CS-171, Intro to A.I. — Mid-term Exam — Fall Quarter, 2015

YOUR NAME: ____

YOUR ID: __________ ID TO RIGHT: __________ ROW: ________ SEAT: ________

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

When you are told to begin the exam, please check first to make sure that you have all ten pages, as numbered 1-10 in the bottom-right corner of each page. We wish to avoid copy problems. We will supply a new exam for any copy problems.

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

<u>Please turn off all cell phones now. No electronics are allowed at any point of the exam.</u>

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 <u>turn it in with your exam</u>.

<u>This page summarizes the points for each question, so you can plan your time.</u>

1. (15 pts total) Problem Solving by search.

2. (15 pts total, -1 for each error, but not negative) A* search showing queue.

3. (10 pts total, -1 for each error, but not negative) Conversion to CNF.

4. (10 pts total, -1 for each error, but not negative) Resolution Theorem Proving.

5. (10 pts total, -1 for each error, but not negative) Mini-Max, Alpha-Beta Pruning.

6. (10 pts total, -1 pt each wrong answer, but not negative) Search Properties.

7. (15 pts total, 5 pts each, -1 each error, but not negative) Bayesian Networks.

8. (10 pts total, 1 pt each) Adversarial (Game) Search Concepts.

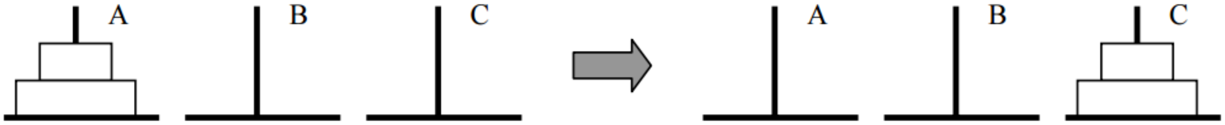
9. (5 pts total, -2 pts for each error, but not negative) Task Environment.

<u>The Exam is printed on both sides to save trees! Work both sides of each page!</u>

1. (15 pts total) Problem Solving by search.

We are now trying to solve the miniature of Tower of Hanoi problem. There are 3 towers (A, B, C), and 2 disks (small one and large one). The purpose of this problem is to move both disks from the tower A to tower C (as illustrated in the figure below), subject to following two conditions:

- You can move only one disk at a time.
- You can move only the top disk in a stack of disks.
- You cannot put the large disk on top of the small disk.



The possible states can be denoted as follows (A = tower 1, B = tower 2, C = tower 3):

a:(b c) where a is the state number, b is the tower number for the large disk, and c is the tower number for the small disk. E.g., 3:(1 3) implies that in state 3 the large disk is on tower 1 and the small disk is on tower 3.

If we use this notation, the following nine states are possible.

1:(1 1), 2:(1:2), 3:(1 3), 4:(2 1), 5:(2 2), 6:(2 3), 7:(3 1), 8:(3 2), 9:(3 3)

1.a (4 pts total, 2 pts each) Which states are the initial state and goal state?

1.a.1 (2 pts) Initial state: _______________

1.a.2 (2pts) Goal state: ________________

1.b. (6 pts total, -1 for each error, but not negative) Enumerate all one-move transitions between states.

For example, 1->2, 3 means that it is possible to make a transition from state 1 to state 2 or state 3 in one move.

The first one is done for you as an example.

1-> 2, 3

2-> _______________

3-> _______________

4-> _______________

5-> _______________

6-> _______________

7-> _______________

8-> _______________

9-> _______________

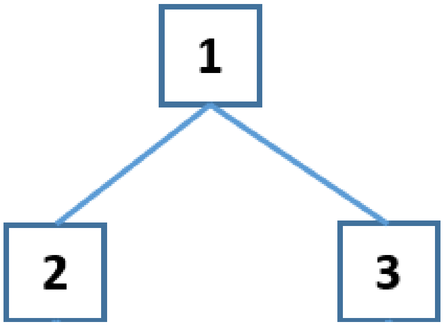
**** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ****

1.c. (5 pts total, -1 for each error, but not negative) Problem Solving by search (continued).
Find a solution using breath-first search and draw the search tree. Stop when you reach goal state 9.

Use Tree Search, i.e., do not remember visited nodes. Children of a node are returned in increasing numerical order by state; i.e., the children of a node are ordered left-to-right by increasing state number.

Assume that cycles are detected and eliminated by never expanding a node containing a state that is repeated on the path back to the root. That is, if a newly-generated child node state also occurs on the path from the parent back to the root, then that child is pruned immediately. This pruning strategy is a “light-weight” way to avoid many cycles and repeated states in Tree Search without the full memory burden of remembering all visited nodes. However, it is an approximate strategy and does not fully solve the repeated node problem.

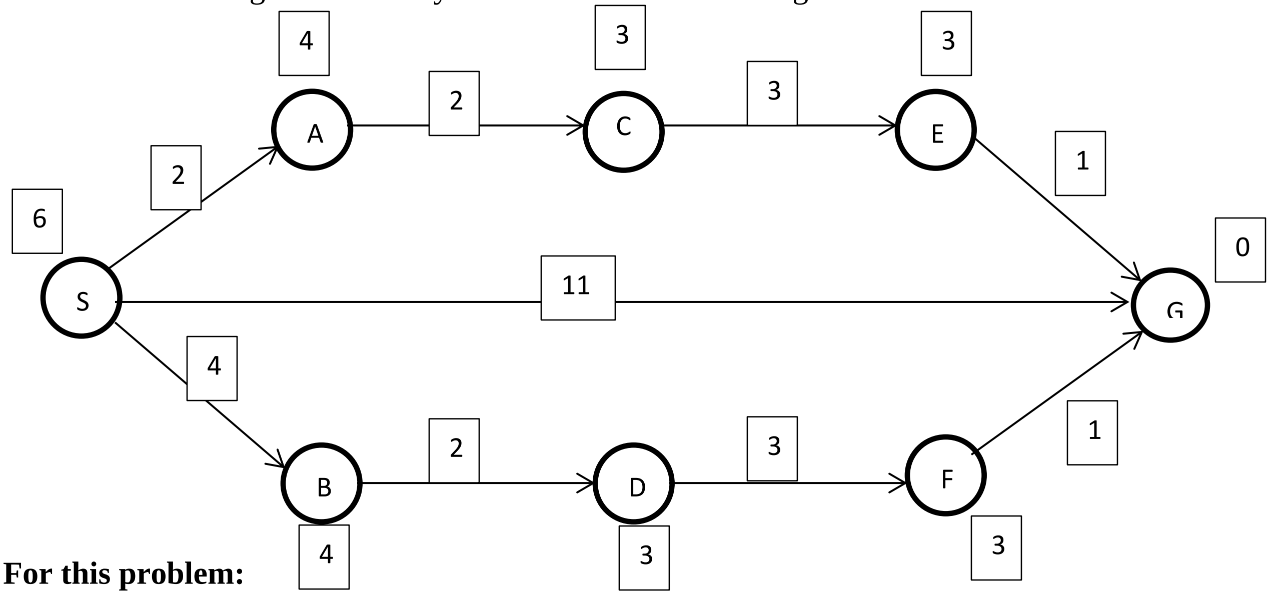
The first expansion is done for you as an example.



2. (15 pts total, -1 for each error, but not negative) A* search showing queue.

Use the A* algorithm on the graph below to find a path from node S to node G.

- Each node is labeled by a capital letter and the value of a heuristic function at that node.
- Each edge is labeled by the cost to traverse that edge.



For this problem:

- Perform the A* algorithm on this graph, filling in the table below. Indicate the f values of each node on the queue as shown in the first two rows of the table. You need not to order the contents of the (priority) queue in the table. You should not need all the lines provided in the table.
- Assume that you are doing Tree Search, i.e., do not remember visited nodes.
- Write the priority queue as <node-label>=<f-value>
- The first one is done for you, as an example.

iteration	Node expanded	Priority queue at the end of this iteration
0		S=6
1	S=6	A=6; B=8; G=11
2		
3		
4		
5		
6		
7		
8		
9		
10		

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

3. (10 pts total, -1 for each error, but not negative) Conversion to CNF.
Convert the following Propositional Logic expression to Conjunctive Normal Form (CNF).
SHOW YOUR WORK. Correct work = full credit, correct answer without work = zero credit.

$(X \Leftrightarrow (Y \wedge Z))$

4. (10 pts total, -1 for each error, but not negative) Resolution Theorem Proving.
Your Knowledge Base (KB) contains the following statements:

(¬ A)
((¬ A) <=> ((¬ B) ∧ (¬ C)))
((¬ B) => ((¬ D) ∧ (¬ E)))
((¬ C) => (D ∨ E ∨ (¬ F)))

You are asked to prove the goal statement (¬ F).

After converting your KB to CNF and adjoining the negated goal statement you have (in clausal form):

(NOT A)　　(A (NOT B))　　(A (NOT C))　　(B (NOT D))
(B (NOT E))　　(C D E (NOT F)))　　F

Produce a resolution proof that (NOT F) is true. I.e., run resolution on the statements above to yield ().
Think about your problem, and find a proof that mirrors how you think. You know that (NOT A) so it is easy to prove (AND (NOT B) (NOT C)). From (NOT B) it is easy to prove (AND (NOT D) (NOT E)). From (NOT C) it is easy to prove (OR D E (NOT F)). You just have proved (AND (NOT D) (NOT E)), so only (NOT F) remains.
The shortest proof I know if is only eight lines long (Bonus Point for a shorter proof).

Resolve ________________________with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

Resolve ________________________ with ___________________________ to produce: ____________________

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

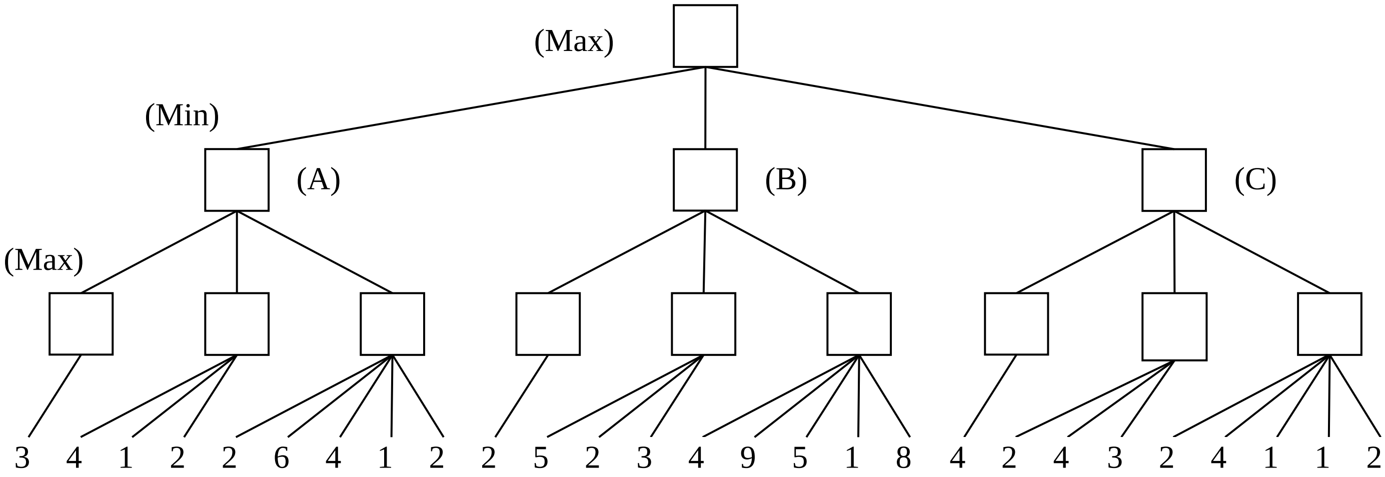
5. (10 pts total, -1 for each error, but not negative) Mini-Max, Alpha-Beta Pruning. In the game tree below it is **Max**'s turn to move. At each leaf node is the estimated score of that resulting position as returned by the heuristic static evaluator.

(1) Perform Mini-Max search and label each branch node with its value.

(2) Cross out each leaf node that would be pruned by alpha-beta pruning.

(3) What is Max's best move (A, B, or C)? __C__________

<u>Cross out</u> each leaf node that would be pruned by Alpha-Beta Pruning. Do not just draw pruning lines.



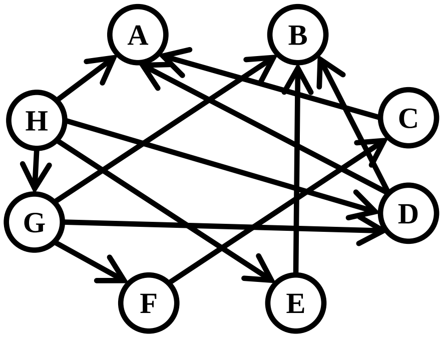
6. (10 pts total, -1 pt each wrong answer, but not negative) Search Properties.

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 use breadth-first search.

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

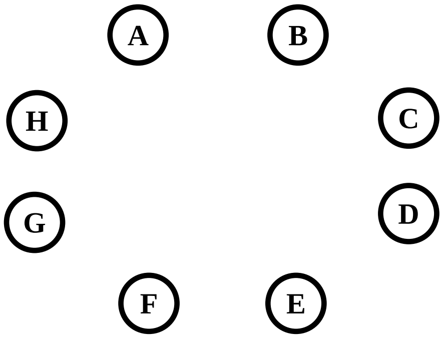
Criterion	Complete?	Time complexity	Space complexity	Optimal?
Breadth-First				
Uniform-Cost				
Depth-First				
Iterative Deepening				
Bidirectional (if applicable)				

7. (15 pts total, 5 pts each, -1 each error, but not negative) Bayesian Networks.
7.a. (5 pts) Write down the factored conditional probability expression that corresponds to the graphical Bayesian Network shown.

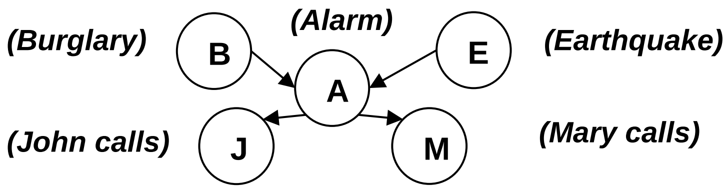


7.b. (5 pts) Draw the Bayesian Network that corresponds to this conditional probability:

P(A | B,C,E) P(B | D,G) P(C | E,F,H) P(D | G) P(E| G,H) P(F | H) P(G | H) P(H)



7.c. (5 pts) Shown below is the Bayesian network corresponding to the Burglar Alarm problem,
P(J | A) P(M | A) P(A | B, E) P(B) P(E).



P(E)
.002

A	P(M)
t	.70
f	.01

B	E	P(A)
t	t	.95
t	f	.94
f	t	.29
f	f	.001

P(B)
.001

A	P(J)
t	.90
f	.05

The probability tables show the probability that variable is True, e.g., P(M) means P(M=t).
Write down an expression that will evaluate to P(J=t ∧ M=f ∧ A=t ∧ B=f ∧ E=t). Express your answer as a series of numbers (numerical probabilities) separated by multiplication symbols. You do not need to carry out the multiplication to produce a single number (probability). **SHOW YOUR WORK.**

P (J=t ∧ M=f ∧ A=t ∧ B=f ∧ E=t)
=

8. (10 pts total, 1 pt each) ADVERSARIAL (GAME) SEARCH CONCEPTS. For each of the following terms on the left, write in the letter corresponding to the best answer or the correct definition on the right.

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

9. (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.

P_______________ E_______________ A_______________ S_______________

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