For each question on the Final Exam, “Zero” below gives the fraction of students who scored zero, “Partial” gives the fraction who got partial credit, and “Perfect” gives the fraction who scored 100%.

(The percentages and raw numbers are approximate as we may have missed recording exams while tallying, and some students had dropped or did not take the exam.)

Problem 1
Zero: 0% (~0 students), Partial: 78% (~141 students), Perfect: 22% (~39 students)

Problem 2
Zero: 0% (~0 students), Partial: 72% (~130 students), Perfect: 28% (~50 students)

Problem 3
Zero: 0% (~0 students), Partial: 77% (~138 students), Perfect: 23% (~42 students)

Problem 4
Zero: 0% (~0 students), Partial: 86% (~148 students), Perfect: 14% (~26 students)

Problem 5
Zero: 4% (~7 students), Partial: 62% (~112 students), Perfect: 13% (~23 students)
(the stats are very off for this question, probably because 3 of us graded and kept separate tallies)

Problem 6
Zero: 9% (~15 students), Partial: 34% (~62 students), Perfect: 57% (~102 students)

Problem 7
Zero: 12% (~17 students), Partial: 50% (~90 students), Perfect: 38% (~68 students)

Problem 8
Zero: 0% (~0 students), Partial: 33% (~52 students), Perfect: 67% (~121 students)

Problem 9
Zero: 0% (~0 students), Partial: 100% (~180 students), Perfect: 0% (~0 students)

Problem 10
Zero: 13% (~25 students), Partial: 68% (~123 students), Perfect: 11% (~20 students)

Problem 11
Zero: 5% (~9 students), Partial: 29% (~52 students), Perfect: 66% (~118 students)

Problem 12
Zero: 7% (~12 students), Partial: 16% (~29 students), Perfect: 74% (~134 students)
Bonus Points (shorter proof): 7% (~12 students)

CS-171, Intro to A.I. — Final Exam — Fall Quarter, 2014

YOUR NAME: ____

YOUR ID: ________ ID TO RIGHT: ________ ROW: ______ SEAT NO.: ________

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 12 pages, as numbered 1-12 in the bottom-left 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, an optional blank piece of paper (for optional scratch pad use), and an optional water bottle.

Please turn off all cell phones now.

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

1. (12 pts total, 2 pts each) k-NearestNeighbor (k-NN) and Cross-validation.
2. (4 pts total, 1 pt each) Task Environment.
3. (10 pts total, 1/2 pt each) Search Properties.
4. (10 pts total, 1 pt each) Probability Rules and Independence.
5. (14 pts total, 2 pts each) Knowledge Representation in FOPL.
6. (5 pts total) Hierarchical Agglomerative Clustering.
7. (5 pts total) k-Means Clustering.
8. (10 points total, 2 pts each) Constraint Satisfaction Problems.
9. (10 pts total, 1 pt each) State-Space Search.
10. (5 pts total, -1 for each wrong answer, but not negative) Mini-Max, Alpha-Beta Pruning.
11. (5 pts total) Bayesian Networks.
12. (10 pts total) Christmas Angel Resolution Theorem Proving in Propositional Logic.

The Exam is printed on both sides to save trees! Work both sides of each page!

1. (12 pts total, 2 pts each) k-NearestNeighbor (k-NN) and Cross-validation. Consider this training data set. Examples are A-E, the single attribute is X, and class labels are 0 or 1.

Example	A	B	C	D	E
Attribute Value (X)	0.1	0.6	0.8	2.0	3.0
Class Label	0	0	0	1	1

See Section 18.8.1

1.a. (2 pts) Using 1-NearestNeighbor, what class label would be assigned to unseen example F, which has attribute value $X_F = 0.3$? (Write 0 or 1) _0_____

1.a. F's nearest neighbor is A (class 0)

1.b. (2 pts) Using 3-NearestNeighbor, what class label would which has attribute value $X_F = 0.3$? (Write 0 or 1) _0_____

1.b. F's 3 nearest neighbors are A (class 0), B (class 0), and C (class 0)

1.c. (2 pts) Using 1-NearestNeighbor, what class label would be assigned to unseen example G, which has attribute value $X_G = 1.5$? (Write 0 or 1) _1_____

1.c. G's nearest neighbor is D (class 1)

1.d. (2 pts) Using 3-NearestNeighbor, what class label would which has attribute value $X_G = 1.5$? (Write 0 or 1) _0_____

1.d. G's 3 nearest neighbors are B (class 0), C (class 0), and D (class 1)

1.e. A is closest to B, B to C, C to B, D to E, and E to D; all are predicted correctly

1.e. (2 pts) -Validation, what is the cross-validated accuracy of write a fraction, as N/5) _5/5____

1.f. D and E are both errors, since their 3 nearest neighbors include B and C (class 0) under cross-validation

1.f. (2 pts) Cross-Validation, what is the cross-validated accuracy of Write a fraction, as N/5) _3/5____

2. (4 pts total, 1 pt each) Task Environment. Your book defines a task environment as a set of four things, with acronym PEAS. Fill in the blanks with the names of the P

See Section 2.3.1

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

3. (10 pts total, 1/2 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 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

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

See Chapter 13.

4. (10 pts total, 1 pt each) Probability Rul
Consider the following full joint distribution

You do not need to add up the numbers to produce a single numerical result. It is sufficient for you to write an arithmetical expression that will evaluate to the correct numerical result.

A	*B*	*C*	*P(a,b,c)*
t	*t*	*t*	*0.03*
t	*t*	*f*	*0.12*
t	*f*	*t*	*0.17*
t	*f*	*f*	*0.18*
f	*t*	*t*	*0.03*
f	*t*	*f*	*0.12*
f	*f*	*t*	*0.24*
f	*f*	*f*	*0.11*

4.a. P(A=f) = P(A=f,B=t,C=t)+P(f,t,f)+P(f,f,t)+P(f,f,f)
= 0.03+0.12+0.24+0.11 = 0.50

4.b. P(B=t) = P(A=t,B=t,C=t)+P(t,t,f)+P(f,t,t)+P(f,t,f)
= 0.03+0.12+0.03+0.12 = 0.30

4.c. P(B=t, C=t) = P(A=t,B=t,C=t)+P(f,t,t)
= 0.03+0.03 = 0.06

4.d. P(A=f, C=t) = P(A=f,B=t,C=t)+P(f,f,t)
= 0.03+0.24 = 0.27

Calculate the following probabilities (writ

4.e. P(A=t | B=t) = P(A=t,B=t)/P(B=t)
P(A=t,B=t) = P(A=t,B=t,C=t)+P(t,t,f) = 0.03+0.12=0.15
P(B=t) = P(A=t,B=t,C=t)+P(t,t,f)+P(f,t,t)+P(f,t,f)
= 0.03+0.12+0.03+0.12 = 0.30
P(A=t,B=t)/P(B=t) = 0.15/0.30 = 0.50

4.a. (1 pt) *P(A = f)* = ______0.50___

4.b. (1 pt) *P(B = t)* = ______0.30___

4.f. P(C=f | B=t) = P(B=t,C=f)/P(B=t)
P(B=t,C=f) = P(A=t,B=t,C=f)+P(f,t,f) = 0.12+0.12=0.24
P(B=t) = 0.30 (above, 2.b)
P(B=t,C=f)/P(B=t) = 0.24/0.30 = 0.80

4.c. (1 pt) *P(B = t, C = t)* = ________0.06

4.d. (1 pt) *P(A = f, C = t)* = ________0.27

4.g. P(A=t,B=t) = P(A=t,B=t,C=t) + P(t,t,f) = 0.03+0.12 = 0.15
P(A=t) = 1 - P(A=f) = 1 - 0.50 = 0.50 (above, 2.a)
P(B=t) = 0.30 (above, 2.b)
0.15 = P(A=t,B=t) = P(A)*P(B) = 0.50*0.30 = 0.15; Yes, independent

4.e. (1 pt) *P(A = t | B = t)* = ________0.50

4.f. (1 pt) *P(C = f | B = t)* = ________0.80___

4.g. (1 pt) Are A and B independent of each other? (Y=Yes, N=No): _____Y_____

4.h N____

4.h. P(B=t,C=t) = 0.06 (above, 2.c)
P(B=t) = 0.30 (above, 2.b)
P(C=t) = P(A=t,B=t,C=t)+P(t,f,t)+P(f,t,t)+P(f,f,t)
= 0.03+0.17+0.03+0.24 = 0.47
0.06 = P(B=t,C=t) /= P(B)*P(C) = 0.30*0.47 = 0.141; No, not independent

4.i : _____N_____

4.j): _____Y_____

4.i. P(B=t,C=t|A=t) = P(A=t,B=t,C=t) / P(A=t) (definition)
P(A=t,B=t,C=t) = 0.03
P(A=t) = 1 - P(A=f) = 1 - 0.50 = 0.50 (above, 2.a)
P(B=t |A=t) = P(A=t,B=t) / P(A=t) (definition)
P(A=t,B=t) = P(A=t,B=t,C=t)+P(t,t,f) = 0.03+0.12 = 0.15
P(C=t |A=t) = P(A=t,C=t) / P(A=t) (definition)
P(A=t,C=t) = P(A=t,B=t,C=t)+P(t,f,t) = 0.03+0.17 = 0.20
0.06 = 0.03/0.50 = P(A=t,B=t,C=t) / P(A=t) = P(B=t,C=t|A=t)
/= P(B=t |A=t)*P(C=t|A=t) = [0.15/0.5]*[0.20/0.5] = 0.12
No, B and C are not conditionally independent given A.

4.j. P(A=t,C=t|B=t) = P(A=t,B=t,C=t) / P(B=t) (definition)
P(A=t,B=t,C=t) = 0.03
P(B=t) = 0.30 (above, 2.b)
P(A=t |B=t) = 0.50 (above, 2.e)
P(C=t |B=t) = P(B=t,C=t) / P(B=t) (definition)
P(B=t,C=t) = P(A=t,B=t,C=t)+P(f,t,t) = 0.03+0.03= 0.06
0.10 = 0.03/0.30 = P(A=t,B=t,C=t) / P(B=t) = P(A=t,C=t|B=t)
= P(A=t |B=t)*P(C=t|B=t) = [0.50]*[0.06/0.3] = 0.10
Yes, A and C are conditionally independent given B.

5. (14 pts total, 2 pts each) Knowledge Representation in FOPL. Consider a vocabulary with the following symbols:

Occupation(p, o) : Predicate. Person *p* has occupation *o*.
Customer(p1, p2) : Predicate. Person *p1* is a customer of person *p2*.
Boss(p1, p2) : Predicate. Person *p1* is a boss of person *p2*.
Doctor, Surgeon, Lawyer, Actor : Constants denoting occupations.
Emily, Joe : Constants denoting people.

Use these symbols to write the following assertic

See Chapter 8. This problem is Exercise 8.10, R&N, p. 317.

It is OK if you use variables *x, y, z,* or any other variable names you like, instead of those shown here. Indeed, a real KB would standardize the variables apart anyway, by using different variable names in each sentence.

5.a. (2 pts) Emily is either a surgeon or a lawyer.

$Occupation(Emily, Surgeon) \vee Occupation(Emily, Lawyer)$

or

$Occupation(Emily, Surgeon) \Leftrightarrow \neg Occupation(Emily, Lawyer)$

This question was taken without change from Exercise 8.10.a, page 317, in your textbook. The textbook authors perhaps intended the phrase "either ... or" to indicate the exclusive OR; but after reflection I agree that it is ambiguous, and that either parse should be correct. <u>You received full credit whether you assumed the inclusive, or the exclusive, OR.</u>

The most common error for this question was to use '*Doctor*, *Surgeon*, *Lawyer*, *Actor*' as predicate symbols, even though the problem clearly stated that they were constant symbols to be used as an argument in the *Occupation* predicate. For example, a common error was:

$Occupation(Surgeon(Emily)) \vee Occupation(Lawyer(Emily))$

Not only does this error use the constant symbols *Surgeon/Lawyer* as predicate symbols, it also violates the arity of *Occupation*, which is 2 as stated in the problem but is 1 as in the error above. Furthermore, if *Surgeon/Lawyer* are treated as predicate symbols, then their result is a truth value T/F, and a truth value T/F is not a valid argument to the *Occupation* predicate.

5.b. (2 pts) Joe is an actor, but he holds another job.

$Occupation(Joe, Actor) \wedge \exists\, o\ [Occupation(Joe, o) \wedge \neg\ (o = Actor)]$

or

$Occupation(Joe, Actor) \wedge [\, Occupation(Joe, Doctor) \vee Occupation(Joe, Surgeon)$
$\vee\, Occupation(Joe, Lawyer)\ ]$

The most common error for this question was to forget to specify that Joe's other *Occupation* was not an *Actor*. For example, a common error was:

$Occupation(Joe, Actor) \wedge \exists\, o\ [Occupation(Joe, o)]$

which is true if $o=Actor$; i.e., it does not require Joe to hold <u>another</u> job, but only to hold his original job..

5.c. (2 pts) All surgeons are doctors.

$\forall p\ [Occupation(p, Surgeon) \Rightarrow Occupation(p, Doctor)]$

The most common error for this question was to use '*Doctor, Surgeon*' as predicate symbols, even though the problem clearly stated that they were constant symbols to be used as an argument in the *Occupation* predicate. For example, a common error was:

$\forall p\ Surgeon(p) \Rightarrow Doctor(p)$

5.d. (2 pts) Joe does not have a lawyer (i.e., Joe is not a customer of any lawyer).

$\forall p\ [Occupation(p, Lawyer) \Rightarrow \neg\ Customer(Joe, p)]$

or

$\neg\ \exists\ p\ [Occupation(p, Lawyer) \wedge Customer(Joe, p)]$

or

$\forall p\ [Customer(Joe, p) \Rightarrow \neg\ Occupation(p, Lawyer)]$

It is easy to prove that these three formulae are all equivalent to each other.

The most common error for this question was to use conjunction instead of implication. For example, a common error was:

$\forall p\ [\neg\ Customer(Joe, p) \wedge Occupation(p, Lawyer)]$

but this statement asserts that Joe is not a customer of anyone and everyone is a lawyer; obviously false.

Another common error was to use *Lawyer* as a predicate, even though, as above, it is clearly designated to be a constant symbol and *Customer* does not accept truth values T/F as arguments. A common error was:

$\forall p1\ \forall p2\ \ [\ (Joe = p1) \wedge (\neg\ Customer(p1, Lawyer(p2))]$

but this statement is not even a grammatical well-formed-formula (wff), as noted above.

As well, if you have universally quantified x and then assert "$(Joe = x)$"for further operations on x to mean that they pertain to *Joe*, then you might as well just avoid the universal quantification on x and simply use "*Joe*" alone as a constant symbol instead of x wherever x occurs. To universally quantify x and assert "$(Joe = x)$" for operations on x is not an error, but it is inelegant; replace it by operations on *Joe*.

5.e. (2 pts) Emily has a boss who is a lawyer.

$\exists\ p\ [Boss(p, Emily) \wedge Occupation(p, Lawyer)]$

The most common error for this question was to use '*Lawyer*' as a predicate symbol, even though the problem clearly stated that it was a constant symbol to be used as an argument in the *Occupation* predicate. For example, a common error was:

$\exists\ p\ [Boss(p, Emily) \wedge Occupation(Lawyer(p))]$

5.f. (2 pts) There exists a lawyer all of whose clients are doctors (i.e., all of whose customers are doctors).

$$\exists\ p1\ \forall\ p2\ Occupation(p1, Lawyer) \land [Customer(p2, p1) \Rightarrow Occupation(p2, Doctor)]$$

or

$$\exists\ p1\ Occupation(p1, Lawyer) \land [\forall\ p2\ Customer(p2, p1) \Rightarrow Occupation(p2, Doctor)]$$

The most common error for this question was to reverse ∧ and ⇒. A common error was:

$$\exists\ p1\ \forall\ p2\ Occupation(p1, Lawyer) \Rightarrow [Customer(p2, p1) \land Occupation(p2, Doctor)]$$

But this statement will be true for p1=anything that is not a lawyer (because the antecedent is false).

5.g. (2 pts) Every surgeon has a lawyer (i.e., every surgeon is a customer of a lawyer).

$$\forall\ p1\ \exists\ p2\ Occupation(p1, Surgeon) \Rightarrow [Customer(p1, p2) \land Occupation(p2, Lawyer)]$$

or

$$\forall\ p1\ Occupation(p1, Surgeon) \Rightarrow [\exists\ p2\ Customer(p1, p2) \land Occupation(p2, Lawyer)]$$

The most common error for this question was to reverse ∧ and ⇒. A common error was:

$$\forall\ p1\ \exists\ p2\ Occupation(p1, Surgeon) \land [Customer(p1, p2) \Rightarrow Occupation(p2, Lawyer)]$$

But this statement asserts that everything in the world is a surgeon; obviously false.

****

PLEASE REMEMBER:

- * The natural connective for ∀ is ⇒.
- * The natural connective for ∃ is ∧.

These natural connectives are explained in the text and lecture notes, which please review. Also, please think about it critically, and understand why this circumstance is the case.

****** TURN PAGE OVER. EXAM CONTINUES ON THE REVERSE ******

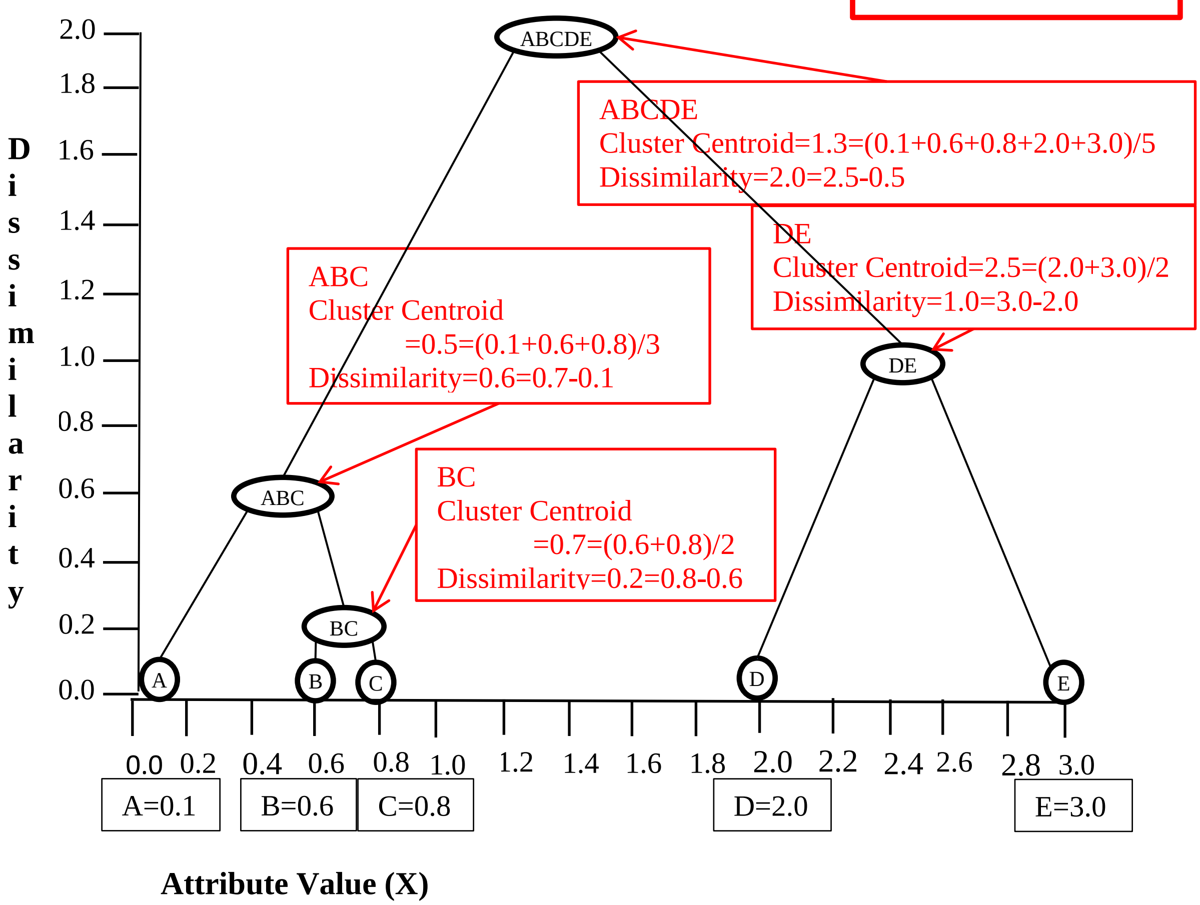
6. (5 pts total) Hierarchical Agglomerative Clustering. Consider this training data set (it is the same as in problem 1). Examples are A-E, the single attribute is X, and class labels are 0 or 1.

Example	A	B	C	D	E
Attribute Value (X)	0.1	0.6	0.8	2.0	3.0
Class Label	0	0	0	1	1

Draw the dendogram (clustering tree) that results from applying hierarchical agglomerative clustering to this data. When two clusters are merged, replace them with their cluster centroid, i.e., the statistical mean of all cluster members. This rule means, (1) each cluster is represented by its cluster centroid which is the numerical mean (average) of all of its cluster members; and (2) dissimilarity between clusters is computed as the distance between their cluster centroids using Euclidean distance. **(Note: A better measure of dissimilarity is the root-mean-squared-deviation [RMSD] of each cluster member from its cluster centroid; but that is infeasible in an exam like this.)** Label the cluster centroids by drawing an oval around the data points that are included in that cluster centroid. The first one is done for you as an example.

You are only obliged draw the clustering tree (dendogram) tha
to write in the Cluster Centroid and Dissimilarity information shown i
which is provided only for your information about how to work the pr

It is also OK to draw the tree rectangularly, e.g., as shown in the class lecture notes.



Full credit for each sub-problem if your answers for it would be correct given your answers for previous sub-problems, even if prevuious sub-problems were wrong.

7. (5 pts total) k-Means Clustering. Consider this problems 1 and 6). Examples are A-E, the single at

Example	A	B	C	D	E
Attribute Value (X)	0.1	0.6	0.8	2.0	3.0
Class Label	0	0	0	1	1

Apply k-Means Clustering to this data set for k=2, i.e., you will produce two data clusters.

7.a. (1 pt) You have randomly chosen two data points with which to initialize your two clusters. Randomly, you chose example A to initialize cluster #1 and example B to initialize cluster #2.

Write down the cluster assignments that result. Write C, D, and E in the blanks below according to which cluster they are assigned (A and B are already assigned).

cluster #1: ___A____________ cluster #2: ___B, C, D, E___

C, D, and E are all closer to B than to A.

7.b. (1 pt) After assigning examples to clusters in 7.a, you recompute the cluster centroids (means) to be the mean (average) of the examples currently assigned to each cluster.

For each cluster, w

cluster #1 contains only A, so its centroid is A's X.

he new

cluster #2 centroid is the mean X of B, C, D, & E.

of the examples that were assig above.

cluster #1: ___0.1____________ cluster #2: ___1.6 = (0.6+0.8+2.0+3.0)/4___

7.c. (1 pt) After recomputing the cluster centroids (means) in 7.b, you reassign the examples to the clusters to which they are closest (i.e., the example is assigned to the closest cluster centroid).

Write down the cluste

A, B, and C are closer to 0.1 than to 1.6.

lt. Write A, B, C,

D and E are closer to 1.6 than to 0.1.

blanks below according to wh signed.

cluster #1: ___A, B, C____________ cluster #2: ___D, E____________

7.d. (1 pt) After assigning examples to clusters in 7.c, you recompute the cluster centroids (means) to be the mean (average) of the examples currently assigned to each cluster.

For each cluster

cluster #1 new centroid is the mean X of A, B, & C.

the new cluste

cluster #2 new centroid is the mean X of D & E.

examples that were ass above.

cluster #1: ___0.5 = (0.1+0.6+0.8)/3___ cluster #2: ___2.5 = (2.0+3.0)/2____________

7.e. (1 pt) After recomputing the cluster centroids (means) in 7.d, you reassign the examples to the clusters to which they are closest (i.e., the example is assi).

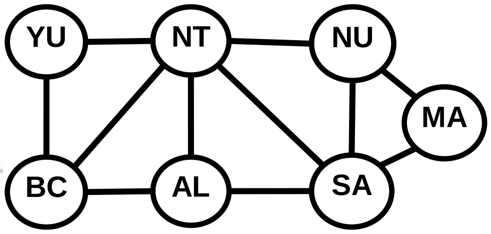
Write down the cluster assignments that result. Wr blanks below according to which cluster they are assigned

cluster membership doesn't change, so the clustering process is quiescent and terminates.

cluster #1: ___A, B, C____________ cluster #2: ___D, E____________

****** TURN PAGE OVER. EXAM CONTINUES ON THE REVERSE ******

8. (10 points total, 2 pts each) CONSTRAINT SATISFACTION PROBLEMS. See Chapter 6.



AL = Alberta
BC = British Columbia
MA = Manitoba
NT = Northwest Territories
NU = Nunavut
SA = Saskatchewan
YU = Yukon

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

8.a. (2 pts) FORWARD CHECKING. Cross out all values that would be See section 6.3.2. Forward Checking, after variable NT has just been assigned value G, as shown:

AL	BC	MA	NT	NU	SA	YU
R ~~G~~ B	R ~~G~~ B	R G B	G	R ~~G~~ B	R ~~G~~ B	R ~~G~~ B

8.b. (2 pts) ARC CONSISTENCY.
AL and MA have been assigned values, but no constraint propagation See section 6.2.2.
Cross out all values that would be eliminated by Arc Consistency (AC-3 in your book).

AL	BC	MA	NT	NU	SA	YU
B	~~R~~ G ~~B~~	R	R ~~G~~ ~~B~~	~~R~~ ~~G~~ B	~~R~~ G ~~B~~	~~R~~ ~~G~~ B

8.c. (2 pts) MINIMUM-REMAINING-VALUES HEURISTIC. Consider the assignment below. YU is assigned and constraint propagation has been done. List all unassigned variables that might be selected by the Minimum-Remaining-Values (M See section 6.3.1.
BC, NT .

AL	BC	MA	NT	NU	SA	YU
R G B	G B	R G B	G B	R G B	R G B	R

8.d. (2 pts) DEGREE HEURISTIC. Consider the assignment below. (It is the same assignment as in problem 8.c. above.) YU is assigned and constraint propagation has been done. List all unassigned variables that might be selected by the See section 6.3.1.
Heuristic: **NT, SA** .

AL	BC	MA	NT	NU	SA	YU
R G B	G B	R G B	G B	R G B	R G B	R

8.e. (2 pts) MIN-CONFLICTS HEURISTIC. Consider the complete but inconsistent assignment below. AL has just been selected to be assigned a new valu See section 6.4. search for a complete and consistent assignment. What new value woul below for AL by the Min-Conflicts Heuristic? **G** .

AL	BC	MA	NT	NU	SA	YU
?	B	G	R	G	B	B

9. (10 pts total, 1 pt each) State-Space Search. Execute Tree Search through this graph (do not

Note that for DFS, BFS, DLS, and IDS, the goal test is done when the child node is generated; see Figs. 3.11 & 3.17. In Fig. 3.11 the goal test is done before the node is inserted into the frontier. In Fig. 3.17 the goal test is done before the test of limit=0.

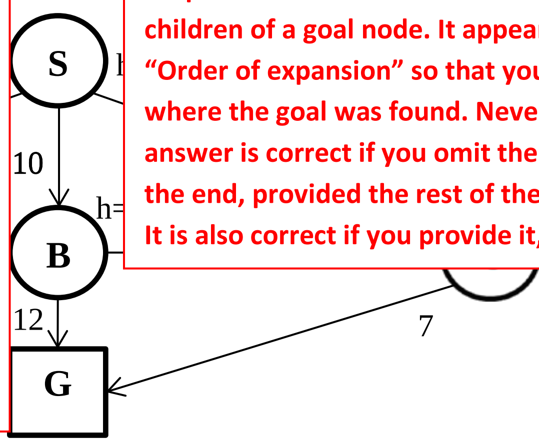
For GBFS the behavior is the same whether the goal test is done when the node is generated or when it is removed from the queue, because h(G)=0 so G is at the top of the queue anyway, and so G will be found as the next node in either case.

For UCS and A* the goal test is done when the node is removed from the queue; see Figs. 3.14 & Sec. 3.5.2. This precaution avoids finding a short expensive path before a long cheap path.

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.

poss … value … that … are A,B,C, and successors of B are G,C, in that order). … s G. For e … d **(2) the** … ound. The

Note that, technically, the goal node G is never "expanded" in the sense that we never generate children of a goal node. It appears below in the "Order of expansion" so that you may see easily where the goal was found. Nevertheless, your answer is correct if you omit the goal node G from the end, provided the rest of the answer is correct. It is also correct if you provide it, as shown below.



9.a. DEPTH-FIRST SEARCH:

See Section 3.4.3.

9.a.(1) Order of expansion: S A B G

9.a.(2) Path to goal found: S A B G Cost of path to goal: 35

9.b. BREADTH-FIRST SEARCH:

See Section 3.4.1.

9.b.(1) Order of expansion: S A B G

9.b.(2) Path to goal found: S B G Cost of path to goal: 22

9.c. ITERATIVE DEEPENING SEARCH:

See Section 3.4.5.

9.c.(1) Order of expansion: S S A B G

9.c.(2) Path to goal found: S B G

For IDS, please first review Fig. 3.17. We begin with S at depth=0; we call Recursive-DLS (RDLS), S is not a goal, limit=0, so we return without expanding any node. On depth=1, we expand S, call RDLS on A, B, & C, goal test them, then limit=0, so we return. On depth=2 we expand S, call RDLS on A, goal test A, expand A, call RDLS on B, goal test B, then limit=0, so we return; next we call RDLS on B, goal test B, expand B, call RDLS on G, goal test G, and succeed. So, the order of node expansion is S S A B G (where G is not really expanded, as discussed above, but is provided so you may see easily where it is found).

9.d. UNIFORM COST SEARCH:

See Section 3.4.2.

9.d.(1) Order of expansion: S C B G

9.d.(2) Path to goal found: S C G

9.e. GREEDY BEST FIRST SEARCH:

See Section 3.5.1.

9.e.(1) Order of expansion: S C G

9.e.(2) Path to goal found: S C G

9.f. A* SEARCH:

See Section 3.5.2.

9.f.(1) Order of expansion: S C G

9.f.(2) Path to goal found: S C G Cost of path to goal: 11

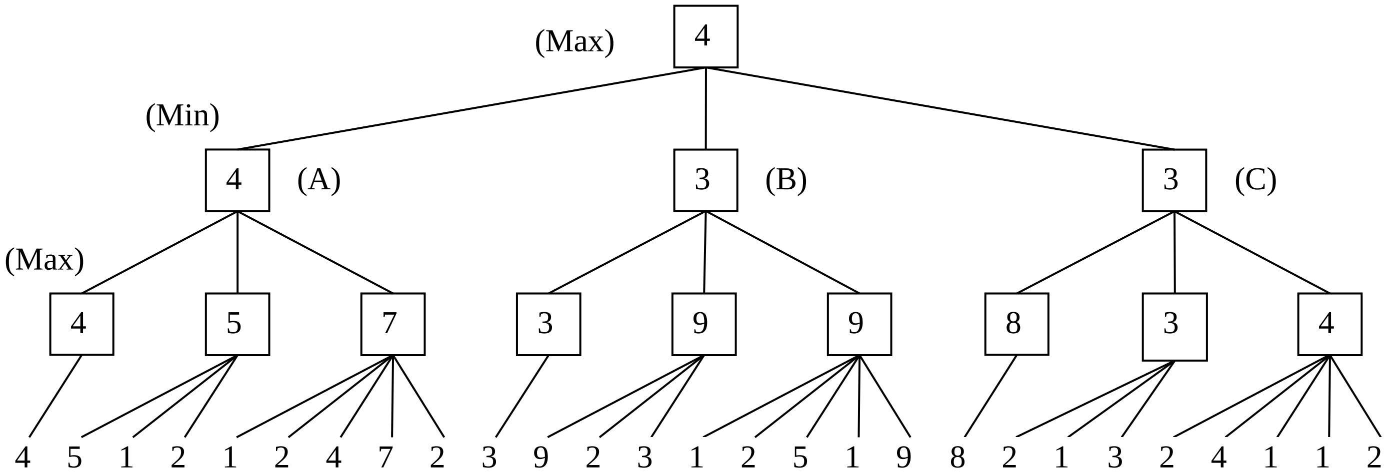
****** TURN PAGE OVER. EXAM CONTINUES ON THE REVERSE ******

10. (5 pts total, -1 for each wrong answer, 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.

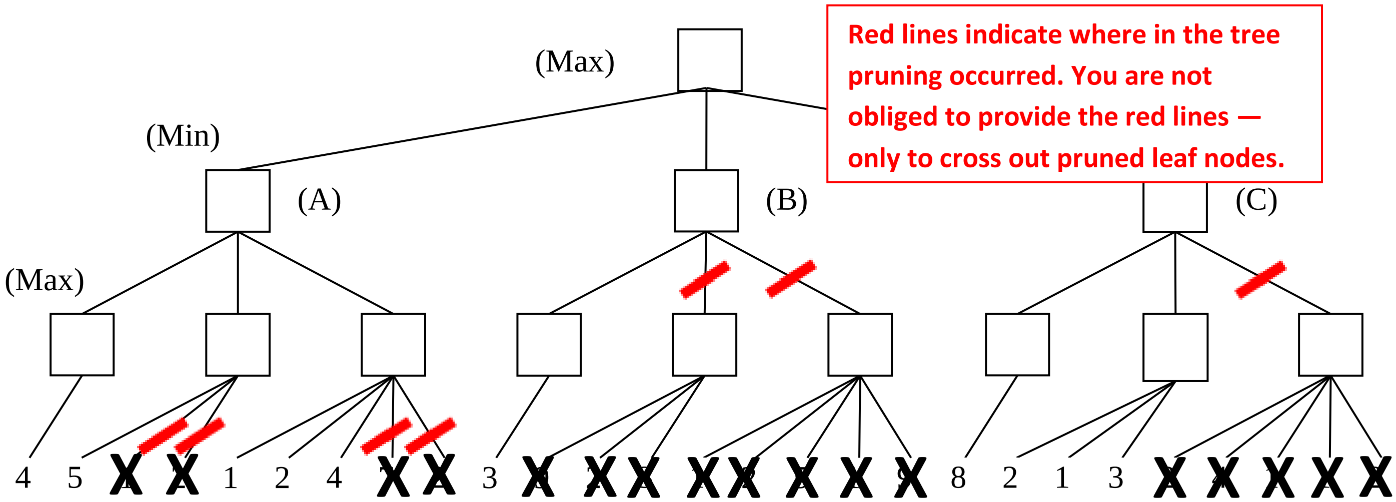
(2) What is Max's best move (A, B, or C)? A

See Section 5.2-3.



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

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.



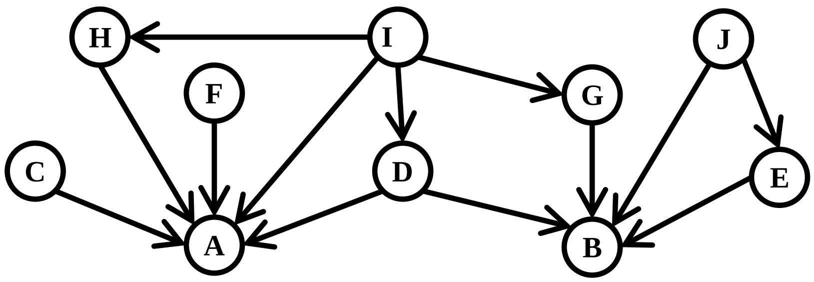
11. (5 pts total) Bayesian Networks.

11a. (1 pt) Write down the factored conditional probability expression that correspon

See Section 14.1-2.

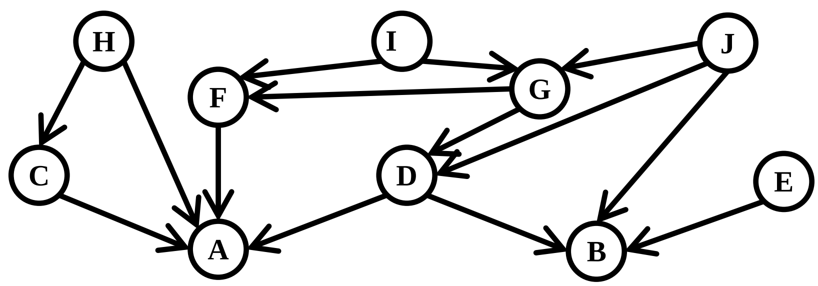
graphical Bayesian Network shown.

<u>P(A | C,D,F,H,I) P(B | D,E,G,J) P(C) P(D | I) P(E | J) P(F) P(G | I) P(H | I) P(I) P(J)</u>

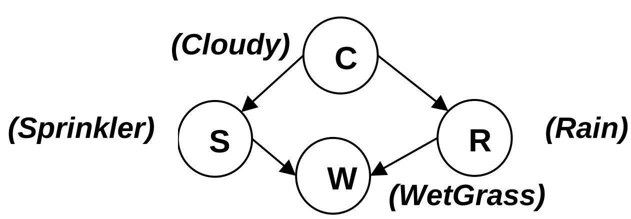


11b. (1 pt) Draw the Bayesian Network that corresponds to this conditional probability:

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



11.c. (3 pts) Below is the Bayesian network for the WetGrass problem [Fig. 14.12(a) in R&N].



P(C)
.5

C	P(S)
t	.1
f	.5

C	P(R)
t	.8
f	.2

S	R	P(W)
t	*t*	.99
t	*f*	.90
f	*t*	.90
f	*f*	.00

Write down an expression that will evaluate to P(C=f $\wedge$ R=f $\wedge$ S=t $\wedge$ W = t).
The probability tables show the probability that variable is True, e.g., P(M) means P(M=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(C=f $\wedge$ R=f $\wedge$ S=t $\wedge$ W = t)
= P(W=t | R=f $\wedge$ S=t) * P(R=f | C = f) * P(S=t | C=f) * P(C=f)
= .90 * .8 * .5 * .5

****** TURN PAGE OVER. EXAM CONTINUES ON THE REVERSE. ******

12. (10 pts total) Christmas Angel Resolution Theorem Proving in Propositional Logic.
(adapted from http://brainden.com/logic-puzzles.htm)

Several bright and clever students constructed a shorter proof than I was able to find. Two examples are:

1. Resolve (B2 B3) and (¬B3 ¬B4) to give (B2 ¬B4)
2. Resolve (¬B1 ¬B2 ¬B4) and (B2 ¬B4) to give (¬B1 ¬B4)
3. Resolve (¬B1 ¬B4) and (B1) to give (¬B4)
4. Resolve (¬B4) and (B4) to give ( )

1. Resolve (B1) and (¬B1 ¬B2 ¬B4) to give (¬B2 ¬B4)
2. Resolve (¬B2 ¬B4) and (B2 B3) to give (B3 ¬B4)
3. Resolve (B3 ¬B4) and (¬B3 ¬B4) to give (¬B4)
4. Resolve (¬B4) and (B4) to give ( )

Four angels sat on the Christmas tree amidst other ornaments. Two had blue halos and
...nia, of course our colors are blue and
...prefix form) as:

See Section 7.5.2.

...B2) B3 (¬ B4))
...B2 B3 (¬ B4))
...(¬ B2) B3 B4)))
...heir views were obscured by branches.
...and a gold halo, but I can't tell which."
...and a gold halo, but I can't tell which."

...ix form) as:
...(¬ B4)) (and (¬ B3) B4)) B4
...?"
...1)" and form the negated goal as "B1."

Your knowledge base (KB) in CNF plus negated goal (in clausal form) is:

(B1 B2 B3)	((¬ B1) (¬ B2) (¬ B3))
(B1 B2 B4)	((¬ B1) (¬ B2) (¬ B4))
(B1 B3 B4)	((¬ B1) (¬ B3) (¬ B4))
(B2 B3 B4)	((¬ B2) (¬ B3) (¬ B4))
(B2 B3)	((¬ B2) (¬ B3))
(B3 B4)	((¬ B3) (¬ B4))
B4	B1

Think about what you are trying to prove, and find a proof that mirrors how you think. You know that Angel 4 has a blue halo, so Angel 3 must have a gold halo, so Angel 2 must have a blue halo, so Angel 1 must have a gold halo.

Write a resolution proof that Angel 1 has a gold halo.

For each step of the proof, fill in the first two blanks with CNF sentences from KB that will resolve to produce the CNF result that you write in the third (resolvent) blank. The resolvent is the result of resolving the first two sentences. Add the resolvent to KB, and repeat. Use as many steps as necessary, ending with the empty clause.

The shortest proof I know of is only five lines long. (A Bonus Point for a shorter proof.)

Resolve ___B4___ with ___((¬ B3) (¬ B4))___ to produce: ___(¬ B3)___

Resolve ___(¬ B3)___ with ___(B2 B3)___ to produce: ___B2___

Resolve ___B4___ with ___((¬ B1) (¬ B2) (¬ B4))___ to produce: ___((¬ B1) (¬ B2))___

Resolve ___B2___ with ___((¬ B1) (¬ B2))___ to produce: ___(¬ B1)___

Resolve ___(¬ B1)___ with ___B1___ to produce: ___( )___

Resolve ________ with ________ to produce: ________

Resolve ________ with ________ to produce: ________

****** THIS IS THE END OF THE FINAL EXAM. HAPPY HOLIDAYS!! ******