CSEP 573 Midterm Exam – Feb 1, 2024

Name:

This exam is take home and is due on **Sun Feb 18**. Please submit through the course Gradescope. 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. Please feel free to post clarification questions to the class message board, but please do not discuss solutions there.

Partial Credit: If you show your work and *briefly* describe your approach to the longer questions, we will happily give partially credit, where possible. We reserve the right to take off points for overly long answers. Please do not just write everything you can think of for each problem.

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 – Minimax is optimal against perfect opponents. (3 pt)

3. True / False – There exist problems for which an admissible heuristic cannot be found. (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 – Alpha-Beta pruning can introduce errors during mini-max search. (3 pt)

6. True / False – Each state can only appear once in a state graph. (3 pt)

7. True / False – Policy Iteration always finds 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 – For MDPs, adapting the policy to depend on the previous state, in addition to the current state, can lead to higher expected reward. (3pt)

10. True / False – Graph search can sometimes expand more nodes than tree search. (3pt)

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 – Explain the relation between tree and graph search and when you would choose one over the other. (5 pts)

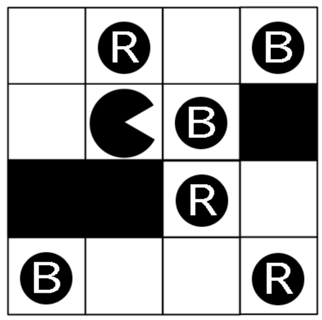
4. Short Answer – Briefly describe the difference between UCS and A* search. When would you prefer to use each, and why? (5 pts)

5. Short Answer – Describe a simple problem that breaks the Markov assumption of MDPs. (5 pts)

6. Short Answer – Briefly describe the difference between value iteration and policy iteration. Describe conditions under which one algorithm might be preferred to the other, in practice. (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 non-trivial upper bound on the size of the state space required to model this problem. Briefly describe your reasoning. [10 pts]

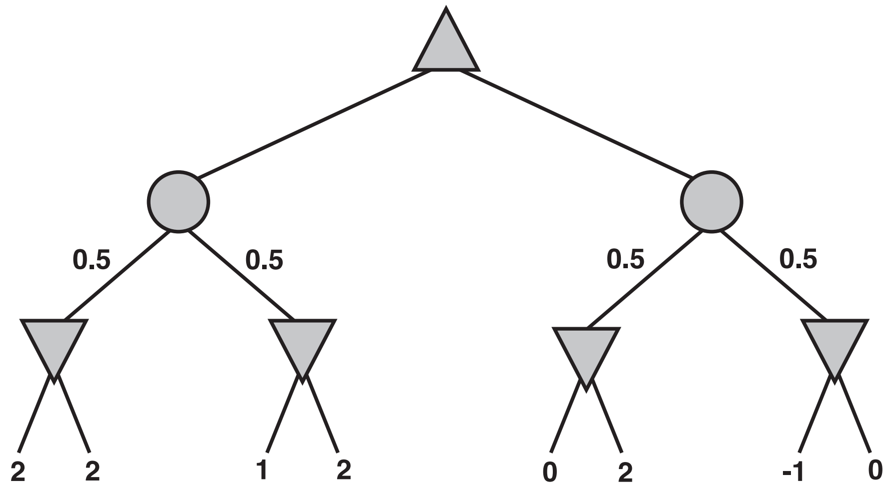
2. Give a non-trivial upper bound on the branching factor of the state space. Briefly describe your reasoning. [5 pts]

3. Name a search algorithm pacman could execute to get the optimal path? Briefly justify your choice (describe in one or two sentences) [5 pts]

4. Give a non-trivial 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]

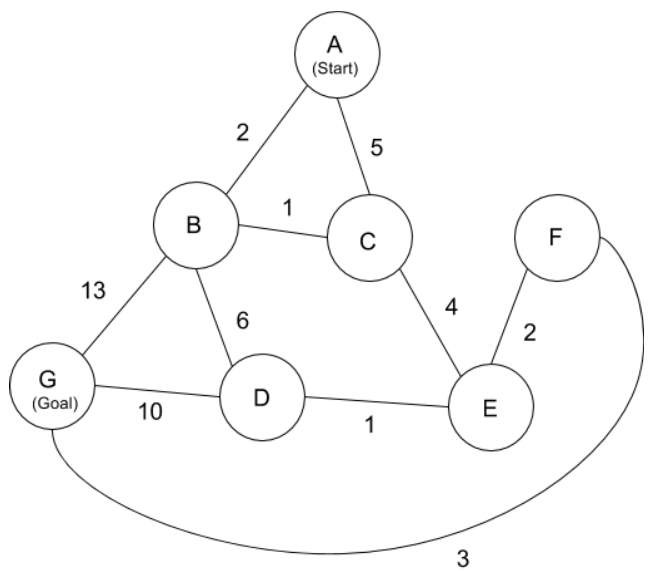
3. If we knew the values of the first six leaves (from left), would we need to evaluate the seventh and eighth leaves? Why or why not? [5 pts]

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 – Tree Search – 30 points

Refer to the provided state graph and execute each of the specified algorithms. Document the sequence in which the nodes are expanded. A node is deemed expanded when it is removed from the fringe. The numbers adjacent to each edge represent the cost of moving between states.

In case of a tie, use alphabetical order as the tie-breaker (for instance, 'A' should precede 'B' in the fringe, assuming all other factors are equal). You should use cycle checking to ensure you never expand the same state twice. Every ordering should always start with the start node and end with the goal node.



State s	H1(s)	H2(s)
A (start)	10	12
B	8	11
C	7	8
D	4	4
E	3	4
F	2	3
G (goal)	0	0

1. Breadth first search [5 pts]

2. Depth first search [5 pts]

3. Iterative deepening [5 pts]

4. Uniform cost search [5 pts]

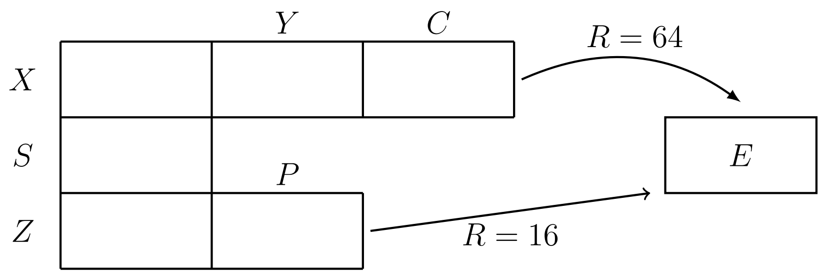
5. Provide the expansion ordering for A^* search with heuristic H2. [5 pts]

6. List which, if any, of the two heuristics are admissible [2.5 pts]

7. List which, if any, of the two heuristics are consistent [2.5 pts]

Question 6 – MDP Exploration – 30 points

Pacman is now a CS student at UW. He finds himself in a (very) simplified, *deterministic* grid world MDP representation of UW depicted below, starting at state S. Pacman can take actions up, down, left or right. If an action moves him into a wall, he will stay in the same state. States C and P represent the CS building and the Party building respectively (their labels both appear above the relevant grid square). Pacman will study at the CS building or he will party at the Party building. At states C and P Pacman can take the exit action to receive the indicated reward and enter the terminal state, E. $R(s, a, s') = 0$ otherwise. Once in the terminal state the game is over and no actions can be taken. Let the discount factor $\gamma = \frac{1}{2}$ for this problem, unless otherwise specified.



1. What is the optimal policy for Pacman on the grid world above? [5 pts]

2. Given an arbirtrary initial policy, what is the maximum number of iterations k it will take before $V^{\pi_k}(S) = V^{\pi^*}(S)$? [5 pts]

3. What are all the values that $V^{\pi_k}(S)$ will take on during the entire process of Policy Iteration given every possible initial policy. [5 pts]

4. Let us mess with Pacman's ability to study. Your task is to change some of the MDP parameters so that Pacman no longer desires to visit the CS building. S is where Pacman starts (the square to the right of the label S). All subquestions are independent of each other so consider each change on its own.

   (a) What discount factor forces Pacman to be indifferent between studying and partying given that he starts at state S? [5 pts]

   (b) Tweak the reward function such that Pacman will always choose partying over studying. Write a bound on $R(C, exit, E)$, that guarantees Pacman exits from P instead of C. [5 pts]

   (c) Let us make the reward of studying so distant that Pacman no longer exits from C. We'll accomplish this by adding a certain number of grid positions, x of them, in between Y and C as depicted below. Give a lower bound for x that **guarantees** Pacman does not exit from C. [5 pts]