CS-171, Intro to A.I. — Mid-term Exam — Fall Quarter, 2013

YOUR NAME AND ID NUMBER: ______________________________

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.

When you are told to begin the exam, please check first to make sure that you have all seven pages, as numbered 1-7 in the bottom-right corner of each page.

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

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 available for each question so you can plan your time.

1. (10 pts total, -2 pts each error, but not negative) STATE SPACE SEARCH.

2. (5 pts total, -2 pts for each error, but not negative) TASK ENVIRONMENT.

3. (5 pts total, -1 pt each wrong answer, but not negative) SEARCH PROPERTIES.

4. (15 pts total, 5 pts each) ALPHA-BETA PRUNING NODE ORDERING.

5. (20 pts total, -5 for each error, but not negative) RESOLUTION THEOREM PROVING.

6. (20 points total, 4 pts each) CONSTRAINT SATISFACTION PROBLEMS.

7. (25 pts total, 5 pts each) SEARCH.

1. (10 pts total, -2 pts each error, but not negative) STATE SPACE SEARCH. The ***Missionaries and Cannibals*** problem is a classic brainteaser. It is a member of a class of river-crossing brainteasers that includes *Farmer to Market*, *Jealous Husbands,* and others (see https://en.wikipedia.org/wiki/River-crossing_problems).

See Section 3.1.1 and Figure 3.3

Three missionaries and three cannibals must cross a river using a boat which can carry at most two people. The constraint is that, for both river banks, if there are missionaries present on the bank, they cannot be outnumbered by cannibals (if they were, the cannibals would eat the missionaries). The boat cannot cross the river by itself without people on board. **How can all six people get safely across the river?**

We will call states where anyone gets eaten "forbidden" and consider them as part of the state space, but from which there is no return; i.e., there is no way to go to any other state (including back to the previous state) from such a forbidden state. All other states are "allowed." It is always possible to go from an allowed state to some other state(s).

One way to represent this problem is as a state vector with components (M,C,B), where M=the number of missionaries on the wrong side, C=the number of cannibals on the wrong side, and B=the number of boats on the wrong side. Since everything starts on the wrong side, the start state is (3,3,1). The single goal state is (0,0,0).

1.a. (2 pts) What is the maximum branching factor counting BOTH allowed and forbidden states as children?

______5__________.

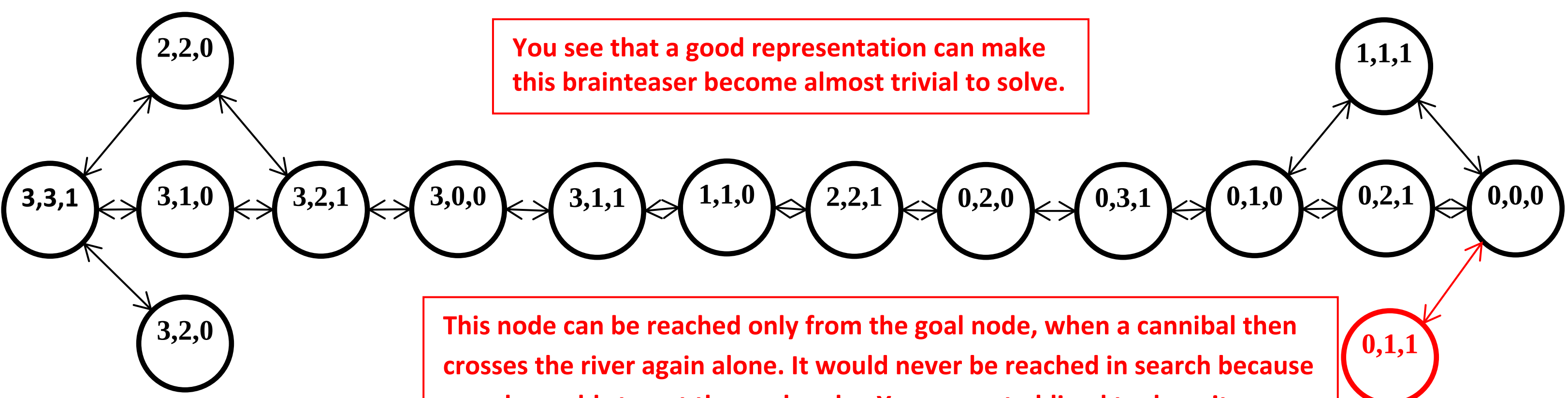
The boat can carry (1) a missionary alone, (2) a cannibal alone, (3) a missionary and a cannibal, (4) two missionaries, or (5) two cannibals.

1.b. (8 pts, -2 for each error, but not negative) Draw the state space showin ALLOWED states. I.e., do NOT show the FORBIDDEN states.

Represent an allowed state as a circle enclosing its state vector (M,C Connect allowed states that are possible successors of each other. The star states are shown. The first node expansion is done for you as an example.

You are not obliged to draw the state space in a straight line. If your graph is correct topologically, you will receive full credit.

You see that a good representation can make this brainteaser become almost trivial to solve.



This node can be reached only from the goal node, when a cannibal then crosses the river again alone. It would never be reached in search because search would stop at the goal node. You are not obliged to draw it.

If you solved the problem "in English," i.e., wrote out the boat moves in English that solved the brainteaser, then you will receive some credit, but very little. The point of the problem was to reason about transitions in state space, and to illustrate the general and powerful state-space approach to problem-solving --- not to get the missionaries and cannibals safely across the river.

2. (5 pts total, -2 pts for each error, but not negative) 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.

See Section 2.3.1.

P<u>erformance (measure)</u> E<u>nvironment</u> A<u>ctuators</u> S<u>ensors</u>

3. (5 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 identical and equal to some positive ε; and in Bidirectional search both directions using breadth-first search.

See Figure 3.21.

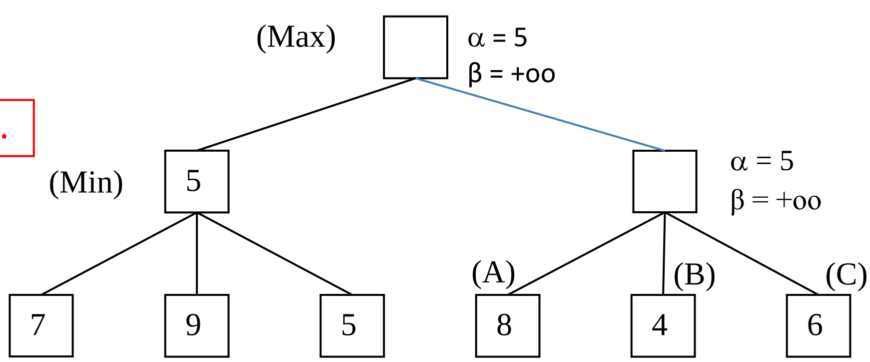
Note that these conditions satisfy all of the footnotes of Fig. 3.21 in your book.

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

4. (15 pts total, 5 pts each) ALPHA-BETA PRUNING NODE ORDERING. Alpha-beta pruning effectiveness depends on the order in which node values are encountered. A bad node order can yield no pruning. A good node order can yield maximal pruning. This question asks about node ordering in relation to alpha-beta pruning effectiveness.

In the game tree below, the agent has searched the left branch and determined it is worth 5. It is about to search the right branch. How does the left-to-right order of the node values on the right branch affect how many nodes can be pruned in that branch? **Below, write A, B, and C in an order that gives a correct answer to each question.**

See Section 5.3.



You are only obliged to provide one correct node order for each question. However, for each question there are two correct answers.

4.a. (5 pts) What node ordering prunes two nodes? <u>B, A, C or B, C, A</u>

4.b. (5 pts) What node ordering prunes one node? <u>A, B, C or C, B, A</u>

4.c. (5 pts) What node ordering prunes zero nodes? <u>A, C, B or C, A, B</u>

5. (20 pts total, -5 for each error, but not negative) RESOLUTION THEOREM PROVING. You are engaged in Knowledge Engineering for the Wumpus Cave. You have interviewed an expert on the Wumpus Cave who told you, among other things, “A breeze in square (1,1) is equivalent to a pit in square (1,2) or a pit in square (2,1).” You translated this into propositional logic as, “(B11 ⇔ P12 ∨ P21),” and then into Conjunctive Normal Form as “(¬B11 ∨ P12 ∨ P21) ∧ (¬P12 ∨ B11) ∧ (¬P21 ∨ B11).”

Now it is time for the first “live” test of your system. An agent has been lowered down into the Wumpus cave, and reports back by radio, “Square (1,1) has a breeze. Also, I went into square (1,2) and I did not die, so it does not have a pit.” You translate this knowledge into propositional logic as “(B11) ∧ (¬ P12)” and add it to your knowledge base.

Next your system is asked to perform inference. The agent asks by radio, “Is it true that square (2,1) has a pit?” You translate this query into propositional logic as the goal sentence “(P21).” You form the negated goal as “(¬ P21).” Your knowledge base plus negated goal is:

(¬B11 ∨ P12 ∨ P21)
(¬P12 ∨ B11)
(¬P21 ∨ B11)
(B11)
(¬ P12)
(¬ P21)

See Sections 7.2 and 7.5.2.

Run resolution on this knowledge base until you produce the null clause, “( )”, thereby proving that the goal sentence is true. The shortest proof I know of is only three lines long. It is OK to use more lines, if your proof is correct. SHOW YOUR WORK.

Repeatedly choose two clauses, write one clause in the first blank space on a line, and the other clause in the second. Apply resolution to them. Write the resulting clause in the third blank space, and insert it into the knowledge base.

Resolve ___(¬B11 ∨ P12 ∨ P21)___ and ___(B11)___ to give ___(P12 ∨ P21)___

Resolve ___(P12 ∨ P21)___ and ___(¬P12)___ to give ___(P21)___

Resolve ___(P21)___ and ___(¬P21)___ to give ___( )___

Resolve ______ and ______ to give ______

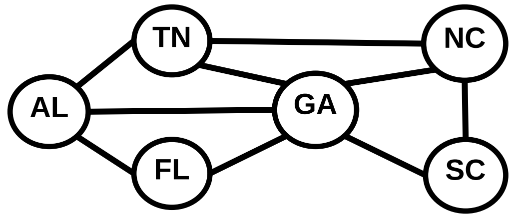
Resolve ______

Other proofs are OK as long as they are correct. For example, you might perform the resolution steps above in any other order you choose.

Resolve ______ and ______ to give ______

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

6. (20 points total, 4 pts each) CONSTRAINT SATISFACTION PROBLEMS.



See Chapter 6.

You are a map-coloring robot assigned to color this Southeast USA map. Adjacent regions must be colored a different color (R=Red, B=Blue, G=Green). The constraint graph is shown.

6a. (4pts total, -2 each wrong answer, but not negative) FORWARD CHECKING. Cross out all values that would be eliminated by Forward Checking, after variable GA has just been assigned value G, as shown:

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

6b. (4pts total, -2 each wrong answer, but not negative) ARC CONSISTENCY. AL and FL have been assigned values, but no constraint propagation has been done. Cross out all values that would be eliminated by Arc Consistency (AC-3 in your book).

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

6c. (4pts total, -2 each wrong answer, but not negative) MINIMUM-REMAINING-VALUES HEURISTIC. Consider the assignment below. TN is assigned and constraint propagation has been done. List all unassigned variables that might be selected by the Minimum-Remaining-Values (MRV) Heuristic:______**AL, GA, NC**______.

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

6d. (4pts total, -2 each wrong answer, but not negative) DEGREE HEURISTIC. Consider the assignment below. (It is the same assignment as in problem 6c above.) TN is assigned and constraint propagation has been done. List all unassigned variables that might be selected by the Degree Heuristic:.______**GA**______.

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

6e. (4pts total) MIN-CONFLICTS HEURISTIC. Consider the complete but inconsistent assignment below. GA has just been selected to be assigned a new value during local search for a complete and consistent assignment. What new value would be chosen below for GA by the Min-Conflicts Heuristic?.______**R**______.

AL	TN	FL	GA	NC	SC
B	G	R	**?**	G	B

7 (25 pts total, 5 pts each) SEARCH. Execute Tree Search through this graph (i.e., do not remember visited nodes, so repeated nodes are possible). It is not a tree, but pretend that you don't know that.

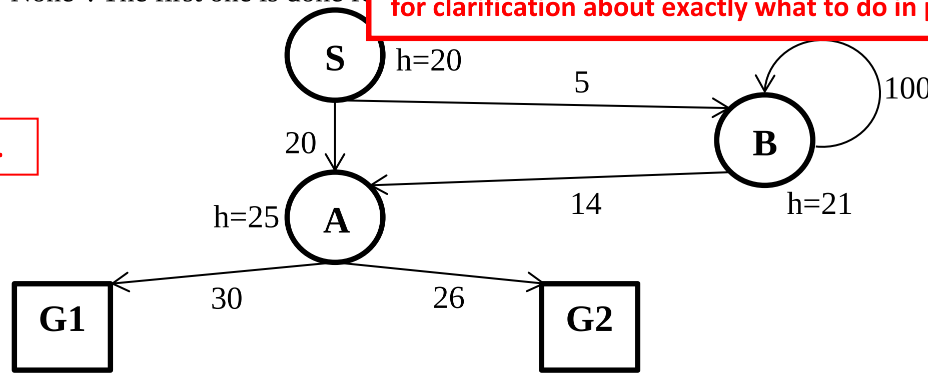
Step costs are given next to each arc. Heuristic values are given next to each node (as h=x). The successors of each node are indicated by the arrows out of that node.

Successors are returned in left-to-right order. (Note: B is a successor of itself).

For each search strategy below, show the ord node means that its children are generated), ending w start to goal, or write "None". The first one is done fo

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.

See Chapter 3.



7a. DEPTH FIRST SEARCH.

See Section 3.4.3 and Fig. 3.17.

of node expansion: S A G1

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

Path found: S A G1

7b. (5 pts) BREADTH FIRST SEARCH.

See Section 3.4.1 and Fig. 3.11.

of node expansion: S A G1

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

Path found: S A G1

7c. (5 pts) UNIFORM COST SEARCH.

See Section 3.4.2 and Fig. 3.14.

of node expansion: S B A A G2

UCS does goaltest when node is popped off queue.

Path found: S B A G2

pts) GREEDY (BEST-FIRST) SEARCH.

See Section 3.5.1 and Fig. 3.23.

of node expansion: S B B B B B B ... etc.

B always has lower h(=21) than any other node on queue.

Path found: None

7d. (5 pts) ITERATED DEEPENING SEARCH.

See Sections 3.4.4-5 and Figs. 3.18-19.

ode expansion: S S A G1

IDS does goaltest when node is generated. Goal G1 is found when A is expanded.

Path found: S A G1

7e. (5 pts) A* SEARCH.

See Section 3.5.2 and Figs. 3.24-25.

of node expansion: S B A A G2

A* does goaltest when node is popped off queue.

und: S B A G2

****** THIS IS THE END OF THE MID-TERM EXAM ******