing a color.

Part A: Backtracking search (15 pts)

You will use simple backtracking search to find a solution to this constraint satisfaction problem. The **variable ordering heuristic** is to make assignments to variables in alphabetical order (A, B, C, D, E); the **value ordering heuristic** is to consider the values in the order blue, green, red. Indicate instantiations with *A=red*, *B=green*, etc.

Assignment:

1. Show the complete search tree, and circle the solution node if there is one.
2. Show the final coloring, if one is found.
3. How many variable instantiations (search steps) are tried by this search method?

Part B: Solution spaces (5 pts)

Assignment: Answer the following questions, in your own words.

4. How large is the search space for this problem? That is, how many different colorings, *legal or illegal*, are there for this map?
5. For this map, how many different *solutions* (legal colorings) are there?

PART II. GAME PLAYING: ROLLING DICE (20 POINTS)

Consider a two-player dice-rolling game where two players alternate rolling a fair six-sided die. The rules are as follows:

- If the roll is 1 or 2, the player who rolled gains two points.
- If the roll is 3 or 4, the player who rolled loses one point.
- If the roll is 5 or 6, the opponent gains one points.
- If a player exceeds five points, they automatically lose all of their points, and the game ends.

- On their turn, *before* rolling, a player can choose to stop the game, in which case both players keep their current scores.
- The goal is to beat the other player **by as many points as possible**.

Assignment: Answer the following questions about playing this game.

6. Draw the 4-ply (two moves for each player) expectiminimax tree for this problem.
7. Using the static evaluation function (score(player1) - score(player2)), back up the leaf values to the root of the tree. Annotate each of the nodes in the tree with its backed-up minimax value.
8. Circle the nodes that would be pruned by alpha-beta pruning using depth-first (left-to-right) search. (You should assume that the alpha and beta values are initialized to -1 and +1, rather than +/- infinity.)
9. What is the best action for the first player to take? (Play or stop?)
10. If player 1 rolls a five, what should player 2 do? Why? (1-3 sentences)
11. Would you describe this game as fair? Why or why not? (1-3 sentences)

PART III. BAYES' NETS AND PROBABILITY (20 POINTS)

Assignment: Provide a complete solution to Russell & Norvig, Exercise 14.6. Show all work.

PART IV. CALCULATING PROBABILITIES PROGRAMMATICALLY (20 POINTS)

Assignment: For the Bayesian network shown, implement a program that computes and returns the probability of any conjunction of events given any other conjunction of events. Your function should take five arguments, in the order A, B, C, D, E, with values as follows:

0	given false
1	given true
2	query false
3	query true
4	unspecified

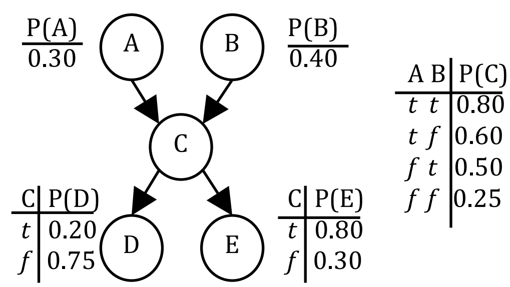

Figure 2: A five-variable Belief Net with associated complete CPTs.

In this formulation, 0 and 1 mean a variable is observed to be false or true; 2 or 3 means we are querying the probability of assigning false or true to that variable; and 4 means the variable doesn't appear in the probability statement we are calculating.

Some examples:

(A,B,C,D,E)	Calculate	Meaning
(3,4,4,4,4)	P(A)	P(A=*t*)
(2,4,3,4,4)	P(¬A ∧ C)	P(A=*f* ∧ C=*t*)
(3,4,2,1,4)	P(A ∧ ¬C | D)	P(A=*t* ∧ C=*f* | D=*t*)
(2,1,0,3,4)	P(D ∧ ¬A | B ∧ ¬C)	P(D=*t* ∧ A=*f* | B=*t* ∧ C=*f*)
(2,3,4,4,0)	P(B ∧ ¬A | ¬E)	P(B=*t* ∧ A=*f* | E=*f*)

Your code should use the probability values in the tables shown, and appropriate formulas to evaluate the probability of the specified event(s). It is OK to hardcode values *from the tables* in your code, but not values for possible arguments or probability values for all possible atomic events.

There are possible combinations of values that don't resolve to meaningful probability queries; be sure to test for these as error cases.