4511W, Spring-2018
ASSIGNMENT 3 :
Assigned: 03/04/18 Due: 03/11/18 at 11:55 PM (submit via moodle, you may scan or take a picture of your paper answers in a zip if you have multiple files)

Problem 1. (15 points)
Assume you have a binary tree and are running Monte-Carlo tree search. If there is a tie between the branches/children, select the left branch. Show the current Monte-Carlo tree and corresponding UCB values for every node if the roll-outs result in (in this order): win, win, loss, win, win, loss, win, win, loss.

When solving this problem, please show work by giving a snapshot of the tree after every loss (i.e. after 3rd and 6th rollout, not just after the 9th.)

Problem 2. (10 points)
Consider the following game:

	Player 2, Action 1	Player 2, Action 2
Player 1, Action 1	(4, 3)	(5, 2)
Player 1, Action 2	(9,1)	(2, 8)

Answer the following questions:
(1) What are the pure Nash equilibrium for the game above.
(2) What are the mixed Nash equilibrium for the game above.
(3) What are the pure Pareto optimum for the game above.
(4) What are the mixed Pareto optimum for the game above.

Problem 3. (25 points)
Suppose you are on an airplane from MSP to YYZ. The flight time is 120 minutes. Assume the plane will ascend or descend at 400 m/min. Assume the maximum cruise height is 16,000 m and the minimum cruise height is 4,000 m (anything lower can only happen at takeoff or landing).

(1) Formula the above flight description as a mathematical CSP. Use variables at all the 10 minute interval marks of the flight. So the variables are “0 minute mark”, “10 minute mark”, “20 minute mark”, ... “120 minute mark”. The “0 minute mark” and “120 minute mark” must be at height 0 (for takeoff and landing). (Note: you can assume the plane will fully ascend/descend at the maximum speed for the whole 10 minute intervals.)

(2) Suppose that there are additional constraints: There is turbulence at the 50 and 60 minute mark in the flight. The plane must not be above 12,000 m at these to time marks. Additionally, to avoid people from feeling sick in flight, the plane must not ascend at one minute mark, then descend at the next. So choosing something like: 10 mark = 8K, 20 mark = 12K, 30 mark = 8K, would not be allowed. Show how you would mathematically incorporate these new constraints in the CSP.

(3) Using your CSP from parts (1) and (2), show the domains after applying 1-consistency constraints. Show the resulting domains (and some work for at least some of the variables).

(4) After doing part (3), make the domains 2-consistent and show the domains.

(5) After doing part (4), make the domains 3-consistent and show the resulting domains.

(6) If the plane wants to cruise at the highest hight possible, what altitude should it be at at every 10 minute mark? (In other words, what are the largest values in your domains after part (5)).

Problem 4. (30 points)
Suppose you have a full binary tree (each node has 2 children and all leaves are on the same depth). You are playing a zero-sum game against an opponent, where you go first (as a maximizer) and the opponent goes next. Turns will keep alternating until the end of the game. Assume the game always ends at a fixed depth. If you could freely choose the value of the terminal states, what is the most you can prune using alpha-beta pruning. In other words, this question is asking for a full binary tree of depth "d", what is the maximum that you can prune (or minimum searched)? Assume the depth "d" will be even (so 2 actions, 4 actions, 6 actions, ...).

(Note: this has to be a formula involving "d", not just one specific depth. This also does not need to be exact, but you will be graded on how close you can get, so poorer approximations will receive less points.)

(Hint: I would work out a few depths by hand and look for patterns.)

Problem 5. (10 points)
For this problem, we will use this webpage:
http://ncase.me/trust/

This question will pertain to only the "7. Sandbox" part, but it might be interesting to play through the whole thing (it doesn't take very long). For this question, keep the payoff matrix at the default and only change the "mistake chance" in the "rules" section.

(1) Set the "mistake chance" in the "rules" section to zero. If you only have only "Simpletons" and "Cheaters", what is the most number of "Cheaters" you can have where the "Simpletons" still win?

(2) If you have only one "Cheater" and 24 "Simpletons", sample every increment of 5% "mistake chance" and report who dominates long term. (i.e. set mistake chance to 0%, then 5%, then 10%, ...)

Programming (python/lisp):

This time we will look at the constraint satisfaction code in:
/root/csp.py
/root/tests/test_csp.py

Problem 6. (10 points)
We will again test the N-queens problem (framed as a constraint satisfaction problem this time).

(1) Report the run-time for board sizes = {11, 20, ~~40~~} when solving the N-Queens CSP problem with backtracking search. (Just do for sizes 11 and 20... 40 takes a very long time.)

(2) Report the run-time for board sizes = {11, 20, 40} when solving the N-Queens CSP problem with

min-conflicts.

(3) Find what board size takes approximately 10 seconds to solve with min-conflicts (make sure it is not ending early).