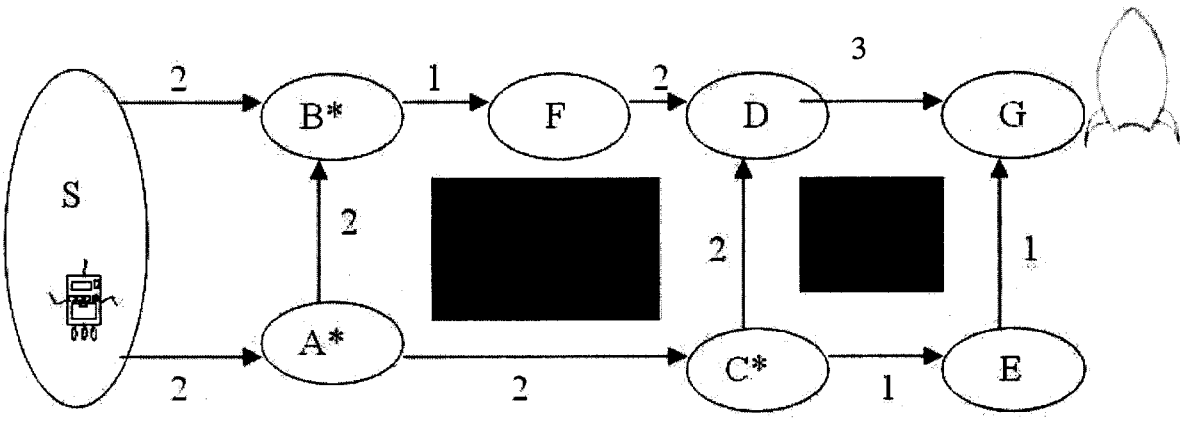
6.034 Quiz 1 SOLUTIONS
October 8, 2003

Name	
EMail	

Problem number	Maximum	Score	Grader
1	30		
2	16		
3	20		
4	24		
5	10		
Total	100		

Question 1: Search (30 points)

Wallace and Gromit have just finished their vacation on the moon and are about to head back to Earth in their rocket ship (located at position G below). The local robot disparately wants to go with them, but must hurry to get to the rocket ship in time from position S below. He has to navigate around buildings on his path. His navigation also has to be smart to avoid the traffic lights, as the traffic lights slow him down if he goes through them straight or makes a 90° left turn (90° right turns do not slow him down).



There are 3 sets of lights located around the two buildings. (The buildings are designated as solid rectangles.) The lights are located at junctions A, B, C (which are designated by *).

All path lengths are as shown on the diagram. Streets are one-way in the direction of the arrows.

Assume that for every traffic light:

- Going straight takes time equivalent to traveling a path of length 1 (he has bad luck and always has to wait for a light cycle).
- Turning 90° left takes time equivalent to traveling a path of length 2 (evidently, he has to wait for a green light and then for traffic to clear).
- Turning 90° right takes no time.

Heuristic distances to the goal are associated with each node as follows:

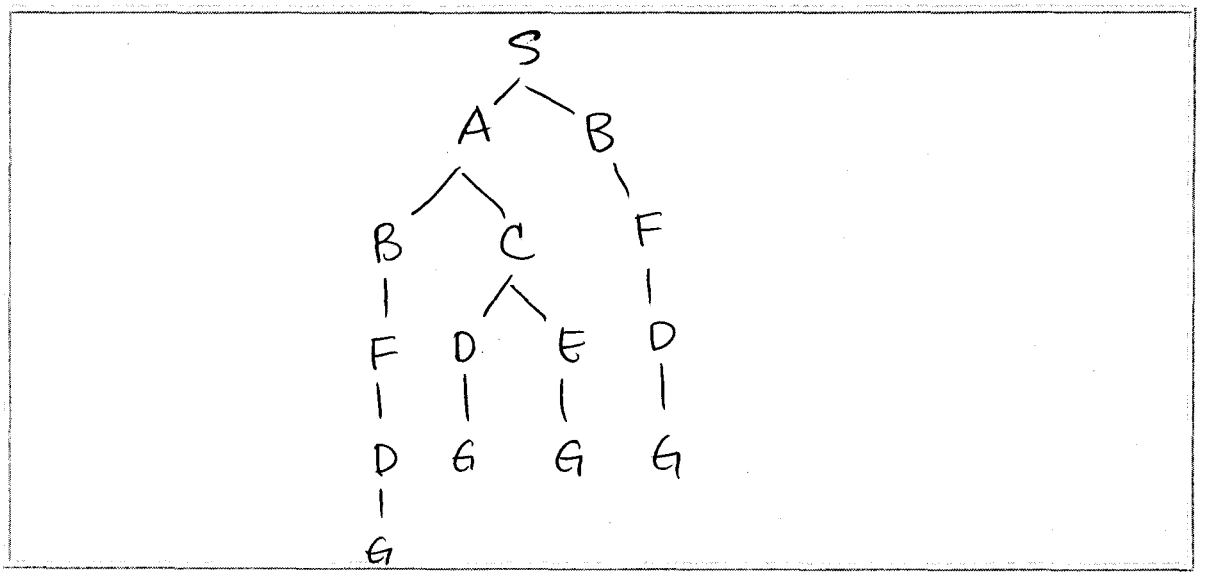
S=5, A=3, B=2, C=3, D=1, F=2, E=3, G=0.

The objective is to find the path that gets him to his destination in the shortest time possible.

Break ties alphabetically.

Part 1.1: Search Tree (4 points)

Draw the search tree, ordering the descendants of each node alphabetically from left to right.



Part 1.2: Depth First, no backup, no enqueued list (4 points)

List the order in which nodes are extended during search.

S A B F D G

Part 1.3: Breadth First, with enqueued list (4 points)

List the order in which nodes are extended during search.

S A B C F D E G

Part 1.4: Hill climbing, with backup, no enqueued list (4 points)

Use the heuristic distances to the goal provided above. List the nodes in the path found.

S B F D G

Part 1.5: Branch and bound, with extended list, no consideration of distance to goal (4 points)

What is the path found to the rocket ship?

S A C E G

Part 1.6: A* [Branch and bound with extended list and distance to goal] (4 points)

Now, you are told to use branch and bound, with an extended list, and using the remaining distance to the goal provide above.

What is the path found?

S B F D G

Part 1.7: Admissible Heuristic (3 points)

The heuristic provided is not an admissible heuristic (defined as an estimate of remaining distance that is less than the actual remaining distance) at which of the following nodes (circle)?

A	C	(E)	none	all

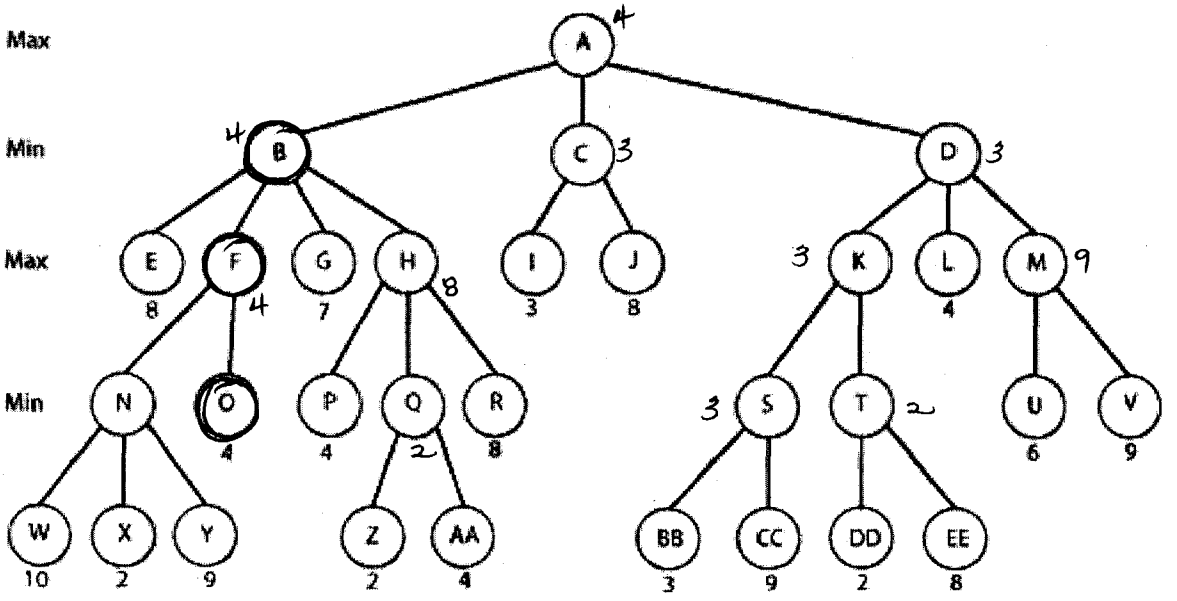
Part 1.8: Shortest Path (3 points)

What is the shortest path (least distance traveled, not necessarily the path that reaches the goal most quickly)?

S A C E G

Question 2: Game Search (16 points)

You are playing a game of chess against your online boyfriend/girlfriend on Yahoo! Games. It's almost the end of the game when your opponent requests to pause the game in order to use the bathroom. You agree but while you wait you decide to sketch a game tree of the remaining moves. The game tree is displayed below:



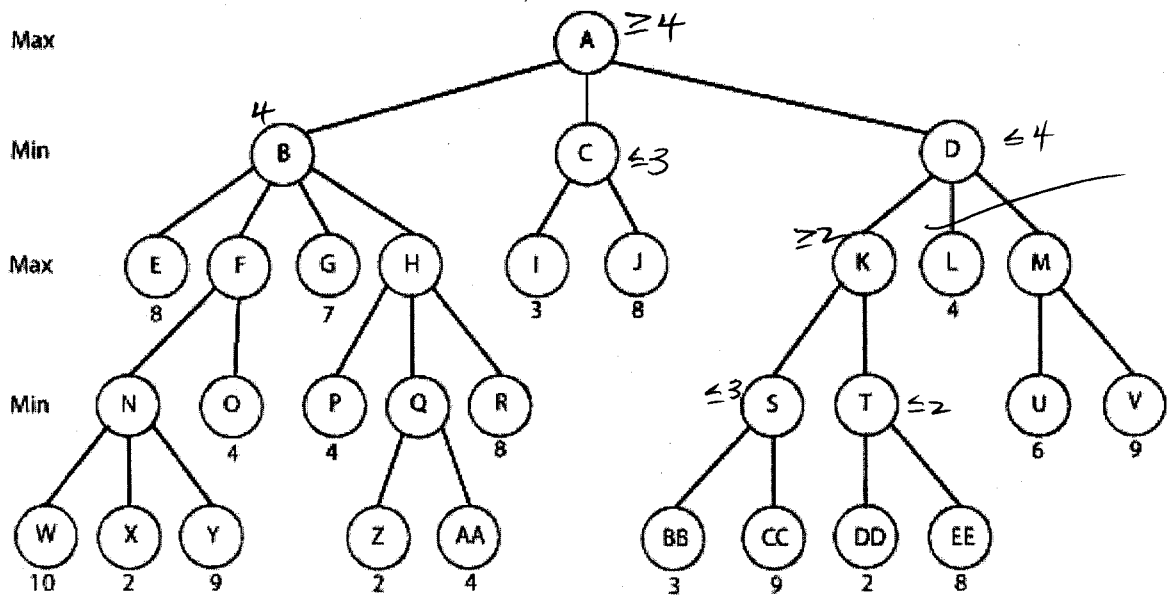
Part 2.1: Minimax (8 points)

After looking at the game tree, you decide to figure out what your best move is. Use minimax to determine the optimal path through the game tree. Show the minimax value for each node on the diagram, then list the nodes on the best-move path below:

A B F O

Part 2.2: Alpha-Beta (8 points)

Your partner still hasn't returned so you look at your game tree and see which nodes could be cut off via alpha-beta. For your convenience the game tree is repeated below:



You are to assume that you have already done the search down the left branch of the tree, discovering that the minimax value of node B is 4. Continue processing the tree from left to right, and **circle the nodes** below for which static values **will be** computed by alpha-beta. Also, **cross out the nodes** for which static values **will not be** computed.

I	X	BB	CC	DD	EE	X	X	X

Question 3: Rules (20 points)

Given the recent spate of lawsuits filed by the Recording Industry of America (RIAA) against people who use peer-to-peer networks, you decide to employ your 6.034 knowledge to help your friends determine whether they are at risk of getting sued. The following is a list of assertions that you know about your friends and their song-swapping habits:

```
(Alice is college-student)
(Bob is college-student)
(Charlie is college-student)
(Charlie uses Kazaa)
(Alice buys CDs)
(Bob uses Morpheus)
(Bob shares files)
```

Through some connections, you have uncovered the set of rules that the RIAA uses to decide whom to sue:

```
R1:
        IF (?x uses Kazaa)
        THEN (?x is file-swapper)

R2:
        IF (?x uses Morpheus)
        THEN (?x is file-swapper)

R3:
        IF (?x buys CDs)
        THEN (?x pays for music)

R4:
        IF (?x is file-swapper)
           (?x shares files)
        THEN (?x will be sued)

R5:
        IF (?x pays for music)er
        THEN (?x will not be sued)

R6:
        IF (?x is college-student)
           (?x is file-swapper)
        THEN (?x will be sued)
```

You decide to implement a forward-chaining rule system to determine whether your friends will be sued.

Important note:

The term **triggered rule** is used in two ways: sometimes it means **the first rule** that has antecedents that match the database and a consequent that is not in the data base; sometimes it means **any rule** that has antecedents that match the database and a consequent that is not in the data base. **In this problem, you are to use the any-rule interpretation**. Thus, there may be more than one triggered rule at any given step in forward chaining.

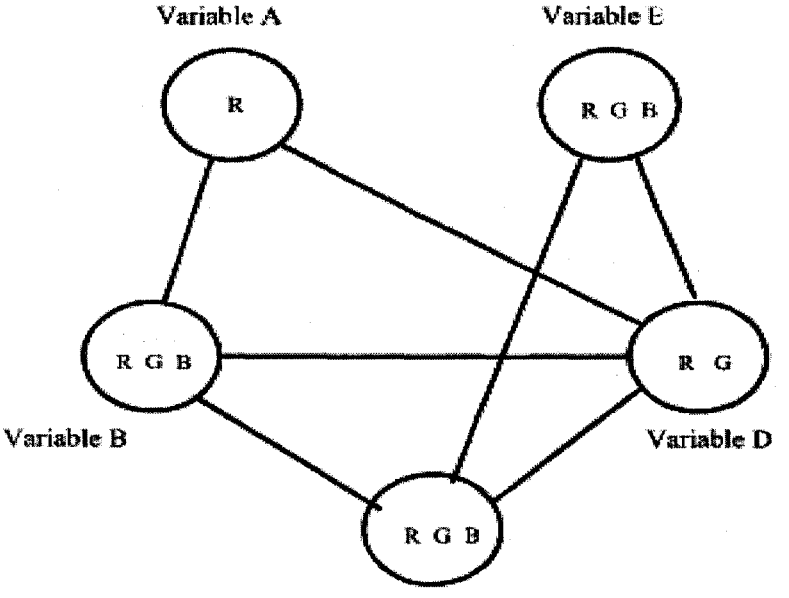
Fill out the following table to show what happens when you run the forward chainer. Use rule ordering for the conflict resolution strategy. Assume new assertions are added to the end of the assertion database. Read the important note in the previous paragraph. Terminate when no further assertions can be made. You may abbreviate clauses as long as there is no ambiguity. (Note: there may be more steps in the table than you need.)

Step	Triggered rules	Rule instance bindings	Rule fired	Assertion added
1	R1	?x = C	R1	(C is file-swapper)
	R2	?x = B		
	R3	?x = A		
2	R2	?x = B	R2	(B is file-swapper)
	R3	?x = A		
	R6	?x = C		
3	R3	?x = A	R3	(A pays for music)
	R4	?x = B		
	R6	?x = B		
	R6	?x = C		

4	R4	?x = B	R4	(B will be sued)
	R5	?x = A		
	R6	?x = B		
	R6	?x = C		
5	R5	?x = A	R5	(A will not be sued)
	R6	?x = C		
6	R6	?x = C	R6	(C will be sued)
7				

Question 4: Constraint Propagation (24 points)

The following constraint network represents a coloring problem with five variables and one, two, or three colors in the domain of each variable. Lines represent constraints requiring that two variables have different colors.

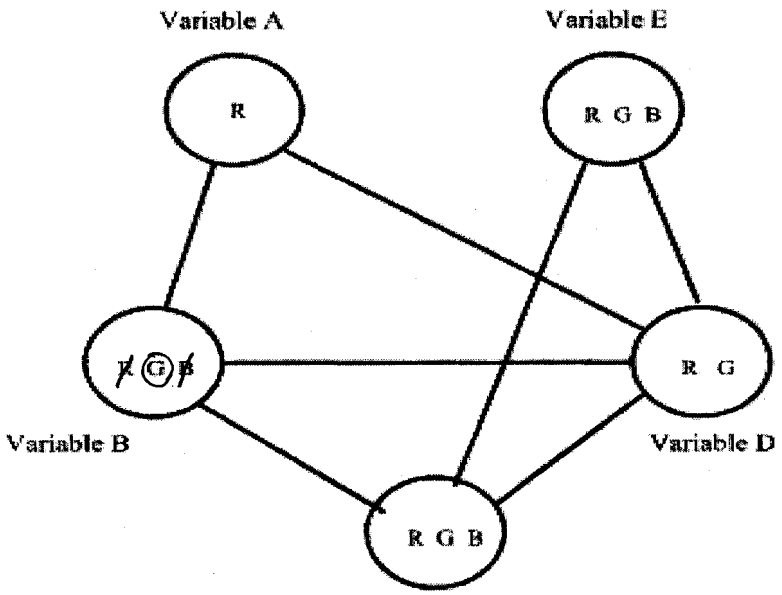


For the first parts of this problem, you are to assume that you are just getting started in your search for a set of assignments that violate no constraint. You perform the prescribed search, working on variables in alphabetical order, and you find an assignment for variable B that permits you to move on to Variable C.

You are to circle the assignment you find for variable B and draw a line through all values that are deleted from variable domains at the point when you move on to Variable C.

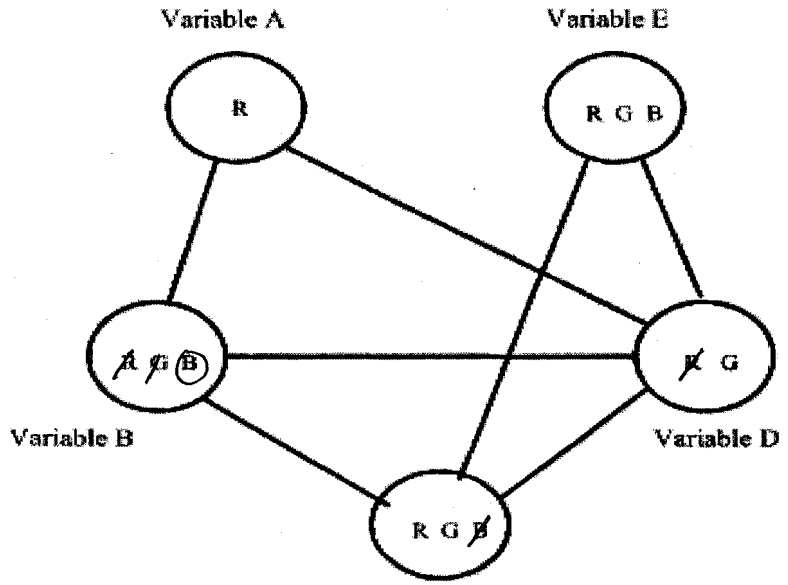
Part 4.1: Backchecking [also known as pure backtracking] (4 points)

The assignment is checked only to be sure it violates no constraint imposed by any previous assignment.



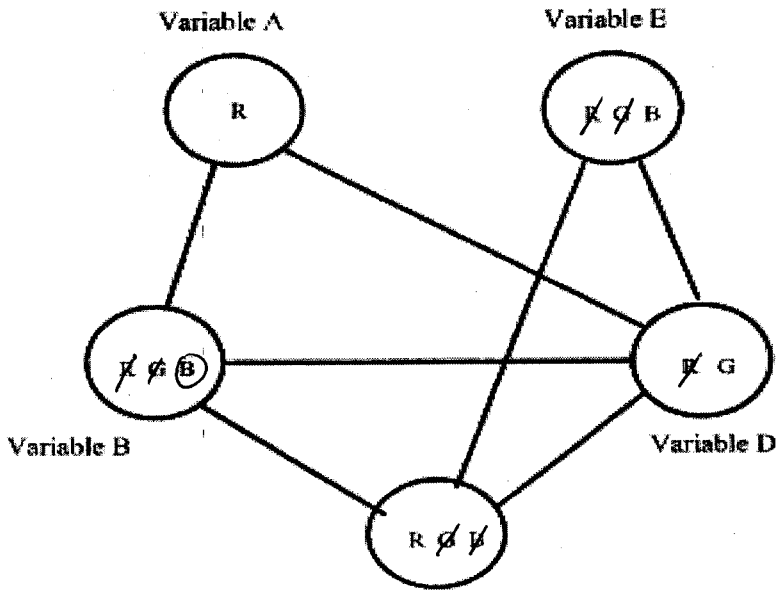
Part 4.2: Immediate Neighbors Forward-Checking (5 points)

The assignment is checked to be sure it violates no constraint imposed by any previous assignment. Also, you check the assigned variable's immediate neighbors.



Part 4.3: Aggressive Forward-Checking (5 points)

The assignment is checked to be sure it violates no constraint imposed by any previous assignment. Also, you check the assigned variable's immediate neighbors, and continue checking on all neighbors of any variable whose domain is reduced to a single value, until no further value can be eliminated.



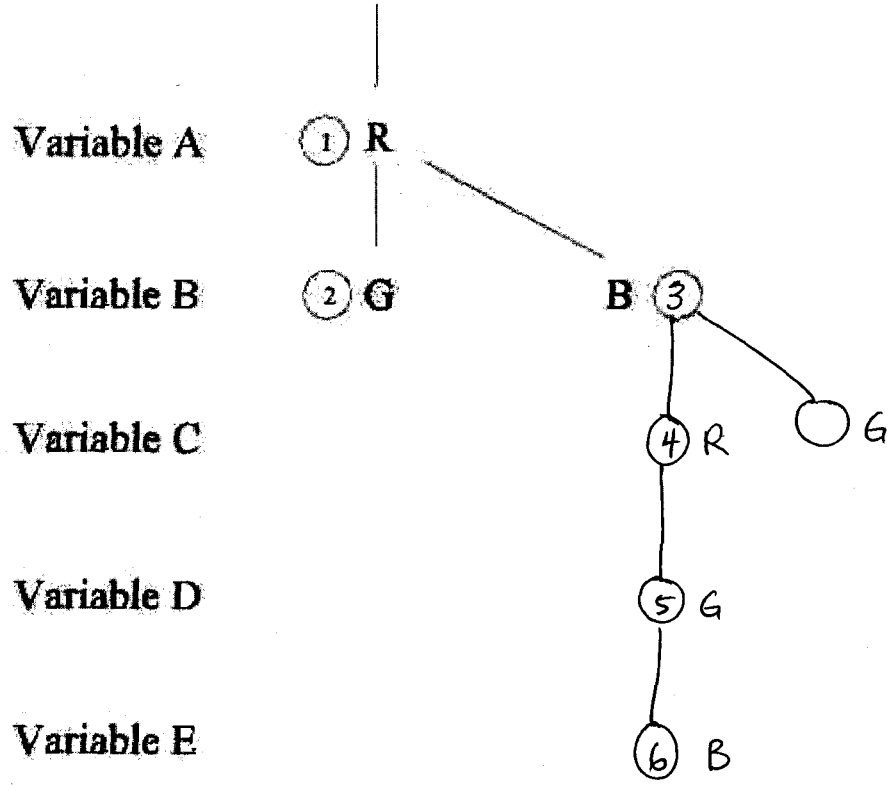
Part 4.4: Complete search (6 points)

In what follows, assume constraints between variables are as previously specified.

You are to find one of the valid solutions for this problem, starting from the beginning, with no assignments made, using backtracking with forward checking, checking immediate neighbors only.

You are to assume that a variable assignment is considered, forward constraints are checked, and then, if ok, the variable assignment then is made. No assignment is considered if it conflicts with a previous assignment.

Examine variables in alphabetical order (A, B, C, D, E) and values in reverse alphabetical order (R, G, B). **Draw the search tree below**; we have started the tree for you. For every node in your tree, draw only its valid descendents. **Number each node** to show when a variable value is considered; draw a node's descendents when a variable assignment is made.



Part 4.5: Existence of solution (2 points)

If you allow four colors, will immediate neighbors, forward checking always find a solution, given any constraint network consisting of variables and connecting lines? Circle your answer:

yes (no)

Part 4.6: Ordering (2 points)

Given any constraint network consisting of variables and connecting lines, is it always best to order the variable assignments such that the variables with more connections are labeled before those with fewer connections? Circle your answer:

yes (no)

Question 5: Miscellaneous (10 points)

A certain deduction rule system has no stop rule. It is observed to run without stopping, creating an ever-growing set of assertions. You can always make such a system stop by:

- Randomly selecting which rule to fire next
- Changing the ordering of rules in the rules database
- Changing the ordering of assertions in the assertions database
- All of the above
- ⊙ None of the above

Solving the general Air Travel Planning problem requires at most:

- Constant time
- Polynomial time
- Exponential time and polynomial space
- Exponential time and exponential space
- ⊙ It can be unsolvable

Airlines have complicated pricing and booking systems because:

- Pricing and booking air travel is provably hard
- Federal, state, and international regulations interact badly
- ⊙ Economists believe flat rates aren't profitable enough
- They hate you
- All of the above
- None of the above

Which of these statements about Air Travel Planning is false?

- A fare may be used to pay for many different flights
- A flight may be paid for using many different fares
- A priceable unit is composed of one or several fare components
- A fare may prohibit using certain airlines on connecting flights
- All of the above
- ⊙ None of the above

Progressive deepening, also known as iterative deepening, is useful for:

- ⊙ Performing search under a strict time-limit
- Reducing the space used by breadth-first search
- Taking maximum advantage of alpha-beta pruning
- Searching trees with a very low branching factor
- All of the above
- None of the above