Below, for each problem on this Midterm Exam, “Perfect” is the percentage of students who received full credit, “Partial” is the percentage who received partial credit, and “Zero” is the percentage who received zero credit.

(Due to rounding or other exceptional reasons, values below may be only approximate estimates.)

Problem 1:
Perfect: ~50% (~30 Students), Partial: ~32% (~19 Students), Zero: ~18% (~11 Students)
Problem 2:
Perfect: ~78% (~47 Students), Partial: ~20% (~12 Students), Zero: ~2% (~1 Student)
Problem 3:
Perfect: ~53% (~32 Students), Partial: ~47% (~28 Students), Zero: ~0% (~0 Students)
Problem 4:
Perfect: ~57% (~34 Students), Partial: ~43% (~26 Students), Zero: ~0% (~0 Students)
Problem 5:
Perfect: ~55% (~33 Students), Partial: ~45% (~27 Students), Zero: ~0% (~0 Students)
Problem 6:
Perfect: ~78% (~47 Students), Partial: ~22% (~13 Students), Zero: ~0% (~0 Students)
Problem 7:
Perfect: ~87% (~52 Students), Partial: ~10% (~6 Students), Zero: ~3% (~2 Students)
Problem 8:
Perfect: ~25% (~15 Students), Partial: ~68% (~41 Students), Zero: ~7% (~4 Students)
Problem 9:
Perfect: ~80% (~48 Students), Partial: ~15% (~9 Students), Zero: ~5% (~3 Students)
Problem 10:
Perfect: ~77% (~46 Students), Partial: ~20% (~12 Students), Zero: ~3% (~2 Students)
Problem 11:
Perfect: ~95% (~57 Students), Partial: ~5% (~3 Students), Zero: ~0% (~0 Students)
Problem 12:
Perfect: ~93% (~56 Students), Partial: ~7% (~4 Students), Zero: ~0% (~0 Students)
Problem 13:
Perfect: ~65% (~39 Students), Partial: ~35% (~21 Students), Zero: ~0% (~0 Students)
Problem 14:
Perfect: ~90% (~54 Students), Partial: ~10% (~6 Students), Zero: ~0% (~0 Students)

CS171, Intro to A.I. — Final Exam —Summer Quarter, 2016

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 13 pages, as numbered 1-13 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.

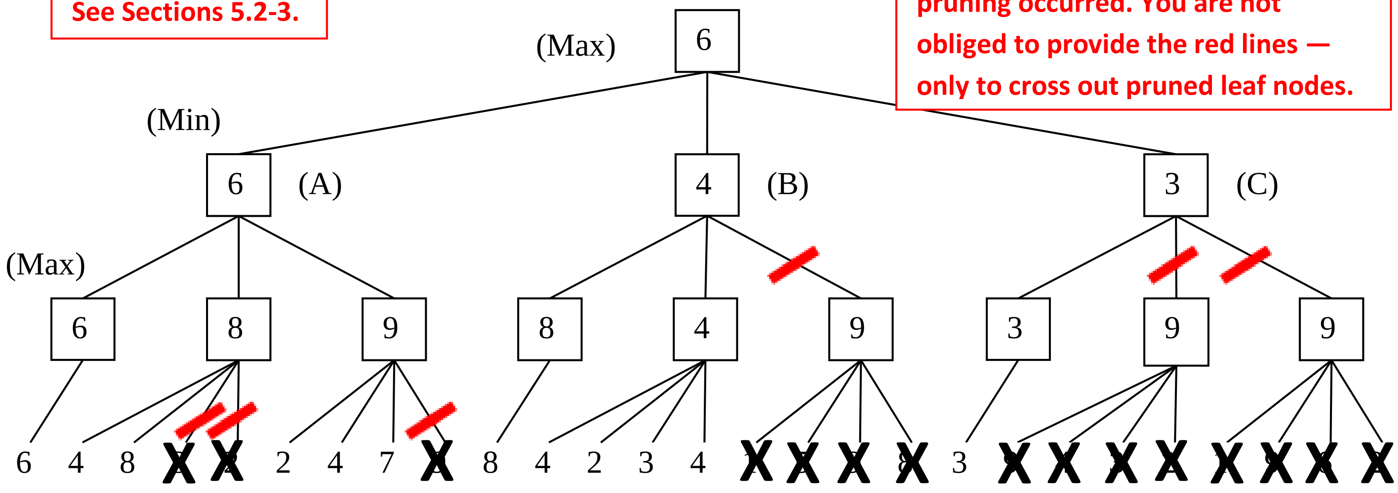
1. (5 pts total, -1 for each wrong answer, but not negative) Mini-Max, Alpha-Beta Pruning.
2. (6 pts total, 2 pts each) Machine Learning Classifier Decision Boundaries.
3. (12 pts total) Decision tree, underfitting and overfitting.
4. (10 pts total, 2 pts each) Constraint Satisfaction Problems. NorCal Counties
5. (10 pts total, 2 pts each) Execute Tree Search through this graph.
6. (10 pts total, 1 pt each) Bayesian Networks.
7. (5 pts total, -1 pt for each error, but not negative) Resolution Theorem Proving: Cake Theft.
8. (5 pts total, 1 pt each) English to FOL Conversion.
9. (3 pts total, 1 pt each) Model Complexity, Underfitting, Overfitting.
10. (8 pts total, -1 pt each wrong answer, but not negative) Search Properties.
11. (4 pts total, 1 pt each) Task Environment.
12. (12 pts total, 1 pt each) Mini-Max Game Search.
13. (6 pts total, 1 pt each) Valid, Unsatisfiable, Satisfiable.
14. Probability formulae (4 pts total, 1 pt each).

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

1. (5 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.

See Sections 5.2-3.

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.

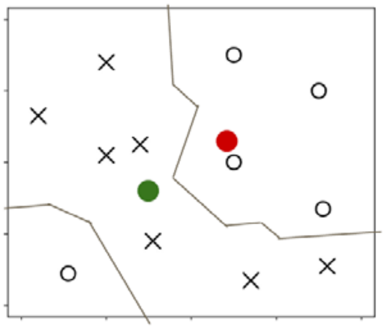


2. (6 pts total, 2 pts each) Machine Learning Classifier Decision Boundaries.
Match each decision boundary below with the classifier type most likely to produce such a decision boundary. The class means (centroids) are shown as filled circles.

2.a. (2 pts) Decision Tree Classifier. (Write A, B, or C) ________C_____

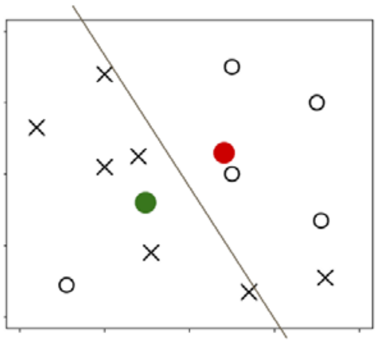
2.b. (2 pts) Minimum Distance Classifier. (Write A, B, or C) __________B_____

2.c. (2 pts) Nearest Neighbor Classifier. (Write A, B, or C) ______A_____



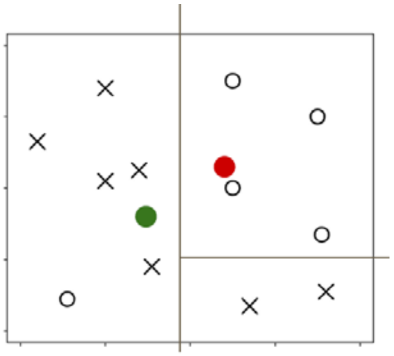
(A)

See sections 18.8.1-3.



(B)

See sections 18.6.3 and 18.7.2.



(C)

See section 18.3.

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

3. (12 pts total) Decision tree, underfitting and overfitting. Consider the following set of training examples. There are two features: A and B. The binary target variable (also known as the class label) is Y ∊ {1, 0}. You came up with a decision tree classifier that is shown in Figure 1. (It is labeled DT2 because it has two splits, as will be used below).

See section 18.3.

A	B	Y
3	3	0
3	6	0
5	5	0
7	2	0
9	7	0
9	3	0
2	7	1
7	6	1
8	5	1
10	5	1
10	7	1
11	2	1

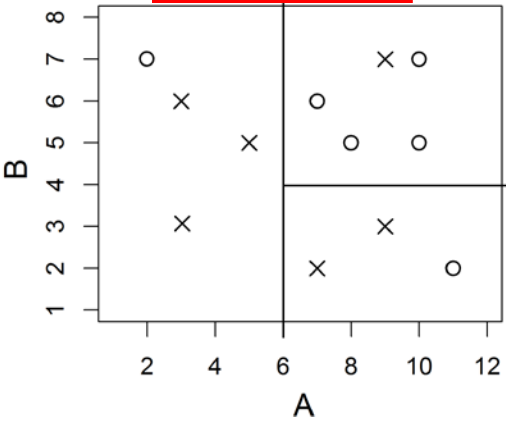
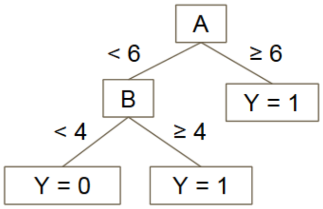


Figure 1: Decision Tree DT2 (2 splits)

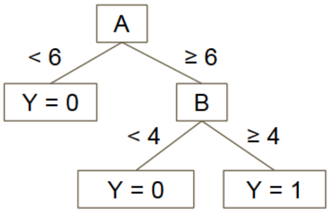
3.a. (2 pts) In Figure 1, do circles represent Y=0 or Y=1? **(Write 0 or 1)** ____1_____

3.b. (2 pts) Consider the three decision trees shown below (labeled 1, 2, and 3). Which tree below represents the same decision tree as that which is shown in Figure 1? When a leaf node is not pure, take the majority class as its label.

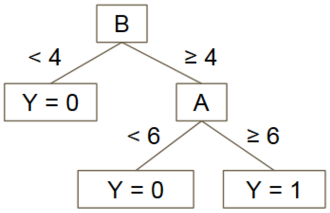
(Write 1, 2, or 3) ______2_____



(1)



(2)



(3)

3.c. (2 pts total, 1 pt each) Classify the new examples below using the decision tree shown in Figure 1. When a decision region is not pure, take the majority class as its label. Indicate your answer as a circle ○ or an X.

3.c.i. (1 pt) What class is [A = 2, B = 7] ? (write circle ○ or X) ______X_______

3.c.ii. (1 pt) What class is [A = 10, B = 2]? (write circle ○ or X) ______X_______

****** THIS PROBLEM CONTINUES ON THE NEXT PAGE ******

Note that [A = 2, B = 7] also occurs in the training data. This duplication is intentional, to reflect the real-world case whereby instances may have duplicate feature vectors but different class labels. This situation may arise for many reasons, e.g., noise in the data, feature vectors that are not sufficiently informative to distinguish classes completely, or natural variability in the world.

Your friends came up with 2 different decision trees, shown below in Figures 2 and 3. Together with Figure 1, you now have 3 decision trees. These decision trees are named by their number of splits: The decision tree in Figure 1 has 2 splits and is named DT2, the decision tree in Figure 2 has 1 split and is named DT1, and the decision tree in Figure 3 has 3 splits and is named DT3.

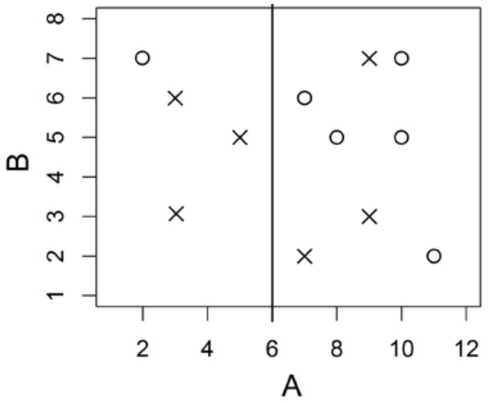


Figure 2: Decision Tree DT1 (1 split)

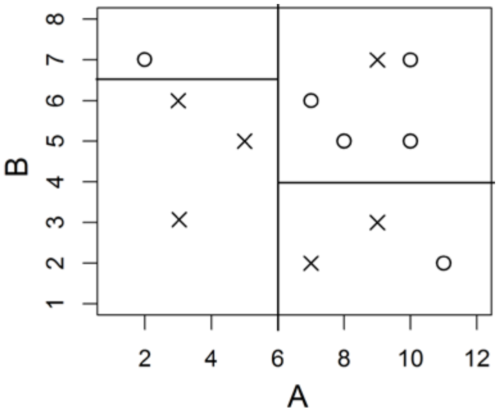


Figure 3: Decision Tree DT3 (3 splits)

In order to evaluate whose classifier performs the best, you decided to examine the training error and the test error of the 3 decision trees. Here you may simply use the fraction of misclassified data cases as "error" (eg, there are 12 training examples, so if 1 training data case is classified incorrectly, then training error = 1/12) . Two test data cases are given to you in the table below. (Assume you don't know their class label before classifying them into classes. Only use the label to examine your test error).

A	B	Y
4	7	0
7	3	0

3.d. (4 pts total, 1 pt each) Fill in the table below. Use fractions o (training set error) or N/2 (test set error). When a decision region is majority class as its label. DT1 (Figure 2) is done for you as an exam

See sections 18.3.5 and 18.4.

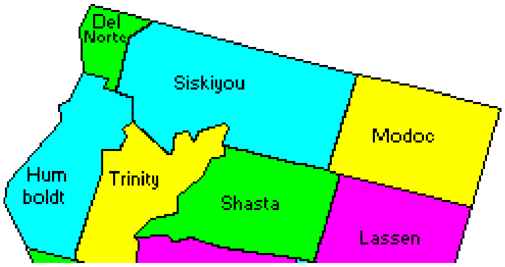
	DT1 (1 split)	DT2 (2 splits)	DT3 (3 splits)
Training error	4/12	3/12	2/12
Test error	1/2	0/2	1/2

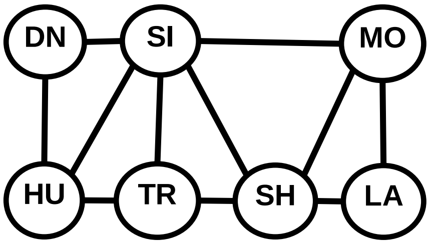
3.e. (2 pts total, 1 pt each)
Which decision tree classifier is likely overfitting? (Write DT1, DT2, or DT3) ___DT3__

Which decision tree classifier is likely underfitting? (Write DT1, DT2, or DT3) __DT1___

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

4. (10 pts total, 2 pts each) Constraint Satisfaction Problems. NorCal Counties.





DN = Del Norte
HU = Humboldt
LA = Lassen
MO = Modoc
SH = Shasta
SI = Siskiyou
TR = Trinity

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

4.a. (2pts total, -1 each wrong answer, but not negative) FORWARD CHECKING. Cross out all values that would be eliminated by Forward Checking, after variable SH has just been assigned value G as shown:

See section 6.3.2.

DN	HU	LA	MO	SH	SI	TR
R G B	R G B	R ~~G~~ B	R ~~G~~ B	G	R ~~G~~ B	R ~~G~~ B

4.b. (2pts total, -1 each wrong answer, but not negative) ARC CONSISTENCY. SH and SI have been assigned values, but no constraint propaga e. Cross out all values that would be eliminated by Arc Consistency (AC-3 in your book).

See section 6.2.2.

DN	HU	LA	MO	SH	SI	TR
R ~~G~~ ~~B~~	~~R~~ G ~~B~~	~~R~~ ~~G~~ B	R ~~G~~ ~~B~~	G	B	R ~~G~~ ~~B~~

4.c. (2pts total, -1 each wrong answer, but not negative) MINIMUM-REMAINING-VALUES HEURISTIC. Consider the assignment below. HU is assigned and constraint propagation has been done. List all unassigned variables that might be sel Minimum-Remaining-Values (MRV) Heuristic:_______**DN, SI, TR**____________________.

See section 6.3.1.

DN	HU	LA	MO	SH	SI	TR
R B	G	R G B	R G B	R G B	R B	R B

4.d. (2pts total, -1 each wrong answer, but not negative) DEGREE HEURISTIC. Consider the assignment below. (It is the same assignment as in problem 4.c above.) HU is assigned and constraint propagation has been done. List all u variables that might be selected by the Degree Heuristic:._______**SH, SI**______________

.

See section 6.3.1.

DN	HU	LA	MO	SH	SI	TR
R B	G	R G B	R G B	R G B	R B	R B

4.e. (2pts total) MIN-CONFLICTS HEURISTIC. Consider the complete but inconsistent assignment below. SI has just been selected to be assigned a new v l search for a complete and consistent assignment. What new value would be chosen below for SI by the Min-Conflicts Heuristic?.__________________**R**_____________________.

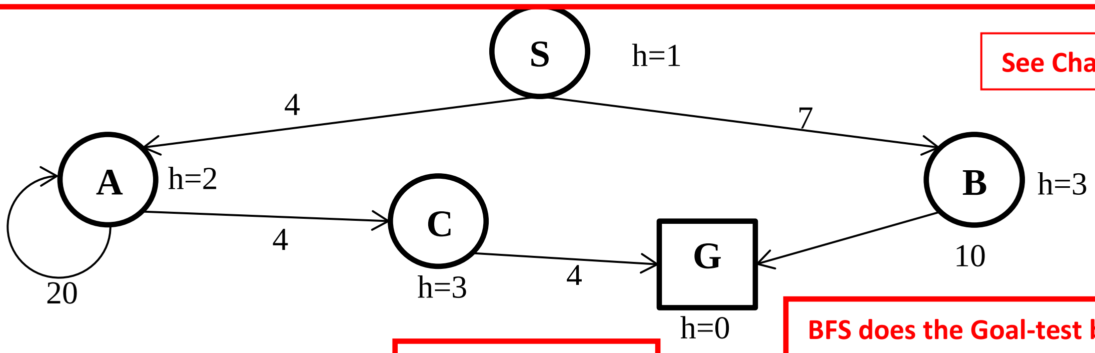
See section 6.4.

DN	HU	LA	MO	SH	SI	TR
B	G	R	G	G	**?**	B

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. (10 pts total, 2 pts each) Execute Tree Search through this graph. (Do not remember visited nodes.) Step costs are given next to each arc. Heuristic values are next to each node (as h=x). The

TECHNICAL NOTE: Technically, the goal node is not expanded, because no children of a goal node are generated. The goal node is listed in "Order of node expansion" for your convenience. Your answer is correct if you do not show the goal node in "Order of node expansion" — but it is a nicety to do so. Nevertheless, "Path found" *always* must show the goal node, because a path to a goal always must end in a goal.



See Chapter 3.

5.a. BREADTH FIRST SEARCH:

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 B is expanded.

Order of node expansion: S A B G

Path found: S B G Cost of path found: 17

5.b. (*2* pts) DEPTH FIRST SEARCH:

See Section 3.4.3 and Fig. 3.17.

DFS can get caught in loops during Tree Search (= do not remember visited nodes).

Order of node expansion: S A A A A A A etc.

Path found: None Cost of path found: None

5.c. (*2* pts) UNIFORM COST SEARCH:

UCS does goaltest when node is popped off queue.

Order of node expansion: S A B C G

See Section 3.4.2 and Fig. 3.14.

Path found: S A C G Cost of path found: 12

5.d. (*2* pts) GREEDY (BEST-FIRST) SEARCH:

A always has lower h(=2) than any other node on queue.

Order of node expansion: S A A A A A A etc.

See Section 3.5.1 and Fig. 3.23.

Path found: None Cost of path found: None

5.e. (*2* pts) ITERATED DEEPENING SEARCH:

Order of node expansion: S S A B G

IDS does the goal-test before the child is pushed onto the queue. The goal is found when B is expanded.

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

Path found: S B G 17

5.f. (*2* pts) A* SEARCH:

Order of node expansion: S A B C G

A* does goal-test when node is popped off queue.

See Section 3.5.2 and Figs. 3.24-25.

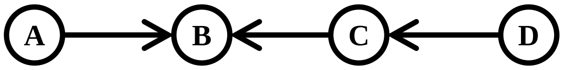
Path found: S A C G Cost of path found: 12

A question arose about IDS, above. At depth limit L=0, it simply does a goal-test of S, but does not expand any nodes. At L=1 it expands S (the first S in SSABG) to yield children A and B, goal-tests them, but does not expand them. At L=2 it expands S (the second S in SSABG) to yield children A and B; expands A to yield children A and C, goal-tests them, but does not expand them; then expands B to yield child G, and the goal-test succeeds.

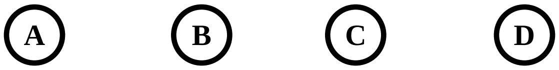
See Section 14.2.

6. (10 pts total, 1 pt each) Bayesian Networks.
Draw the Bayesian Network that corresponds to the conditional probability equation.

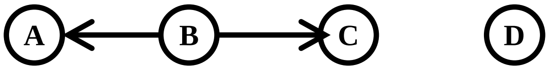
6.a. P(B|A,C) P(A) P(C|D) P(D)



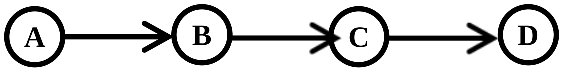
6.b. P(A) P(B) P(C) P(D)



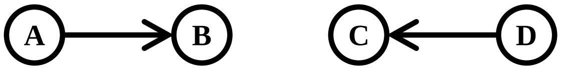
6.c. P(A|B) P(C|B) P(B) P(D)



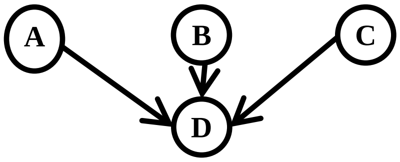
6.d. P(D|C) P(C|B) P(B|A) P(A)



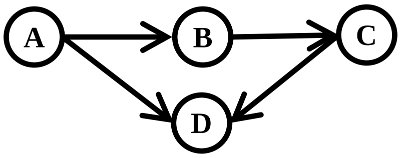
6.e. P(B|A) P(A) P(C|D) P(D)



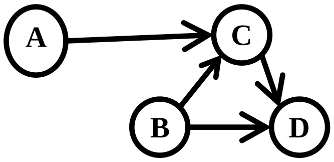
Write down the factored conditional probability equation that corresponds to the graphical Bayesian Network shown.



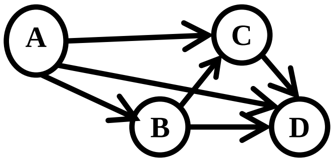
6.f. P(D|A,B,C) P(A) P(B) P(C)



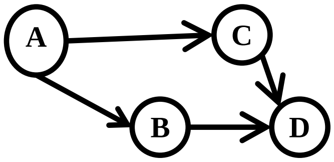
6.g. P(D|A,C) P(C|B) P(B|A) P(A)



6.h. P(D|B,C) P(C|A,B) P(B) P(A)



6.i. P(D|A,B,C) P(C|A,B) P(B|A) P(A)



6.j. P(D|B,C) P(C|A) P(B|A) P(A)

7. (5 pts total, -1 pt for each error, but not negative) Resolution Theorem Proving: Cake Theft. (http://www.brainbashers.com) Chief Inspector Parker interviewed five local burglars to identify who stole Mrs. Archer's cake.

See Section 7.5.2

It was well known that each suspect told exactly one lie:

Arnold: *It was not Edward. It was Brian.* *Brian:* *It was not Charlie. It was not Edward.*
Charlie: *It was Edward. It was not Arnold.* *Derek:* *It was Charlie. It was Brian.*
Edward: *It was Derek. It was not Arnold.*

Use these propositional variables:

A=It was Arnold. **B**=It was Brian. **C**=It was Charlie. **D**=It was Derek. **E**=It was Edward.

You translate the evidence into propositional logic (recall that each suspect told exactly one lie):

Arnold: (E ∧ B) ∨ (¬E ∧ ¬B) *Brian:* (C ∧ ¬E) ∨ (¬C ∧ E)
Charlie: (¬E ∧ ¬A) ∨ (E ∧ A) *Derek:* (¬C ∧ B) ∨ (C ∧ ¬B)
Edward: (¬D ∧ ¬A) ∨ (D ∧ A)

At most one burglar stole the cake:

(A ⇒ ¬B ∧ ¬C ∧ ¬D ∧ ¬E) (B ⇒ ¬A ∧ ¬C ∧ ¬D ∧ ¬E) (C ⇒ ¬A ∧ ¬B ∧ ¬D ∧ ¬E)
(D ⇒ ¬A ∧ ¬B ∧ ¬C ∧ ¬E) (E ⇒ ¬A ∧ ¬B ∧ ¬C ∧ ¬D)

After converting to Conjunctive Normal Form, your Knowledge Base (KB) consists of:

(E ∨ ¬B) (¬E ∨ B) (C ∨ E) (¬C ∨ ¬E) (¬E ∨ A) (E ∨ ¬A)
(¬C ∨ ¬B) (C ∨ B) (¬D ∨ A) (D ∨ ¬A)
(¬A ∨ ¬B) (¬A ∨ ¬C) (¬A ∨ ¬D) (¬A ∨ ¬E) (¬B ∨ ¬C)
(¬B ∨ ¬D) (¬B ∨ ¬E) (¬C ∨ ¬D) (¬C ∨ ¬E) (¬D ∨ ¬E)

From Brian, it was Charlie or Edward. From Derek, it was Charlie or Brian. **Thus, it was Charlie.** You will be asked to prove, **"It was Charlie."** The goal is (C). You adjoin the negated goal to your KB:

(¬C)

Produce a resolution proof, using KB and the negated goal, that "It was Charlie." 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. Continue until you produce ( ). If you cannot produce ( ), then you have made a mistake. The shortest proof I know is only three lines. It is OK to use more lines, if your proof is correct.

It is OK if you used abbreviated CNF, i.e., (¬ A ¬ B) instead of (¬ A ᵛ ¬ B). It is OK to omit the parentheses.

Resolve ______(¬B ∨ ¬E)______ and ________(C ∨ E)________ to give ________(¬B ∨ C)______.

Resolve ______(¬B ∨ C)________ and ________(C ∨ B)________ to give ________(C ∨ C) = (C) .

Resolve ______(C)____________ and ________(¬C)__________ to give ________( )____________.

Resolve ______

Resolve ______

Resolve ______

****

Other proofs are OK as long as they are correct. For example, another proof is:
Resolve (¬ C) and (C ᵛ B) to give (B).
Resolve (B) and (¬ B ᵛ ¬ E) to give (¬ E).
Resolve (¬ E) and (C ᵛ E) to give (C).
Resolve (C) and (¬ C) to give ( ).

Yet another is:
Resolve (¬ C) and (C ᵛ E) to give (E).
Resolve (¬ C) and (C ᵛ B) to give (B).
Resolve (E) and (¬ B ᵛ ¬ E) to give (¬ B).
Resolve (B) and (¬ B) to give ( ).

8. (5 pts total, 1 pt each) English to FOL Conversion. For each English sentence below, write the FOL sentence that best expresses its intended meaning. Use Person(x) for “x is a person,” Food(x) for “x is food,” and Likes(x, y) for “x likes y.”

The first one is done for you as an example.

See Section 8.2.6

8.example. “Every person likes every food.”

$$\forall x\ \forall y\ [\ \mathrm{Person}(x) \land \mathrm{Food}(y)\ ] \Rightarrow \mathrm{Likes}(x, y)$$

8.a. (1 pt) “For every food, there is a person who likes that food.”

$$\forall y\ \exists x\ \mathrm{Food}(y) \Rightarrow [\ \mathrm{Person}(x) \land \mathrm{Likes}(x, y)\ ]$$

8.b. (1 pt) “There is a person who likes every food.”

$$\exists x\ \forall y\ \mathrm{Person}(x) \land [\ \mathrm{Food}(y) \Rightarrow \mathrm{Likes}(x, y)\ ]$$

8.c. (1 pt) “Some person likes some food.”

$$\exists x\ \exists y\ \mathrm{Person}(x) \land \mathrm{Food}(y) \land \mathrm{Likes}(x, y)$$

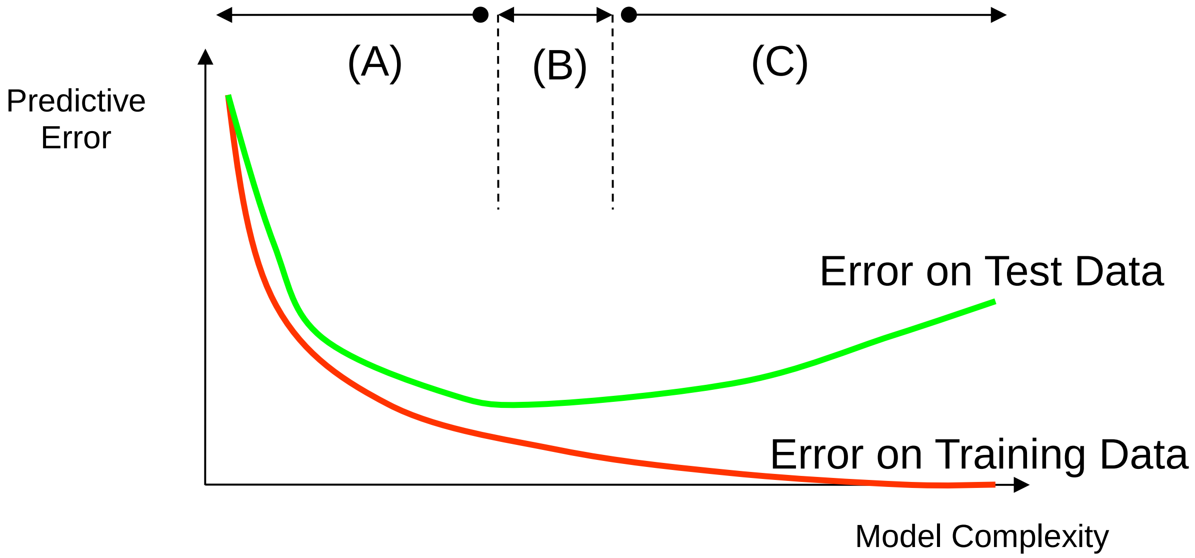
8.d. (1 pt) “There is a food that every person likes.”

$$\exists y\ \forall x\ \mathrm{Food}(y) \land [\ \mathrm{Person}(x) \Rightarrow \mathrm{Likes}(x, y)\ ]$$

8.e. (1 pt) “For every person, there is a food that the person likes.”

$$\forall x\ \exists y\ \mathrm{Person}(x) \Rightarrow [\ \mathrm{Food}(y) \land \mathrm{Likes}(x, y)\ ]$$

9. (3 pts total, 1 pt each) Model Complexity, Underfitting, Overfitting.
Consider the hypothetical graph below of Predictive Error (y-axis) vs. Model Complexity (x-axis), and how Error on Test/Training Data varies as Model Complexity increases.



Label the following regions as A, B, or C, depending on whether they correspond to Overfitting, Underfitting, or Ideal Model Complexity:

9.a. Overfitting (1 pt) (Write A ,B, or C) ______ C ______

9.b. Underfitting (1 pt) (Write A, B, or C) _____ A _____

9.c. Ideal Model Complexity (Write A, B, or C) ______ B ______

See Section 18.4 and Figure 18.9.

10. (8 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.

See Figure 3.21.

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 ****

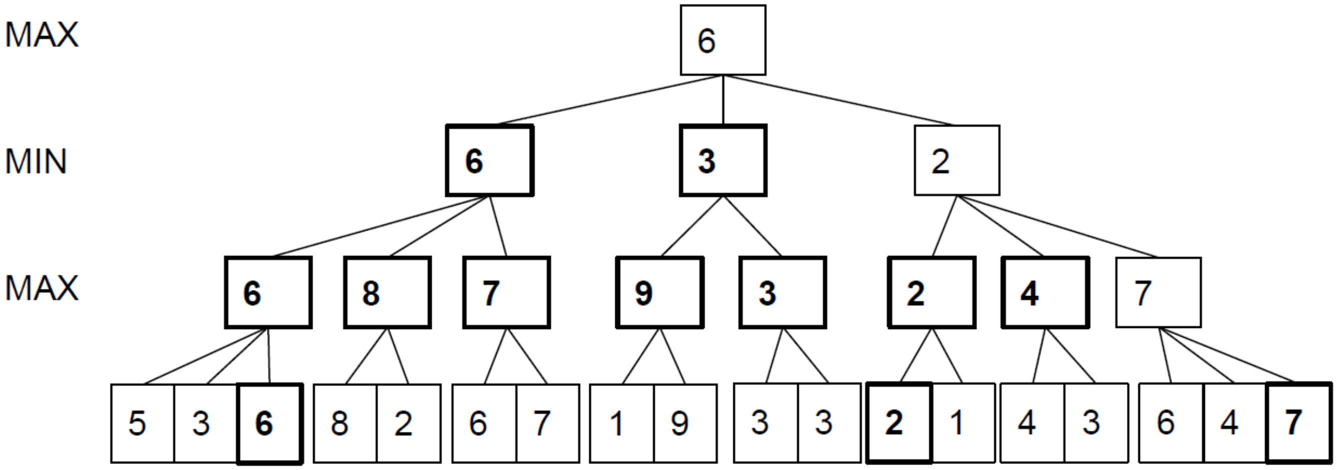
11. (4 pts total, 1 pt each) Task Environment. Your book defines a task environment as a set of four things, with the acronym PEAS. Fill in the blanks with the names of the PEAS components.

See Section2.3.1.

Performance (measure)   Environment   Actuators   Sensors

12. (12 pts total, 1 pt each) Mini-Max Game Search. 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. Each empty box in the game tree below has a single unique value that is consistent with the rest of the tree.
Fill in the values of the empty boxes below so that the tree is consistent.

See Section 5.2.1.



13. (6 pts total, 1 pt each) Valid, Unsatisfiable, Satisfiable.
For each sentence below, write V=valid if the sentence is valid, U=unsatisfiable if the sentence is unsatisfiable, and S=satisfiable if the sentence is satisfiable but is neither valid nor unsatisfiable.

13.a. (1 pt) (write V, U, or S) ___V___ (A OR (¬ A)) True in all worlds.

13.b. (1 pt) (write V, U, or S) ___U___ (A AND (¬ A)) False in all worlds.

13.c. (1 pt) (write V, U, or S) ___S___ (A AND (¬ B)) True only in worlds with A=TRUE and B=FALS

13.d. (1 pt) (write V, U, or S) ___V___ ((A AND (¬ A)) => B) True in all worlds, because antecede is false so implication is always true

13.e. (1 pt) (write V, U, or S) ___S___ ((A OR (¬ A)) => B) True only in worlds with B=TRUE.

13.f. (1 pt) (write V, U, or S) ___U___ ((A OR (¬ A)) => (A AND (¬ A)))

False in all worlds,
because antecedent is always true
and consequent is always false.

See Chapter 13.

14. Probability formulae (4 pts total, 1 pt each).
For each term on the left, write the letter of the best description on the right.
The first one is done for you as an example.

A	Probability Theory	A	Assigns each sentence a degree of belief ranging from 0 to 1
C	Conditional independence	B	$P(a \wedge b) = P(a)\, P(b)$
B	Independence	C	$P(a \wedge b \mid c) = P(a \mid c)\, P(b \mid c)$
E	Product rule (chain rule)	D	$P(a \mid b) = P(b \mid a)\, P(a) / P(b)$
D	Bayes' rule	E	$P(a \wedge b \wedge c) = P(a \mid b \wedge c)\, P(b \mid c)\, P(c)$

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