Name	1	2	3	4	5	total
	30	40	50	25	30	**175**

UMBC CMSC 471 01 Midterm Exam 2019-03-13

Write your answers on this exam, which is closed book and consists of five problems, summing to 175 points. You have the entire class period, seventy-five minutes, to work on this exam. There are two blank pages at the end you can use for working out problems, please hand in with your exam. Good luck.

1. True/False [30 points]

Circle either **T** or **F** in the space before each statement to indicate whether the statement is **true** or **false**.

T F In the AIMA framework, a rational agent is one that uses a logical approach to reasoning.

T F An agent's environment is said to be **stochastic** if the next state is completely determined by the current state and the agent’s action.

T F Depth-first search is an **optimal, uninformed** search technique.

T F The greedy search algorithm works by always computing the successors of the unexpanded search space node that a heuristic estimates to be closest to a goal node.

T F A search procedure is considered to be **sound** if and only if it will always find a solution to a search problem if one exists.

T F If f1(s) and f2(s) are two admissible A* heuristics, then their sum f(s) = f1(s)+f2(s) is also be admissible.

T F Solving many Constraint Satisfaction Problems (CSPs) using either forward checking or arc consistency still requires some kind of backtracking search.

T F A problem constraint involving three variables (e.g., X+Y < Z) cannot be modeled in a constraint graph because a graph can only represent unary or binary relations.

T F Using the minimax procedure with and without alpha-beta pruning will always identify the best move for the player whose turn it is to move.

T F Minimax with alpha-beta pruning will maximize the number branches pruned if the moves under a node are ordered from best to worst.

T F Simple hill climbing is a complete algorithm for solving constraint satisfaction problems.

T F In a zero-sum two player game there is necessarily always a winner and a loser.

T F Minimax with and without alpha-beta pruning can sometimes return different results.

T F The game-theory framework can model games that are not zero-sum.

T F A Nash equilibrium in a game is a collection of player strategies where no single player can improve their outcome by changing their strategy.

2. Short answers [40: 10 points each]

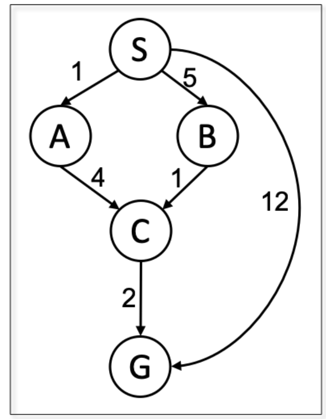
(a) Describe potential advantages and disadvantages of using **hill climbing** to solve a state search problem.

(b) Briefly describe the main advantage of the **depth-first iterative deepening** search algorithm.

(c) Explain why we cannot use traditional minimax for games with an element of chance, such as backgammon.

(d) How many **leaf nodes** will be in a game tree at level X if, at every turn, a player has Y possible moves?

3. Problem solving as search [50 points]



node	h1	h2
S	6	20
A	5	8
B	2	6
C	1	10
G	0	0

Consider the search graph shown on the right. S is the start state and G is the goal state. Edges are annotated with their cost. The table shows the values for each node for two different heuristic functions: h1 and h2. For each of the following search strategies, give (1) the path that will be returned, or write none if no path will be returned; (2) the nodes that are added to the graph, and (3) the nodes that are expanded in the order expanded. If there are ties, assume alphabetical tie-breaking (i.e., nodes for states earlier in the alphabet are expanded first).

(a) [10] Depth-first graph search ignoring arc costs**.** Assume an algorithm that stops as soon as a goal node is added to the graph

(a1)

(a2)

(a3)

(b) [10] Breadth-first graph search ignoring arc costs. Assume an algorithm that stops as soon as a goal node is added to the graph

(b1)

(b2)

(b3)

(e) [10] Algorithm A graph search using the heuristic function h1. Assume we stop only when a goal is reached and there is no possibility of finding a shorter path with the given heuristic.

(e1)

(e2)

(e3)

(e) [10] Algorithm A graph search using the heuristic function h2. Assume we stop only when a goal is reached and there is no possibility of finding a shorter path with the given heuristic.

(e1)

(e2)

(e3)

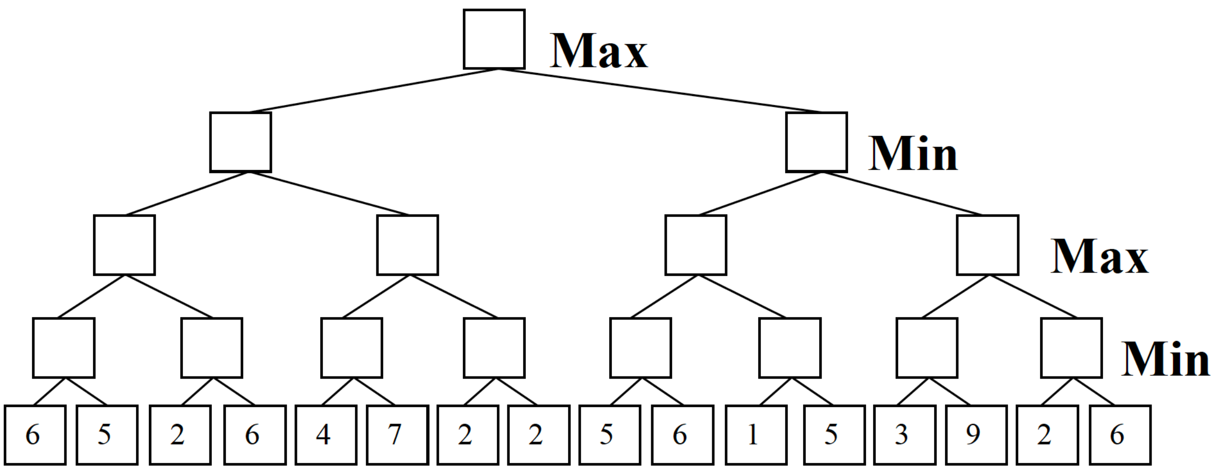
(f) [5] is heuristic h1 admissible?

(f) [5] is heuristic h2 admissible?

4. Game tree search [25 points]

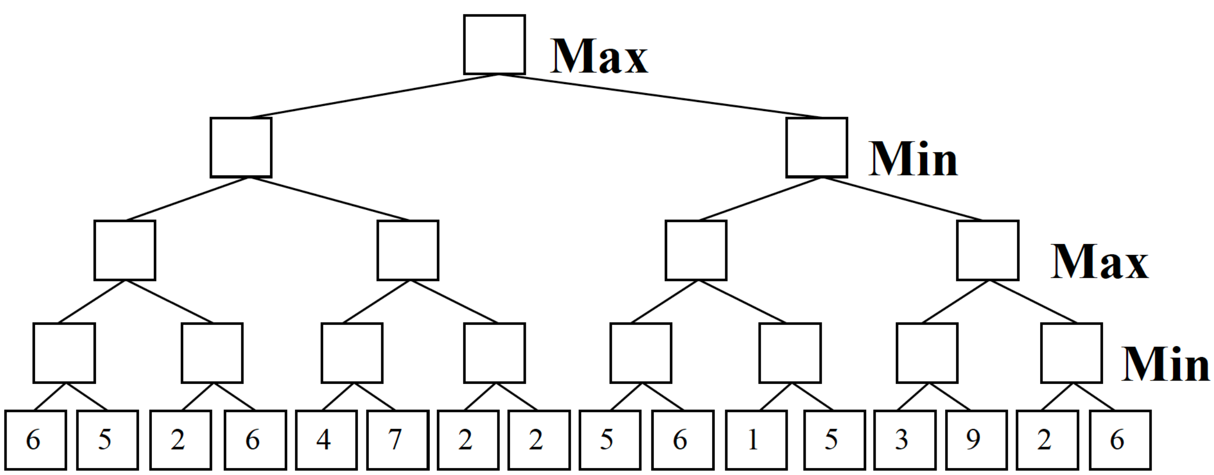
Consider the game tree below in which the first player is trying to maximize her score and the number at the leaves are the values returned by a static evaluator for the board positions reached.

(a) [10] Fill in each box with the value returned by the standard minimax algorithm



(b) [5] Circle the best initial move for the first player: **left** **right**

c) [10] In the copy of this game tree below, fill in each box with the value returned by the standard **alpha-beta algorithm** if the tree is processed from **left to right**. Cross out both leaves and non-leaf nodes that need not be examined or considered.



5. Constraint satisfaction [30 points]

Consider coloring a map of the south western part of the U.S. shown to the right with three colors (R, G, B) so that no two adjacent regions sharing a border have the same color. We can represent this as a CSP graph with six variables, one for each state: CA (California), OR (Oregon), NV (Nevada), AZ (Arizona), ID (Idaho) and UT (Utah), where domain of each is {R, G, B}.

(a) [5] Draw the constraint graph with the variables as nodes and an edge between two nodes that share a constraint.

(a) [5] This table shows the possible values for each variable. Cross out all values that would be eliminated by **forward checking**, after variable NV has just been assigned value G, as shown.

CA	OR	NV	AZ	ID	UT
R G B	R G B	G	R G B	R G B	R G B

(b) [5] NV and UT have been assigned values, but no constraint propagation has been done. Cross out all values that would be eliminated by applying both **forward checking** and **arc consistency**

CA	OR	NV	AZ	ID	UT
R G B	R G B	G	R G B	R G B	R

(c) [5] Can this map be colored with just two colors, say R and B?

(d) [10] If it can be colored with just two colors, show an assignment that satisfies it. If it cannot, give a simple argument to show that it is not possible.

scratch page