CS-171, Intro to A.I. — Mid-term Exam — Winter Quarter, 2018

I have changed some text to clarify some questions that students found confusing.

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 1-14 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.

Clear your desk except for pen, pencil, eraser, & water bottle. Put backpacks under your seat. 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. (16 pts total) LOCAL SEARCH: 8-QUEENS PROBLEM.

3. (15 pts total, 3 pts each) ADMISSIBLE AND CONSISTENT HEURISTICS.

4. (15 pts total, 3 pts each) STATE-SPACE SEARCH.

5. (15 points total, 3 pts each) CONSTRAINT SATISFACTION PROBLEMS.

6. (10 pts total, 1 pt each) ADVERSARIAL (GAME) SEARCH CONCEPTS.

7. (10 pts total, 1/2 pt each) SEARCH PROPERTIES.

8. (5 pts total, -1 pt for each error, but not negative) MINI-MAX SEARCH IN GAME TREES.

9. (10 pts total, -1 pt for each error, but not negative) ALPHA-BETA PRUNING.

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.

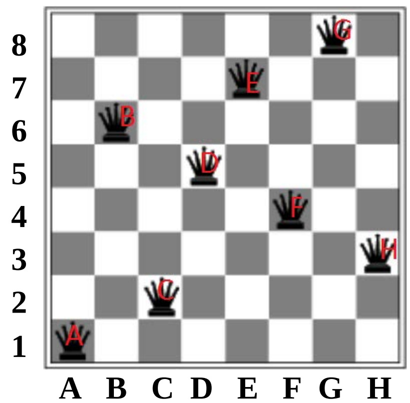
See Section 2.3.1

P<u>erformance (measure)</u>   E<u>nvironment</u>   A<u>ctuators</u>   S<u>ensors</u>

2. (16 pts total) LOCAL SEARCH: 8-QUEENS PROBLEM. You are a robot that is assigned to solve the 8-Queens Problem. Recall that each column contains exactly one queen. 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 to the 8-Queens problem is a board position in which each queen in each column is assigned to a row and no pair of queens attack each other. Your heuristic for the cost of a state is the number of queen pairs in conflict, and so lower values are better.

2.a. (6 pts total, 2 pts each) Please consider this state (= board position):

See Section 4.1.1 and Fig. 4.3



2.a.i. (2 pts) Which queens are in conflict? Write "None" if no queens are in conflict. Else, list all queen pairs in conflict, as *(col1 col2)* where *col*1, *col*2 ∈ {A B C D E F G H} and queens in columns *col*1 and *col*2 conflict.

____________(D G)____________________

2.a.ii. (2 pts) Based on your answer to 2.a.i above, is the state above a solution? Write Y (= yes) or N (= no).

Your answer will be considered correct if it is consistent with your answer to 2.a.i. ____________N_____

2.a.iii. (2 pts) Simulate Hill-Climbing on the state above to select a next state that reduces conflicts in one action (i.e., do local search only to a depth of one). Your available actions are to select one column and move the queen in that column to a different row. What actions might Hill-Climbing take to reduce cost (= number of queen pairs in conflict) in one move?

Write "None" if Hill-Climbing cannot improve the state above in one action. Otherwise, list all Hill-Climbing actions that would reduce conflict in one action, as *(col row)* where *col* ∈ {A B C D E F G H} and 0 < *row* < 9.

____________None____________________

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

2.b. (6 pts total, 2 pts each) Please now consider this state (= board position):

2.b.i. (2 pts) Which queens are in conflict? Write "None" if no queens are in conflict. Else, list all queen pairs in conflict, as *(col1 col2)* where *col*1, *col*2 ∈ {A B C D E F G H} and queens in columns *col*1 and *col*2 conflict.

_____________(F H), (G H)_____________________

2.b.ii. (2 pts) Based on your answer to 2.b.i above, is the state above a solution? Write Y (= yes) or N (= no).

Your answer will be considered correct if it is consistent with your answer to 2.b.i. ____________N_____

2.b.iii. (2 pts) Simulate Hill-Climbing on the state above to select a next state that reduces conflicts in one action (i.e., do local search only to a depth of one). Your available actions are to select one column and move the queen in that column to a different row. What actions might Hill-Climbing take to reduce cost (= number of queen pairs in conflict) in one move?

Write "None" if Hill-Climbing cannot improve the state above in one action. Otherwise, list all Hill-Climbing actions that would reduce conflict in one action, as *(col row)* where *col* ∈ {A B C D E F G H} and 0 < *row* < 9.

_____________(H 1), (H 5), (H 8)_________________

2.c. (4 pts total, 1 pt each) Which statements are true for Local Search in general? T (= true) and F (= false).

2.c.i. (1 pt) __F______ Local Search guarantees to find a global optima, i.e., it is an optimal search method.

2.c.ii. (1 pt) __T______ Local Search encounters problems of local minima, plateaus, and ridges, among others.

2.c.iii. (1 pt) __F______ Hill-Climbing can escape from local minima but Simulated Annealing cannot.

2.c.iv. (1 pt) __T______ Local Search keeps one or a few states, and so has bounded controllable memory needs.

3. (15 pts total, 3 pts each) ADMISSIBLE AND CONSISTENT HEURISTICS.

3.a. (9 pts total, 3 pts each) Recall that a heuristic function h(n) is <u>consistent</u> if, for every node n and each n′ that is a child node of n,

See Section 3.5.2, especially pp. 94+

$$h(n) \leq c(n, n') + h(n'),$$

where c(n, n′) is the step cost to go from n to n′. The proof below shows that if $h_1(n)$ and $h_2(n)$ are both consistent, then $h_{max}(n) = \max(h_1(n), h_2(n))$ also is consistent.

The lines of the proof have been labelled A through E. Unfortunately, the lines have been scrambled.

A: $h_{max}(n)$

D: $= \max(h1(n), h2(n))$ // By definition of $h_{max}(n)$

C: $\leq \max(c(n,n') + h_1(n')), c(n,n') + h_2(n'))$ // By definition of consistent

B: $= c(n,n') + \max(h_1(n'), h_2(n'))$ // By definition of max

E: $= c(n,n') + h_{max}(n')$ // By definition of $h_{max}(n')$

Fill in the blanks with the letters B, C, and D in the correct order to prove that $h_{max}$is consistent. The first and last letters, A and E, have been done for you as an example.

____A____ ____D____ ____C____ ____B____ ____E____

3.b. (6 pts total, 3 pts each) Recall that a heuristic function h(n) is <u>admissible</u> if, for every node n,

$$h(n) \leq h^*(n),$$

where h*(n) = the true optimal cost to reach the goal state from n. The proof below shows that if $h_1(n)$ and $h_2(n)$ are both admissible, then $h_{max}(n) = \max(h_1(n), h_2(n))$ also is admissible.

The lines of the proof have been labelled A through D. Unfortunately, the lines have been scrambled.

A: $h_{max}(n)$

C: $= \max(h1(n), h2(n))$ // By definition of $h_{max}(n)$

B: $\leq \max(h^*(n), h^*(n))$ // Because h1 and h2 are both admissible

D: $= h^*(n)$ // By definition of max

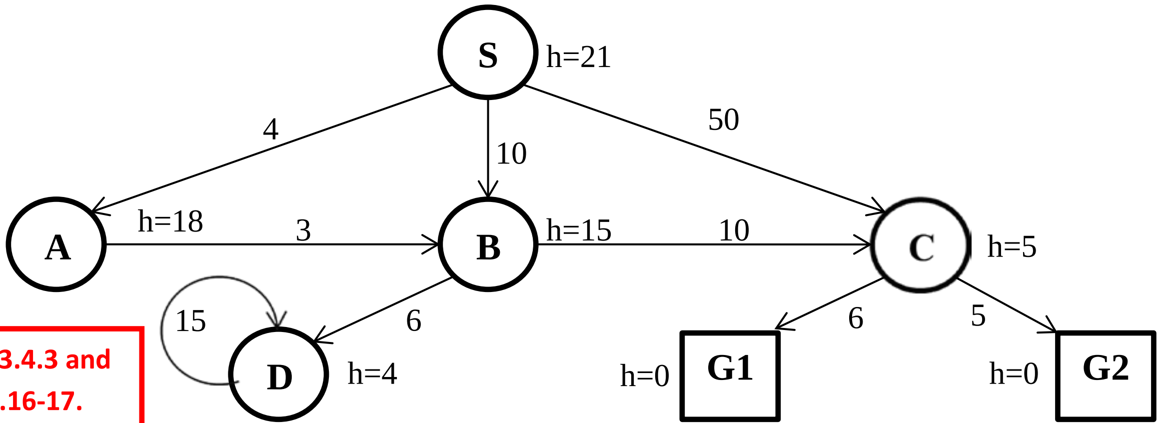
Fill in the blanks with the letters B and C in the correct order to prove that $h_{max}$is consistent. The first and last letters, A and D, have been done for you as an example.

____A____ ____C____ ____B____ ____D____

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

4. (15 pts total, 3 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: D is a successor of itself).** As usual, successor nodes are returned in left-to-right order. (The successor nodes of S are A, B, C; the successor nodes of B are D, C; and the successor nodes of C are G1, G2. For LIFO and FIFO queues, 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.



See Section 3.4.3 and Figs. 3.7 & 3.16-17.

4.a. (example) DEPTH-FIRST SEARCH:

4.a.i Order of expansion: S A B D D D D ...

4.a.ii Path to goal found: None   Cost of path found: None

4.b. (3 pts) BREADTH-FIRST SEARCH:

See Section 3.4.1 and Fig. 3.11.

on: S A B C (G1)

4.b.ii Path to goal found: S C G1   Cost of path found: 56

4.c. (3 pts) ITERATIVE DEEPENING SEARCH:

See Sections 3.4.4-5 and Figs. 3.18-19.

S S A B C (G1)

4.c.ii Path to goal found: S C G1   Cost of path found: 56

4.d. (3 pts) UNIFORM COST SEARCH:

See Section 3.4.2 and Fig. 3.14.

on: S A B D C (G2) or S A B B D D C C (G2)

nd: S A B C G2   Cost of path found: 22

4.e. (3 pts) GREEDY BEST FIRST SEARCH:

See Section 3.5.1 and Fig. 3.23.

on: S C (G1)

d: S C (G1)   Cost of path found: 56

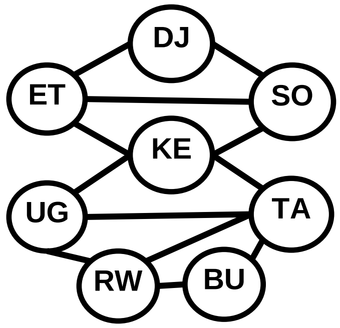
4.f. (3 pts) A* SEARCH:

See Section 3.5.2 and Figs. 3.24-25.

n: S A B D C (G2)

d: S A B C G2   Cost of path found: 22

5. (15 points total, 3 pts each) CONSTRAINT SATISFACTION PROBLEMS.



BU = Burundi
DJ = Djibouti
ET = Ethiopia
KE = Kenya
RW = Rwanda
SO = Somalia
TA = Tanzania
UG = Uganda

See Chapter 6.

You are a map-coloring robot assigned to color this East Africa map. Adjacent regions must be colored a different color (R=Red, B=Blue, G=Green). The constraint graph is shown. See Section 6.3.2.

5.a. (3 pts total) FORWARD CHECKING. Variable KE just now has been assigned value G, as shown. Cross out all values that would be eliminated by Forward Checking.

BU	DJ	ET	KE	RW	SO	TA	UG
R G B	R G B	R ~~G~~ B	G	R G B	R ~~G~~ B	R ~~G~~ B	R ~~G~~ B

See Section 6.2.2.

5.b. (3 pts total) ARC CONSISTENCY.
Variables KE and UG have been assigned values, as shown, but no constraint propagation has been done. Cross out all values that would be eliminated by Arc Consistency (AC-3 in your book).

BU	DJ	ET	KE	RW	SO	TA	UG
R ~~G~~ ~~B~~	R G B	R ~~G~~ B	G	~~R~~ G ~~B~~	R ~~G~~ B	~~R~~ ~~G~~ B	R

5.c. (3 pts total) MINIMUM-REMAINING-VALUES HEURISTIC. Cor See Section 6.3.1. ent below. TA is assigned and constraint propagation has been done. List all unassigned variables that might be selected by the Minimum-Remaining-Values (MRV) Heuristic: **BU, KE, RW, UG** .

BU	DJ	ET	KE	RW	SO	TA	UG
R B	R G B	R G B	R B	R B	R G B	G	R B

5.d. (3 pts total) DEGREE HEURISTIC. Consider the See Section 6.3.1. . (It is the same assignment as in problem 5.c above.) TA is assigned and constraint propagation has been done. Ignore MRV. List all unassigned variables that might be selected by the Degree Heuristic: **ET, KE, SO**

BU	DJ	ET	KE	RW	SO	TA	UG
R B	R G B	R G B	R B	R B	R G B	G	R B

5.e. (3 pts total) MIN-CONFLICTS HEURISTIC IN LOCAL SEARC See Section 6.4. complete but inconsistent assignment below. UG has just been selected to be assigned a new value during local search for a complete and consistent assignment. What new value would be chosen below for UG by the Min-Conflicts Heuristic? Write R, G, or B. **R** .

BU	DJ	ET	KE	RW	SO	TA	UG
B	G	G	G	G	B	B	**?**

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

6. (10 pts total, 1 pt each) ADVERSARIAL (GAME) 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.

See Chapter 5.

D	Game Strategy	A	Approximates the value of a game state (i.e., of a game position)
H	Cut-off Test	B	In all game instances, total pay-off summed over all players is a constant
E	Alpha-Beta Pruning	C	Tree where nodes are game states and edges are game moves
G	Weighted Linear Function	D	Function that specifies a player's move in every possible game state
J	Terminal Test	E	Returns same move as MiniMax, but may prune more branches
I	ExpectiMiniMax	F	Optimal strategy for 2-player zero-sum games of perfect information, but impractical given limited time to make each move
C	Game Tree	G	Vector dot product of a weight vector and a state feature vector
A	Heuristic Evaluation Function	H	Function that decides when to stop exploring this search branch
B	Zero-sum Game	I	Generalizes MiniMax to apply to games with chance (stochastic games)
F	MiniMax Algorithm	J	Function that says when the game is over

7. (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 ε; and in Bidirectional search both directions use breadth-first search.

See Figure 3.21

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	Yes	O(b^d)	O(b^d)	Yes
Uniform-Cost	Yes	O(b^(1+floor(C*/ε))) O(b^(d+1)) also OK	O(b^(1+floor(C*/ε))) O(b^(d+1)) also OK	Yes
Depth-First	No	O(b^m)	O(bm)	No
Iterative Deepening	Yes	O(b^d)	O(bd)	Yes
Bidirectional (if applicable)	Yes	O(b^(d/2))	O(b^(d/2))	Yes

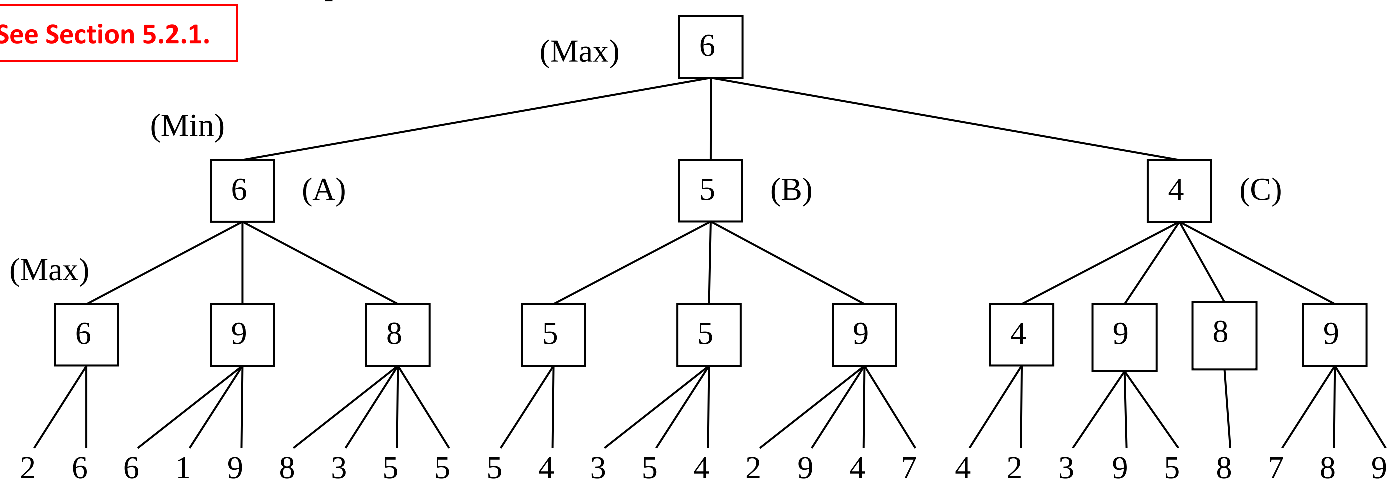
8. (5 pts total, -1 pt for each error, but not negative) MINI-MAX SEARCH IN GAME TREES. The game tree below illustrates a position reached in the game. Process the tree left-to-right. It is **Max**'s turn to move. At each leaf node is the estimated score returned by the heuristic static evaluator.

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

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

8.c. What score does Max expect to achieve? ____6____

See Section 5.2.1.

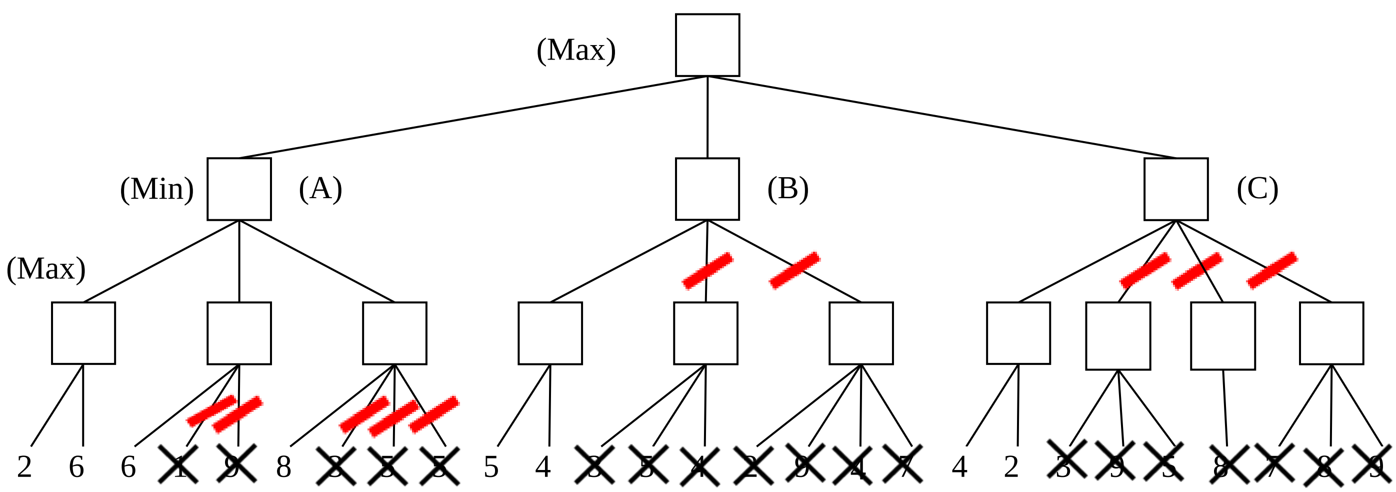


9. (10 pts total, -1 pt for each error, but not negative) ALPHA-BETA PRUNING. Process the tree left-to-right. This is the same tree as above (1.a). You do not need to indicate the branch node values again.

<u>**Cross out each leaf node that will be pruned by Alpha-Beta Pruning.**</u>

See Section 5.3.

Please explicitly cross out leaf nodes. Please do not simply prune branches.



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