CS-171, Intro to A.I. — Mid-term Exam — Fall Quarter, 2017

YOUR NAME: ____

YOUR ID: __________ ID TO RIGHT: __________ ROW: ________ SEAT: ________

<u>Please turn off all cell phones now.</u>

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

When you are told to begin the exam, please check first to make sure that you have all eight pages, as numbered 1-8 in the bottom-right corner of each page. We wish to avoid copy problems. We will supply a new exam for any copy problems.

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

Please clear your desk entirely, except for pen, pencil, eraser, a blank piece of paper (for scratch pad use), and an optional water bottle. Please write your name and ID# on the blank piece of paper and <u>turn it in with your exam</u>.

You may turn in your Midterm exam and leave class when you are finished. Show your UCI ID to the CS-171 Teaching Staff for verification, and deposit your exam in the box at the front. <u>After you first stand up from your seat, your exam is over and must be turned in immediately.</u>

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

1. (10 pts total, 1 pt each) HILL-CLIMBING LOCAL SEARCH.

2. (16 pts total, 1 pt each) HILL-CLIMBING LOCAL SEARCH.

3. (16 pts total) CONSTRAINT SATISFACTION PROBLEMS (CSPs)

4. (16 pts total) A* HEURISTIC SEARCH

5. (6 pts total, 2 pts each) CONVERSION TO CNF

6. (10 pts total, -1 pt each wrong answer, but not negative) MINIMAX WITH ALPHA-BETA PRUNING

7. (8 pts total, 2 pts each) RESOLUTION OF CLAUSES

8. (8 pts total, 2 pts each) TASK ENVIRONMENT

9. (10 pts total, 1/2 pt each, fractional scores rounded up in your favor) SEARCH PROPERTIES

<u>The Exam is printed on both sides to save trees! Work both sides of each page!</u>

1. (10 pts total, 1 pt each) HILL-CLIMBING LOCAL SEARCH. You are a robot that is playing 8-Puzzle. The only actions are to move the blank cell UP (U), DOWN (D), RIGHT (R), or LEFT (L). The size of the game board is 3x3. The Start and Goal states are shown below. The heuristic value of a state is its number of misplaced tiles from the goal state (**h1**), which is the sum over each tile of {if it is in its goal position 0, else 1}. For instance, tile number 1 in the Start state is not in its goal position so counts 1, while tile number 2 is in its goal position so counts 0. The heuristic value of the Start State is: h1(Start) = 1 + 0 + 1 + 1 + 0 + 1 + 0 + 1 = 5.

1.a (6 pts total, 1 pt each) Fill in heuristic h1(n) values below (n = current node). The first is done for you.

[Start State]

	2	8
1	5	4
7	3	6

(state a) h1(a) = __**5**__.

[Goal State]

1	2	3
4	5	6
7	8	

Action = R

2		8
1	5	4
7	3	6

(state b) h1(b) = __6__.

Action = D

1	2	8
	5	4
7	3	6

(state c) h1(c) = __4__.

Action = R

2	8	
1	5	4
7	3	6

(state d) h1(d) = __6__.

Action = D

2	5	8
1		4
7	3	6

(state e) h1(e) = __7__.

Action = R

1	2	8
5		4
7	3	6

(state f) h1(f) = __5__.

Action = D

1	2	8
7	5	4
	3	6

(state g) h1(g) = __5__.

1.b (1 pt) You are doing Hill-Climbing local search. In which state does the search terminate? ____C_____

1.c (3 pts total, 1 pt each) Answer True (T) or False (F) to the following questions:

_F_ Hill climbing looks ahead beyond the immediate neighbors of the current state.

_F_ Hill-climbing search always guarantees to find a globally optimal solution within finite time.

_T_ Given a finite space and infinite time, Hill-climbing with random restart will find a globally optimal solution with probability 1.0.

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

2. (16 pts total, 1 pt eac

The only actions are to

game board is 3x3. **NOT**

Note: In order to break the dependency of your answers to following sub-problems that depend upon your answers to previous sub-problems, your answer to following sub-problems will be graded based only upon your answer to the previous sub-problem.

[Start State]

1	8	4
7	3	2
5	6	

[Goal State]

1	2	3
4	5	6
7	8	

The heuristic value of each state is its Manhattan distance (**h2**) from the goal state, which is the sum of the Manhattan distances of each tile from its goal position. For instance, tile number 2 in the Start state requires two moves (L, U) in order to get to its goal position in the Goal state. The heuristic value of the Start State is.

$$h2(Start) = 0 + 2 + 2 + 3 + 2 + 2 + 1 + 2 = 14$$

For each sub-problem below, write the heuristic value of the state that results from actions U, D, R, L in the previous state (begin with the Start State above; if an action is not possible, write NONE). Then, write the action that Hill-Climbing would choose, and draw the resulting state as a new 8-Puzzle configuration.

2.a (4 pts total, 1 pt each; two values are done for you as examples)

h2(result(Start, U)) = **15** . h2(result(Start, D)) = **None** h2(result(Start, R)) = None h2(result(Start, L)) = 13 .

Chosen action (write one of U, D, R, or L) = **A1** = L . result(Start, A1) = **S1** =

1	8	4
7	3	2
5		6

2.b (6 pts total, 1 pt each)

h2(result(**S1**, U)) = 14 . h2(result(**S1**, D)) = None h2(result(**S1**, R)) = 14 . h2(result(**S1**, L)) = 12 .

Chosen action (write one of U, D, R, or L) = **A2** = L . result(S1, A2) = **S2** =

1	8	4
7	3	2
	5	6

2.c (6 pts total, 1 pt each)

h2(result(**S2**, U)) = 11 . h2(result(**S2**, D)) = None h2(result(**S2**, R)) = 13 . h2(result(**S2**, L)) = None

Chosen action (write one of U, D, R, or L) = **A3** = U . result(S2, A3) = **S3** =

1	8	4
	3	2
7	5	6

3. (16 pts total) CONSTRAINT SATISFACTION PROBLEMS (CSPs). DBSICS recently built a new building with 6 faculty offices. You are a robot that is in charge of room assignments. You choose to use a CSP. The CSP variables are six professors: Rick Lathrop (**R**), Michael Goodrich (**M**), David Eppstein (**D**), Alexander Ihler (**A**), Wayne Hayes (**W**), and Chen Li (**C**). The domains are rooms {1, 2, 3, 4, 5, 6}. The constraints are:

(a) No two professors can stay in the same room

(b) R > 3	(f) D is even
(c) A is less than R	(g) W is not 1 or 6
(d) M is either 5 or 6	(h) |W-C| = 1
(e) R > M	(i) |R-D| = 2

3.a (5 pts total, -1 each wrong answer, but not negative) Unary Constraints. Mark X below by each constraint that is a Unary Constraint.

[ ] a [X] b [ ] c [X] d [ ] e [X] f [X] g [ ] h [ ] i

3.b (3 pts total, -1 each wrong answer, but not negative) Minimum Remaining Value (MRV) heuristic). After initially enforcing the constraints above, the domains of the variables are:

Domain(R) = {4, 5, 6}	Domain(A) = {1, 2, 3, 4, 5, 6}
Domain(M) = {5, 6}	Domain(W) = {2, 3, 4, 5}
Domain(D) = {2, 4, 6}	Domain(C) = {1, 2, 3, 4, 5, 6}

Mark X below by each variable that might be selected by the MRV heuristic to be assigned next.

[ ] R [X] M [ ] D [ ] A [ ] W [ ] C

3.c (5 pts total, -1 each wrong answer, but not negative) Forward Checking (FC). For the purposes of decoupling this problem from your solution to the previous problem, assume we arbitrarily choose to assign R first. <u>We assign R the value R = 6.</u> **Cross out the values from each domain that will be eliminated by enforcing Forward Checking of R = 6.**

Domain(R) = { **<u>6</u>** }	Domain(A) = { 1, 2, 3, 4, 5, ~~6~~ }
Domain(M) = { 5, ~~6~~ }	Domain(W) = { 2, 3, 4, 5 }
Domain(D) = { ~~2~~, 4, ~~6~~ }	Domain(C) = { 1, 2, 3, 4, 5, ~~6~~ }

3.d (3 pts total) Least Constraining Value (LCV). Assume that we have assigned R = 4 and M = 5, the remaining unassigned variables are D, A, W, and C. After enforcing Forward Checking their domains are:

Do

Problem 3.d was cancelled due to an error (a constraint said R>M but the problem assigned R = 4 and M = 5, which violates the constraint).

<u>Everyone scores problem 3.d correct, automatically.</u>

Later, we will come back to fix problem 3.d, for use as a study guide. For now, it is simply cancelled.

Domain(C) = {1, 2, 3, 6}

(2 pts) Va **igned next by LCV:**

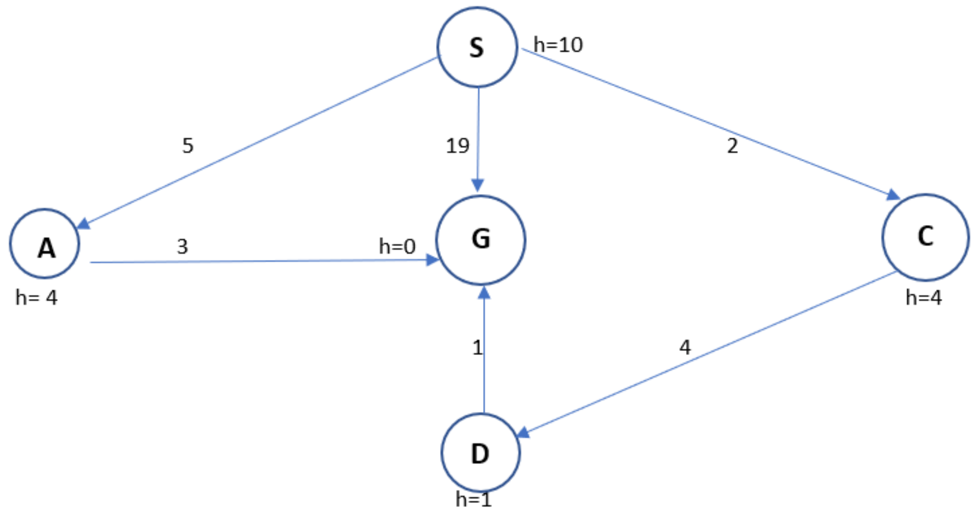
<u>6 .</u>

(2 pts) Ple

If D=2, we eliminate 2 from A, 2 from W, 2 from C, for a total of 3 values removed from unassigned variables.
If D=6, we eliminate 6 from A, 6 from C, for a total of 2 values are removed.
So according to the Least Constraining Value heuristic, we assign D=6.

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

4. (16 pts total) A* HEURISTIC SEARCH. Execute A* Tree Search through this graph (i.e., do not remember visited nodes). S is the start node and G is the goal node. Step costs are given next to each arc. 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. Successors are returned in left-to-right order, i.e., successors of S are (A, G, C).



4.a (12 pts total) Show the order in which nodes are expanded in A* search (to expand a node means its children are generated), ending with the goal node found, or indicate the repeating cycle if the search gets stuck in a loop. Show the path from start to goal, or write "None." Give the cost of the path found, or write "None."

(8 pts) Order of node expansion:__________ S C D (G) __________

(3 pts) Path found:___ S C D (G) __________ **(1 pt)** Cost of path found: __________ 7 _____

4.b (4 pts total) The A* search above actually has reached the goal node twice, once first with a bad score, and later with an optimal score. Please briefly explain why A* search does not always terminate when the first goal node is found, but instead delays until the optimal goal node is found and returned?

Even though a sub-optimal path to a goal node is reached early, there might be an optimal (= better) path to the goal whose cost is less. This possibility exists as long as there is any node whose heuristic value is less than that of the already-found but sub-optimal goal node, which will cause that node to sort in front of the sub-optimal goal node on the priority queue.

5. (6 pts total, 2 pts each) CONVERSION TO CNF. Convert these expressions to CNF.

5.a. (2 pts) $(A \Leftrightarrow (B \lor C))$ ______ $((A \neg B)(A \neg C)(\neg A\ B\ C))$ ______

See section 7.5.

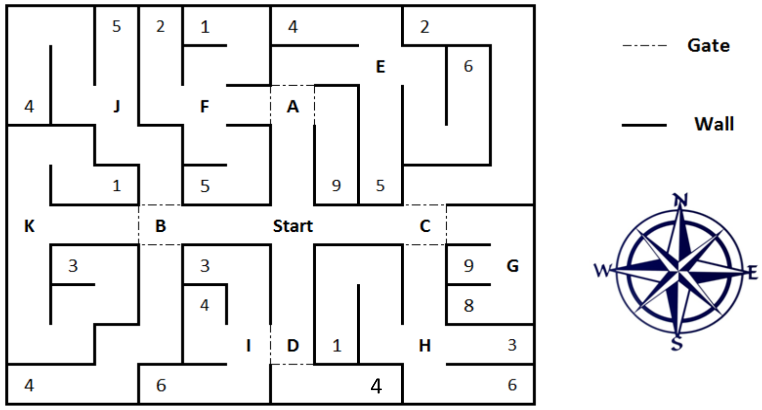
5.b. (2 pts) $((C \land D) \Rightarrow \neg E)$ ______ $(\neg C \neg D \neg E)$ ______

5.c. (2 pts) $((A \Rightarrow B) \Rightarrow C)$ ______ $((A\ C)(\neg B\ C))$ ______

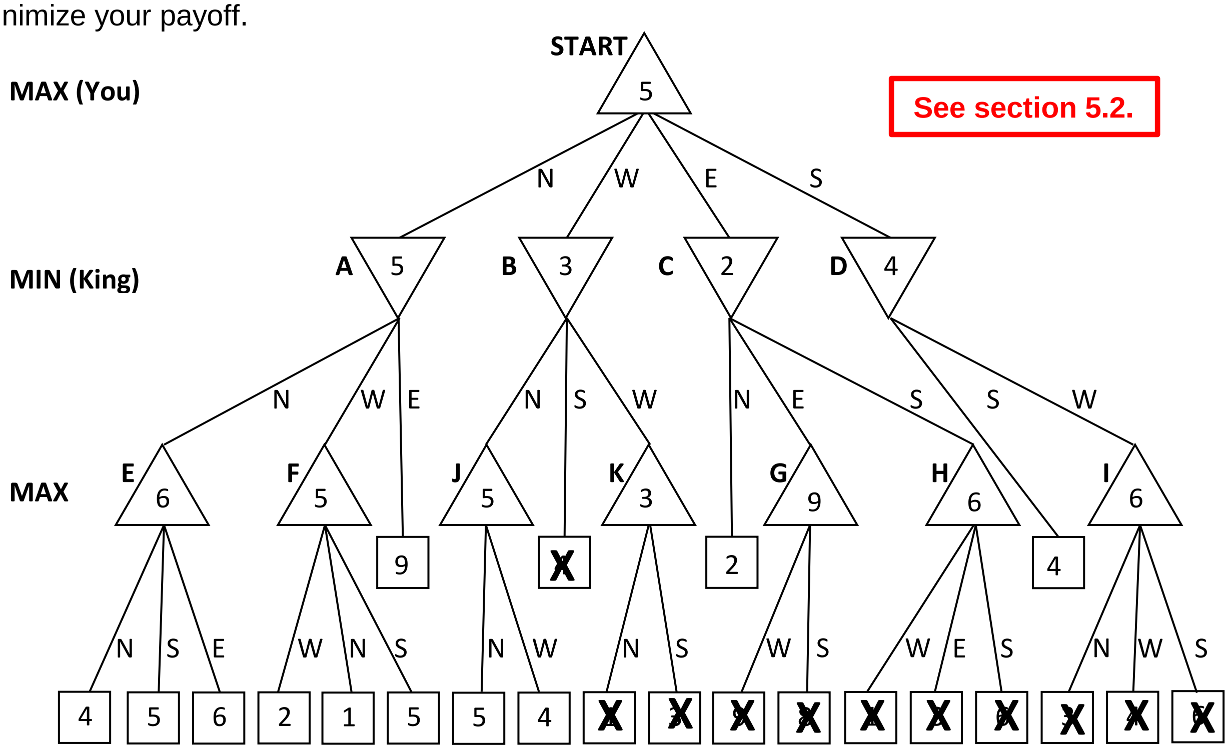
3.a is sentence S1, and 3.b is sentence S3, of problem 7.20, p. 283, in your textbook, after variable relabeling.

It is OK if you put in "∨" and "∧" above. For example, it is OK if you answered 3.a. as $((A \lor \neg B) \land (A \lor \neg C) \land (\neg A \lor B \lor C))$, and so on.

6. (10 pts total, -1 pt each wrong answer, but not negative) MINIMAX WITH ALPHA-BETA PRUNING. While visiting Crete, you are challenged by a passing king to what he calls the “Labyrinth Challenge”. The rules are simple: you must make your way through a maze to find the largest prize for yourself. You are given the following map to plan your route:



You will start in the maze at the location labeled START and may travel North (N), South (S), East (E), or West (W). Your goal is to secure the largest, single prize for yourself, represented by the numbers spread across the maze. At four specific intersections (A, B, C, D), the king will be able to close off all but one pathway by closing gates around you, forcing you to take the path he gives you. Backtracking is not allowed. The king acts to minimize your payoff.



6.a. Fill in each blank triangle with its Mini-Max value. Process the game tree left-to-right.

6.b. Cross out each leaf node that will be pruned by Alpha-Beta pruning. Go left-to-right.

6.c. What is the best move for MAX? (write N, W, E, or S) ____N______ See section 5.3.

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

7. (8 pts total, 2 pts each) RESOLUTION OF CLAUSES. Use resolution to resolve the following pairs of clauses, simplify, and write the resulting clause in simplified form. If no resolution is possible write “None”. If the resolvent simplifies to True write “True.” Remember that (A B C) is shorthand for (A or B or C).

7.a. (2 pts) Resolve (A B C) with (¬B) to yield ___(A C)___________ **See section 7.5.**

7.b. (2 pts) Resolve (A B C) with (¬B ¬C ¬D) to yield __True______________________

7.c. (2 pts) Resolve (A B C) with (B ¬C ¬D) to yield ____(A B ¬D)________________

7.d. (2 pts) Resolve (A B C) with (B C ¬D) to yield _____None______________________

8. (8 pts total, 2 pts each) TASK ENVIRONMENT. Your book defines a task environment as a set of four things, with the acronym PEAS. Fill in the blanks w **See Section 2.3.1.** he PEAS components.

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

9. (10 pts total, 1/2 pt each, fractional scores rounded up in your favor) SEARCH PROPERTIES. Fill in the values of the four evaluation criteria for each search strategy shown. 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 using breadth-first search.

Note that these conditions satisfy all of the footnotes of Fig. 3.21 in yo **See Figure 3.21.**

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

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