CSEP 573 Midterm Exam – February 6, 2016

Name:

This exam is take home and is due on **Sunday February 14th at 11:45 pm**. You can submit it in the online DropBox or to one of the TAs. This exam should not take significantly longer than 3 hours to complete if you have already carefully studied all of course material. Studying while taking the exam may take longer. :)

This exam is open book and open notes, but you must complete all of the work yourself with no help from others.

If you show your work and *briefly* describe your approach to the longer questions, we will happily give partially credit, where possible.

There are 10 pages in this exam.

Scores						
Q.1 (30)	**Q.2 (30)**	**Q.3 (25)**	**Q.4 (30)**	**Q.5 (20)**	**Q.6 (30)**	**Total (165)**

Question 1 – True/False – 30 points

Circle the correct answer each True / False question.

1. True / False – A* Tree Search requires a consistent heuristic for optimality. (3 pt)

2. True / False – In blind search (uninformed search), each node in a search tree corresponds to a node in the state graph. (3 pt)

3. True / False – Greedy search can take longer to terminate than uniform cost search. (3 pt)

4. True / False – Uniform cost search with costs of 1 for all transitions is the same as depth first search. (3 pt)

5. True / False – There exist problems for which an admissible heuristic cannot be found. (3 pt)

6. True / False – Alpha-Beta pruning is an exact algorithm that has been observed to, in practice, double the depth of the minimax tree that can be built. (3 pt)

7. True / False – Policy Iteration always find the optimal policy, when run to convergence. (3 pt)

8. True / False – Higher values for the discount (γ) will, in general, cause value iteration to converge more slowly. (3pt)

9. True / False – Q-learning with constant, ϵ-greedy exploration ($\epsilon = 0.05$) will always converge to the optimal policy. (3 pt)

10. True / False – In probabilistic inference, the random variables are divided into two classes: observed and query. (3 pt)

Question 2 – Short Answer – 30 points

These short answer questions can be answered with a few sentences each.

1. Short Answer – Briefly describe the relationship between admissible and consistent heuristics. When would you use each, and why? (5 pts)

2. Short Answer – Briefly describe when you would use Alpha-beta pruning in minimax search. (5 pts)

3. Short Answer – For Q-learning, when would you prefer to use linear function approximation and when would you just use the tabular version? Is there ever any drawback to using the linear version? (5 pts)

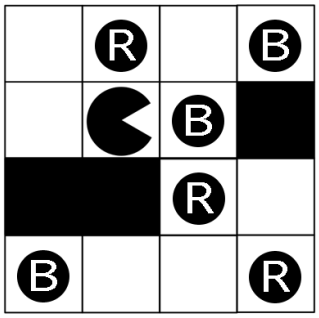
4. Short Answer – Briefly describe the difference between UCS and A* search. When would you prefer to use each, and why? (5 pts)

5. Short Answer – For Q-learning, briefly describe the conditions needed to ensure convergence. Is it guaranteed for any exploration policy? (5 pts)

6. Short Answer – Describe a situation in which you might use Bayes rule during probabilistic inference (5 pts)

Question 3 – Ordered Pacman Search – 25 points

Consider a new Pacman game where there are two kinds of food pellets, each with a different color (red and blue). Pacman has peculiar eating habits; he strongly prefers to eat all of the red dots before eating any of the blue ones. If Pacman eats a blue pellet while a red one remains, he will incur a cost of 100. Otherwise, as before, there is a cost of 1 for each step and the goal is to eat all the dots. There are K red pellets and K blue pellets, and the dimensions of the board are N by M.



$K = 3$, $N = 4$, $M = 4$

1. Give a tight upper bound on the size of the state space required to model this problem. Briefly describe your reasoning. [10 pts]

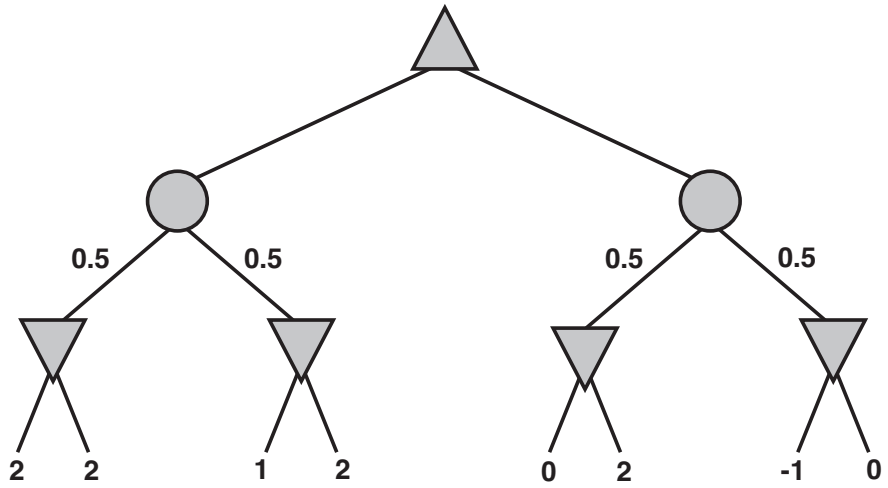
2. Give a tight upper bound on the branching factor of the state space. Briefly describe your reasoning. [5 pts]

3. Which search algorithm would pacman execute to get the optimal path? Why? (describe in one or two sentences) [5 pts]

4. Give an admissible heuristic for this problem. [5 pts]

Question 4 – Game Trees – 30 points

Consider the following game tree, which has min (down triangle), max (up triangle), and expectation (circle) nodes:



1. In the figure above, label each tree node with its value (a real number). [7 pts]

2. In the figure above, circle the edge associated with the optimal action at each choice point. [7 pts]

3. If we knew the values of the first six leaves (from left), would we need to evaluate the seventh and eighth leaves? Why or why not? [5 pts]

4. Suppose the values of leaf nodes are known to be in the range $[-2, 2]$, inclusive. Assume that we evaluate the nodes from left to right in a depth first manner. Can we now avoid expanding the whole tree? If so, why? Circle all of the nodes that would need to be evaluated (include them all if necessary). [11 pts]

Question 5 – Modeling an MDP – 20 points

Consider the following elevator scenario, where you are a rider trying to leave a building at the end of the day.

The building has four floors (ranging in numbers from 1-4) and there is one elevator with four buttons, one for each floor. After a button is pressed, the elevator will move directly to the desired floor 80% of the time, but will move to one of the other two floors with equal probability. For example, if the elevator is on floor 3 and the rider presses 4, there is a 80% chance of arriving at floor 4, 10% chance of floor 2, and 10% chance of floor 1.

In general, the rider will have an equal probably of starting out at floors 2-4, but never starts on floor 1. Furthermore, this is a toll elevator. It costs 10 cents every time you press a button. Finally, since it is late in the day, we will assume that the rider wants to get to floor 1 to go home (stop riding).

1. Model this problem as an MDP. Specify all of the necessary parameters. [15 pts]

2. What is the optimal policy for this problem? [5 pts]

Question 6 – Stutter Step MDP and Bellman Equations – 30 points

Consider the following special case of the general MDP formulation we studied in class. Instead of specifying an arbitrary transition distribution $T(s, a, s')$, the stutter step MDP has a function $T(s, a)$ that returns a next state s' deterministically. However, when the agent actually acts in the world, it often stutters. It only actually reaches s' half of the time, and it otherwise stays in s. The reward $R(s, a, s')$ remains as in the general case.

1. Write down a set of Bellman equations for the stutter step MDP in terms of $T(s, a)$, by defining $V^*(s)$, $Q^*(s, a)$ and $\pi^*(s)$. Be sure to include the discount γ. [20 pts]

2. Consider the special case of the stutter step MDP where $R(s, a, s')$ is zero for all states except for a single good terminal state, which has reward 1, and a single bad terminal state, with reward -100. Furthermore, assume all states s are connected to both terminal states (there exists some sequence of actions that will go from s to the terminal state with non-zero probability).

   If $\gamma = 1$, briefly describe what the optimal values $V^*(s)$ for all states would look like. [5 pts]

3. Again, set the rewards as in the previous question, but now consider $\gamma = 0.1$ and describe $V^*(s)$. Would the optimal policy $\pi^*(s)$ change? [5 pts]