CS-171, Intro to A.I. — Final Exam — Winter Quarter, 2020

NAME (Print Darkly & Clearly): _____ UCI NetID: _____

YOUR ID#: _____

The following Pledge was suggested to faculty by ICS Dean Marios Papaefthymiou:

<u>The front page of your exam must be accompanied by the following (truthful) pledge written/typed out and signed by you:</u>

"On my honor, I have neither given nor received any unauthorized aid on this examination."

(Signature) _____

The Final Exam is take-home, open-book, open notes. Since you will not be able to ask for clarifications about any problems, if you have questions or confusions about any problem please write out your interpretation of the problem and we will take that statement into account when grading.

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

1. (15 pts total, 5 pts each) Constraint Satisfaction Problems (CSPs).

2. (18 pts total, 3 pts each) First Order Logic (FOL, FOPC, FOPL).

3. (18 pts total, 3 pts each) Bayesian Networks.

4. (12 pts total, 3 pts each) Machine Learning, Decision Tree Classifier.

5. (12 pts total) Minimax Search and Alpha-Beta Pruning.

6. (15 pts total, 3 pts each) Probability.

7. (10 pts total, 2 pts each) State-Space Search.

NAME (Print Darkly & Clearly): _____ UCI NetID: _____

1. (15 pts total, 5 pts each) Constraint Satisfaction Problems (CSPs)
You are a robot tasked to assign room locations to the Spring 2020 classes. Assume you must schedule 7 classes into 3 rooms. Obviously, you cannot assign the same room to two different classes that have a time-conflict.

The class times are given below (C means class):

C1: 7:30 – 9:50 am, Monday
C2: 8 – 8:50 am, Monday
C3: 9 - 10:20 am, Monday
C4: 7 - 10:20 am, Monday
C5: 10:15 – 10:50 am, Monday
C6: 11 - 11:20 am, Monday
C7: 10:25-11:20 am, Monday

The rooms are (R means room): R1, R2, R3

1.a. (5 pts total, -1 for each error, but not negative) Define this problem as a constraint satisfaction problem. Specify the variables, domains, and binary constraints.

1.a.i. (1 pt) The variables are: C1, C2, C3, C4, C5, C6, C7

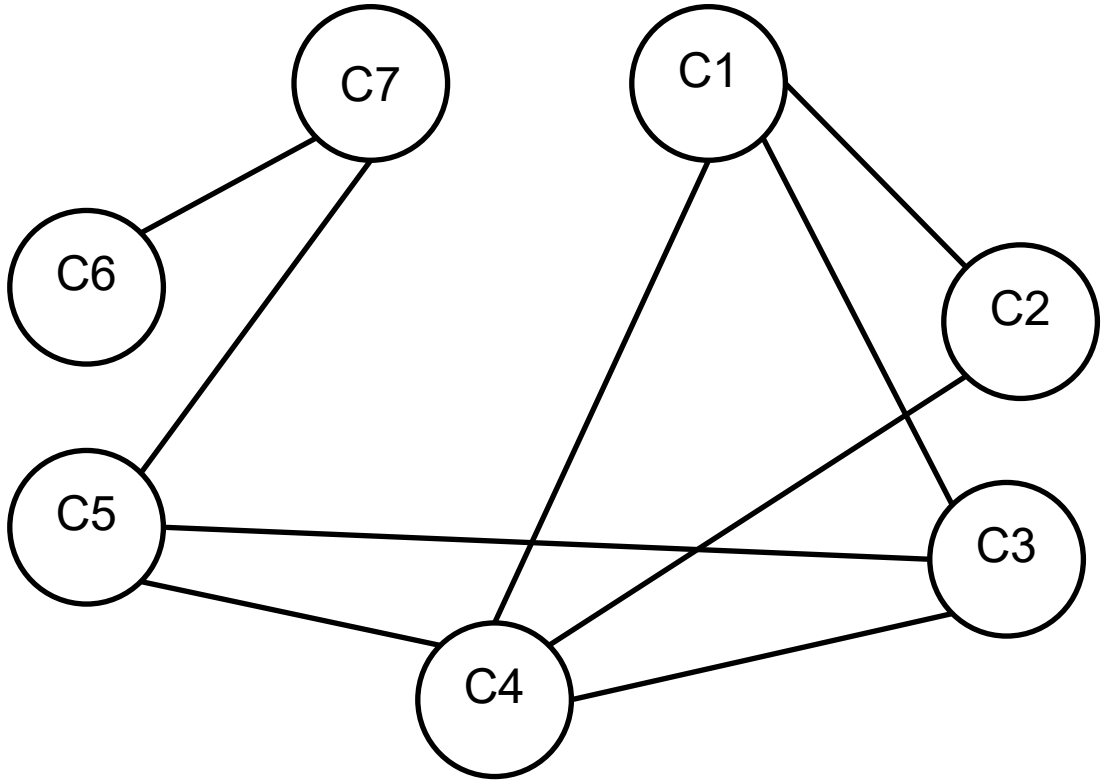
1.a.ii. (1 pt) The domains are: R1, R2, R3 Also OK: {R1, R2, R3}

1.a.iii. (3 pts total, -1 for each error, but not negative) (Write a constraint as Ci≠Cj, where Ci and Cj have a time-conflict.) The constraints are:

C1≠C2, C1 ≠ C3, C1≠C4, C2≠C4, C3≠C4, C3≠C5, C4≠C5, C5≠C7, C6≠C7

NAME (Print Darkly & Clearly): _____ UCI NetID: _____

1.b. (5 pts total, -1 for each error, but not negative) Draw the constraint graph for this problem.



1.c. (5 pts total) Next, we need to assign **all the room locations of all UCI classroom buildings** to **all the classes of all UCI departments** in the Spring quarter, so we have a much bigger constraint graph with many more nodes and edges. Assume we want to assign D available rooms to N classes.

Sara decides to do backtracking search with no heuristic on the constrain graph to find an answer.
Anna decides to perform 2-consistency (arc consistency) preprocessing on the constraint graph.
Bob decides to perform 3-consistency (path-consistency) preprocessing on the graph.

1.c.i. (1 pt) What is the complexity of the arc-consistency (AC-3 in your textbook) approach?

(use O(...) notation with only parameters N and D) $O(N^2D^3)$

1.c.ii. (1 pt) Whose approach is the simplest in terms of implementation?

(write one of Sara, Anna, Bob) Sara

1.c.iii. (1 pt) Whose approach is more likely to result in the smallest branching factors?

(write one of Sara, Anna, Bob) Bob

1.c.iv. (2 pts) Order the three approaches by how much preprocessing time each will take, from least to most preprocessing time. (fill in each blank below with one of Sara, Anna, Bob)

Sara < Anna < Bob

NAME (Print Darkly & Clearly):_____ UCI NetID: _____

2. (18 pts total, 3 pts each) First Order Logic (FOL, FOPC, FOPL)

2.a. (6 pts total, 3 pts each) In the following two questions, you will be asked to choose a FOL sentence that is equivalent to the one provided to you. Please write your answers to the multiple-choice questions so they are centered on the line, to assist the Readers in automatic grading.

2.a.i. (3 pts) Which of the following is equivalent to **¬∃x Unicorn(x)**? ___D____

A. ∀x Union(x)
B. ¬∀x Union(x)
C. ¬∀x ¬Union(x)
D. ∀x ¬Union(x)

2.a.ii. (3 pts) Which of the following is equivalent to **∀x Student(x) ⇒ GetA(x)**? ___B____

A. ¬∃x Student(x) ⇒ GetA(x)
B. ¬∃x Student(x) ∧ ¬GetA(x)
C. ¬∃x Student(x) ⇒ ¬GetA(x)
D. ¬∀x Student(x) ∧ ¬GetA(x)

NAME (Print Darkly & Clearly):_____ UCI NetID: _____

2.b. (12 pts total, 3 pts each) For each English sentence below, write the FOL sentence that best expresses its intended meaning. Use the following symbols:

Use **Student(x)** to mean that x is a student.
Use **Course(x)** to mean that x is a course.
Use **IsMaliciousDemon(x)** to mean that x is a malicious demon.
Use **IsLazy(x)** to mean that x is lazy.
Use **Passed(x, c)** to mean that x passed c.

Translate the following English sentences into FOL sentences:

2.b.example. Some student passed CS171.

∃x Student(x) ∧ Passed(x, CS171)

Any answer that is logically equivalent to an answer shown below is also a correct answer.

2.b.i. (3 pts) John is a lazy student

Student(John) ∧ IsLazy(John)

2.b.ii. (3 pts) If all the students passed CS171, then ProfLathrop is not a malicious demon.

(∀x Student(x) ⇒ Passed(x, CS171)) ⇒ ¬ IsMaliciousDemon(ProfLathrop)

2.b.iii. (3 pts) Not all lazy students failed CS171. (Note: Failed means Not Passed.)

∃x Student(x) ∧ Lazy(x) ∧ Passed(x, CS171)

Or: ¬∀x (Student(x) ∧ Lazy(x)) ⇒ ¬ Passed(x, CS171)

2.b.iv. (3 pts) All the students passed CS171, as well as some other class.

∀x∃c Student(x) ⇒ Passed(x, CS171) ∧ Course(c) ∧ Passed(x, c) ∧ ¬ (c = CS171)

Or: ∀x Student(x) ⇒ ∃c Passed(x, CS171) ∧ Course(c) ∧ Passed(x, c) ∧ ¬ (c = CS171)

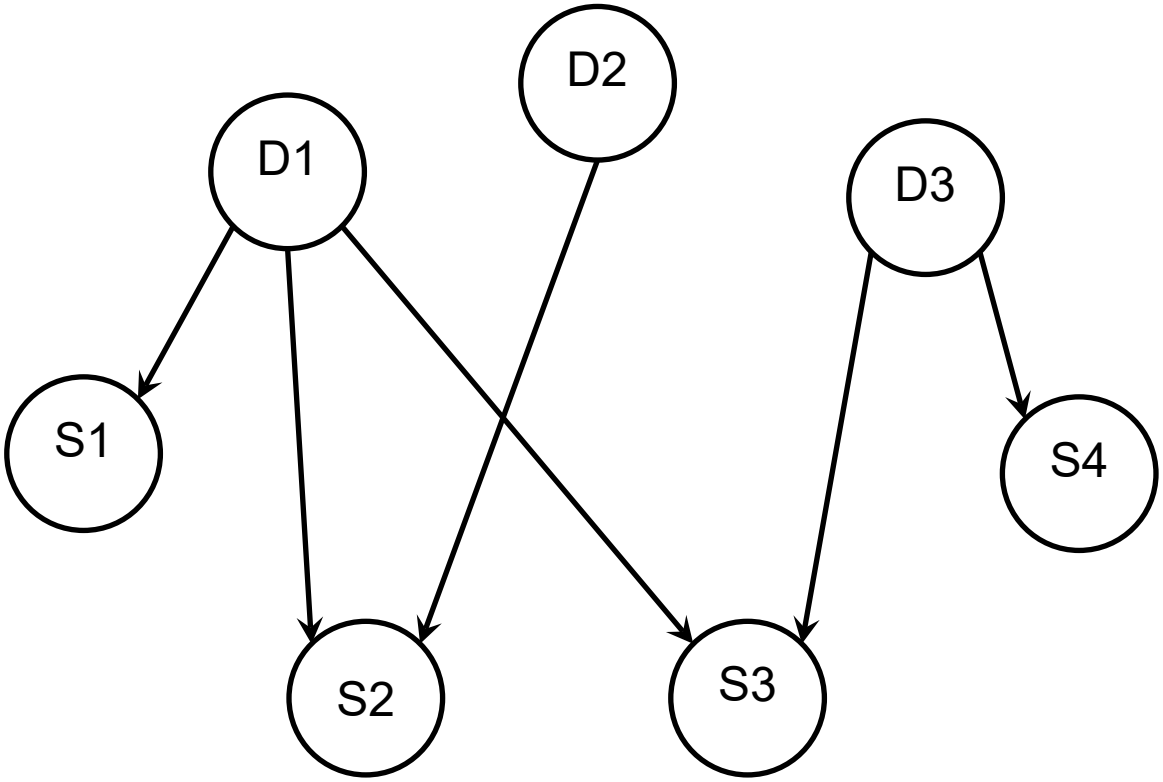
Or: ∀x Student(x) ⇒ Passed(x, CS171) ∧ ∃c Course(c) ∧ Passed(x, c) ∧ ¬ (c = CS171)

3. (18 pts total, 3 pts each) Bayesian Networks
A patient goes to the doctor for a medical condition. The doctor suspects three diseases, D1, D2, and D3, as the cause of the condition. The three diseases D1, D2, and D3 are marginally independent of each other. There are four symptoms S1, S2, S3, and S4 that the doctor wants to check. The symptoms are conditionally dependent to the three diseases as follows:

- S1 depends only on D1
- S2 depends on both D1 and D2
- S3 depends on both D1 and D3
- S4 depends only on D3.

Assume all random variables are Boolean, i.e., they are either 'true' or 'false'.

3.a. (3 pts, -1 for each error, but not negative) Draw the graph structure of the Bayesian Network that corresponds to the assumptions above.



3.b. (3 pts) Write the fully factored form of the full joint distribution that corresponds to the assumptions above. Order the variables as S1, S2, S3, S4, D1, D2, D3, left-to-right. (This ordering is designed to simplify automatic grading. You won't be penalized if you use a different ordering and your answer is otherwise correct. However, the Readers would be grateful if you helped in this way.)

P(S1 | D1) P(S2 | D1, D2) P(S3 | D1, D3) P(S4 | D3) P(D1) P(D2) P(D3)

3.c. (3 pts total, -1 for each error, but not negative) Each of the nodes in the Bayesian Network of 3.a above has an associated probability table that holds some probabilities (numbers, parameters). Fill in the table below with the number of probabilities needed at each node. The first one is done for you, as an example.

Node (Variable)	Number of Probabilities (Parameters)
S1	2 (= P(S1 | D1))
S2	4 (= P(S2 | D1, D2))
S3	4 (= P(S3 | D1, D3))
S4	2 (= P(S4 | D1))
D1	1 (= P(D1)
D2	1 (= P(D2)
D3	1 (= P(D2)
Total sum of above	15

3.d. (3 pts) Assume there were no conditional independence at all between the variables. Then how many total independent probabilities (numbers, parameters) would be needed?

127 (= 2^7 – 1, if represented as a Bayesian Network)

Also OK: 128 (if represented as a full joint distribution in table form)

3.e. (3 pts total, -1 for each error, but not negative) List all variables that are part of the Markov Blanket of variable S2.

D1 D2

3.f. (3 pts total, -1 for each error, but not negative) Suppose we observe symptom S4 = true. List all disease variables (D1, D2, D3) for which their posterior probability would be different from their prior probability.

D3

NAME (Print Darkly & Clearly): _____ UCI NetID: _____

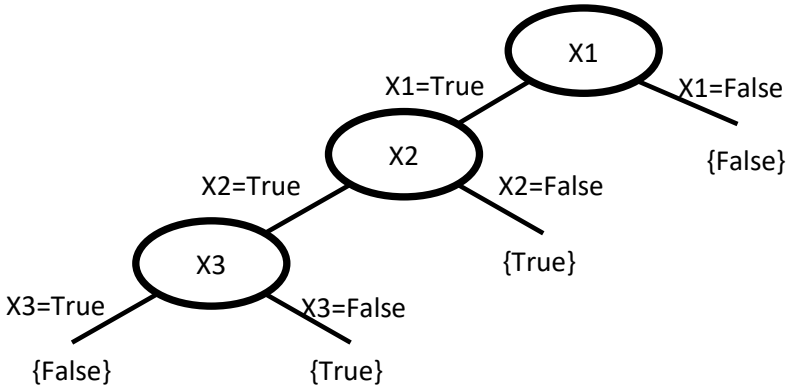
4. (12 pts total, 3 pts each) Machine Learning, Decision Tree Classifier.

The next two questions ask about Decision Trees (DTs) and Disjunctive Normal Form (DNF). A Decision Tree that has only Boolean features and a Boolean class (target) variable is equivalent to a Boolean formula expressed in Disjunctive Normal Form (they are dual representations).

4.a. (3 pts) DT to DNF Conversion. In the Decision Tree shown below, X1, X2, and X3 are Boolean variables, and the truth labels at the leaf of the decision tree are represented by {True} and {False}.

Write the DNF formula that corresponds to the tree shown.

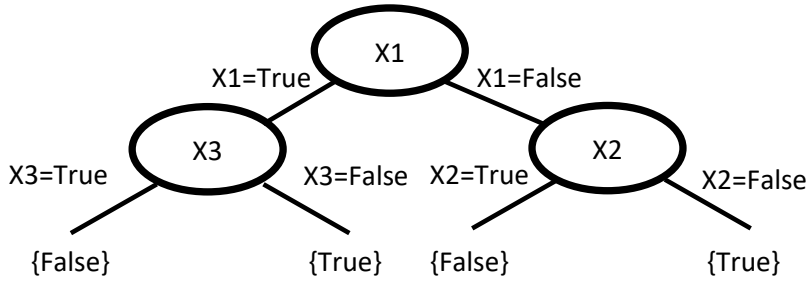
(X1 AND (NOT X2)) OR (X1 AND X2 AND (NOT X3))



4.b. (3 pts) DNF to DT Conversion. The DNF formula shown below uses Boolean variables X1, X2, and X3. You are asked to convert it into a DT. Along each path from root to leaf, order the variables as X1 (root), X2, X3 (last branch node), as is done above in 4.a; omit any variables you do not need along that path. At each branch node Xi, draw the Xi=True branch to the left and the Xi=False branch to the right, as is done above in 4.a. Represent the truth labels at the leaves by {True} and {False}.

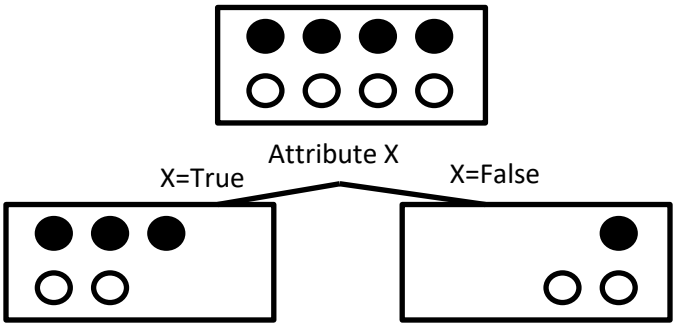
Draw the DT that corresponds to the DNF formula shown.

(X1 AND (NOT X3)) OR ((NOT X1) AND (NOT X2))



NAME (Print Darkly & Clearly): _____ UCI NetID: _____

4.c. (3 pts) Information Gain. Consider the figure below. There are 4 black and 4 white balls, initially evenly mixed. At a branch node, a Boolean attribute X splits the balls into two subgroups with uneven distributions, as shown in the figure.



Compute the information gain for splitting on attribute X above. (Use the approximations that log 1 = 0, log 2 = 1, log3 ≈ 1.58, log 4 = 2, log 5 ≈ 2.32; this is a take-home exam, so you also may use a calculator or write a computer program to compute your answer.) Please write your answers to the multiple-choice questions so they are centered on the line, to assist the Readers in automatic grading.

Which number is closest to the correct answer? (Write A, B, C, or D) ____C____

A. −0.05
B. 0.0
C. 0.05
D. 0.1

4.d. (3 pts total, -1 for each error, but not negative) DT True/False. Fill in each blank below with T (= True) or F (= False). Please write your answers to the True/False questions so they are centered on the line, to assist the Readers in automatic grading.

__F__ There are 2^n decision trees with n Boolean attributes.

__F__ For any Boolean function, there is a compact decision tree.

__T__ Overfitting occurs when the model complexity is too high for the available data.

__F__ Information gain from an attribute test is the expected gain in entropy.

NAME (Print Darkly & Clearly): _____ UCI NetID: _____

5. (12 pts total) Minimax Search and Alpha-Beta Pruning. Please write your answers to the multiple-choice questions so they are centered on the line, to assist the Readers in automatic grading.

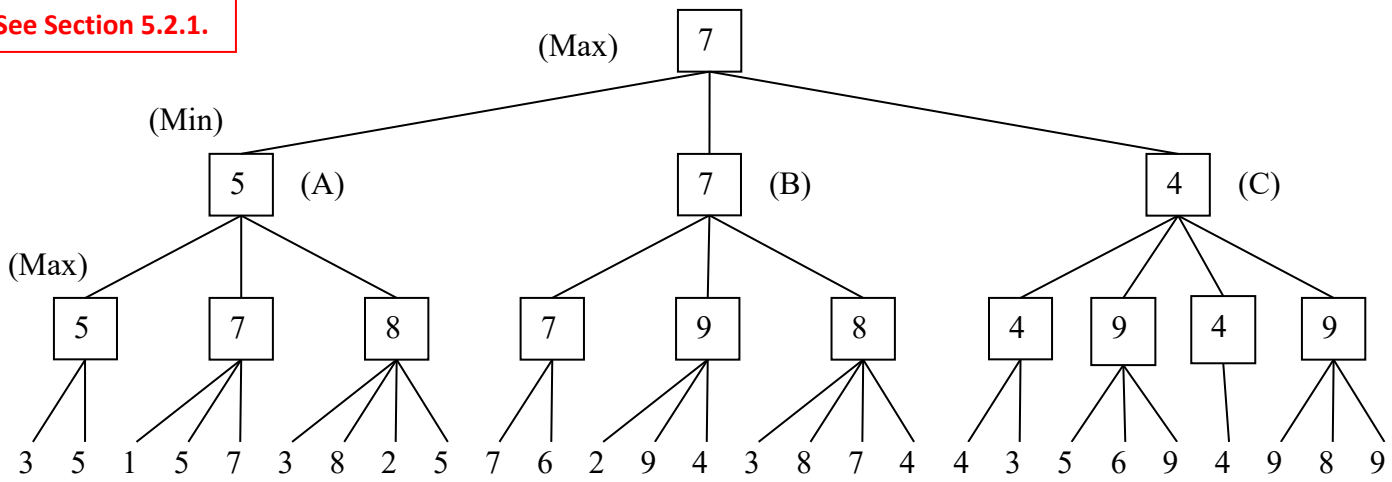
5.a. (3 pts total, 1 pt each) Minimax Search The game tree below illustrates a position reached in the game. Process the tree left-to-right. It is **Max**'s turn to move. At each leaf node is the estimated score returned by the heuristic static evaluator.

5.a.i. (1 pt) Fill in each blank square with the proper mini-max search value.

5.a.ii. (1 pt) What is the best move for Max? (write A, B, or C) B

5.a.iii. (1 pt) What score does Max expect to achieve? 7

See Section 5.2.1.

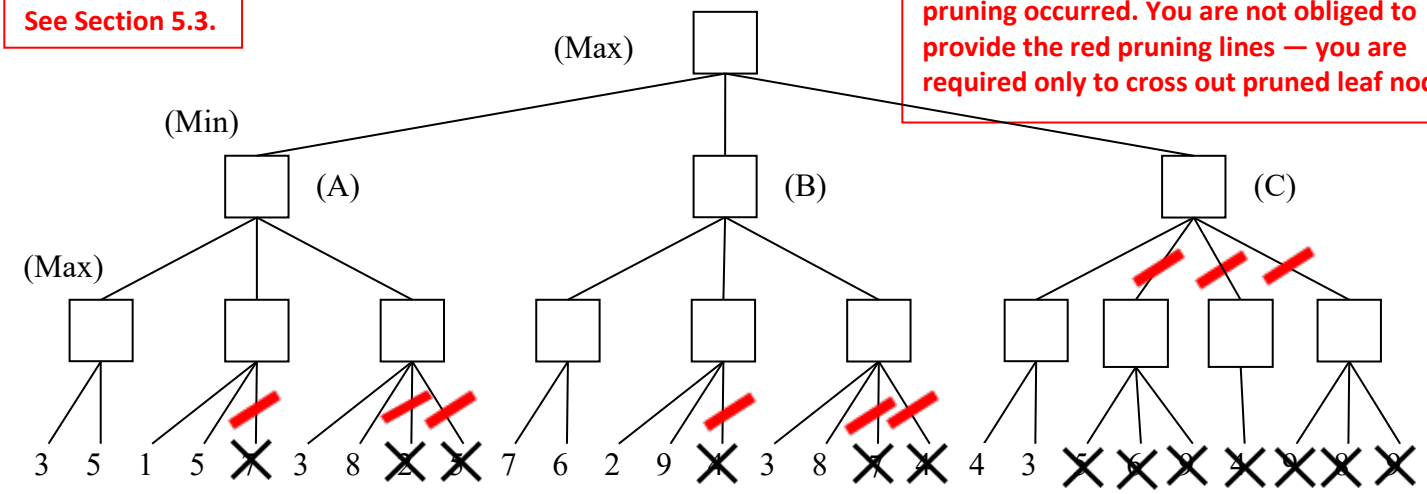


5.b. (5 pts total, -1 for each error, but not negative) ALPHA-BETA PRUNING. Process the tree left-to-right. This is the same tree as above (5.a). You do not need to indicate the branch node values again.

Cross out each leaf node that will be pruned by Alpha-Beta Pruning. Do not just draw pruning lines.

See Section 5.3.

Red pruning lines indicate where in the tree pruning occurred. You are not obliged to provide the red pruning lines — you are required only to cross out pruned leaf nodes.



5.c. (4 pts total, 1 pt each)

5.c.i. (1 pt) Minimax Search
What is the time complexity of minimax search? C

A. $O(bm)$
B. $O(m)$
C. $O(b^m)$
D. $O(m^b)$

5.c.ii. (1 pt) Basic Minimax Search
In the minimax search algorithm in its most basic form, which of the following is false? D

A. MAX assumes the opponent (MIN) will always choose the move that is worst for MAX
B. It is usually infeasible to search the entire game tree each turn for realistic games
C. The algorithm avoids all worst-case solutions for Max
D. It prunes all branches/moves that are best for our opponent

5.c.iii. (1 pt) Alpha-Beta Pruning Condition
What is one pruning condition of alpha-beta pruning? B

A. $\alpha > \beta$
B. $\alpha \geq \beta$
C. $\beta > \alpha$
D. $\beta \geq \alpha$

5.c.iv. (1 pt) Alpha-Beta Pruning Time Complexity
In the best case, alpha-beta pruning has a time complexity of approximately (use O(...) notation)

$O(b^{d/2})$

NAME (Print Darkly & Clearly):_____ UCI NetID: _____

6. (15 pts total, 3 pts each) Probability.
John is dismayed that 60% of the calls he receives are spam calls. As a result, John decides to make an intelligent call blocker. After weeks of development, John's program is ready to be tested. The results showed that the call blocker successfully blocked 70% of the spam calls. However, the call blocker also blocked 30% of calls from friends and family (= not spam). Use these random variables:

- S = the call is a spam call
- B = the call is blocked by John's program

Given the scenario provided above, calculate the following probabilities. **Show your work.**

6.a. (3 pts) What's the probability that the call blocker does not block calls made from friends and family?

$P(\neg B \mid \neg S) = 1 - P(B \mid \neg S) = 1 - 0.3 = 0.7$

6.b. (3 pts) What's the probability that the call blocker blocks a call and the call is a spam call?

$P(B, S) = P(B \mid S)\, P(S) = 0.7 * 0.6 = 0.42$

6.c. (3 pts) What's the probability that the call blocker blocks a call?

$P(B) = P(B, S) + P(B, \neg S) = P(B \mid S)\, P(S) + P(B \mid \neg S)\, P(\neg S)$
$= 0.7 * 0.6 + (1 - 0.7) * (1 - 0.6) = 0.7 * 0.6 + 0.3 * 0.4$
$= 0.54$

6.d. (3 pts) What's the probability that the call is from friends and family given that the call has been blocked by the call blocker?

$P(\neg S \mid B) = P(B \mid \neg S)\, P(\neg S) / P(B)$
$= 0.3 * (1 - 0.6) / 0.54 = 0.3 * 0.4 / 0.54$
$= 0.2222$

6.e. (3 pts) Is the random variable B independent of the random variable S?
(answer Y=Yes or N=No) N

$P(B, S) = 0.42$

$P(B) * P(S) = 0.54*0.6 = 0.32$

Since P(B) * P(S) does not equal P(B, S), B and S are not independent of each other.

NAME (Print Darkly & Clearly): _____ UCI NetID: _____

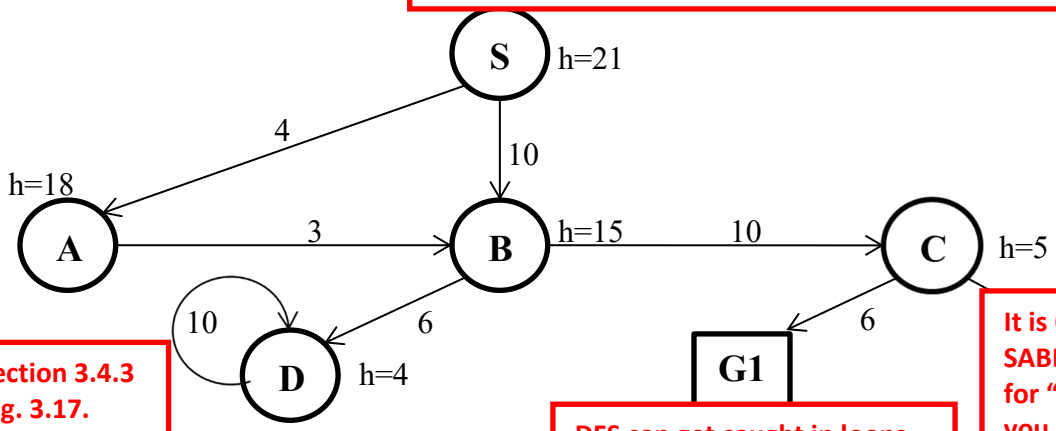
7. (10 pts total, 2 pts each) STATE-SPACE SEARCH. Execute Tree Search through this graph (do not remember visited nodes, so repeated nodes are possible). It is not a tree, but pretend you don't know that. Step costs are given next to each arc, and heuristic values are given next to each node (as h=x). The successors of each node are indicated by the arrows out of that node. **(Note: D is a successor of itself).** As usual, successor nodes are returned in left-to-right order. (The successor nodes of S are A,B; and the

The start node is S and there are tw

(1) the order in which nodes are expanded

Write "None" for the path and cost if the g

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 Section 3.4.3 and Fig. 3.17.

It is OK if you wrote SABDDD... instead of None for "Path found." It is OK if you said N/A for "Cost of path found," or left it blank.

7.a. (example) DEPTH-FIRST SEARCH:

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

7.a.i Order of expansion: S A B D D D D ...

7.a.ii Path to goal found: None Cost of path found: None

7.b. (2 pts) 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 G1 is found when C is expanded.

on: S A B D C G1

7.b.ii Path to goal found: S B C G1 Cost of path found: 26

7.c. (2 pts) ITERATIVE DEEPENING SEARCH:

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

IDS does the Goal-test iteratively on each child as generated, keeping the queue on the stack.

S S A B S A B B D C G1

7.c.ii Path to goal found: S B C G1 Cost of path found: 26

7.d. (2 pts) UNIFORM COST SEARCH:

See Section 3.4.2 and Fig. 3.14.

UCS does Goal-test when node is popped off

on: S A B D C G2

nd: S A B C G2 Cost of path found: 22

7.e. (2 pts) GREEDY BEST FIRST SEARCH

See Section 3.5.1 and Fig. 3.23.

GBFS can get caught in loops during Tree Search (= do not remember visited nodes). The heuristic value at node D (h=4) is lower than any other heuristic value on the queue.

on: S B D D D D ...

nd: None Cost of path found: None

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

See Section 3.5.2 and Figs. 3.24-25.

A* does Goal-test when node is popped off

n: S A B D C G2

d: S A B C G2 Cost of path found: 22

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.