Final

June 10, 2015

Directions

You have 110 minutes to complete this exam.

- The exam is closed book. No calculators are needed.

- If you have trouble with a question, by all means move on to the next problem—or the next part of the same problem.

- In answers incurring numbers, feel free to not resolve fractions and sums.

- If a question is unclear, feel free to ask me for clarification!

- **Please do not turn the page until I indicate that it is time to begin.**

NAME: _____Abdelaziz Salah Mohammed Abdo_____

NUMBER: _____194037_____

1. (16 points) **True or False**

   Please circle the correct answer.

   (a) Given three random variables A, B, C. $A \perp\!\!\!\perp B$ implies that $A \perp\!\!\!\perp B \,|\, C$. . . . .   T **F**

   (b) $A \perp\!\!\!\perp B \,|\, C$ if and only if: $P(A|B,C) = P(A|C)$. There are other cases, but this case is mendator, so here it should be true **T** F

   (c) Assume your agent uses <u>Minimax search</u>, which results in a certain utility score against an <u>optimally behaving agent</u>. It is *not* possible that your agent gets a <u>lower score</u>, no matter which strategy the other agent follows. . . . .   T **F**

   (d) Breadth first search and uniform cost search are equivalent when every edge has the <u>exact same positive path cost</u>. . . . .   T **F**

   (e) Iterative deepening depth first search is <u>complete and ~~optimal~~</u>. . . . .   **T** F

   (f) One can use variable elimination on a Hidden Markov Model (HMM) to perform exact inference in time *linear* in the number of time steps covered by the HMM.   T F

   (g) Performing a step of the forward algorithm in a Hidden Markov Model takes time linear in the size of the hidden state space. . . . .   T F

   (h) The temporal difference learning (TD-learning) algorithm for reinforcement learning requires that the <u>transition model for the MDP</u> is provided. . . . .   T **F**

2

2. (8 points) **Search**
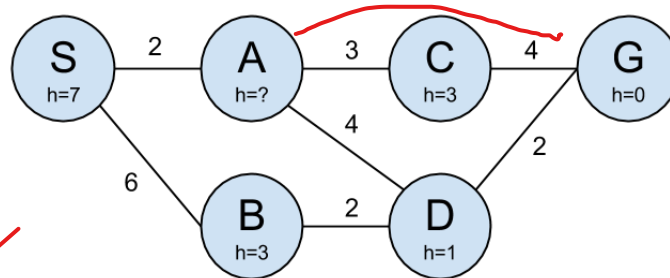
(a) Given the graph above, show the order in which the states are visited by the following **graph search** algorithms listed below. Path cost of an edge is given by the number next to the edge. **Consistent** and **admissible** heuristic estimates h of path cost from a state to the goal state are indicated in the state.
Here are some rules to disambiguate the problem:

- If a state is visited more than once, make sure to write it down each time.
- Ties (e.g. which child to first explore in depth first search) should be resolved according to alphabetic order (i.e. A is expanded before Z).
- Remember to include the start and goal states in your answer (you can denote them as S and G, respectively).
- Treat the goal state as G when you alphanumerically break ties.
- Assume that all search algorithms execute the goal check when nodes are visited, not when their parent is expanded to create them as children.
- Assume graph search does **not** insert a node into the frontier if it is already present in the frontier.
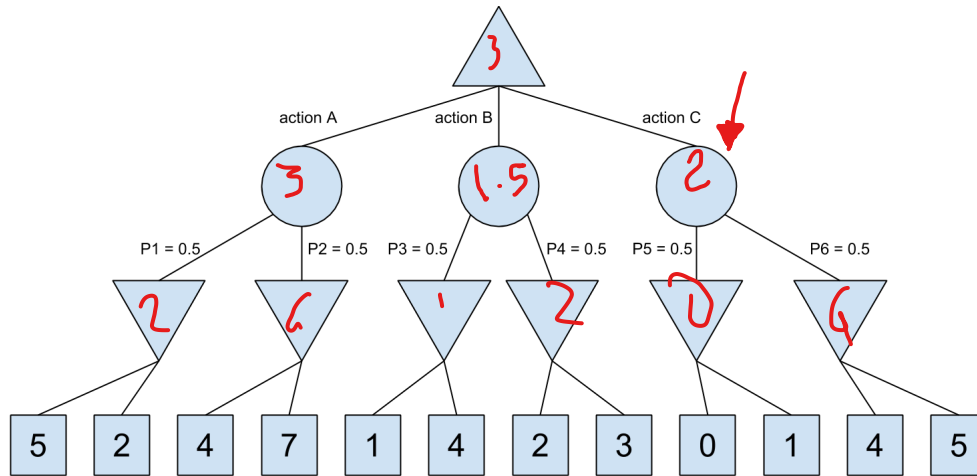
i. (2 points) **Iterative deepening depth first search**

ii. (2 points) **Uniform cost search**

(b) (2 points) The heuristic value of state A is missing. Find the *most general* range of $h(A)$ such that h is **admissible**.

(c) (2 points) Find a value of $h(A)$ such that A* graph search is guaranteed to output the shortest path.

3. (8 points) **Adversarial Search**

In the diagram above, an upward triangle indicates a max node, circles indicate chance nodes, downward triangles indicate min nodes, and squares indicate estimated utility values. The numbers next to the edges under a chance node are the probability of moving from the chance node to subsequent min nodes.

(a) (4 points) Fill in the values of all nodes and determine the utility value at the root node.

(b) (2 points) Instead of the listed value $P6 = 0.5$, find the range of values for $P6$ such that the maximizer always chooses action C (assume $P6 + P5 = 1$).

(c) (2 points) Reverting to the provided value $P6 = 0.5$, using the same reasoning as alpha-beta pruning employs for standard minimax search, are there edges you can prune in this tree? If so, put a line through any such edge. If not, briefly explain why not.

4. (9 points) **Constraint Satisfaction Problems**

A computer science department has three supercomputers C_1, C_2 and C_3 for its research groups. There are also three research groups: Artificial Intelligence, Computational Biology, and Natural Language processing.

Each research group has a list of one or more tasks:

- Artificial Intelligence: AI_1, AI_2, AI_3
- Computational Biology: CMB_1, CMB_2
- NLP: NLP_1

Each task takes 1 full day to complete, and they all need to be scheduled in the same 2-day period of time. The tasks also need to satisfy the following list of constraints:

- Only one task can be run on each computer each day.
- AI_2 needs to be completed before AI_1
- AI_1 cannot be completed on C_3
- AI_3 needs to be run on C_1 on day 2
- CMB_1 and CMB_2 cannot be done on the same computer
- NLP_1 needs to be done on day 2

Each task corresponds to a variable, where the domain for each variable is is the set $\{C_i^{day}\}$ where $i \in \{1, 2, 3\}$ and $day \in \{1, 2\}$.

(a) (3 points) Write down all the constraints using the following notation. Use $ALLDIFF(X_1, ..., X_n)$ to say that variables $\{X_1, ..., X_n\}$ need each have a unique assignment. Use $COMPUTER(X)$ to extract $i \in \{1, 2, 3\}$ representing which computer and $DAY(X)$ to extract $day \in \{1, 2\}$ from an assignment to variable X. For example, if variable $CMB_1 = C_2^1$, then: $DAY(CMB_1) = 1$.

	AI_1	AI_2	AI_3	CMB_1	CMB_2	NLP_1
C_1^1						
C_2^1						
C_3^1						
C_1^2						
C_2^2						
C_3^2						

(b) (3 points) On the table above, apply only the UNARY constraints and cross out the illegal assignments **using a vertical line**.

(c) (3 points) On the table above, check if values are arc-consistent. Treat ALLDIFF as a set of equivalent binary arc constraints. Cross out values that would be pruned by running AC3 using **a horizontal line**.

5. (8 points) **Markov Decision Processes: Value Iteration**

Consider the Markov Decision Process represented by the following graph. Actions provide the listed immediate reward. States with no outgoing actions are terminal states. You may refer to actions as E for east and S for south. Note that these are the *only* available actions. Remaining stationary in the current state is **not** an available action (In state C, it is action E which returns to C with an immediate reward of -4).
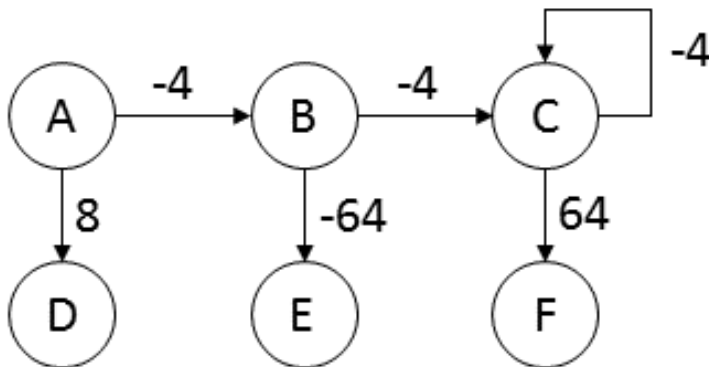
(a) (4 points) If $\gamma = 0.5$ and transitions are deterministic, fill out the following table with the values computed for each iteration of value iteration for the listed states.

	A	B	C
V_0	0	0	0
V_1	8	-4	64
V_2	8	28	64
V_3	10	28	64

(b) (4 points) Now let $\gamma = 1.0$ and transitions are stochastic, where with probability 0.25, an taking an action results in actually performing the *other* of the two available actions. Fill out the following table with the values computed for each iteration of value iteration for the listed states.

	A	B	C
V_0	0	0	0
V_1	5	-19	
V_2			

7

6. (8 points) **Reinforcement Learning**

Consider the grid world shown. Each grid cell has a reward which is earned by entering that cell. The available actions are N, S, E, W for north, south, east, and west. Remember that the process may be stochastic, where an action does not necessarily result in the expected next state.
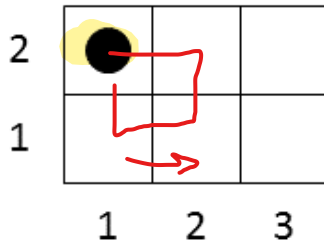
(a) (4 points) Consider the following episodes performed in the grid world. These tuples are of the form $[s, a, s', r]$, where we start in state s, perform action a, end up in state s', and receive immediate reward r which is determined by the state entered. Let $\gamma = 1.0$ for this MDP. Fill in the values computed by the Q-learning algorithm with a learning rate of $\alpha = 0.5$. All Q values are initially 0, and you should fill out each row using values you have computed in previous rows.

1	$[(1,2), E, (2,2), 2]$	$Q((1,2), E) =$	1
2	$[(2,2), S, (2,1), 4]$	$Q((2,2), S) =$	2
3	$[(2,1), W, (1,1), 8]$	$Q((2,1), W) =$	4
4	$[(1,1), N, (2,1), 4]$	$Q((1,1), N) =$	2
5	$[(1,2), E, (1,1), 8]$	$Q((1,2), E) =$	6.5
6	$[(1,1), E, (2,1), 4]$	$Q((1,1), E) =$	4
7	$[(2,1), E, (3,1), 64]$	$Q((2,1), E) =$	32
8	$[(3,1), N, (3,2), 16]$	$Q((3,1), N) =$	

(b) (2 points) Which of the 8 experiences, if any, show that the MDP is stochastic?

(c) (2 points) If instead the learning rate α was 1.0, would Q-learning converge to the true value function for a stochastic MDP if provided with infinite episodes?

7. (10 points) **Probabilistic reasoning**

You are a witness of a night-time hit-and-run accident involving a taxi in Athens. All taxis in Athens are either blue or green, with 9 out of 10 Athenian taxis being green. You perceived the taxi as blue. Extensive testing shows that under the dim lighting conditions, discrimination between blue and green is 75% reliable (i.e. in 75% of the cases, you see blue (or green) when the car *is* actually blue (or green)).

(a) (2 points) Draw the Bayes' net that models this setting.

(b) (4 points) Provide all probability tables necessary to specify the Bayes' net.

(c) (4 points) Provide the equation needed to compute the probability that the taxi is blue given your observation (feel free to not multiply out the different terms).

8. (8 points) **Bayes' Net Parameter Estimation**

Consider the following Bayes' net, where both variables are Boolean:

(a) (2 points) Consider the following training set, where each example provides a value for A and B in the form $< A, B >$ where T is *true* and F is *false*:

$$\{< F, T >, < F, T >, < F, F >, < T, T >\}$$

This data may be equivalently represented as the following table of counts:

$< A, B >$	Count
$< T, T >$	1
$< T, F >$	0
$< F, T >$	2
$< F, F >$	1

What are the maximum likelihood estimates for the CPTs in the network?
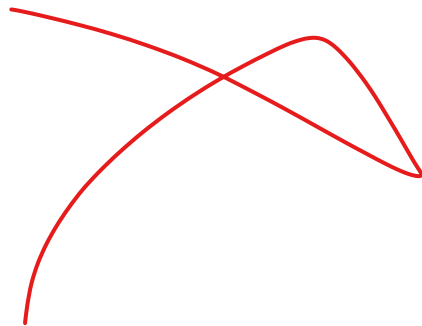
(b) (4 points) Now imagine instead that you have no training values for A and are only given the following examples for $B : \{F, F, F, F, T, T, T\}$. Assume the CPTs of the Bayes' net are initialized by the tables below. Compute the expectations for the missing values for A. In other words, compute $P(A|B = true)$ and $P(A|B = false)$.

A	P(A)
false	0.5
true	0.5

A	B	P(B|A)
false	false	0.6
false	true	0.4
true	false	0.2
true	true	0.8

(c) (2 points) Please interpret your answer to the previous question as (possibly fractional) counts for each of the following four pairs, which *could* be used to estimate an updated set of parameters for the CPTs:

$< A, B >$	Count
$< T, T >$	
$< T, F >$	
$< F, T >$	
$< F, F >$	

Additional space for notes.