EECS 492, Winter 2010
Introduction to Artificial Intelligence
Midterm 1 (65 points)

Name: ____

Uniqname: ____

This exam is closed book: you may not use the textbook, lecture slides, notes, or any type of computing device. Write your name at the top of every page.

"I have neither given nor received aid on this examination, nor have I concealed any violations of the honor code."

signed ____

1	
2	
3	
4	
5	
6	
7	
8	
total	

Problem 1. Agents and Environments

1.1. (2 points) What type of agent predicts the effects of its actions before it performs them?

A) Reflexive
B) Autonomous
C) Model-based
D) Genetic

1.2. (4 points) Characterize the environments below by circling the properties that apply:

The wumpus world:

Fully Observable	Partially Observable
Stochastic	Deterministic
Discrete	Continuous
Episodic	Sequential

A system that solves instances of the Traveling Salesman problem:

Fully Observable	Partially Observable
Stochastic	Deterministic
Discrete	Continuous
Episodic	Sequential

Problem 2. Basic Search

2.1. (3 points) What type of queue (e.g., FIFO, LIFO) do the algorithms use below?

BFS _________ DFS _________ IDS ___________

2.2. (2 points) Breadth-first search (BFS) and Iterative Deepening Search (IDS) have similar computational complexities, but they differ by a factor of 'd'. What are the complexities of the two algorithms?

BFS ________ IDS _______

2.3. (2 points) What accounts for this difference of a factor of 'd'? (Circle all that apply)

A) BFS expands nodes up to a level deeper than the solution.
B) IDS expands shallow nodes multiple times.
C) IDS checks for goal nodes *before* inserting them into the fringe; BFS checks for goals only when removing nodes from the fringe.
D) BFS's heuristic function prunes portions of the search tree.

2.4. (2 points) Which of the following statements are true? (Circle all that apply)

A) A consistent heuristic function is always admissible.
B) An admissible heuristic function is always consistent.
C) Graph-search A* requires a consistent heuristic function to be optimal.
D) Given two admissible heuristic functions h1() and h2(), we can derive a new admissible heuristic function h3() = h1() + h2().
E) A*, like IDS, has O(n) memory complexity.

2.5. (4 points) Suppose you are given not only a heuristic function h() that returns an lower bound on the cost-to-go, but a second heuristic function k() that returns a upper *bound* on the cost-to-go. Can k() be used to improve the performance of the algorithm, without sacrificing optimality or completeness? If so, how do you use k()? If not, why not?

Problem 3. Paper, Rock, Scissors.

Suppose we are playing paper-rock scissors with an opponent that has a simple strategy: it simply cycles through 20 pre-determined actions (see below) though its position is unknown to you. The state of the other agent is simply a number 1-20 which encodes the position of its next move.

```
00000000011111111112
12345678901234567890
--------------------
RPPSRPSSRPSRRRPSSRPR
```

3.1. (2 points) The first time you play, you observe an S. What is the belief state (remember, the state reflects its *next* move)?

3.2. (2 points) The second time you play, you observe an R. What is the belief state?

3.3. (4 points) Supposing that the sequence of pre-determined actions is actually N steps long (as opposed to 20 in the case above). About how many moves will you need to observe to determine its state, assuming there is no particular pattern to the sequence?

 Your uniqname: ____________________________

Problem 4. Genetic Algorithms

4.1. (2 points) Which of the following statements are true?

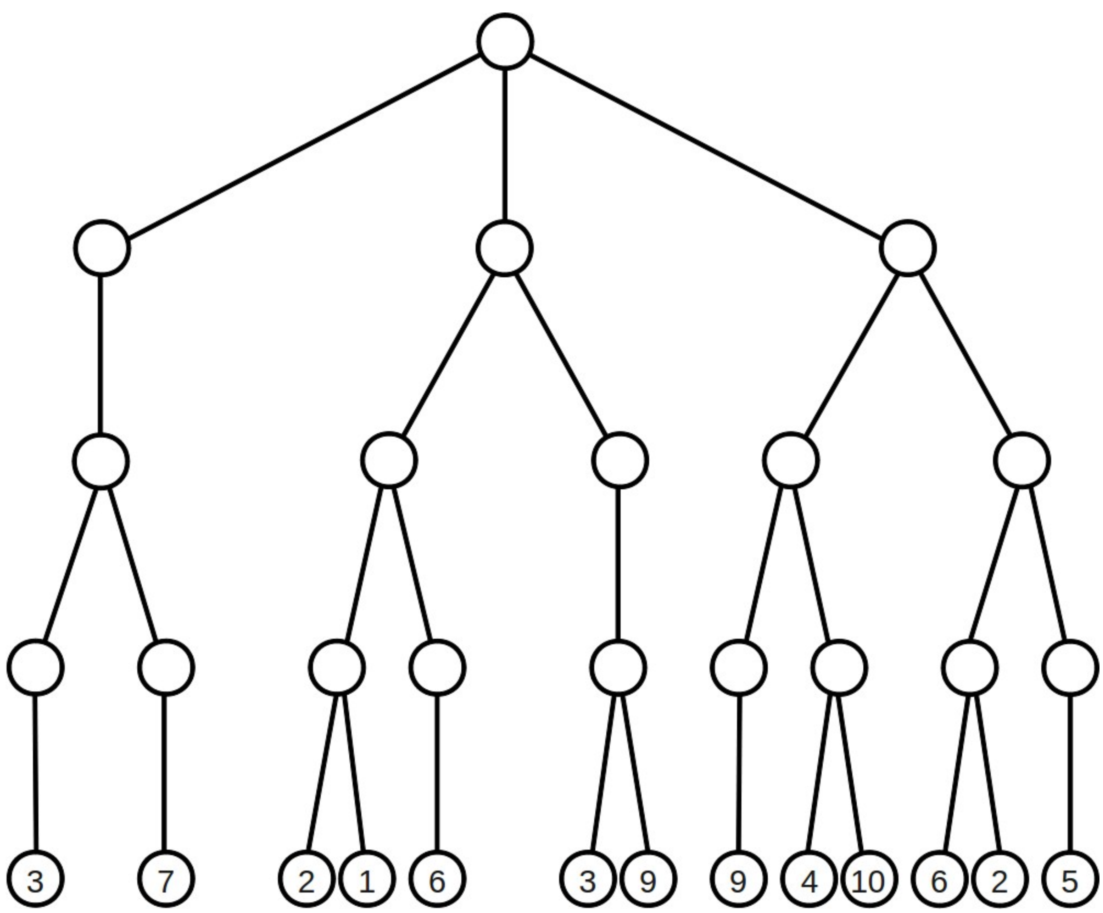
A) Genetic search is a form of local search.
B) Genetic search is complete given finite time.
C) An advantage of genetic search is its ability to work well without the need for a carefully designed representation.
D) Genetic algorithms use a constant amount of memory.

Problem 5. Adversarial Search

Consider the game tree below.

5.1. (4 points) Apply the basic mini-max algorithm, labeling the value of every node in the game tree.

5.2. (4 points) Identify nodes (or entire sub-trees) that will not be explored when using the alpha-beta pruning by circling them.



 Your uniqname: ____________________________

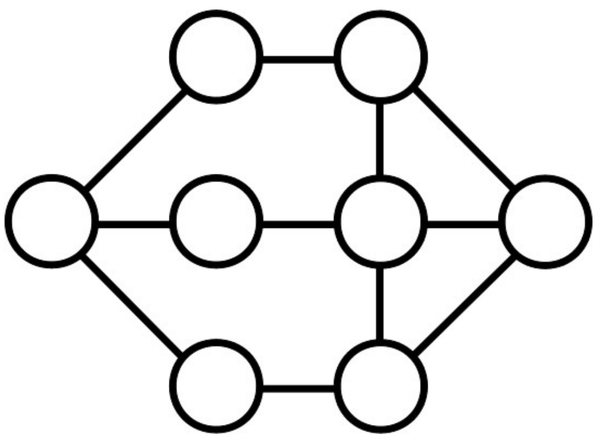
5.3. (2 points) The exploration of mini-max most closely resembles:

A) Breadth-first search
B) Depth-first search
C) Iterative deepening search
D) Hill climbing
E) Best first search

5.4. (4 points) Why might a real-world implementation of mini-max use iterative deepening, rather than minimax to a pre-determined depth?

Problem 6. Constraint Satisfaction

We wish to color the graph below using four different colors.



6.1. (2 points) What is the computational complexity of solving this CSP using TreeSearch, in terms of **n** (the number of nodes) and **d** (the number of colors)?

6.2. (2 points) Identify (by circling them) the smallest subset of the nodes that, once removed from the graph, cause the remainder of the graph to be a tree.

6.3. (3 points) What is the time complexity of solving the CSP (for the whole graph), exploiting the induced tree structure? Assume that the cost of applying the variable consistency check on each edge is K.

 Your uniqname: ______________________________

Problem 7. Propositional Logic

7.1. (5 points) Fill in the truth tables for the logical connectives.

A	B	¬A	A v B	A ^ B	A => B	A <=> B
0	0					
0	1					
1	0					
1	1					

Problem 8. Matching

8.1. (10 points, -1 for each wrong.) Match the terms (letters on left) with the most appropriate phrase on the right:

______ A) Crossover
______ B) Local maximum
______ C) Heuristic function
______ D) Belief States
______ E) Satisfiable
______ F) Forward checking
______ G) Conflict set
______ H) Valid
______ I) Entailment
______ J) Alpha-Beta
______ K) Quiescence
______ L) Implication
______ M) Model

1. With mutation, mechanism for genetic reproduction.
2. Sound tree pruning in adversarial search
3. Eliminate values from domain that are inconsistent with most recent variable assignment.
4. Reduces incomplete knowledge problems to search.
5. A desirable property of a leaf node in mini-max search
6. Determines the truth/falsity of every proposition
7. Used to back-jump further
8. A sentence that is true in at least one model
9. =>
10. An optimization hazard
11. Condition-Action rule
12. Lower bound on cost-to-go
13. Reduces memory requirements of DFS
14. |=
15. A sentence that is true in every model