CS 188
Spring 2011

Introduction to
Artificial Intelligence

Practice Midterm

To earn the extra credit, one of the following has to hold true. Please circle and sign.

A I spent 3 or more hours on the practice midterm.

B I spent fewer than 3 hours on the practice midterm, but I believe I have solved all the questions.

Signature: ____

The normal instructions for the midterm follow on the next page.

- You have 3 hours.
- The exam is closed book, closed notes except a two-page crib sheet.
- Please use non-programmable calculators only.
- Mark your answers ON THE EXAM ITSELF. If you are not sure of your answer you may wish to provide a *brief* explanation. All short answer sections can be successfully answered in a few sentences AT MOST.

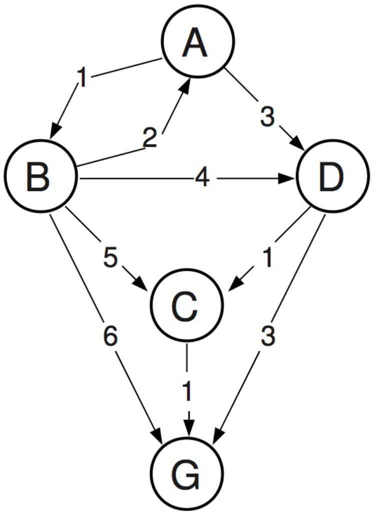
First name	
Last name	
SID	
Login	

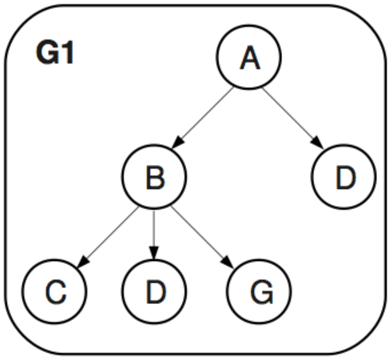
For staff use only:

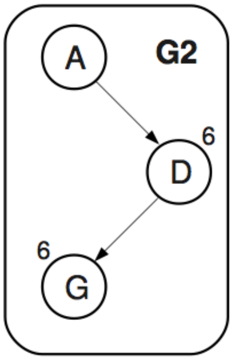
Q1. Search Traces	/15
Q2. Minimax and Expectimax	/12
Q3. n-pacmen search	/10
Q4. CSPs: Course scheduling	/10
Q5. Cheating at cs188-Blackjack	/14
Q6. Markov Decision Processes	/16
Q7. Short answer	/26
Total	/103

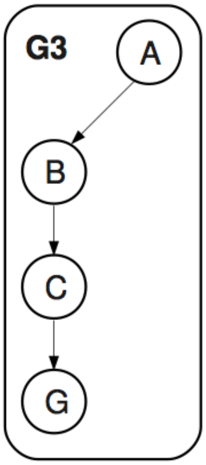
Q1. [15 pts] Search Traces

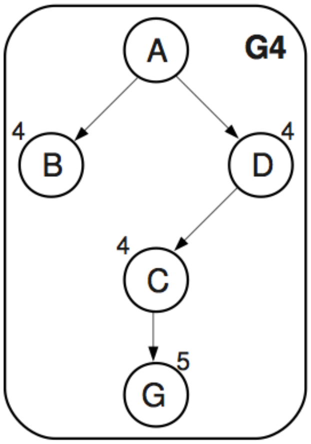
Each of the trees (G1 through G5) was generated by searching the graph (below, left) with a *graph search* algorithm. Assume children of a node are visited in alphabetical order. Each tree shows *only the nodes that have been expanded.* Numbers next to nodes indicate the relevant "score" used by the algorithm's priority queue. The start state is A, and the goal state is G.

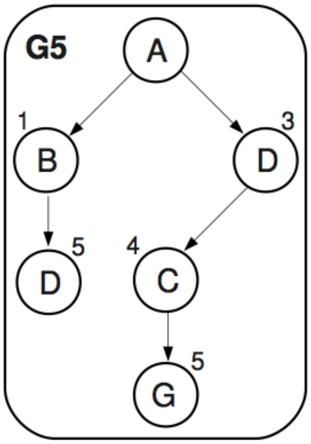












For each tree, indicate:

1. Whether it was generated with depth first search, breadth first search, uniform cost search, or A^* search. Algorithms may appear more than once.
2. *If* the algorithm uses a heuristic function, say whether we used

   $\mathbf{H1} = \{h(A) = 3,\, h(B) = 6,\, h(C) = 4,\, h(D) = 3\}$

   $\mathbf{H2} = \{h(A) = 3,\, h(B) = 3,\, h(C) = 0,\, h(D) = 1\}$
3. For all algorithms, say whether the result was an optimal path (assuming we want to minimize sum of link costs). If the result was *not* optimal, state why the algorithm found a suboptimal path.

Please fill in your answers on the next page.

(a) [3 pts] **G1**:

1. Algorithm:
2. Heuristic (if any):
3. Did it find least-cost path? If not, why?

(b) [3 pts] **G2**:

1. Algorithm:
2. Heuristic (if any):
3. Did it find least-cost path? If not, why?

(c) [3 pts] **G3**:

1. Algorithm:
2. Heuristic (if any):
3. Did it find least-cost path? If not, why?

(d) [3 pts] **G4**:

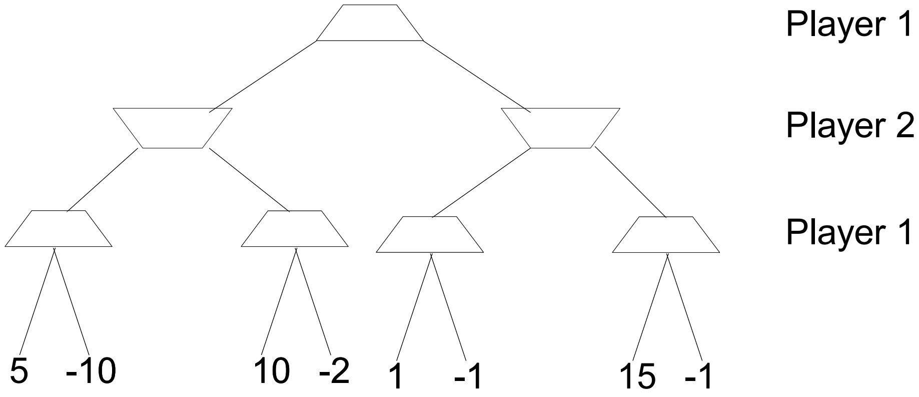
1. Algorithm:
2. Heuristic (if any):
3. Did it find least-cost path? If not, why?

(e) [3 pts] **G5**:

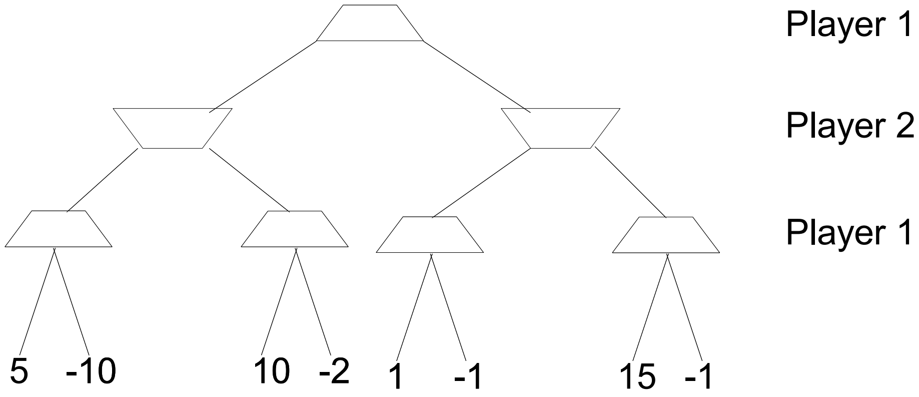
1. Algorithm:
2. Heuristic (if any):
3. Did it find least-cost path? If not, why?

Q2. [12 pts] Minimax and Expectimax

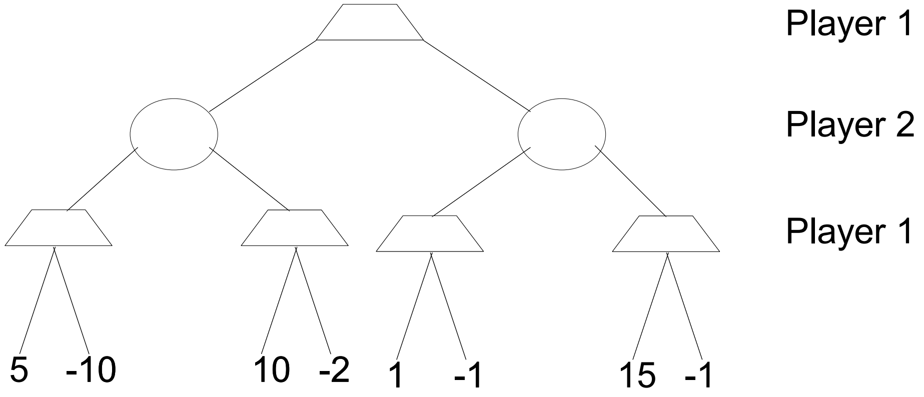
(a) [2 pts] Consider the following zero-sum game with 2 players. At each leaf we have labeled the payoffs Player 1 receives. It is Player 1's turn to move. Assume both players play optimally at every time step (i.e. Player 1 seeks to maximize the payoff, while Player 2 seeks to minimize the payoff). Circle Player 1's optimal next move on the graph, and state the minimax value of the game. Show your work.



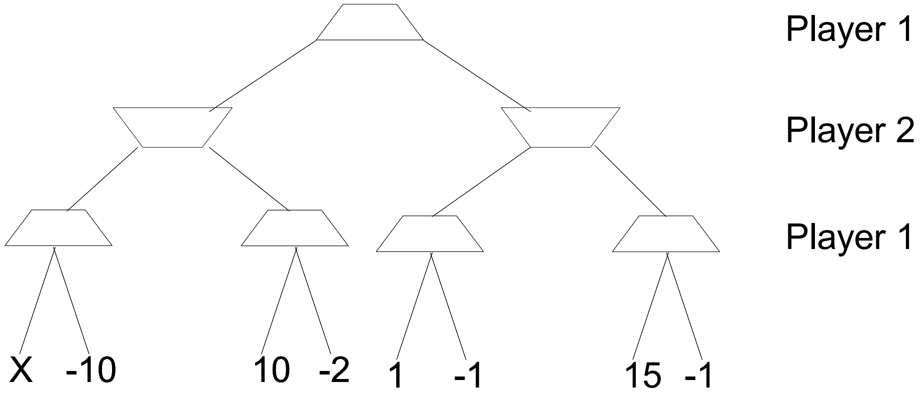
(b) [2 pts] Consider the following game tree. Player 1 moves first, and attempts to maximize the expected payoff. Player 2 moves second, and attempts to minimize the expected payoff. Expand nodes left to right. Cross out nodes pruned by alpha-beta pruning.



(c) [2 pts] Now assume that Player 2 chooses an action uniformly at random every turn (and Player 1 knows this). Player 1 still seeks to maximize her payoff. Circle Player 1's optimal next move, and give her expected payoff. Show your work.



Consider the following modified game tree, where one of the leaves has an unknown payoff x. Player 1 moves first, and attempts to maximize the value of the game.



(d) [2 pts] Assume Player 2 is a minimizing agent (and Player 1 knows this). For what values of x does Player 1 choose the left action?

(e) [2 pts] Assume Player 2 chooses actions at random (and Player 1 knows this). For what values of x does Player 1 choose the left action?

(f) [2 pts] For what values of x is the minimax value of the tree worth more than the expectimax value of the tree?

Q3. [10 pts] n-pacmen search

Consider the problem of controlling n pacmen simultaneously. Several pacmen can be in the same square at the same time, and at each time step, each pacman moves by at most one unit vertically or horizontally (in other words, a pacman can stop, and also several pacmen can move simultaneously). The goal of the game is to have all the pacmen be at the same square in the minimum number of time steps. In this question, use the following notation: let M denote the number of squares in the maze that are not walls (i.e. the number of squares where pacmen can go); n the number of pacmen; and $p_i = (x_i, y_i) : i = 1 \dots n$, the position of pacman i. Assume that the maze is connected.

(a) [1 pt] What is the state space of this problem?

(b) [1 pt] What is the size of the state space (not a bound, the exact size).

(c) [2 pts] Give the tightest upper bound on the branching factor of this problem.

(d) [2 pts] Bound the number of nodes expanded by uniform cost *tree* search on this problem, as a function of n and M. Justify your answer.

(e) [4 pts] Which of the following heuristics are admissible? Which one(s), if any, are consistent? Circle the corresponding Roman numerals and briefly justify all your answers.

1. The number of (ordered) pairs (i, j) of pacmen with different coordinates: $h_1 : \sum_{i=1}^{n} \sum_{j=i+1}^{n} (p_i \neq p_j)$
   (i) Consistent? (ii) Admissible?

2. $h_2 : \frac{1}{2} \max(\max_{i,j} |x_i - x_j|, \max_{i,j} |y_i - y_j|)$
   (i) Consistent? (ii) Admissible?

Q4. [10 pts] CSPs: Course scheduling

An incoming freshman starting in the Fall at Berkeley is trying to plan the classes she will take in order to graduate after 4 years (8 semesters). There is a subset R of required courses out of the complete set of courses C that must all be taken to graduate with a degree in her desired major. Additionally, for each course $c \in C$, there is a set of prerequisites $\text{Prereq}(c) \subset C$ and a set of semesters $\text{Semesters}(c) \subseteq S$ that it will be offered, where $S = \{1, \ldots, 8\}$ is the complete set of 8 semesters. A maximum load of 4 courses can be taken each semester.

(a) [4 pts] Formulate this course scheduling problem as a constraint satisfaction problem. Specify the set of variables, the domain of each variable, and the set of constraints. **Your constraints *need not* be limited to unary and binary constraints.** You may use any precise and unambiguous mathematical notation.

Variables:

Constraints:

(b) [3 pts] The student managed to find a schedule of classes that will allow her to graduate in 8 semesters using the CSP formulation, but now she wants to find a schedule that will allow her to graduate in as few semesters as possible. With this additional objective, formulate this problem as an uninformed search problem, using the specified state space, start state, and goal test.

State space: The set of all (possibly partial) assignments x to the CSP.

Start state: The empty assignment.

Goal test: The assignment is a complete, consistent assignment to the CSP.

Successor function: $\text{Successors}(x) =$

Cost function: $\text{Cost}(x, x') =$

(c) [3 pts] Instead of using uninformed search on the formulation as above, how could you modify backtracking search to efficiently find the least-semester solution?

Q5. [14 pts] Cheating at cs188-Blackjack

Cheating dealers have become a serious problem at the cs188-Blackjack tables. A cs188-Blackjack deck has 3 card types (5,10,11) and an honest dealer is equally likely to deal each of the 3 cards. When a player holds 11, cheating dealers deal a 5 with probability $\frac{1}{4}$, 10 with probability $\frac{1}{2}$, and 11 with probability $\frac{1}{4}$. You estimate that $\frac{4}{5}$ of the dealers in your casino are honest (H) while $\frac{1}{5}$ are cheating ($\neg H$).

Note: You may write answers in the form of arithmetic expressions involving numbers or references to probabilities that have been directly specified, are specified in your conditional probability tables below, or are specified as answers to previous questions.

(a) [3 pts] You see a dealer deal an 11 to a player holding 11. What is the probability that the dealer is cheating?

The casino has decided to install a camera to observe its dealers. Cheating dealers are observed doing suspicious things on camera (C) $\frac{4}{5}$ of the time, while honest dealers are observed doing suspicious things $\frac{1}{4}$ of the time.

(b) [2 pts] Fill in the conditional probability tables for the following random variables: H (honest dealer), D (the card dealt to a player holding 11), and C (suspicious behavior on camera).

H	$P(H)$
1	
0	

H	D	$P(D\|H)$
1	5	
1	10	
1	11	
0	5	
0	10	
0	11	

H	C	$P(C\|H)$
1	1	
1	0	
0	1	
0	0	

(c) [3 pts] What is the probability that a dealer is honest given that he deals a 10 to a player holding 11 and is observed doing something suspicious?

You can either arrest dealers or let them continue working. If you arrest a dealer and he turns out to be cheating, you will earn a \$4 bonus. However, if you arrest the dealer and he turns out to be innocent, he will sue you for -\$10. Allowing the cheater to continue working will cost you -\$2, while allowing an honest dealer to continue working will get you \$1. Assume a linear utility function $U(x) = x$.

(d) [3 pts] You observe a dealer doing something suspicious (C) and also observe that he deals a 10 to a player holding 11. Should you arrest the dealer?

(e) [3 pts] A private investigator approaches you and offers to investigate the dealer from the previous part. If you hire him, he will tell you with 100% certainty whether the dealer is cheating or honest, and you can then make a decision about whether to arrest him or not. How much would you be willing to pay for this information?

Q6. [16 pts] Markov Decision Processes

Consider a simple MDP with two states, S_1 and S_2, two actions, A and B, a discount factor γ of 1/2, reward function R given by

$$R(s,a,s') = \begin{cases} 1 & \text{if } s' = S_1; \\ -1 & \text{if } s' = S_2; \end{cases}$$

and a transition function specified by the following table.

s	a	s'	$T(s,a,s')$
S_1	A	S_1	1/2
S_1	A	S_2	1/2
S_1	B	S_1	2/3
S_1	B	S_2	1/3
S_2	A	S_1	1/2
S_2	A	S_2	1/2
S_2	B	S_1	1/3
S_2	B	S_2	2/3

(a) [2 pts] Perform a single iteration of value iteration, filling in the resultant Q-values and state values in the following tables. Use the specified initial value function V_0, rather than starting from all zero state values. Only compute the entries not labeled "skip".

s	a	$Q_1(s,a)$
S_1	A	
S_1	B	
S_2	A	skip
S_2	B	skip

s	$V_0(s)$	$V_1(s)$
S_1	2	
S_2	3	skip

(b) [2 pts] Suppose that Q-learning with a learning rate α of 1/2 is being run, and the following episode is observed.

s_1	a_1	r_1	s_2	a_2	r_2	s_3
S_1	A	1	S_1	A	-1	S_2

Using the initial Q-values Q_0, fill in the following table to indicate the resultant progression of Q-values.

s	a	$Q_0(s,a)$	$Q_1(s,a)$	$Q_2(s,a)$
S_1	A	$-1/2$		
S_1	B	0		
S_2	A	-1		
S_2	B	1		

(c) [4 pts] Assuming that an ϵ-greedy policy (with respect to the Q-values as of when the action is taken) is used, where $\epsilon = 1/2$, and given that the episode starts from S_1 and consists of 2 transitions, what is the probability of observing the episode from part b? State precisely your definition of the ϵ-greedy policy with respect to a Q-value function $Q(s,a)$.

(d) [4 pts] Given an arbitrary MDP with state set S, transition function $T(s, a, s')$, discount factor γ, and reward function $R(s, a, s')$, and given a constant $\beta > 0$, consider a modified MDP (S, T, γ, R') with reward function $R'(s, a, s') = \beta \cdot R(s, a, s')$. Prove that the modified MDP (S, T, γ, R') has the same set of optimal policies as the original MDP (S, T, γ, R).

(e) [4 pts] Although in this class we have defined MDPs as having a reward function $R(s, a, s')$ that can depend on the initial state s and the action a in addition to the destination state s', MDPs are sometimes defined as having a reward function $R(s')$ that depends only on the destination state s'. Given an arbitrary MDP with state set S, transition function $T(s, a, s')$, discount factor γ, and reward function $R(s, a, s')$ that *does depend* on the initial state s and the action a, define an *equivalent* MDP with state set S', transition function $T'(s, a, s')$, discount factor γ', and reward function $R'(s')$ that depends only on the destination state s'.

By *equivalent*, it is meant that there should be a one-to-one mapping between state-action sequences in the original MDP and state-action sequences in the modified MDP (with the same value). **You do not need to give a proof of the equivalence.**

States: $S' =$

Transition function: $T'(s, a, s') =$

Discount factor: $\gamma' =$

Reward function: $R'(s') =$

Q7. [26 pts] Short answer

Each true/false question is worth 1 point. Leaving a question blank is worth 0 points. **Answering incorrectly is worth -1 point.**

(a) For a search problem, the path returned by uniform cost search may change if we

(i) [*true* or *false*] rescale all step costs by a scalar α: $0 < \alpha < 1$.

(ii) [*true* or *false*] rescale all step costs by a scalar α: $1 < \alpha < 2$.

(iii) [*true* or *false*] add a positive constant C to every step cost.

(b) Assume we are running A^* graph search with a consistent heuristic h. Let p be the node in the fringe about to be expanded in the search. When expanding p, we find that exactly one of its children is the goal state G and the cost of the found path through p to G is K. Let pathcost(p) denote the cost of the path to p that led to p being inserted in the queue. Then we have that

(i) [*true* or *false*] the found path through p to the goal is a shortest path.

(ii) [*true* or *false*] the found path through p to the goal is guaranteed to be at most $K - \text{pathcost}(p)$ longer than the shortest path.

(iii) [*true* or *false*] the found path through p to the goal is guaranteed to be at most $K - \text{pathcost}(p) - h(p)$ longer than the shortest path.

(iv) [*true* or *false*] the found path through p to the goal is guaranteed to be the shortest path going through p to the goal state.

(c) Consider two consistent heuristics, H_1 and H_2, in an A^* search seeking to minimize path costs in a graph. Assume ties do not occur in the priority queue. If $H_1(s) \leq H_2(s)$ for all s, then

(i) [*true* or *false*] A^* search using H_1 will find a lower cost path than A^* search using H_2.

(ii) [*true* or *false*] A^* search using H_2 will find a lower cost path than A^* search using H_1.

(iii) [*true* or *false*] A^* search using H_1 will not expand more nodes than A^* search using H_2.

(iv) [*true* or *false*] A^* search using H_2 will not expand more nodes than A^* search using H_1.

(d) Consider the following statements about α-β pruning. Alpha-beta pruning

(i) [*true* or *false*] may not find the minimax optimal strategy.

(ii) [*true* or *false*] prunes the same number of subtrees independent of the order in which successor states are expanded.

(iii) [*true* or *false*] generally requires more run-time than minimax on the same game tree.

(e) Consider a zero-sum game adversarial game. The minimizer is played by a computer program that is fast enough to perform min-max search all the way to the end of the game and it plays according to the thus-found moves. It is the minimizer's turn to play, and the minimizer's computer program returns a win of some positive value for the minimizer. Then we have that

(i) [*true* or *false*] the minimizer is guaranteed to win the game only if the maximizer also plays the min-max strategy.

(ii) [*true* or *false*] the minimizer is guaranteed to win the game only if the maximizer plays a deterministic strategy.

(iii) [*true* or *false*] if the maximizer is known to make moves uniformly at random every other turn, then the minimizer is not necessarily maximizing pay-off.

(f) Arjun, John, Jonathan, and Lubomir all get to act in an MDP $(S, A, T, \gamma, R, s_0)$. Arjun runs value iteration until he finds V^* which satisfies $\forall s \in S : V^*(s) = \max_{a \in A} \sum_{s'} T(s, a, s')(R(s, a, s') + \gamma V^*(s'))$ and acts according to $\pi_{\text{Arjun}} = \arg\max_{a \in A} \sum_{s'} T(s, a, s')(R(s, a, s') + \gamma V^*(s'))$. John acts according to an arbitrary policy π_{John}. Jonathan takes John's policy π_{John} and runs one round of policy iteration to find his policy π_{Jonathan}. Lubomir takes Arjun's policy and runs one round of policy iteration to find his policy π_{Lubomir}. Then we have that

(i) [*true* or *false*] there are MDP's in which John would outperform Jonathan.

(ii) [*true* or *false*] there are MDP's in which Jonathan would outperform Lubomir.

(iii) [*true* or *false*] there are MDP's in which Lubomir would outperform Arjun.

(iv) [*true* or *false*] there are MDP's in which Arjun would outperform Lubomir.

(v) [*true* or *false*] there are MDP's in which John would outperform Arjun.

(g) Bob notices value iteration converges more quickly with smaller γ and rather than using the true discount factor γ, he decides to use a discount factor of $\alpha\gamma$ with $0 < \alpha < 1$when running value iteration. Then we have that

(i) [*true* or *false*] while Bob will not find the optimal value function, he could simply rescale the values he finds by $\frac{1-\gamma}{1-\alpha}$ to find the optimal value function.

(ii) [*true* or *false*] if the MDP has zero rewards everywhere, except for a single transition at the goal with a positive reward, then Bob will still find the optimal policy.

(iii) [*true* or *false*] if the MDP's transition model is deterministic, then Bob will still find the optimal policy.

(iv) [*true* or *false*] Bob's policy will tend to more heavily favor short-term rewards over long-term rewards compared to the optimal policy.