CS 181 Artificial Intelligence (Fall 2019), Midterm Exam

Name (in Chinese): ________________________

ID#: ________________________

Instructions

- Time: 1–2:40am (100 minutes)
- This exam is closed-book, but you may bring one A4-size cheat sheet. Put all the study materials and electronic devices into your bag and put your bag in the front, back, or sides of the classroom.
- You can write your answers in either English or Chinese.
- Two blank pieces of paper are attached, which you can use as scratch paper. Raise your hand if you need more paper.

1 Multiple choice (10 pt)

Each question has one or more correct answer(s). Select all the correct answer(s). For each question, you get 0 point if you select one or more wrong answers, but you get 0.5 point if you select a non-empty proper subset of the correct answers. Fill your answers in the table below.

1	2	3	4	5

6	7	8	9	10

1. Consider graph search in a finite state space graph. Which of the following statement(s) is/are correct?

   A. BFS is complete and optimal when all the edge costs are the same.

   B. The space complexity of iterative deepening search is lower than that of BFS.

   C. A^* is optimal if the heuristic is admissible.

   D. Greedy search is complete but not optimal.

   E. None of the above

2. For a CSP represented by the following constraint graph, suppose we initially enforce arc consistency and then run backtracking search with no filtering. Which of the following orderings of variable assignments can guarantee that no backtracking will be necessary when finding a solution?

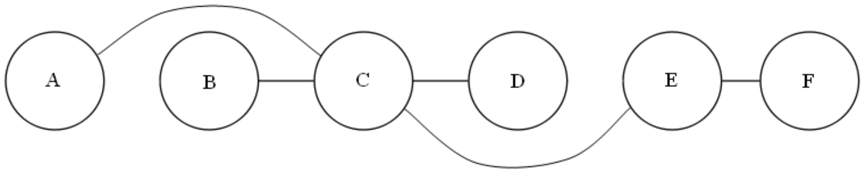
   A. A-B-C-D-E-F

   B. D-E-F-C-B-A

   C. B-C-D-A-E-F

D. B-D-A-F-E-C

E. None of the above



3. Which of the following statement(s) about propositional logic is/are correct?

   A. If a sentence S is unsatisfiable, then $\neg S$ is valid.

   B. If we use the resolution algorithm to prove $A \wedge B \models A$, then it will yield an empty clause in the process.

   C. $\neg A \wedge \neg B \wedge C \wedge D$ is a Horn clause.

   D. Forward chaining and backward chaining are sound and complete with Horn clauses and run linear in space and time.

   E. None of the above

4. Which of the following statement(s) about first-order logic is/are correct?

   A. $\exists x\ At(x, SIST) \Rightarrow Loves(x, AI)$ represents "*Someone at SIST loves AI*".

   B. $\forall x\ Has(x, y)$ satisfies FOL syntax.

   C. If $\forall x\ \exists y\ (x > y)$ is true, then $\exists y\ \forall x\ (x > y)$ is also true.

   D. Unification of $Marries(Alice, x)$ and $Marries(x, Bob)$ fails but the failure can be avoided by standardizing apart.

   E. None of the above

5. Which of the following statement(s) of equivalence is/are correct?

   A. $((A \Rightarrow C) \wedge (B \Rightarrow C)) \equiv (A \wedge B \Rightarrow C)$

   B. $((A \Rightarrow C) \vee (B \Rightarrow C)) \equiv (A \vee B \Rightarrow C)$

   C. $(\neg A \vee \neg B) \equiv ((A \wedge B) \Rightarrow (A \wedge \neg A))$

   D. $[C \vee (\neg A \wedge \neg B)] \equiv [(\neg A \wedge \neg B) \vee (C \wedge \neg B) \vee (\neg A \wedge C) \vee C]$

   E. None of the above

6. Which of the following statement(s) about independence and Bayesian networks is/are correct? X, Y, Z are random variables.

   A. $X \perp\!\!\!\perp Y|Z$ if and only if $P(X|Y, Z) = P(X|Z)$.

   B. $X \perp\!\!\!\perp Y$ can imply $P(X, Y|Z) = P(X|Z)P(Y|Z)$.

   C. In Bayesian networks, every variable is conditionally independent of its non-descendants given one of its parents.

   D. In Bayesian networks, every variable is conditionally independent of all other variables given its Markov blanket.

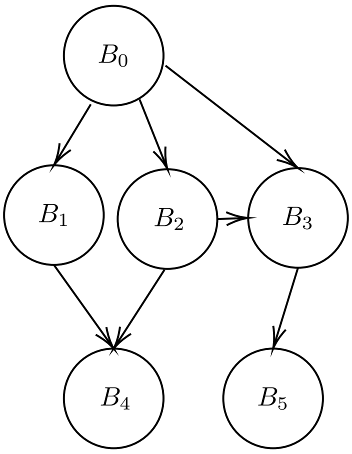
   E. Bayesian networks need to reflect the true causal patterns.

7. Which of the following statement(s) about conditional independence in the following Bayesian network and Markov network is/are correct? B_i, M_i are random variables.

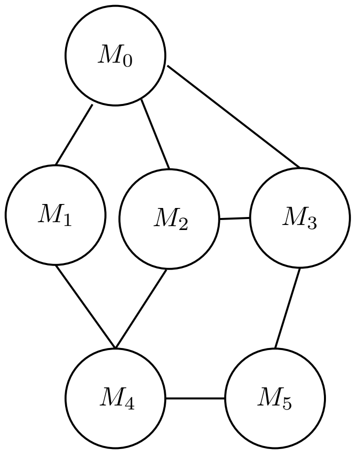
   A. $B_1 \perp\!\!\!\perp B_5|\emptyset$.

   B. $B_1 \perp\!\!\!\perp B_2|B_0$.

   C. $B_0 \perp\!\!\!\perp B_4|B_1, B_2, B_5$.



(a) Bayesian Network



(b) Markov Network

D. $\{M_0, M_1\} \perp\!\!\!\perp \{M_4, M_5\} | \{M_2, M_3\}$

E. If we add an edge from B_4 to B_5 in the Bayesian network, and also we assume B_i and M_i are the same random variable for $i \in \{0, 1, 2, 3, 4, 5\}$, then the two graphical models encode exactly the same set of conditional independence.

8. Consider the Markov network below. Which of the following statement(s) is/are correct?

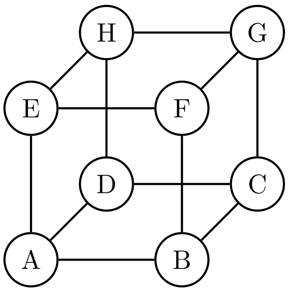
A. The Markov blanket of any node in this network contains exactly 6 nodes.

B. We need to define exactly 6 potential functions on this network.

C. A and G can become conditionally independent given at least 3 other nodes.

D. Given any joint distribution on the 8 nodes, it is always possible to represent the distribution using this network by adding at most 15 edges.

E. None of the above.



9. Consider the Bayesian network below, where $E = e$ is the evidence. Which of the following statement(s) is/are correct?

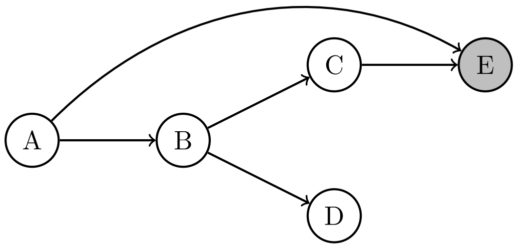
A. In Gibbs sampling, after initialization A must be sampled at least once before the first sampling of D.

B. In Gibbs sampling, assume that the sample is currently $(A, B, C, D, E) = (a, b, c, d, e)$; to sample C in the next step, we should sample from the distribution $P(C|B = b, E = e)$.

C. In variable elimination, A must be eliminated before D.

D. In variable elimination, D must be eliminated before A.

E. None of the above.



10. For a Markov logic network (MLN), which of the following statement(s) is/are correct?

    A. First-order logic is a special case of MLNs.

    B. A formula in the MLN generates exactly one clique in the ground Markov network.

    C. If all the formula weights are finite, the probability of a world is zero as long as it violates a formula.

    D. Every probabilistic relational model can be converted to an equivalent MLN.

    E. None of the above.

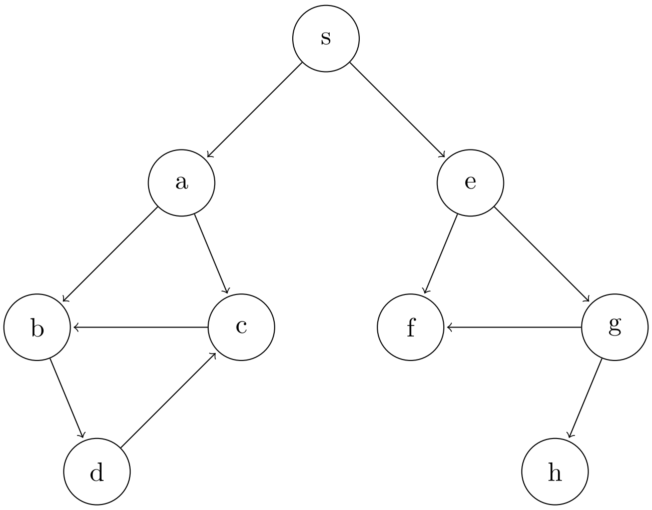
Solution:

1. ABD
2. C
3. ABD
4. D
5. CD
6. AD
7. BC
8. C
9. E
10. AD

2 Search (10 pt)

2.1 BFS and DFS (3 pt)

Consider the graph search problem with the following directed graph, where s is the start state and h is the goal state. Please write down the state visiting order of BFS and DFS. Please use "→" to indicate the order. For example, $s \to a \to b \to \cdots$. Suppose we expand the child nodes from left to right.

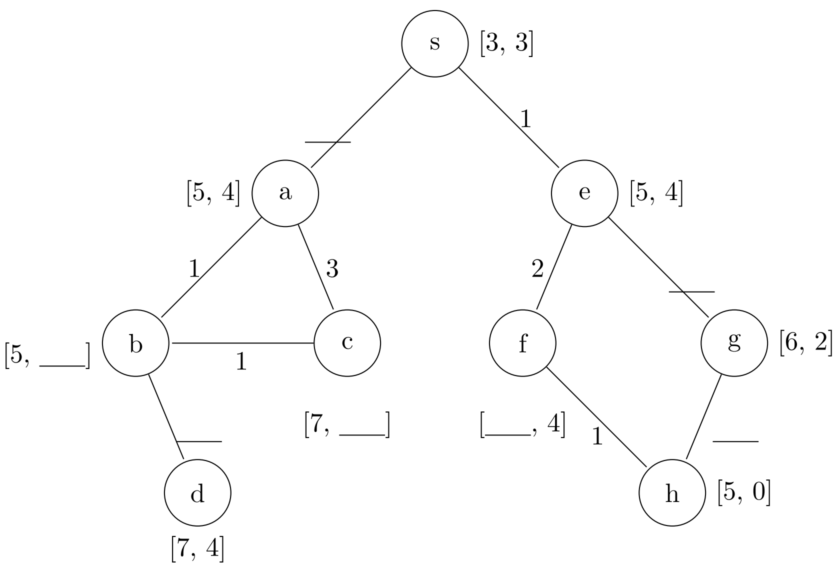


1) BFS: ____

2) DFS: ____

2.2 A^* algorithm (7 pt)

Consider the graph search problem with the following undirected graph, where s is the start state and h is the goal state. Suppose that the A^* algorithm has been applied to the search problem and the goal h has been found. The pair of numbers next to each node n denote $f(n)$ and $h(n)$ in A^*, and the number next to each edge denotes the cost of the edge. Suppose we put the child nodes of an expanded node into the fringe in the left-to-right order and we expand nodes with the same $f(n)$ value in their order of entering the fringe.



1. Please fill the blanks in the figure.
2. In the table below, please write down, for each step of A^*, the content of the fringe and the closed-set, as well as the node that is expanded (we have provided the answer for the first step for you).

	Fringe	Closed-set	Node to expand
1	s	empty	s

3. Is the final solution optimal?

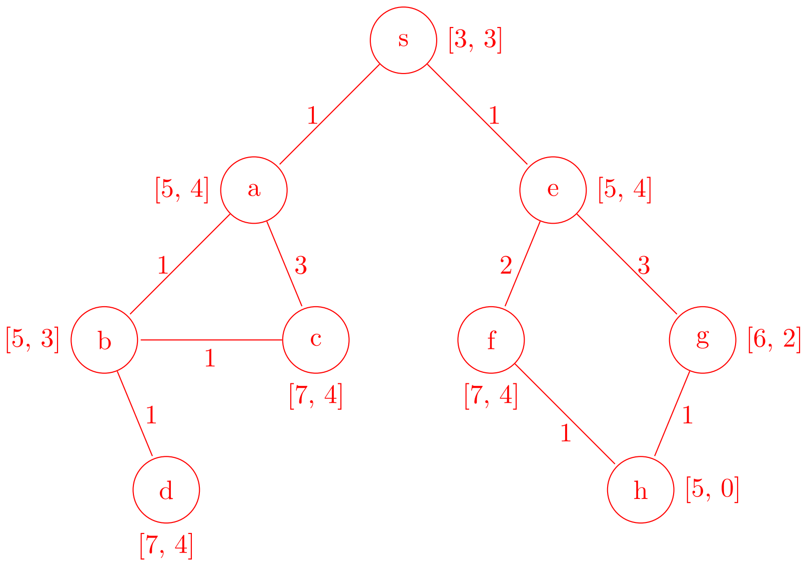
Solution:

2.1

1) BFS: $\underline{s->a->e->b->c->f->g->d->h}$
2) DFS: $\underline{s->a->b->d->c->e->f->g->h}$

2.2 A^* algorithm

1. Please fill in the blank of the figure 2.



2.

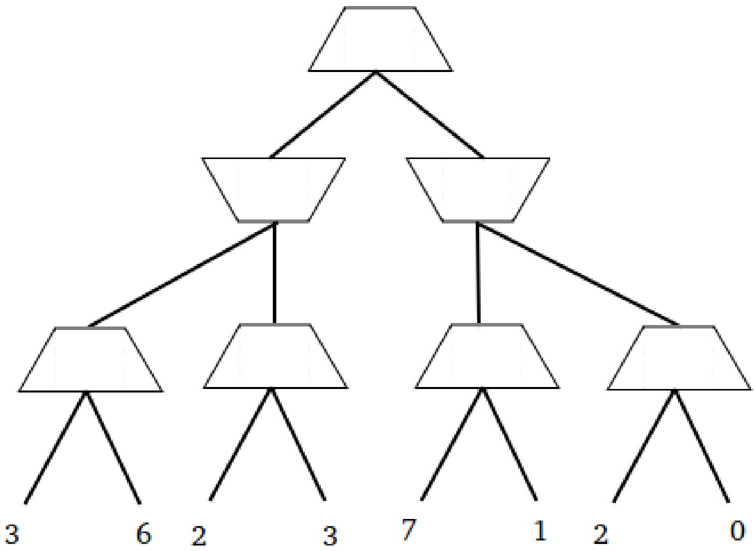
	Fringe	Closed-set	Node to expand
1	s	empty	s
2	a,e	s	a
3	e,b,c	s,a	e
4	b,c,f,g	s,a,e	b
5	c,f,g,d	s,a,e,b	g
6	c,f,d,h	s,a,e,b,g	h

3. Is the final solution optimal?
   No.

3 Game (10 pt)

3.1 Standard Minimax (1 pt)

Fill in the value of each of the nodes in the following Minimax tree, and circle the path of actions that correspond to Minimax play (i.e., the actions of the two players if they follow the optimal policy). The upward pointing shapes correspond to maximizer nodes (layer 1 and 3), and the downward pointing shapes correspond to minimizer nodes (layer 2). Each node has two actions available, Left and Right.



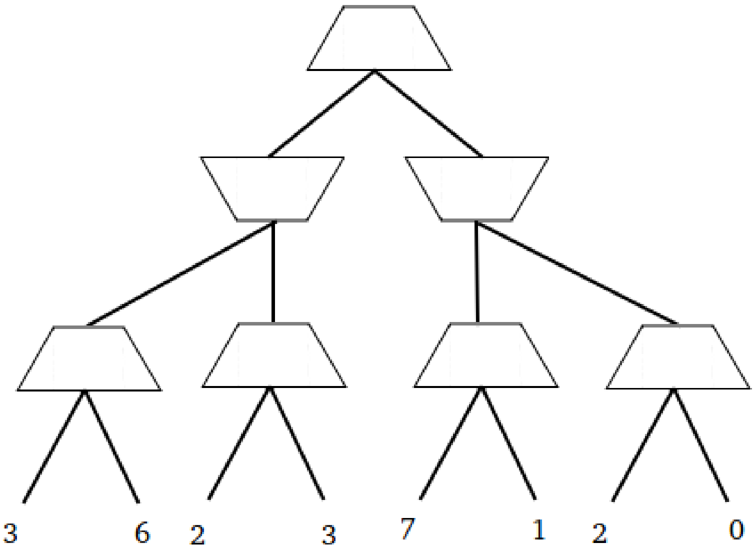
3.2 Dark Magic (9 pt)

Pacman (= maximizer) has mastered some dark magic. With his dark magic skills, Pacman can take control over his opponent's muscles while they execute their move, and in doing so be fully in charge of the opponent's move. But the magic comes at a price: every time Pacman uses his magic, he pays a price of c, which is measured in the same units as the values at the bottom of the tree.

Note: For each of his opponent's actions, Pacman has the choice to either let his opponent act (optimally according to minimax), or to take control over his opponent's move at a cost of c.

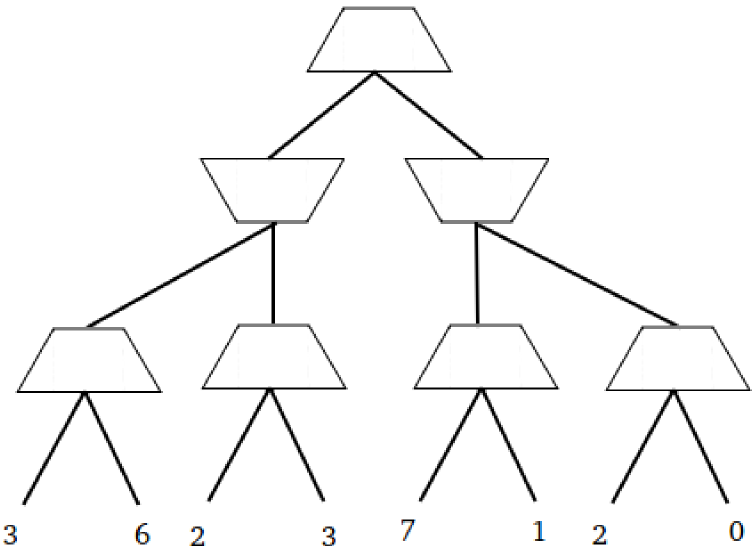
3.2.1 Dark Magic at Cost $c = 2$ (1.5 pt)

Consider the same game as before but now Pacman has access to his magic at cost $c = 2$. Is it optimal for Pacman to use his dark magic? If so, circle the node in the tree below where he will use it. Either way, circle the path of actions that lead to the outcome of the game.



3.2.2 Dark Magic at Cost $c = 5$ (1.5 pt)

Consider the same game as before but now Pacman has access to his magic at cost $c = 5$. Is it optimal for Pacman to use his dark magic? If so, circle the node in the tree below where he will use it. Either way, circle the path of actions that lead to the outcome of the game.



3.2.3 Dark Magic Minimax Algorithm (4 pt)

Now let's study the general case. Assume that the minimizer player has no idea that Pacman has the ability to use dark magic at a cost of c, i.e., the minimizer chooses their actions according to standard minimax. As a starting point we give you below the pseudo-code for a standard minimax agent.

```
function Max-Value(state)
    if state is leaf
        return Utility(state)
    end if
    v ← −∞
    for successor in Successors(state)
```

```
        v ← max(v, Min-Value(successor))
    end for
    return v
end function

function Min-Value(state)
    if state is leaf
        return Utility(state)
    end if
    v ← ∞
    for successor in Successors(state)
        v ← min(v, Max-Value(successor))
    end for
    return v
end function
```

Please complete the following pseudo-code such that it returns the optimal value for Pacman with dark magic.

```
function Max-Value(state)
    if state is leaf
        return (Utility(state), Utility(state))
    end if
    v_min ← -∞
    v_max ← -∞
    for successor in Successors(state)
        vNext_min, vNext_max ← Min-Value(successor)

        v_min ← ______________________

        v_max ← ______________________
    end for
    return (v_min, v_max)
end function

function Min-Value(state)
    if state is leaf
        return (Utility(state), Utility(state))
    end if
    v_min ← ∞
    min_move_v_max ← -∞
    v_magic_max ← -∞
    for successor in Successors(state)
        vNext_min, vNext_max ← Max-Value(successor)
        if v_min > vNext_min

            v_min ← ______________________

            min_move_v_max ← ______________________
        end if

        v_magic_max ← ______________________
    end for

    v_max ← ______________________
    return (v_min, v_max)
end function
```

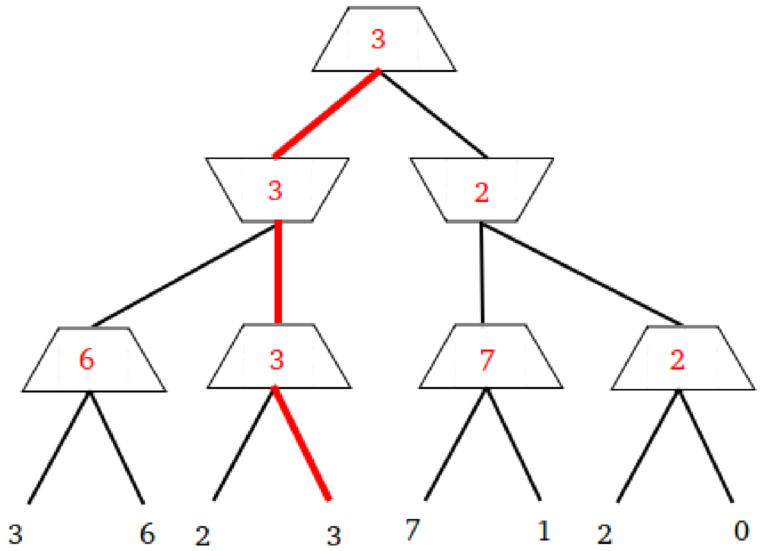
3.2.4 Dark Magic Becomes Predictable (2 pt)

The minimizer has come to the realization that Pacman has the ability to apply magic at cost c. Hence the minimizer now does not play according the regular minimax strategy anymore, but accounts for Pacman's magic capabilities when making decisions. Pacman in turn, is also aware of the minimizer's new way of making decisions. Again complete the following pseudo-code such that it returns the optimal value for Pacman.

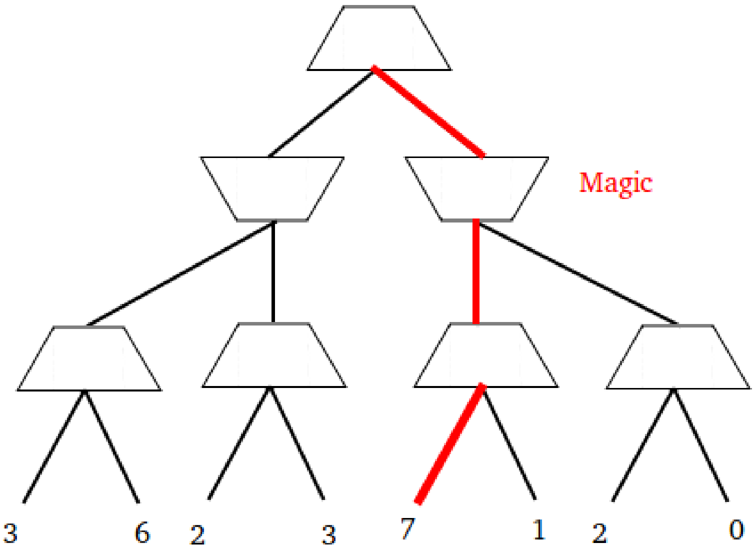
```
function Min-Value(state)
    if state is leaf
        return Utility(state)
    end if
    v ← ∞
    v_m ← −∞
    for successor in Successors(state)
        temp ← Max-Value(successor)

        v ← ____________________

        v_m ← ____________________
    end for

    return ____________________
end function
```

Solution:

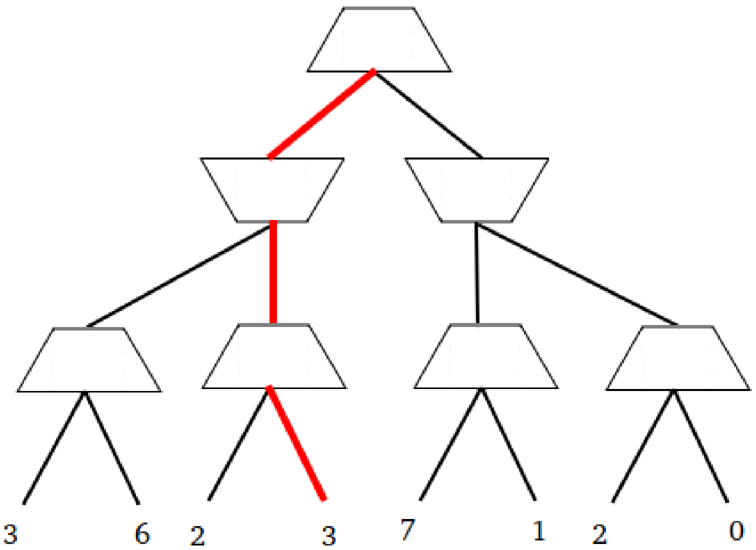
3.1



3.2.1



3.2.2



3.2.3

```
function Max-Value(state)
    if state is leaf
        return (Utility(state), Utility(state))
    end if
    v_min ← -∞
    v_max ← -∞
    for successor in Successors(state)
        vNext_min, vNext_max ← Min-Value(successor)
        v_min ← max(v_min, vNext_min)
        v_max ← max(v_max, vNext_max)
    end for
    return (v_min, v_max)
end function

function Min-Value(state)
    if state is leaf
        return (Utility(state), Utility(state))
    end if
    v_min ← ∞
    min_move_v_max ← -∞
```

```
        v_magic_max ← −∞
        for successor in Successors(state)
            vNext_min, vNext_max ← Max−Value(successor)
            if v_min > vNext_min
                v_min ← vNext_min
                min_move_v_max ← vNext_max
            end if
            v_magic_max ← max(vNext_max, v_magic_max)
        end for
        v_max ← max(min_move_v_max, v_magic_max − c)
        return (v_min, v_max)
end function
```

3.2.4

```
function Min−Value(state)
        if state is leaf
            return Utility(state)
        end if
        v ← ∞
        v_m ← −∞
        for successor in Successors(state)
            temp ← Max−Value(successor)
            v ← min(v, temp)
            v_m ← max(v_m, temp)
        end for
        return max(v, v_m − c)
end function
```

4 Logic (10 pt)

4.1 Propositional Logic (5pt)

KB consists of the following sentences:

$$\neg A \rightarrow B$$
$$B \rightarrow A$$
$$A \rightarrow (C \wedge D)$$

Proposition α is

$$A \wedge B \wedge C$$

4.1.1 CNF

Convert KB to the conjunctive normal form.

Solution:

$(A \vee B) \wedge (\neg B \vee A) \wedge (\neg A \vee C) \wedge (\neg A \vee D)$

4.1.2 Resolution inference rule

Prove $KB \models \alpha$ using resolution. (Note: apply the resolution rule to two clauses each time.)

Solution:

KB does not entail α.

1. convert $KB \wedge \neg\alpha$ to CNF:

$$(A \vee B) \wedge (\neg B \vee A) \wedge (\neg A \vee C) \wedge (\neg A \vee D) \wedge (\neg A \vee \neg B \vee \neg C)$$

2. Repeatedly apply the resolution rule to add new clauses. Until there no new clause that can be added, empty set is not yielded. Therefore, KB does not entail α.

Note: If two clauses resolve, they may have more than one resolvent because there can be more than one way in which to choose the resolvents.

correct:

$$\frac{\{p, q\} \qquad \{\neg p, \neg q\}}{\begin{array}{c}\{p, \neg p\} \\ \{q, \neg q\}\end{array}}$$

wrong:

$$\frac{\{p, q\} \qquad \{\neg p, \neg q\}}{\{\}}$$

There is an example on page 26 of lecture slide Propositional Logic: resolve $P_{1,2} \vee P_{2,1} \vee \neg B_{1,1}$ and $\neg P_{1,2} \vee B_{1,1}$

4.2 First-order Logic (5pt)

Express the following sentences in first-order logic.
Note:

1. You may use binary predicates *Friend*, *Enemy*, and equality $=$.
2. For simplicity, in the case of $Friend(x, y)$ and $Enemy(x, y)$, we assume $x \neq y$, so you do not need to say it explicitly.

1. An enemy's enemy is a friend.

Solution:
$\forall x \forall y \forall z(Enemy(x, y) \wedge Enemy(y, z) \rightarrow Friend(x, z))$

2. Any two people are either enemies or friends.

Solution:
$\forall x \forall y(Friend(x, y) \vee Enemy(x, y))$

3. Bob has at least two friends.

Solution:
$\exists x \exists y(Friend(Bob, x) \wedge Friend(Bob, y) \wedge \neg(x = y))$

4. Anyone is Bob's enemy.

Solution:
$\forall x Enemy(Bob, x)$

5. Bob has exactly two friends, and everyone else is an enemy.

Solution:
$\exists x \exists y(Friend(Bob, x) \wedge Friend(Bob, y) \wedge \neg(x = y) \wedge \forall z(\neg(z = x) \wedge \neg(z = y) \rightarrow Enemy(Bob, z))$

5 Inference on Bayesian Network (10 pt)

Given the following Bayesian network with four random variables A, B, C, D, answer the following questions.

P(A)

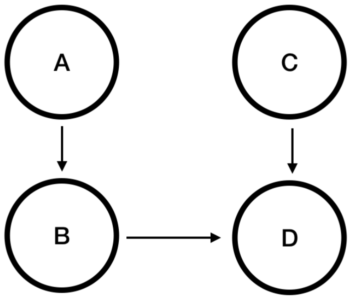
+a	0.2
-a	0.8

P(C)

+c	0.5
-c	0.5

P(B|A)

+a	+b	0.1
+a	-b	0.9
-a	+b	0.3
-a	-b	0.7



P(D|B,C)

+b	+c	+d	0.0
+b	+c	-d	1.0
+b	-c	+d	0.4
+b	-c	-d	0.6
-b	+c	+d	0.2
-b	+c	-d	0.8
-b	-c	+d	0.8
-b	-c	-d	0.2

5.1 Exact Inference(5pt)

Using variable elimination to calculate $P(+d|+a)$. Write down the variable eliminating steps to get full points.

Solution:

Eliminate C

factors: $f(+a)$,$f(B|+a)$, $\underline{f(C)}$, $\underline{f(D|B,C)}$

$f(+d|B) = \sum_c f(C)f(+d|B,C)$

$f(+d|+b) = P(+c)P(+d|+b,+c) + P(-c)P(+d|+b,-c) = 0.5 * 0.0 + 0.5 * 0.4 = 0.20$

$f(+d|-b) = P(+c)P(+d|-b,+c) + P(-c)P(+d|-b,-c) = 0.5 * 0.2 + 0.5 * 0.8 = 0.50$

$f(-d|B) = \sum_c f(C)f(-d|B,C)$

$f(-d|+b) = P(+c)P(-d|+b,+c) + P(-c)P(-d|+b,-c) = 0.5 * 1.0 + 0.5 * 0.6 = 0.80$

$f(-d|-b) = P(+c)P(-d|-b,+c) + P(-c)P(-d|-b,-c) = 0.5 * 0.8 + 0.5 * 0.2 = 0.50$

Then eliminate B

factors: $f(+a)$,$\underline{f(B|+a)}$, $\underline{f(D|B)}$

$f(+d|+a) = \sum_b f(B|+a)f(+d|B)$

$f(+d|+a) = f(+b|+a)f(+d|+b) + f(-b|+a)f(+d|-b) = 0.1 * 0.2 + 0.9 * 0.5 = 0.47$

$f(-d|+a) = \sum_b f(B|+a)f(-d|B)$

$f(-d|+a) = f(+b|+a)f(-d|+b) + f(-b|+a)f(-d|-b) = 0.1 * 0.8 + 0.9 * 0.5 = 0.53$

Optional: joint the remaining factors. After normalize, $P(+d|+a) = 0.47$

5.2 Approximate Inference(5pt)

Now we use approximate inference to calculate $P(+d|+a)$. We select truth values of variables based on a sequence of uniformly sampled random numbers between 0 and 1 shown in the table below. We determine the value (True/False) of random variable A based on the value of random number r in the following way. If $r \leq P(+a)$, then A is True (denoted as $+a$); otherwise A is False (denoted as $-a$). Suppose we use the following topological order: $A \to B \to C \to D$.

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}
0.466	0.175	0.432	0.915	0.594	0.842	0.596	0.570	0.209	0.862	0.980	0.522

5.2.1 Rejection Sampling

Write down the sampled values (e.g $+a, -b, \cdots$) in the following table using rejection sampling. We have already filled the first cell for you.

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}
$-a$											

What is the approximate probability $P(+d|+a)$ computed from these samples? __________

5.2.2 Likelihood Weighting

Write down the sampled values (e.g $+a, -b$) and the likelihood weights in the following table using likelihood weighting. (Hint: in likelihood weighting, we do not need to sample the evidence variable A, so we get 4 samples in total.)

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}
w_1:			w_2:			w_1:			w_3:		

What is the approximate probability $P(+d|+a)$ computed from these samples? __________

5.2.3 Pros and Cons

What are the pros and cons of rejection sampling and likelihood weighting? Please describe them in short text.

Solution:

1. Rejection sampling:

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}
-a	+a	-b	-c	+d	-a	-a	-a	-a	-a	-a	-a

$P(+d|+a) = 1.0$

2. Likelihood weighting:

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}
-b	+c	-d	-b	-c	-d	-b	-c	+d	-b	-c	+d
w_1: 0.2			w_2: 0.2			w_1: 0.2			w_3: 0.2		

$P(+d|+a) = \frac{0.2*2}{0.2*4} = 0.5$

3. Pros and Cons:
Cons of Reject sampling: Reject Sampling may reject lots of sample when evidence is unlikely, also reflect the Pros of Likelihood weighting.