CS-171, Intro to A.I. — Midterm Exam — Spring Quarter, 2011

(1 pt) YOUR NAME AND EMAIL ADDRESS: ______________________________

YOUR ID: ________ ID TO RIGHT:________ ROW:____ NO. FROM RIGHT:____

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

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

Please clear your desk except pen, pencil, eraser, and one blank piece of paper (for scratch pad use).

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

(1 pt) YOUR NAME AND EMAIL ADDRESS.

1. (15 pts total, -3 pts for each error, but not negative) ALPHA-BETA PRUNING.

2. (20 pts total, 1 pt each) SEARCH PROPERTIES.

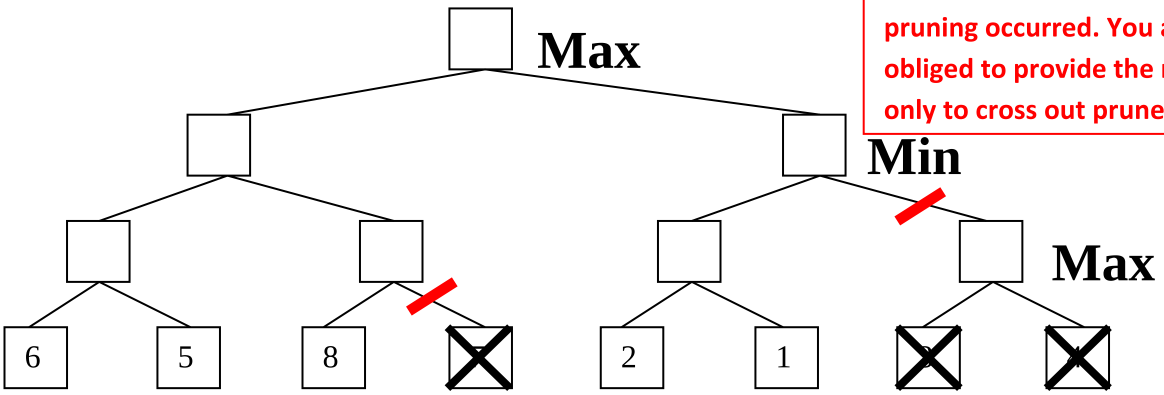
3. (20 pts total, 2 pts each) ADVERSARIAL (GAME) SEARCH CONCEPTS.

4. (24 pts total, 2 pts each) CONSTRAINT SATISFACTION PROBLEMS.

5. (20 pts total, 5 pts each subproblem, -1 pt each error, not negative for any subproblem) GRAPH SEARCH.

1. (15 pts total, -3 pts for each error, but not negative) ALPHA-BETA PRUNING.
The game tree below illustrates a game position. It is **Max**'s turn to move. CROSS OUT EACH LEAF NODE THAT WILL NOT BE EXAMINED BECAUSE IT IS PRUNED BY ALPHA-BETA PRUNING. You do not need to indicate the branch node values. You may wish to hand-si

Red lines indicate where in the tree pruning occurred. You are not obliged to provide the red lines — only to cross out pruned leaf nodes.



2. (20 pts total, 1 pt each) 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 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	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

3. (10 pts total, 1 pt each) ADVERSARIAL (GAME) SEARCH CONCEPTS.
For each term on the left, write in the letter of the best answer or correct definition on the right.

D	Game Strategy	A	Approximates the value of a game state (i.e., of a game position)
H	Cut-off Test	B	In all game instances, total pay-off summed over all players is the same
E	Alpha-Beta Pruning	C	Tree where nodes are game states and edges are game moves
G	Weighted Linear Function	D	Function that specifies a player’s move in every possible game state
J	Terminal Test	E	Returns same move as MiniMax, but may prune more branches
I	ExpectiMiniMax	F	Optimal strategy for 2-player zero-sum games of perfect information, but impractical given limited time to make each move
C	Game Tree	G	Vector dot product of a weight vector and a state feature vector
A	Heuristic Evaluation Function	H	Function that decides when to stop exploring this search branch
B	Zero-sum Game	I	Generalizes MiniMax to apply to games with chance (stochastic games)
F	MiniMax Algorithm	J	Function that says when the game is over

4. (24 pts total, 2 pts each) CONSTRAINT SATISFACTION PROBLEMS.

Label the following statements as T (true) or F (false).

4a. ___T_____ A **constraint satisfaction problem** (CSP) consists of a set of variables, a set of domains (one for each variable), and a set of constraints that specify allowable combinations of values.

4b. ___F_____ A **consistent assignment** is one in which every variable is assigned.

4c. ___F_____ A **complete assignment** is one that does not violate any constraints.

4d. ___F_____ A **partial assignment** is one that violates only some of the constraints.

4e. ___T_____ The nodes of a **constraint graph** correspond to variables of the problem, and a link connects any two variables that participate in a constraint.

4f. ___T_____ A **constraint** consists of a pair <*scope, rel*>, where *scope* is a tuple of variables that participate and *rel* defines the values those variables can take on.

4g. ___T_____ A variable in a CSP is **arc-consistent** iff, for each value in its domain and each of its binary constraints, that constraint is satisfied by that domain value together with some value in the domain of the other variable in that constraint.

4h. ___T_____ The **minimum-remaining-values** (MRV) heuristic chooses the variable with the fewest remaining legal values to assign next.

4i. ___F_____ The **degree heuristic** is used to set the temperature in methods for solving CSPs based on Simulated Annealing.

4j. ___T_____ The **least-constraining-value** heuristic prefers the value that rules out the fewest values for the neighboring variables in the constraint graph.

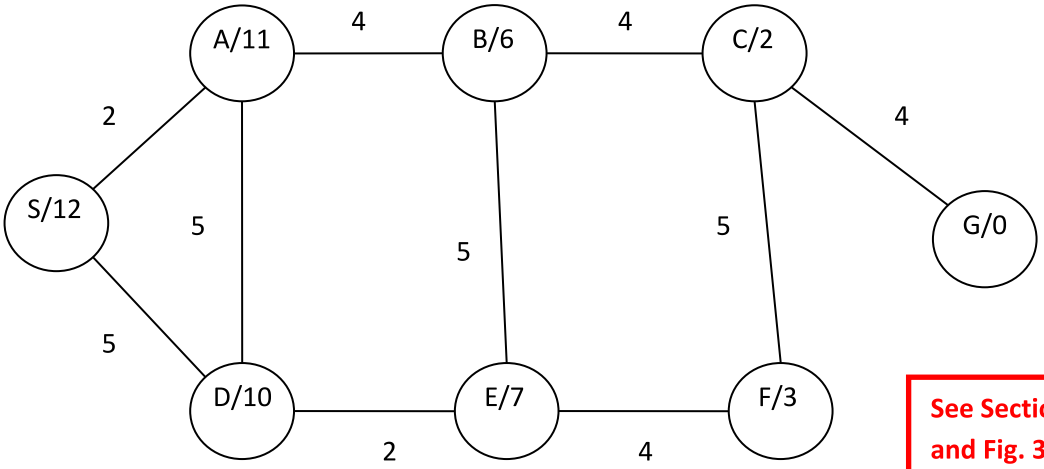
4k. ___T_____ The **min-conflicts** heuristic for local search prefers the value that results in the minimum number of conflicts with other variables.

4l. ___F_____ The **min-conflicts** heuristic is rarely used because it is only effective when the constraint graph is a tree.

Please see the lecture slides for Uninformed Search, topic "When to do Goal-Test? When generated? When popped?" for clarification about exactly what to do in practical cases.

5. (20 pts total, 5 pts each subproblem, -1 pt each error, not negative for any subproblem) GRAPH SEARCH. The following problem asks about this graph. The children of a node are given in ALPHABETICAL ORDER when the node is expanded. You are doing graph search so nodes are never expanded twice (i.e., no cycles; graph search eliminates repeated states).

"S" is the start node, and "G" is the goal node. The number inside each node is an estimate of the remaining distance to the goal from that node. The number by each arc is the step cost for that arc. The first one has been done for you as an example. Hand-simulate the queue.



See Section 3.4.3 and Fig. 3.17.

5.a. (Example subproblem)
Write the order in which Depth-First search expands nodes:

DFS does the Goal-test before the child is pushed onto the queue. The goal is found when C is expanded.

S A B C G

5.b. (**5 pts for this subproblem, -1 pts for each e**
Write the order in which Breadth-First search exp

See Section 3.4.1 and Fig. 3.11.

BFS does the Goal-test before the child is pushed onto the queue. The goal is found when C is expanded.

S A D B E C G

5.c. (**5 pts for this subproblem, -1 pts for each er**
Write the order in which Greedy Best-First search e

See Section 3.5.1 and Fig. 3.23.

GBFS has the same behavior whether the goaltest is done before node is pushed or after node is popped.

S D E F C G

5.d. (**5 pts for this subproblem, -1 pts for each error, but not negative)**
Write the order in which Uniform-Cost search expands nodes:

UCS does goaltest when node is popped off queue.

See Section 3.4.2 and Fig. 3.14.

S A D B E C F G

5.e. (**5 pts for this subproblem, -1 pts for each error, but not negative)**
Write the order in which A* search expands nodes:

See Section 3.5.2 and Figs. 3.24-25.

A* does goaltest when node is popped off queue.

S A B C G