Student Number : _____ UW CSE 473 Midterm Exam, Autumn 2016

Student Name (just on first page):_____

14	9	9	4	7	8	✗

51 full points!

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, leaving the answer blank will give you one free point! Wrong answer = zero points. Only make informed guesses.**

Exam is closed book, but you may bring and use one 8.5 x 11' double-sided page of notes

+14

1. (2 points each) Circle the correct answer.

(T) F The game, Tic-tac-toe, is fully-observable.

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

(T) F In a graph where the goal is neither the root nor at depth one, iterative deepening search will definitely expand more nodes than breadth- first search.

(T) F A* search with the heuristic h(n) = 0 is equivalent to uniform-cost search.

(T) F If there are two sets of admissible heuristic values, h1 and h2, for the same search problem, then h3=mean(h1,h2) will also be admissible. h^* : actual forward cost

if $h_1 < h^*$ and $h_2 < h^* \Rightarrow 0.5(h_1+h_2) < h^*$

(T) F An agent that uses Expectimax search may well achieve a better score when playing against a suboptimal enemy than a Minimax agent would.

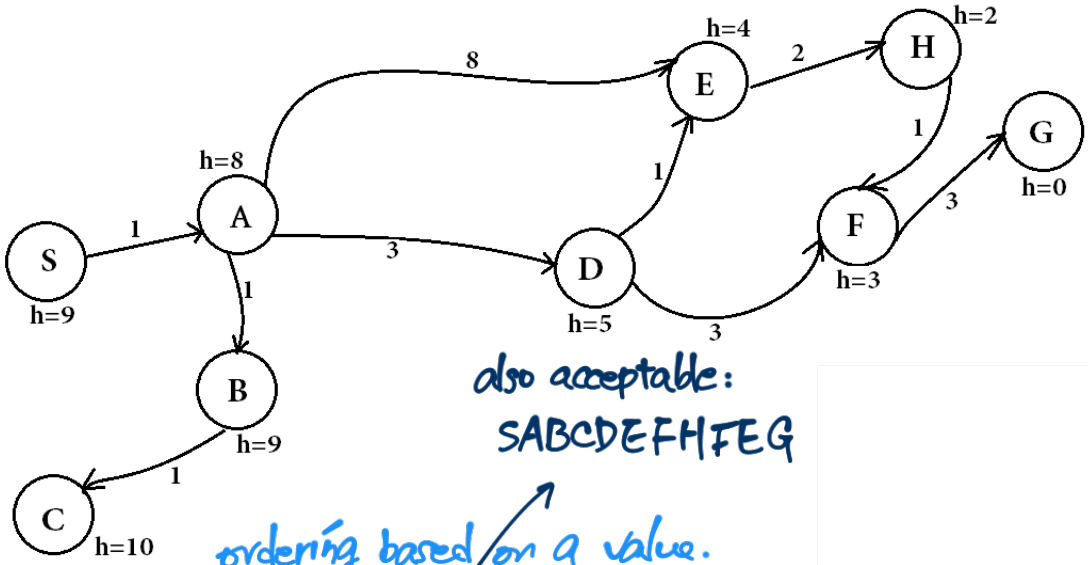
(T) F Decreasing the discount factor, γ, in an MDP tends to make an agent favor short-term rewards over long-term rewards.

small γ : decay faster (e.g. $\gamma=0.1$, $\gamma^2=0.01$ / $\gamma=1$, $\gamma^2=1$)

∴ favors short term rewards

+9

2. Search. Given the graph below, suppose you want to go from start state "S" to goal state "G", write down the order in which the states are *visited* and the path found by the following search algorithms. Ties (e.g., which child to first explore in depth-first search) should be resolved alphabetically (i.e. prefer A before Z). Remember to include the start and goal states in your answer. Assume that algorithms execute the goal check when nodes are visited, not when their parent is expanded to create them as children. Assume a graph search implementation – do not *expand* any node more than once.



also acceptable: SABCDEFHFEG

ordering based on g value.

(a) (3 points) Uniform Cost Search:

Visited order: SABCDEFHG

Solution (path length: 10): SADFG

or 5 : for number of nodes

or 4 : for number of links

ordering based on h value

(b) (3 points) Greedy Search:

Visited order: SAEHFG

Solution (path length: 15): SAEHFG

or 6 : for number of nodes

or 5 : for number of links

ordering based on f value

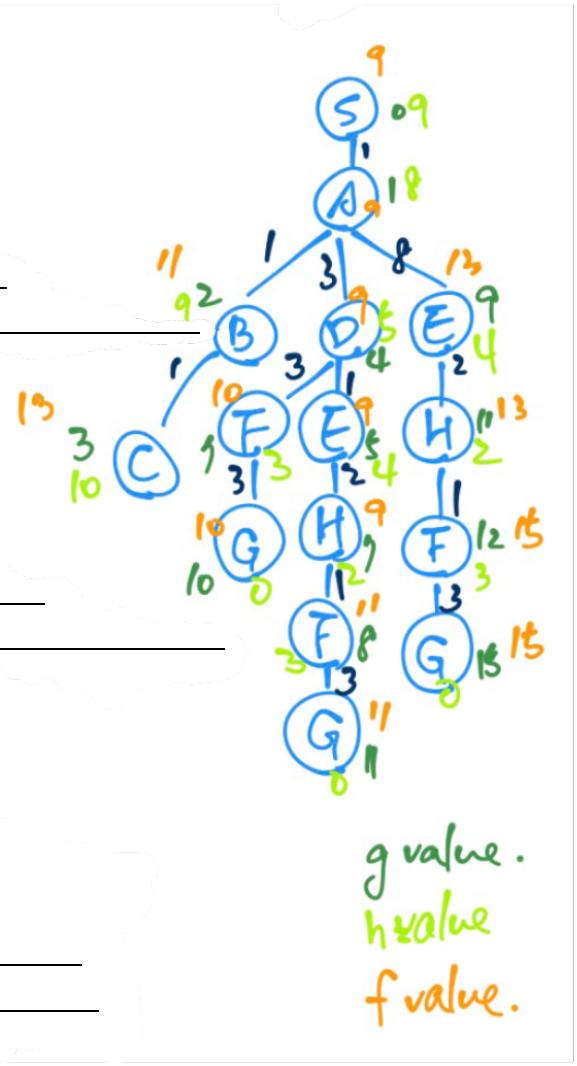
(c) (3 points) A* Search: (where f(n)=g(n)+h(n))

Visited order: SADEHFG

Solution (path length: 10): SADFG

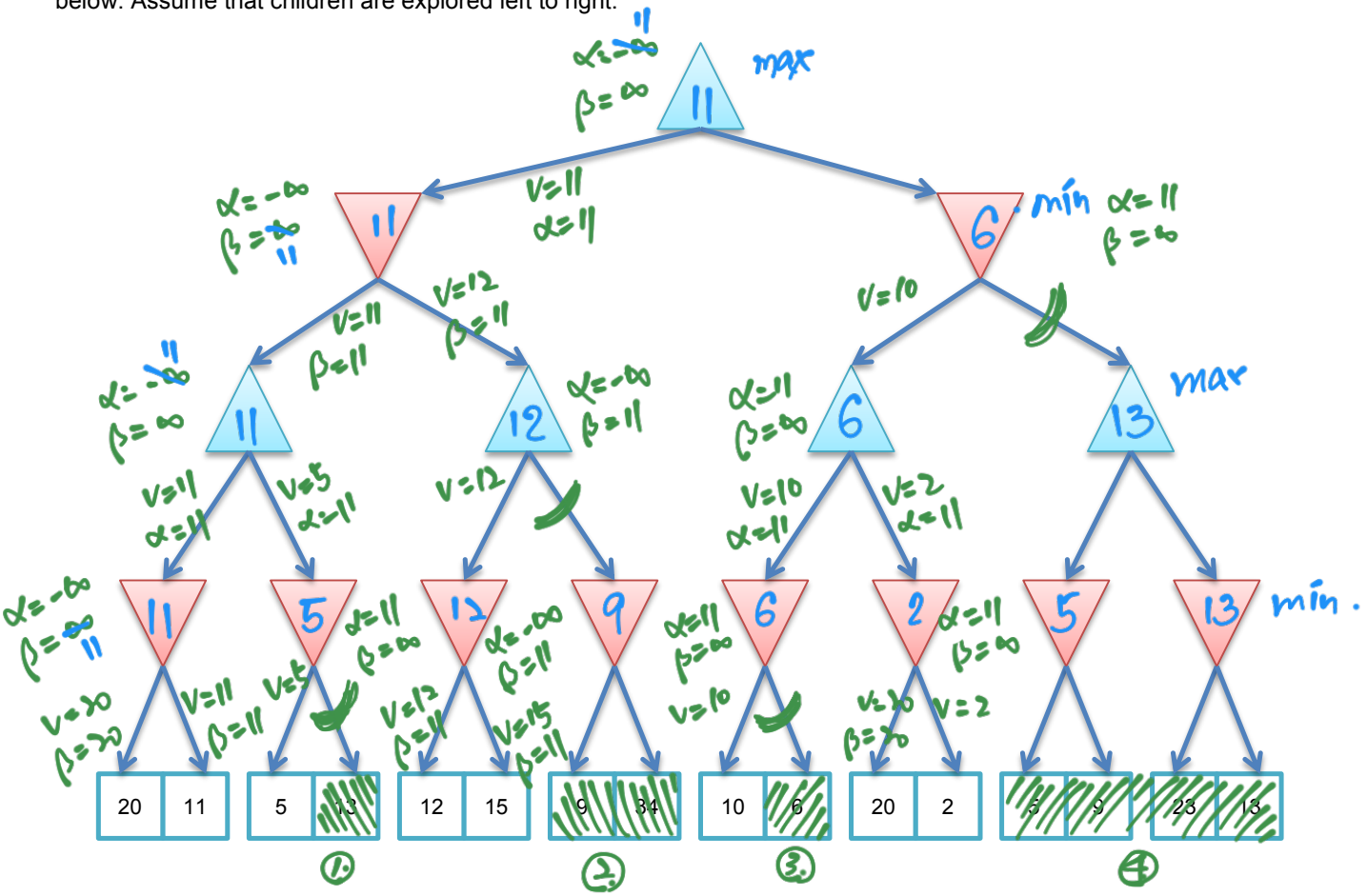
or 5 : for number of nodes

or 4 : for number of links



+9

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. **(6 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.

initialize: $\alpha = -\infty$, $\beta = \infty$

at max node: prune if $v \geq \beta$; else: update $\alpha = \max(\alpha, v)$

at min node: prune if $v \leq \alpha$; else: update $\beta = \min(\beta, v)$

prune:

①: because $(v=5) < (\alpha=11)$

②: because $(v=12) > (\beta=11)$

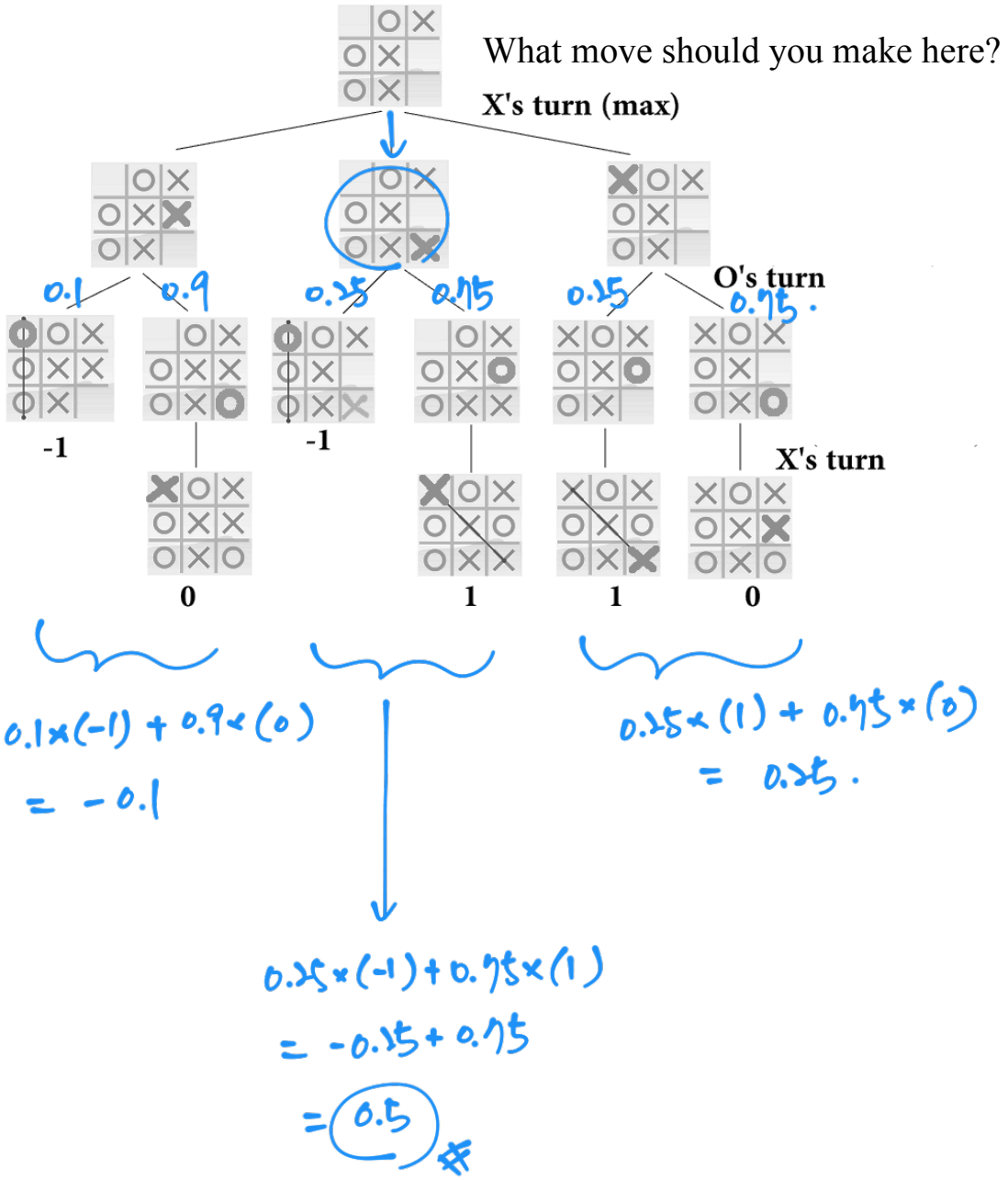
③: because $(v=10) < (\alpha=11)$

④: because $(v=10) < (\alpha=11)$

+4

4 **Expectimax:** (4 points)

You, are playing tic tac toe as the "X" player. You opponent, Olivia ("O" player) is a child, who is playing randomly. Since Olivia is very short, she is much more likely to fill "O"s into lower rows. Specifically, you know that Olivia is three times as likely to choose a space on the middle row than the top row, and three times again more likely to choose a space on the bottom row than the middle row. What move should you make to maximize your chance of winning? (Circle the board position)

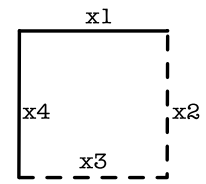


+7

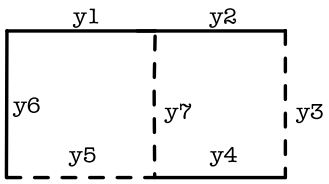
5. Model based object recognition as a CSP

Given a model of an object we want to determine if the object occurs (at least once) anywhere in the given image (with rotation and translation allowed). In this case the model is a square with edges x1, x2, x3, x4. The model has two types of edges: "straight", and "dashed". We can construct a CSP to perform recognition, using one variable for each edge of the model i.e., x1, x2, x3, and x4, whose possible values are a subset of the edges in the image y1, y2, …, y7.

When defining this CSP, you may use the following predicates and any additional ones (e.g. equality) that you find necessary.



TARGET MODEL



CURRENT IMAGE

TYPE(x)	Type of the edge x: "straight" or "dashed".
RIGHTANGLE(x, y)	Whether or not the edges x and y are incident (share a corner point) and perpendicular.

a) (3 points) Write the constraints (unary, binary or n-ary forms are okay). Be sure your CSP will correctly recognize the model object in the images other than the one shown.

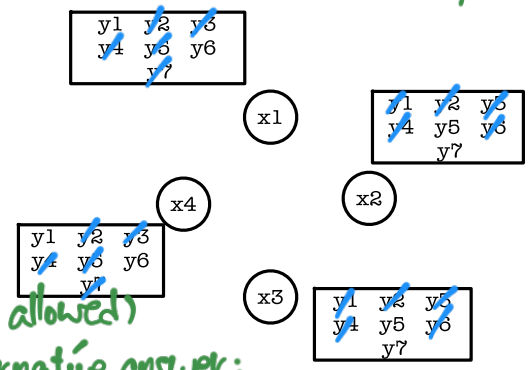
This is the answer we expect:

TYPE(x_1) = TYPE(x_4) = straight

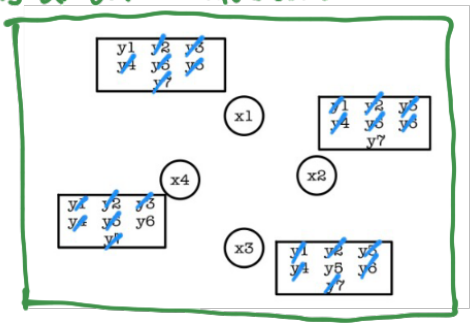
TYPE(x_3) = TYPE(x_2) = dashed

RIGHTANGLE (x_1, x_2)
(x_1, x_4)
(x_3, x_3)
(x_3, x_4)

(assume NO rotation allowed)



This is an alternative answer:



= "not" INCIDENT (x_1, x_3)
(x_2, x_4)

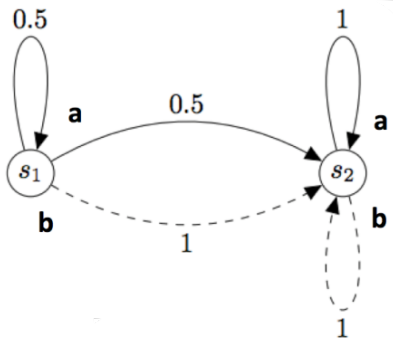
Where INCIDENT(p, q) indicates whether edges p and q are incident or not.

b) (4 points) Is the initial state arc-consistent? NO!

If not, cross out the values for each variable that would be pruned by running AC-3.

+8

6. 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.



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

(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.**

	s_1	s_2
V_0	0	0
V_1	16	-4
V_2	14	-6

assume:
"a" action — solid line
"b" action — dashed line.

$$V_1(s_1) = \max \begin{cases} 0.5(R(s_1,a) + \gamma \cdot V_0(s_1)) + 0.5(R(s_1,a) + \gamma V_0(s_2)) \\ 1(R(s_1,b) + \gamma V_0(s_2)) \end{cases}$$

$$= \max \begin{cases} 0.5(8 + 0.5 \times 0) + 0.5(8 + 0.5 \times 0) = 8 \\ 1(16 + 0.5 \times 0) = 16 \end{cases}$$

$$V_1(s_2) = \max \begin{cases} 1(R(s_2,a) + \gamma V_0(s_1)) \\ 1(R(s_2,b) + \gamma V_0(s_2)) \end{cases}$$

$$= \max \begin{cases} 1(-8 + 0.5 \times 0) = -8 \\ 1(-4 + 0.5 \times 0) = -4 \end{cases}$$

$$V_2(s_1) = \max \begin{cases} 0.5(R(s_1,a) + \gamma V_1(s_1)) + 0.5(R(s_1,a) + \gamma \cdot V_1(s_2)) \\ 1(R(s_1,b) + \gamma V_1(s_2)) \end{cases}$$

$$= \max \begin{cases} 0.5(8 + 0.5 \times (16)) + 0.5(8 + 0.5 \times (-4)) = 11 \\ 1(16 + 0.5 \times (-4)) = 14 \end{cases}$$

$$V_2(s_2) = \max \begin{cases} 1(R(s_2,a) + \gamma V_1(s_2)) \\ 1(R(s_2,b) + \gamma V_1(s_2)) \end{cases}$$

$$= \max \begin{cases} 1(-8 + 0.5 \times (-4)) = -10 \\ 1(-4 + 0.5 \times (-4)) = -6 \end{cases}$$