Name: _____ UW CSE 473 Final Exam, Fall 2014

P1

P6

Instructions

Please answer clearly and succinctly. If an explanation is requested, think carefully before writing. Points may be removed for rambling answers. If a question is unclear or ambiguous, feel free to make the additional assumptions necessary to produce the answer. State these assumptions clearly; you will be graded on the basis of the assumption as well as subsequent reasoning. **On multiple choice questions, incorrect answers will incur negative points** proportional to the number of choices. For example a 1 point true-false question will receive 1 point if correct, -1 if incorrect, and zero if left blank. Only make informed guesses.

There are 8 problems worth 71 points (plus 1 bonus point) on 9 pages.

***0*. Who are you?** Write your name at the top of every page.

1. (1 point each – total of 10) True / False Circle the correct answer.

(a) T F A pattern database helps an agent avoid wasting time in cycles by storing previously-expanded states.

(b) T F Random restarts are often used in local search to diminish the problem of local maxima.

(c) T F If a binary CSP has a tree-structured constraint graph, we can find a satisfying assignment (or prove no satisfying assignment exists) in time that is linear with the number of variables.

(d) T F. One advantage of Q-learning is the need to estimate fewer parameters than would be necessary to fully specify the corresponding MDP.

(e) T F. Consider an HMM specifying a Markov chain of length t (e.g., with states ranging from X_1 to X_t). The number of parameters required to represent its joint distribution grows linearly in t.

(f) T F. In a hidden Markov model, an evidence variable, E_i, is independent of *all* other evidence variables given its associated hidden state, X_i.

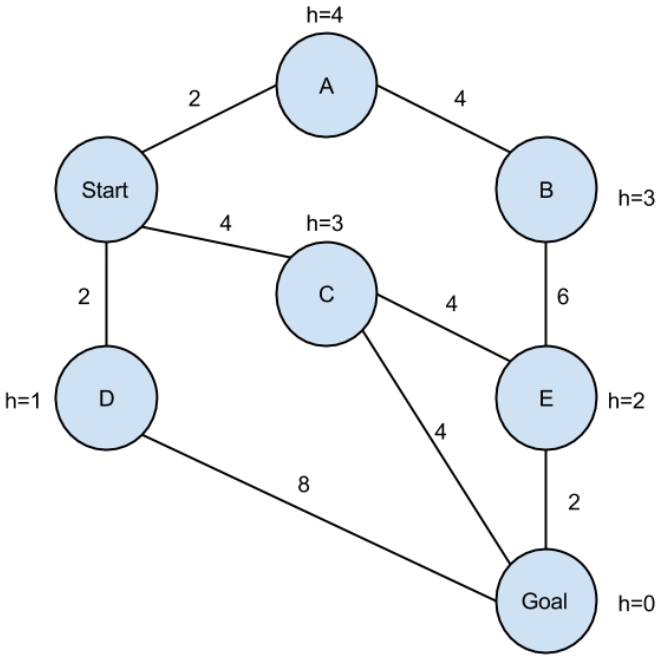
(g) T F. If two random variables in a Bayes net have at least one active path between them (ie they are not D-separated), then the variables are necessarily dependent on each other.

(h) T F When learning the structure of a Bayes net, the scoring function should penalize high connectivity in order to prevent over-fitting of the training data.

(i) T F In an MDP, the larger the discount factor, γ, the more strongly favored are short-term rewards over long-term rewards.

(j) T F In an MDP, a single Bellman backup for one state has worst-case time complexity *O(|States||Actions|).*

2. (2 points each – total of 8) Search. Given the graph to the right, write down the order in which the states are visited by the following search algorithms. If a state is visited more than once, write it **each time**. Ties (e.g., which child to first explore in depth-first search) should be resolved according to alphabetic order (i.e. prefer A before Z). Remember to **include the start** and goal states in your answer. Treat the goal state as G when you break ties. Assume that algorithms execute the goal check when nodes are visited, not when their parent is expanded to create them as children.



(a) Uniform-cost search

___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___

(b) Iterative deepening depth first search

___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___

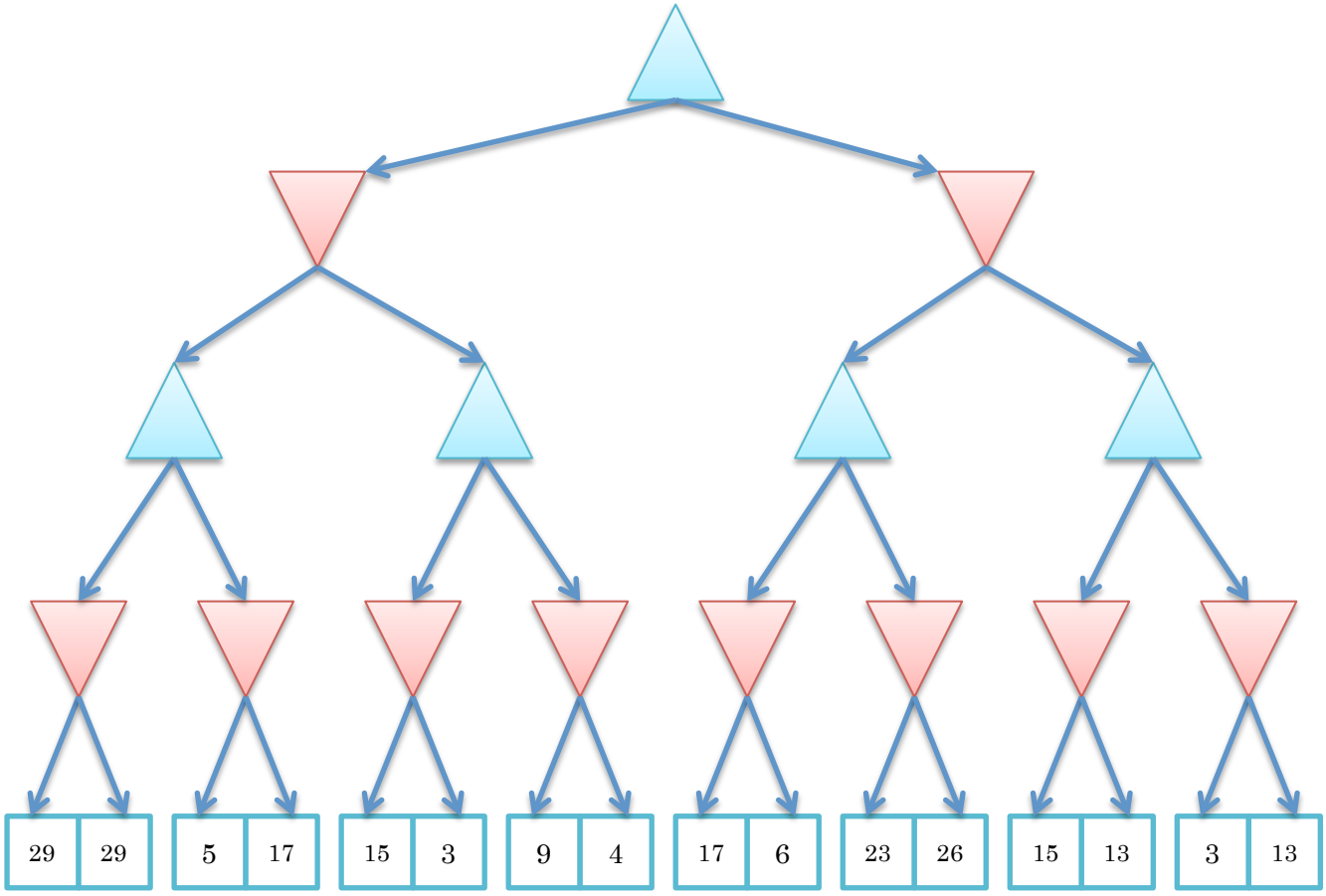
(c) A* search, where f(n)=g(n)+h(n)

___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___

(d) Breadth-first search

___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___

3. **Adversarial Search.** For the next questions, consider the mini-max tree, whose root is a max node, shown below. Assume that children are explored left to right.



a. **(3 points**) Fill in the mini-max values for each of the nodes in the tree that aren't leaf nodes

b. **(5 points)** If alpha-beta pruning were run on this tree, which branches would be cut? Mark the branches with a slash or a swirl (like a cut) and shade the leaf nodes that don't get explored.

c. **(3 points)** Consider the leaf node with the value 26. Suppose you could change the value of that leaf to 3 or 15. Would any of these values let you prune more nodes in the search? If so, which would let you prune the most? Explain.

4. Constraint Satisfaction

Several research are trying to solve CS using two super-computers (X, Y). Each computer can only compute one task within one time slot. There are 6 problems to solve; [ML_1, ML_2, CMB_1, CMB_2, AI, VISION]. "ML_1" must be completed before "ML_2" and "CMB_1" before "CMB_2". Since the AI group brings in the most research money, "AI" needs to take the first time slot. Finally, both "AI" and "VISION" can only be computed by computer Y, which has special software installed. Define a CSP by using each problem-name as a variable, whose domain specifies the computer and time-slot used to solve that task. Let's define X_t to mean that computer X working at time t. Assume that there are only three time slots available (so the domain of each variable is a subset of {X_1, …, Y_3} as shown in the table).

a) (3 points) write the constraints. You may find it useful to use the notation **Time(problem)** to denote when that problem is being solved and **Computer(problem)** to say which computer must be assigned to a task. You may also say that two or more variables have values that are **ALLDIFF**.

ML_1	ML_2	CMB_1	CMB_2	AI	VISION
X_1	X_1	X_1	X_1	X_1	X_1
X_2	X_2	X_2	X_2	X_2	X_2
X_3	X_3	X_3	X_3	X_3	X_3
Y_1	Y_1	Y_1	Y_1	Y_1	Y_1
Y_2	Y_2	Y_2	Y_2	Y_2	Y_2
Y_3	Y_3	Y_3	Y_3	Y_3	Y_3

(a) **(1 point)** Applying JUST unary constraints, rule out illegal values in the table above **using an X**, as shown to the right:

(b) **(3 points)** Is the initial state (from part a) arc-consistent? If not, update the table above **using a diagonal line** (as shown) to cross out values that would be pruned by running AC-3 .

e) (**4 points)** Solve the (reduced, arc-consistent) CSP using backtracking search (without forward checking). Use the minimum remaining values (MRV) variable ordering and least constraining value (LCV) value ordering. Break all ties in alpha-numerical order.

The first variable assigned is _____ it's given value _____

The second variable assigned is _____ it's given value _____

5 **Hidden Markov Models (8 points)** You are using an HMM to track the status of a probe orbiting Neptune. The probe has two states S: collecting data (C) and recharging (R). Each day, it sends a signal back to earth about its status. Due to interference caused by solar flares, the signal is occasionally corrupted, or even lost altogether. Therefore there are three possible signals G: collecting data (c), recharging (r), and no signal received (n). The transition and emission probabilities of the HMM are summarized in these tables:

S_t	C	R
$P(S_t \mid S_{t-1} = C)$	3/4	1/4
$P(S_t \mid S_{t-1} = R)$	2/3	1/3

G_t	c	r	n
$P(G_t \mid S_t = C)$	3/4	0	1/4
$P(G_t \mid S_t = R)$	1/6	1/2	1/3

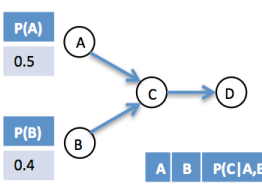
Assume that at time 0, you have no information about the state: $P(S_0 = C) = 1/2$.

You receive the first two signals $G_0 = r, G_1 = n$. Use the forward algorithm to compute the distribution over the probe's state after receiving the second signal $P(S_1 \mid G_0, G_1)$. To show your work, fill in the following table:

S_t	C	R
$P(S_0 \mid G_0)$		
$P(S_1 \mid G_0)$		
$P(S_1 \mid G_0, G_1)$		

What is the most likely state for S_1 given the signals received? [ ]

6. **Bayes Nets** In this domain, the world state has 4 state variables A, B, C and D, and 2 actions, X and Y. The Bayes net represents the dependence of the state variables on each other (each value denotes the probability of the variable being true).



P(A)
0.5

P(B)
0.4

C	P(D|C)
T	0.25
F	0.75

A	B	P(C|A,B)
T	T	0
T	F	1.0
F	T	1.0
F	F	0.5

Reward Matrix

	States of Nature	
Actions	C=T	C=F
X	$1000	$2000
Y	$800	$900

(a) **[1 points]** Is A independent of B?

(b) **[1 points]** Is A independent of B conditioned on C?

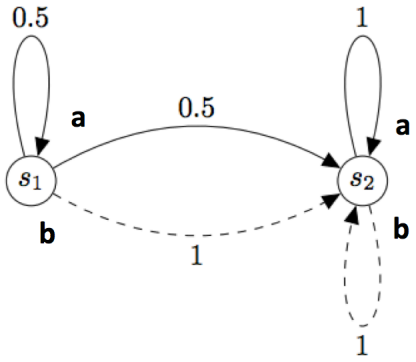
(c) [**1 points]** Is A independent of B conditioned on D?

(d) **[1 points]** Is A independent of D conditioned on C?

(e) [2 points] Compute the probability of P(C=true). Show your work.

(f) [Bonus 1 point] Consider the reward matrix representing the individual rewards for actions the agent might take. Note that the rewards depend on the value of the unobserved variable C. Supposing the agent is a risk-neutral rational agent, What is the utility of executing action X (show your work)?

7. MDPs Consider a simple MDP with two states, s_1 and s_2, and two actions, a (solid line) and b (dashed line); the numbers indicate transition probabilities. Rewards, which just depend on state and action (not the state resulting from the action), are shown in the table below.



$R(s_1, a) = 8$	$R(s_2, a) = -4$
$R(s_1, b) = 16$	$R(s_2, b) = -4$

(a) **(8 points)** Supposing that V_0 of both states is 0 and the discount factor, γ, is ½, fill in the four boxes (V_1 and V_2), but be sure to **show your work below.**

$V_0(s_1) = 0$	$V_0(s_2) = 0$
$V_1(s_1) =$	$V_1(s_2) =$
$V_2(s_1) =$	$V_2(s_2) =$

8. Reinforcement Learning (a) (8 points) A self-driving car needs to decide whether to Accelerate (**A**) or Brake (**B**) so as to drive to a location without hitting other cars. It receives a reward of +1 if the car moves and doesn't hit another car, 0 if it doesn't move, and -2 if it hits another car. The discount factor (γ) is 1. The car wishes to do Approximate Q-Learning to learn a good driving policy. The car has sensors that allow it to observe the distance (**D**) to the nearest object and the current speed (**S**). We'll create a set of four integer-valued features by pairing sensor values with the name of the action we are about to execute, resulting in the following four features: f_{AD}, f_{AS}, f_{BD}, f_{BS}. The first two are for action A, the second two for action B. For example, when executing a brake action, the BD feature might report 1 while the BS feature might be 0. Q values will be approximated by a linear combination of terms, with four weights (one for each feature): W_{AD}, W_{AS}, W_{BD}, W_{BS}. Some learning has already happened so W_{AD} =1, but the others are 0. The learning rate α is 0.5. Below is the stream of data the car receives as it drives. Compute the weights after each step.

Observed Data	Weights after seeing data
	1, 0, 0, 0
Initial Sensors: D=0, S=2 Action: A Reward: -2 Final Sensors: D=1, S=0	
Initial Sensors: D=1, S=0 Action: B Reward: 0 Final Sensors: D=1, S=0	

b) (1 point) Given the learned weights, suppose that the sensors read D=1, S=1. Which action would be preferred? If there is a tie, write "tie". Show your work below.