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.

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.

T F Iterative deepening search typically needs more memory than breadth first search.

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

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

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

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

T F Applying constraint satisfaction to problems using both forward-checking and arc consistency algorithms means that backtracking search is not required to find a solution.

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

T F A perfectly rational backgammon agent using minimax with unlimited lookahead would never lose.
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!

(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.

(c) What are the main advantages of depth-first search over breadth-first search?

(d) What are the main advantages of breadth-first search over depth-first search?


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. 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 graph search* ignoring arc costs
   (a1) 
   (a2) 
   (a3) 

(b) [6] Breadth-first graph search* ignoring arc costs
   (b1) 
   (b2) 
   (b3) 

(e) [6] Algorithm A graph search using the heuristic function h1
   (e1) 
   (e2) 
   (e3) 

(e) [6] Algorithm A graph search using the heuristic function h2
   (e1) 
   (e2) 
   (e3) 

(e) [6] Algorithm A graph search using the heuristic function h3
   (e1) 
   (e2) 
   (e3) 

(f) [3] is heuristic h1 admissible?

(f) [3] is heuristic h2 admissible?

(f) [3] is heuristic h3 admissible? * 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: AL, TN, FL, GA, NC and SC where domain of each is \{R, G, B\}.

(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.

<table>
<thead>
<tr>
<th></th>
<th>AL</th>
<th>TN</th>
<th>FL</th>
<th>GA</th>
<th>NC</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>G</td>
<td>B</td>
<td>G</td>
<td>R</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>G</td>
<td>R</td>
<td>B</td>
<td>R</td>
<td>G</td>
<td>B</td>
<td>R</td>
</tr>
<tr>
<td>B</td>
<td>R</td>
<td>G</td>
<td>B</td>
<td>R</td>
<td>G</td>
<td>B</td>
</tr>
</tbody>
</table>

(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.

<table>
<thead>
<tr>
<th></th>
<th>AL</th>
<th>TN</th>
<th>FL</th>
<th>GA</th>
<th>NC</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>R</td>
<td>G</td>
<td>B</td>
<td>R</td>
<td>G</td>
<td>B</td>
</tr>
</tbody>
</table>

(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.