CS 171, Intro to A.I. — Midterm Exam — Fall Quarter, 2016

YOUR NAME: ____

YOUR ID: __________ ROW: ________ SEAT: ________

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

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 turn off all cell phones now.

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 turn it in with your exam.

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

1. (12 pts total) TRUE / FALSE

2. (8 pts total) SEARCH PROPERTIES.

3. (4 pts total, 1 pt each) TASK ENVIRONMENT.

4. (20 pts total, 5 pts each) STATE-SPACE SEARCH STRATEGIES.

5. (9 pts total) DOMINATING HEURISTICS.

6. (10 pts total) MINIMAX.

7. (8 pts total) ALPHA-BETA PRUNING.

8. (21 pts total) CONSTRAINT SATISFACTION PROBLEMS.

9. (8 pts total) CONSTRAINT SATISFACTION (CSP) CONCEPTS.

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

1. (12 pts total, 1 pt each) TRUE/FALSE. Mark the following statements True (T) or False (F).

	Uniform-cost search will never expand more nodes than A*-search.
	Depth-first search will always expand more nodes than breadth-first search.
	Let $h_1(n)$ and $h_2(n)$ both be admissible heuristics. Then, $\min(h_1, h_2)$ is necessarily admissible.
	Let $h_1(n)$ be an admissible heuristic, and let $h_2(n)$ be an inadmissible heuristic. Then $(h_1 + h_2)/2$ is necessarily admissible.
	Let $h_1(n)$ be an admissible heuristic, and $h_2(n) = 2*h_1(n)$. The solution found by A* tree search with $h_2(n)$ is guaranteed to have a cost at most twice as much as the optimal path.
	RBFS will possibly re-expand some node that it has visited before, but SMA* will not.
	The most-constrained variable heuristic provides a way to select the next variable to assign in a backtracking search for solving a CSP.
	The purpose of the least-constraining value heuristic is to reduce the branching factor of the backtracking search.
	By using the most-constrained variable heuristic and the least-constraining value heuristic we can solve every CSP in time linear in the number of variables.
	When enforcing arc consistency in a CSP, the set of values that remain when the algorithm terminates does not depend on the order in which arcs are processed from the queue.
	When using alpha-beta pruning, the computational savings are independent of the order in which children are expanded.
	When using expectimax to compute a policy, re-scaling the values of all the leaf nodes by multiplying them all by 10 can result in a different policy being optimal.

2. (8 pts total, -1 pt each wrong answer, but not negative) 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 greater than some positive ε; 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)				

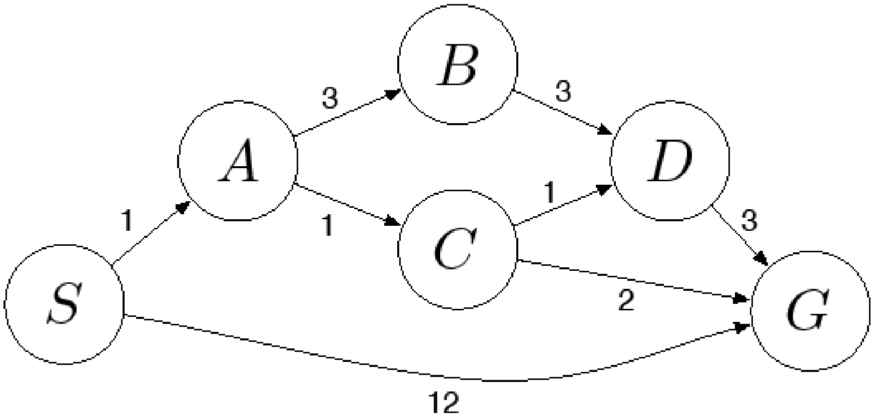
3. (4 pts total, 1 pt each) TASK ENVIRONMENT. Your book defines a task environment as a set of four things, with the acronym PEAS. Fill in the blanks with the names of the PEAS components.

P________________ E________________ A________________ S________________

4. (20 pts total, 5 pts each) STATE-SPACE SEARCH STRATEGIES. Execute Tree Search through this graph (i.e., do not remember visited nodes). Step costs are given next to each arc. Heuristic values are given in the table on the right. 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), successors of A are (B, C), and successors of C are (D, G), in that order.

For each search strategy below, show the order in which nodes are expanded (i.e., to expand a node means that its children are generated), ending with the goal node that is found. Show the path from start to goal, and give the cost of the path that is found.

The first one is done for you as an example.



State	$h(n)$
S	4
A	2
B	6
C	1
D	3
G	0

4.a. DEPTH FIRST SEARCH.

Order of node expansion: S A B D G

Path found: S A B D G Cost of path found: 10

4.b. (5 pts) UNIFORM COST SEARCH.

(2 pts) Order of node expansion:

(2 pts) Path found: **(1 pt)** Cost of path found:

4.c. (5 pts) GREEDY (BEST-FIRST) SEARCH.

(2 pts) Order of node expansion:

(2 pts) Path found: **(1 pt)** Cost of path found:

4d. (5 pts) ITERATED DEEPENING SEARCH.

(2 pts) Order of node expansion:

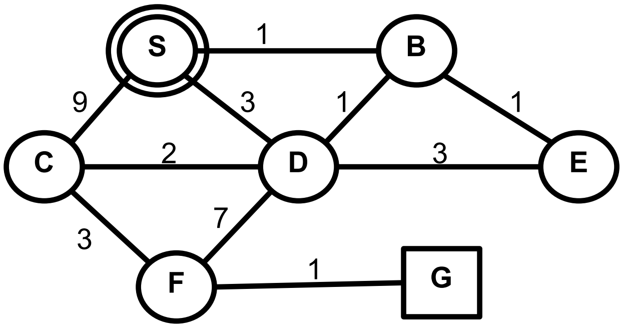
(2 pts) Path found: **(1 pt)** Cost of path found:

4.e. (5 pts) A* SEARCH WITH $h(n)$.

(2 pts) Order of node expansion:

(2 pts) Path found: **(1 pt)** Cost of path found:

5. (9 pts total) DOMINATING HEURISTICS. In this question, you are asked to compare different heuristics and to determine which, if any, dominate each other. You are executing Tree Search through this graph (i.e., you do not remember previously visited nodes). The start node (= initial state) is S, and the goal node is G. Actual step costs are shown next to each link. Heuristics are given in the following table. As is usual in your book, h* is the true (= optimal) heuristic; here, h_i are various other heuristics.



Node	h1	h2	h3	h* (optimal)
S-Start	5	5	5	8
B	4	4	5	7
C	5	3	2	4
D	3	1	4	6
E	4	3	4	8
F	1	0	1	1
G-Goal	0	0	0	0

5.a. (2 pts)
Which heuristic functions are admissible among h1, h2 and h3? _________________

5.b. (2 pt)
Which heuristic functions are consistent among h1, h2 and h3? _________________

5.c. (5 pts, -1 pt for each error but not negative)
Which of the following statements are true? (write T=True, F=False)

(a) h1 dominates h2. (T or F) ____________

(b) h1 dominates h3. (T or F) ____________

(c) h2 dominates h1. (T or F) ____________

(d) h2 dominates h3. (T or F) ____________

(e) h3 dominates h1. (T or F) ____________

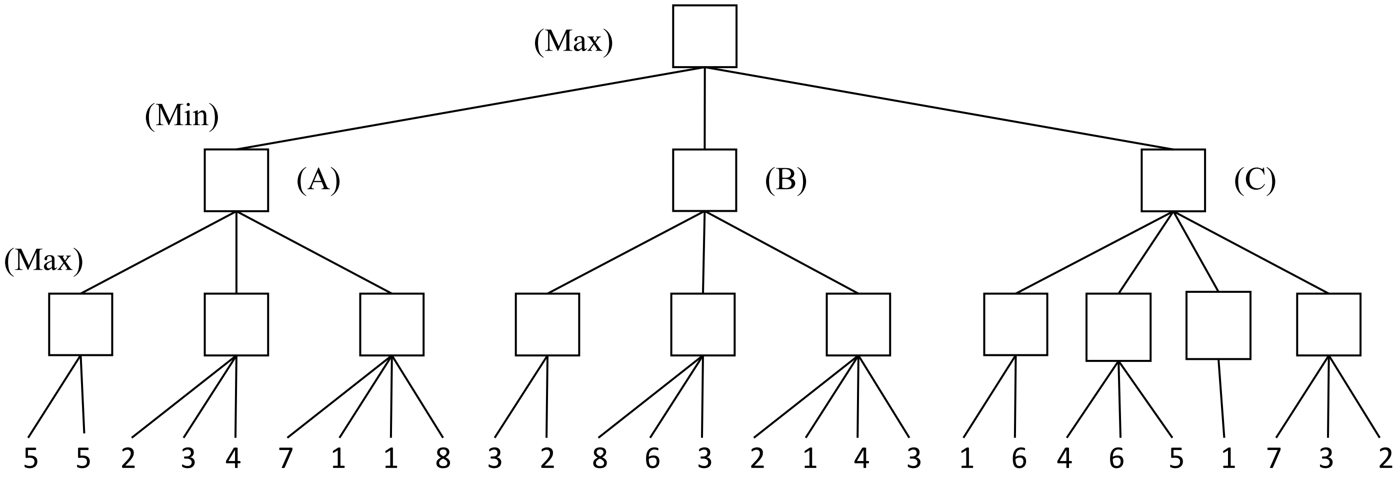
(f) h3 dominates h2. (T or F) ____________

6. (8 pts total, -1 pts 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.

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

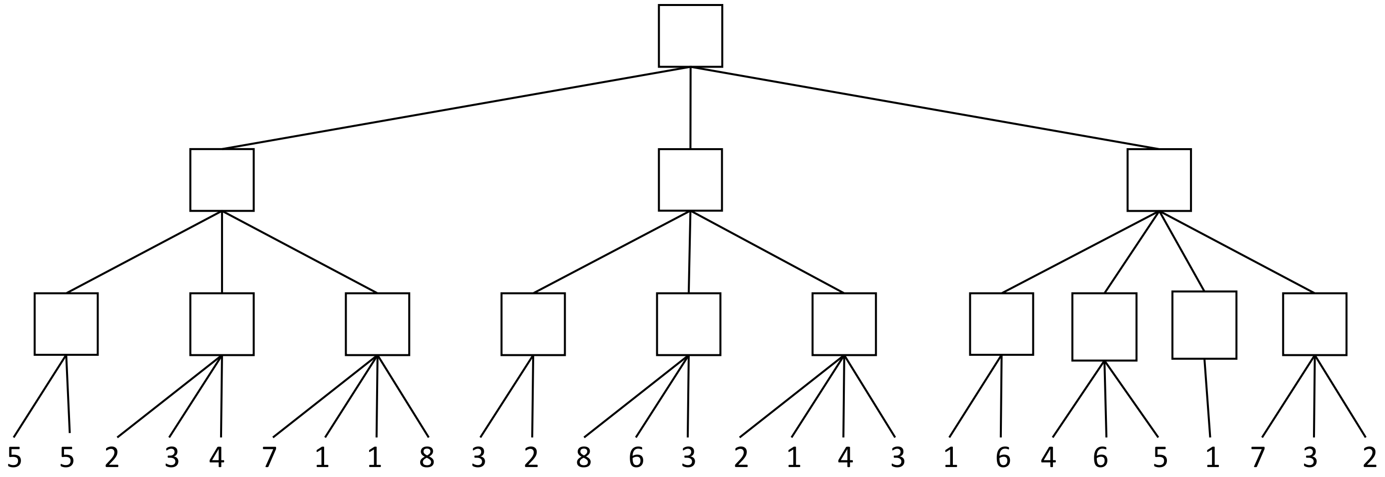
6.b. What is the best move for Max? (write A, B, or C) ______________

6.c. What score does Max expect to achieve? ________________

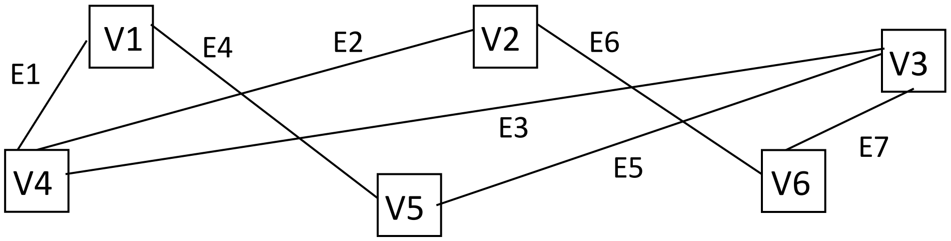


7. (10 pts total, -1 for each error, but not negative) ALPHA-BETA PRUNING. Process the tree left-to-right. This is the same tree as above (1.a). You do not need to indicate the branch node values again.

Cross out each leaf node that will be pruned by Alpha-Beta Pruning. Do not just draw pruning lines.

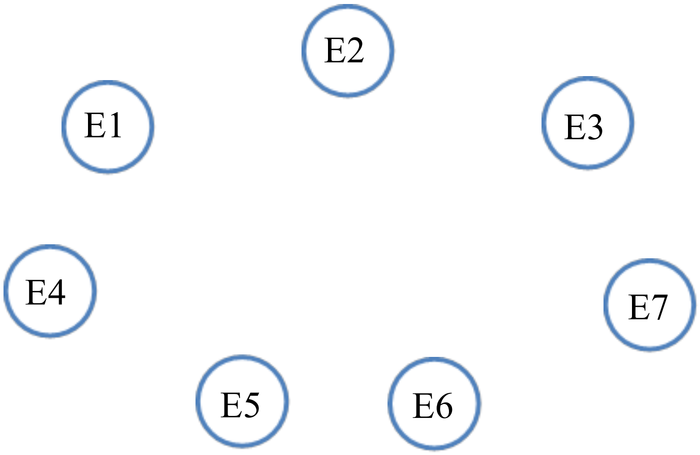


8. (21 pts total) CONSTRAINT SATISFACTION. Consider the following graph with 6 square-shaped vertices and 7 undirected edges. In this problem, you can color each edge using one color from the following set of 3 colors, { **Red**, **Green**, **Blue** }, and you are asked to solve this edge-coloring problem as constraint satisfaction problem.



The edge-coloring of a graph is an assignment of colors to the edges of the graph so that no two adjacent edges have the same color. Let's call this constraint the "edge-coloring" constraint. For example, e1 and e2 cannot have the same color because both are adjacent at the vertex v4. On the other hand, the graph doesn't restrict you to use the same color on e2 and e4 because they are not adjacent at any vertex.

8.a (3 pts) Constraint Graph. Draw the constraint graph associated with your CSP. The nodes are provided for you. Draw the arcs.



8.b (3 pts) Degree Heuristic. Assume that you have not assigned any variables yet.

List all variables that might be selected by the Degree Heuristic: ____________________________

8.c (3 pts) Forward Checking. Consider the assignment below. E2 is assigned R. Cross out all the values that would be eliminated by forward checking:

E1	E2	E3	E4	E5	E6	E7
R G B	R	R G B	R G B	R G B	R G B	R G B

8.d (3 pts) Minimum Remaining Values Heuristic. Consider the assignment below. E7 is assigned R and constraint propagation has been done.

E1	E2	E3	E4	E5	E6	E7
R G B	R G B	R G B	G B	G B	G B	R

List all variables that might be selected by the MRV Heuristic: ______________________________

8.e (3 pts) Least Constraining Value Heuristic. Consider the assignment below. E1 is assigned R, E6 assigned G, and constraint propagation has been done. Assume you have selected E5.

E1	E2	E3	E4	E5	E6	E7
R	B	G	G B	R G B	G	R B

List all values that might be selected by the LCV Heuristic: ______________________________

8.f (3 pts) Arc Consistency. Consider the assignment below; E2 is assigned R, and E7 is assigned B, but no constraint propagation has been done. Cross out all values that would be eliminated by Arc Consistency (AC-1 or AC-3).

E1	E2	E3	E4	E5	E6	E7
R G B	R	R G B	R G B	R G B	R G B	B

8.g (3 pts) Min Conflicts Local Search. Consider the complete but inconsistent assignment below. E1 is selected to be assigned a new value.

E1	E2	E3	E4	E5	E6	E7
R	B	G	G	R	G	G

List all values that could be chosen by the Min-Conflicts Algorithm: ____________________________

9. (8 pts total, 1 pt each) CONSTRAINT SATISFACTION PROBLEM (CSP) 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.

	Minimum Remaining Values Heuristic	A	Specifies the allowable combinations of variable values
	Solution to a CSP	B	The values assigned to variables do not violate any constraints
	Least Constraining Value Heuristic	C	Set of allowed values for some variable
	Domain	D	Every variable is associated with a value
	Constraint	E	Nodes correspond to variables, links connect variables that participate in a constraint
	Consistent Assignment	F	Chooses the next variable to expand to have the fewest legal values in its domain
	Complete Assignment	G	A complete and consistent assignment
	Constraint Graph	H	Prefers to search next the value that rules out the fewest choices for the neighboring variables in the constraint graph