## CS-271P, Intro to A.I. — Mid-term Exam — Winter Quarter, 2018

YOUR NAME: \_\_\_\_\_

YOUR ID: \_\_\_\_\_\_ ID TO RIGHT: \_\_\_\_\_ ROW: \_\_\_\_\_ SEAT: \_\_\_\_\_

# Please turn off all cell phones now.

The exam will begin on the next page. Please, do not turn the page until told.

When told to begin, check first to ensure that your copy has all the pages, as numbered 1-15 in the bottom-right corner of each page. We will supply a new exam for any copy problems.

The exam is closed-notes, closed-book. No calculators, cell phones, electronics.

<u>Clear your desk except for pen, pencil, eraser, & water bottle. Put backpacks under your seat.</u> Please do not detach the provided scratch paper from the exam.

After you first stand up from your seat, your exam is over and must be turned in immediately. You may turn in your Midterm exam early and leave class when you are finished.

This page summarizes the points for each question, so you can plan your time.

- 1. (4 pts total, 1 pt each) TASK ENVIRONMENT.
- 2. (10 pts total) GENETIC ALGORITHMS: N-QUEENS PROBLEM.
- 3. (8 pts total, 2 pts each) HYPOTHETICAL ALPHA-BETA PRUNING.
- 4. (7 pts total, 1 pt each) PROBABILITY FORMULAS.
- 5. (10 pts total) BAYESIAN NETWORKS.
- 6. (10 pts total, 2 pts each) STATE-SPACE SEARCH.
- 7. (12 pts total, 1 pt each) SEARCH QUESTIONS.
- 8. (10 pts total, 1/2 pt each) SEARCH PROPERTIES.
- 9. (5 pts total, -1 pt for each error, but not negative) MINI-MAX SEARCH IN GAME TREES.
- 10. (10 pts total, -1 pt for each error, but not negative) ALPHA-BETA PRUNING.
- 11. (14 pts total, 1 pt each) AGENT/SEARCH CONCEPTS.

### The Exam is printed on both sides to save trees! Work both sides of each page!

**1.** (**4 pts total, 1 pt each**) **TASK ENVIRONMENT.** Your book defines a task environment as a set of four things, with acronym PEAS. Fill in the blanks with the names of the PEAS components.

P\_\_\_\_\_ E\_\_\_\_ A\_\_\_\_ S\_\_\_\_

2. (10 pts total. Fractional points will be accumulated across all problems, and then at the end rounded upward in your favor) GENETIC ALGORITHMS: N-QUEENS PROBLEM. You are a robot that is assigned to solve the N-Queens problem, where N = number of rows = number of columns. You choose to use a <u>Genetic Algorithm</u>. Recall that each column contains exactly one queen. A state is represented as a vector of integers wherein vector element *i* gives the row number of the queen in column *i*. Queens attack each other if they are in the same row or diagonal. (Queens also attack each other if they are in the same column, but the representation exploits this constraint by requiring exactly one queen per column.) A solution is a board position in which each queen in each column is assigned to a row and no queen pairs conflict (= no queens attack each other). The value of a state is the number of queen pairs NOT in conflict, so higher values are better.

This problem is about the 4-Queens problem. The example 4-Queens board below gives the representation and value of the state shown. <u>Note that all vector coordinates and row numbers are one-based.</u>



Representation( state shown ) = [1 2 4 3]Value( state shown ) = 4Note: Four queen pairs (A1:C4, A1:D3,<br/>B2:C4, B2:D3) are not in conflict

**2.a.** (**4 pts total, 1 pt each**) **Genetic algorithm concepts for N-queens.** For each of the genetic algorithm terms on the left, write in the letter corresponding to the best answer or the correct definition on the right <u>as modified</u> to apply to the N-queens problem. The first one is done for you as an example.

А	Individual	А	A vector of N numbers in which each element $\underline{i}$ is the row number of the queen in column $i$ of a given board position
	Fitness function	В	A set of $k$ states, each of which represents a chess board with N queens
	Mutation	С	The act of combining two chess boards by splitting them at the same random position and exchanging the right-hand sides of each board
	Crossover	D	The act of moving a randomly chose queen to a new random row on a given chess board
	Population	Е	The number of queens that do not attack each other on a given chess board

# **2.b.** (6 pts total, 1/2 pt each. Fractional points will be accumulated across all problems, and then at the end rounded upward in your favor) Genetic algorithm application.

**2.b.i.** (3 pts total, 1/2 pt each) Genetic algorithm application: Fitness. Consider the initial population shown below. On the line labeled 'Fitness' indicate the fitness of each individual, as an integer. On the line labeled 'Probability of Selection' indicate the probability that each individual will be selected for the next generation, as a common fraction. The first one is done for you as an example.

	State1	State2	State3	State4
Individual	[2434]	[3 3 1 1]	[2124]	[1 1 1 1]
Fitness	3			
Probability of Selection	3/8			

**2.b.ii.** (1.5 pts total, 1/2 pt each) Genetic algorithm application: Crossover. State1, State2, and State3 have been selected to contribute to the next generation. Indeed, the highly-fit State1 has been selected twice. State1 and State2 have been paired to become the parents of two children via cross-over, as have State1 and State3. For each pair of **Parents**, a Cross-over point has been selected randomly, as indicated by the ' $\underline{X}$ ' position within each individual. Fill in the blanks with the two **Children** that will result from each Crossover operation (for a total of four children). The first one is done for you as an example.

Parents	[2 4 3 <u>X</u> 4]	[3 3 1 <u>X</u> 1]	*	[24 <u>X</u> 34]	[2 1 <u>X</u> 2 4]
Children	[2 4 3 1]		*		

**2.b.iii.** (1.5 pts total, 1/2 pt each) Genetic algorithm application: Mutation. For each of the following individuals, write the state that would result from mutating that individual at the indicated randomly chosen position to the indicated randomly chosen new value. The first one is done for you as an example.

Individual	[3 1 2 3]	[3 3 4 1]	[2 1 2 4]	[1 4 2 3]
Randomly chosen position	3	2	4	1
Randomly chosen new value	4	2	1	3
Resulting mutated vector	[3 1 4 3]			

**3.** (8 pts total, 2 pts each) HYPOTHETICAL ALPHA-BETA PRUNING. For each of the game-trees shown below, indicate for which values of x the dashed branch will be pruned. Write your answer as  $x \le M$  or  $x \ge N$ , where M and N are integers. If the pruning will not happen for any value of x, write 'None'. If pruning will happen for all values of x, write 'All'. Work each game tree using Depth-First Search in order from left to right. An example tree is given for you below.



**4.** (**7 pts total, 1 pt each**) **PROBABILITY FORMULAS.** Write out the following probability formulas. Below, "in terms of X" means X should appear in your answer. All answers should be formulas, not text.

**4.a.** (1 pt) Write the formula for  $P(A \land B)$  in terms of  $P(A \lor B)$  and possibly other terms.

 $P(A \land B) =$ 

**4.b.** (1 pt) Write the formula for the conditional probability P(A | B).

 $P(A \mid B) =$ 

**4.c.** (1 pt) Factor  $P(A \land B \land C)$  completely using the Product Rule (or Chain Rule). You may use any variable ordering you wish.

 $P(A \land B \land C) =$ 

**4.d.** (1 pt) Given a joint probability distribution  $P(A \land B \land C)$ , use the Sum Rule (or Law of Total Probability) to write the marginal probability of P(A).

**P**(**A**) =

4.e. (1 pt) Write Bayes' Rule (or Bayes' Theorem).

 $\mathbf{P}(\mathbf{A} \mid \mathbf{B}) =$ 

**4.f.** (1 pt) Assume that A and B are independent. Write  $P(A \land B)$  in terms of P(A) and P(B) and possibly other terms.

 $P(A \land B) =$ 

**4.g.** (1 pt) Assume that A and B are conditionally independent given C. Write  $P(A \land B | C)$  in terms of P(A | C) and P(B | C) and possibly other terms.

 $\mathbf{P}(\mathbf{A} \land \mathbf{B} \mid \mathbf{C}) =$ 

#### 5. (10 pts total) BAYESIAN NETWORKS.

5.a. (3 pts) Write down the factored conditional probability expression corresponding to this Bayesian Network:



**5.b.** (3 pts) Draw the Bayesian Network corresponding to this factored conditional probability expression: P(A | B, E, H) P(B | F, G) P(C | D, F, H) P(D | E, F, G) P(E | F, G) P(F | G, H) P(G) P(H)



**5.c.** (4 pts) Shown below is the Bayesian network corresponding to the Burglar Alarm problem, i.e., P(J,M,A,B,E) = P(J | A) P(M | A) P(A | B, E) P(B) P(E). This is Fig. 14.2 in your R&N textbook.



Write down an expression that will evaluate to P( $J=t \land M=f \land A=t \land B=f \land E=t$ ). Express your answer as a series of numbers (numerical probabilities) separated by multiplication symbols. You do not need to carry out the multiplication to produce a single number (probability). SHOW YOUR WORK.

$$P(J=t \land M=f \land A=t \land B=f \land E=t)$$

**6.** (10 pts total, 2 pts each) STATE-SPACE SEARCH. Execute Tree Search through this graph (do not remember visited nodes, so repeated nodes are possible). It is not a tree, but pretend you don't know that. Step costs are given next to each arc, and heuristic values are given next to each node (as h=x). The successors of each node are indicated by the arrows out of that node. (Note: A, D are successors of themselves.) Successor nodes are returned in left-to-right order. (The successor nodes of S are A, B, C; the successor nodes of A are A, B; the successor nodes of B are D, C; and the successor nodes of C are G1, G2. For LIFO and FIFO queues the children will be processed in those node orders, i.e., assume that the child list is concatenated to the front or back of the queue in the order stated above. Priority queues are always sorted by the queue sort function.)

The start node is S and there are two goal nodes, G1 and G2. For each search strategy below, indicate (1) the order in which nodes are expanded, and (2) the path and cost to the goal that was found, if any. Write "None" for the path and cost if the goal was not found. The first one is done for you, as an example.

(s)	h=21
$\begin{array}{c c} 4 \\ \hline 10 \\ \hline \\ A \\ \hline \\ 15 \\ \hline \\ B \\ \hline \\ h=4 \\ \end{array}$	$\begin{array}{c} 50 \\ h=15 \\ \hline 6 \\$
<ul><li>6.a. (example) DEPTH-FIRST SEARCH:</li><li>6.a.i Order of expansion: <u>S A A A A A A A</u></li></ul>	
<ul><li>6.a.ii Path to goal found: <u>None</u></li><li>6.b. (2 pts) BREADTH-FIRST SEARCH:</li></ul>	Cost of path found: None
6.b.i Order of expansion:	
6.b.ii Path to goal found: 6.c. (2 pts) ITERATIVE DEEPENING SEARCH:	Cost of path found:
6.c.i Order of expansion:	
6.c.ii Path to goal found: 6.d. (2 pts) UNIFORM COST SEARCH:	Cost of path found:
6.d.i Order of expansion:	
6.d.ii Path to goal found: 6.e. (2 pts) GREEDY BEST FIRST SEARCH:	Cost of path found:
6.e.i Order of expansion:	
6.e.ii Path to goal found: 6.f. (2 pts) A* SEARCH:	Cost of path found:
6.f.i Order of expansion:	
<b>6.f.ii</b> Path to goal found:	Cost of path found:

7. (12 pts total, 1 pt each) SEARCH QUESTIONS. Label the following as T (= True) or F (= False).

**7.a.** (1 pt) \_\_\_\_\_\_ An admissible heuristic NEVER OVER-ESTIMATES the remaining cost (or distance) to the goal.

**7.b.** (1 pt) \_\_\_\_\_\_ Greedy search is both complete and optimal when the heuristic is optimal.

**7.c.** (1 pt) \_\_\_\_\_\_ A consistent heuristic NEVER VIOLATES the triangle inequality.

**7.d.** (1 pt) \_\_\_\_\_\_ If the search space contains only a single local maximum (i.e., the global maximum is the only local maximum), then hill-climbing is guaranteed to climb that single hill and will find that global maximum.

**7.e.** (1 pt) \_\_\_\_\_\_ A\* search is both complete and optimal for both tree and graph search whenever the step cost is bounded away from zero by a small positive number and the heuristic function is consistent.

7.f. (1 pt) \_\_\_\_\_\_ Hill-climbing has very attractive space properties because it uses only *O*(*bd*) space.

**7.g.** (1 pt) \_\_\_\_\_\_ Simulated annealing will accept more and worse bad moves at a low temperature than at a high temperature.

**7.h.** (1 pt) \_\_\_\_\_\_ Simulated annealing uses *O*(*constant*) space and sometimes can escape from local optima.

7.i. (1 pt) \_\_\_\_\_\_ The simulated annealing temperature increases as the search progresses.

**7.j.** (1 pt) \_\_\_\_\_\_ If the search space is small enough for Mini-Max search to go all the way down to the leaves (e.g., in tic-tac-toe), then it will play a perfect (= optimal) game.

**7.k.** (1 pt) \_\_\_\_\_\_ Uniform-cost search is both complete and optimal when the minimum step cost is bounded away from zero by a small positive constant.

**7.1.** (1 pt) \_\_\_\_\_\_ Mini-Max search assumes that the opponent plays optimally, so the opponent can defeat it by playing sub-optimally.

8. (10 pts total, 1/2 pt each) SEARCH PROPERTIES. Fill in the values of the four evaluation criteria for each search strategy. Assume a tree search where b is the finite branching factor; d is the depth to the shallowest goal node; m is the maximum depth of the search tree; C\* is the cost of the optimal solution; step costs are identical and equal to some positive  $\varepsilon$ ; and in Bidirectional search both directions use breadth-first search. Note that these conditions satisfy all of the footnotes of Fig. 3.21 in your book.

Criterion	Complete?	Time complexity	Space complexity	Optimal?
Breadth-First				
Uniform-Cost				
Depth-First				
Iterative Deepening				
Bidirectional				
(if applicable)				

### 9. (5 pts total, -1 pt for each error, but not negative) MINI-MAX SEARCH IN GAME TREES.

The game tree below illustrates a position reached in the game. Process the tree left-to-right. It is **Max**'s turn to move. At each leaf node is the estimated score returned by the heuristic static evaluator.

### 9.a. Fill in each blank square with the proper mini-max search value.

9.b. What is the best move for Max? (write A, B, or C)



**10.** (10 pts total, -1 pt for each error, but not negative) ALPHA-BETA PRUNING. Process the tree left-toright. This is the same tree as above (1.a). You do not need to indicate the branch node values again.

Cross out each leaf node that will be pruned by Alpha-Beta Pruning.



8 9

**11.** (14 pts total, 1 pt each) AGENT/SEARCH CONCEPTS. For each of the following terms on the left, write in the letter corresponding to the best answer or the correct definition on the right. The first one is done for you as an example.

Α	Agent		Perceives environment by sensors, acts by actuators
	Percept	В	All states reachable from the initial state by a sequence of actions
	Rational Agent	C	Guaranteed to find a solution if one is accessible
	State Space	D	Process of removing detail from a representation
	Search Node	E	Maximum number of successors of any node
	Link between nodes	F	Set of all leaf nodes available for expansion at any given time
	Path	G	Estimates cost of cheapest path from current state to goal state
	Abstraction	Η	Guaranteed to find lowest cost among all accessible solutions
	Optimal Search	Ι	Represents a state in the state space
	Complete Search	J	Sequence of states connected by a sequence of actions
	Expand a state	K	Agent's perceptual inputs at any given instant
	Frontier	L	Agent that acts to maximize its expected performance measure
	Search Strategy	M	Apply each legal action to a state, generating a new set of states
	Branching Factor	N	Represents an action in the state space
	Heuristic Function	0	How a search algorithm chooses which node to expand next

\*\*\*\* THIS IS THE END OF THE MID-TERM EXAM \*\*\*\*