Name

1	2	3	4	5	total
20	40	40	25	25	**150**

UMBC CMSC 471 Midterm Exam 2018-03-18

Write your answers on this exam, which is closed book and consists of five problems, summing to 150 points. You have the entire class period, seventy-five minutes, to work on this exam. Good luck.

1. True/False [20 points]

Circle either **T** or **F** in the space before each statement to indicate whether the statement is true or false. If you think the answer is simultaneously true and false, quit while you are ahead.

T F A "rational agent" is one that always performs the best action in a given situation.
TRUE or FALSE. I had intended this to be FALSE, since a rational agent may know which action will result in the best utility if it does not have complete information. However, virtually everyone answered TRUE and, on reflection, I think the meaning of questions is ambiguous.

T F Breadth-first search where all arcs have a cost of one will always find the shortest path to a goal if one exists. **TRUE**

T F Iterative deepening search needs more memory than breadth first search. **FALSE. Iterative deepening performs a series of depth-first searches until a goal is found, so the amount of memory needed is proportional to the maximum depth of the graph. Breadth first search requires space in the worst case proportional to b**d where b is the branching factor and d is the depth of the solution.**

T F If h1(n) and h2(n) are two different admissible heuristics, then (h1(n) + h2(n))/2 is necessarily an admissible heuristic. **TRUE**

T F An advantage of hill-climbing search is that it requires only a constant amount of memory when solving a problem. **TRUE**

T F Depth-first search always expands at least as many nodes as A* search with an admissible heuristic**.**
FALSE

T F Hill climbing search algorithms only work for search spaces that are two-dimensional or have solution-preserving projections onto two-dimensions. **FALSE**

T F Solving constraint satisfaction problems with forward-checking and arc consistence algorithms means that backtracking search is not required. **FALSE**

T F A prisoner's dilemma game is an example of a two-person, partial-information, zero sum game
FALSE. It is not a zero-sum game.

T F A perfectly rational backgammon agent using minimax with unlimited lookahead would never loose. **FALSE. A player's available moves in backgammon is determined by rolling dice, so an unlucky sting of rolls might lead to a player losing.**

2. Short answers [40: 10 points each]

(a) When defining algorithms to find a shortest path in problem where arcs have a cost, we usually specify that the cost must be non-negative. Why is this constraint important? Be concise and specific!

If we allowed arc negative costs, then if there are loops that contain a negative-cost arc, there may be no lower bound on the cost of a solution.

(b) The minimax algorithm (and its variant alphabeta) cannot be applied to all games. Briefly identify at least four properties a game must have for minimax to be applicable.

- **two players**
- **full information (The game state is completely visible to both players)**
- **zero-sum (If one player wins, the other loses)**
- **deterministic (A move has a given outcome; no chance effects like dice rolls)**
- **turn-based = asynchronous (players alternate moves, i.e. no simultaneous moves)**

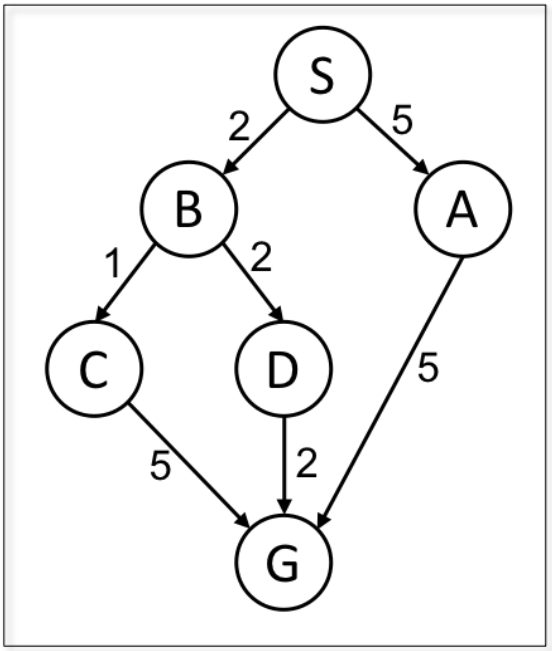
(c) What is a main advantage of depth-first search over breadth-first search search?

DFS has two advantages: (1) it requires less space and (2) it may find a solution must faster

(d) What is a main advantage of breadth-first search over depth-first search?

BFS has several advantages: (1) it finds an optimal solution of one exists, (2) it is complete in that it always finds a solution if one exists, even for infinite problem spaces.

3. Problem solving as search [40 points]



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 three different heuristic functions: h1, h2 and h3.

node	h1	h2	h3
S	0	5	6
A	0	3	5
B	0	4	2
C	0	5	3
D	0	2	5
G	0	0	0

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) [6] Depth-first search ignoring arc costs*

(a1) **S-A-G**

(a2) **S B C G (left) / S A G (right)**

(a3) **S A (right)**

(b) [6] Breadth-first graph search, ignoring arc costs*

(b1) **S-A-G**

(b2) **S A B G (Right child) / S B A C D G (left)**

(b3) **S A (although B is expanded not part of solution) / S B A (for left)**

(e) [6] Algorithm A graph search using the heuristic function h1

(e1) **S-B-D-G**

(e2) **S A B C D G**

(e3) **S B C D A**

(e) [6] Algorithm A graph search using the heuristic function h2

(e1) **S-B-D-G**

(e2) **S A B C D G**

(e3) **S B D**

(e) [6] Algorithm A graph search using the heuristic function h3

(e1) **S-B-C-G**

(e2) **S A B C D G**

(e3) **S B C**

(f) [3] is heuristic h1 admissible? **YES**

(f) [3] is heuristic h2 admissible? **YES**

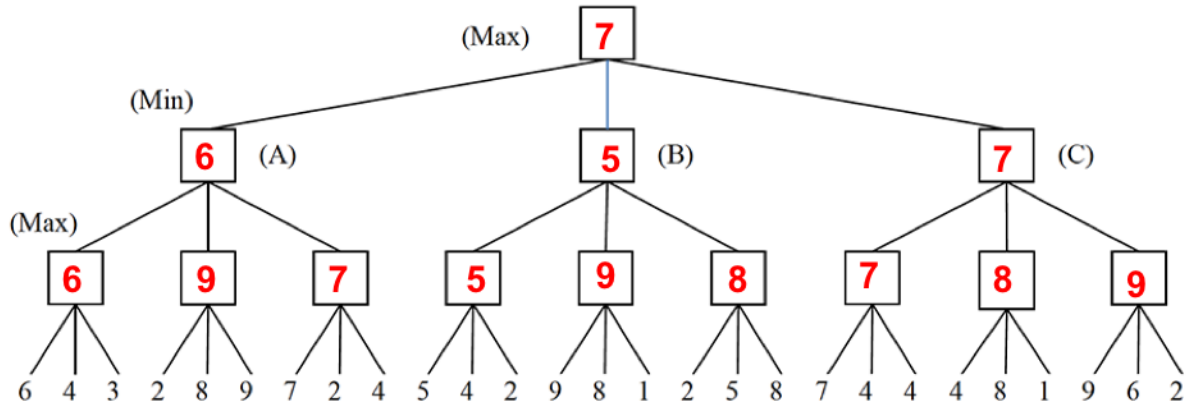
(f) [3] is heuristic h3 admissible? **NO**

* assume an algorithm that stops as soon as a goal node is added to the graph

4. Game tree search [25 points]

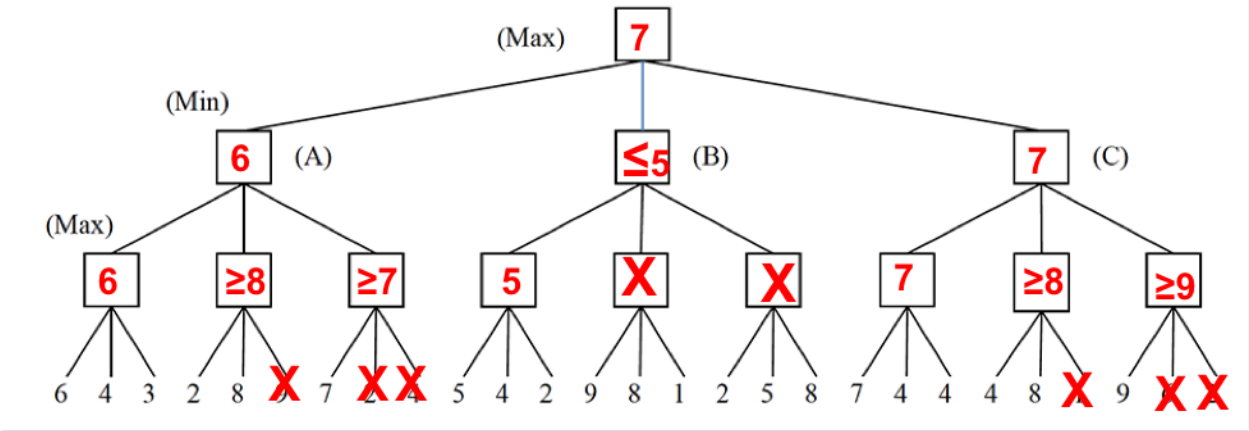
Consider the game tree below in which the first player is trying to maximize her score and the number at the leaves are the values returned by a static evaluator for the board positions reached.

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



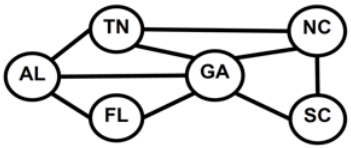
(b) [5] Circle the best initial move for the first player: A B C

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**. And cross out both leaves and non-leaf nodes that need not be examined or considered.



5. Constraint satisfaction [25 points]

Consider coloring a map of the south eastern part of the U.S. shown to the right with three colors: R, G, B so that no two adjacent regions that share a border have the same color. We can represent this as a CSP graph with six variables.



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

AL	TN	FL	GA	NC	SC
R ~~G~~ B	G	R G B	R ~~G~~ B	R ~~G~~ B	R G B

(b) [5] AL and FL 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**

AL	TN	FL	GA	NC	SC
B	R ~~G~~ ~~B~~	R	~~R~~ G ~~B~~	~~R~~ ~~G~~ B	R ~~G~~ ~~B~~

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

NO

(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.

If a map is a triad subgraph of three variables where each is connected to the other two (e.g., AL, TN, GA) then the subgraph cannot be colored with two colors, say C1 and C2. If we color one of the nodes in the triad with color C1, then both of the other nodes must be colored with C2. However, they are connected so they cannot both be C2.