Exam 2 - Requires Respondus LockDown Browser + Webcam

Due Apr 8 at 11:40am **Points** 15 **Questions** 8
Available Apr 8 at 10:30am - Apr 8 at 11:45am about 1 hour **Time Limit** 60 Minutes
Requires Respondus LockDown Browser

This quiz was locked Apr 8 at 11:45am.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	60 minutes	12.5 out of 15

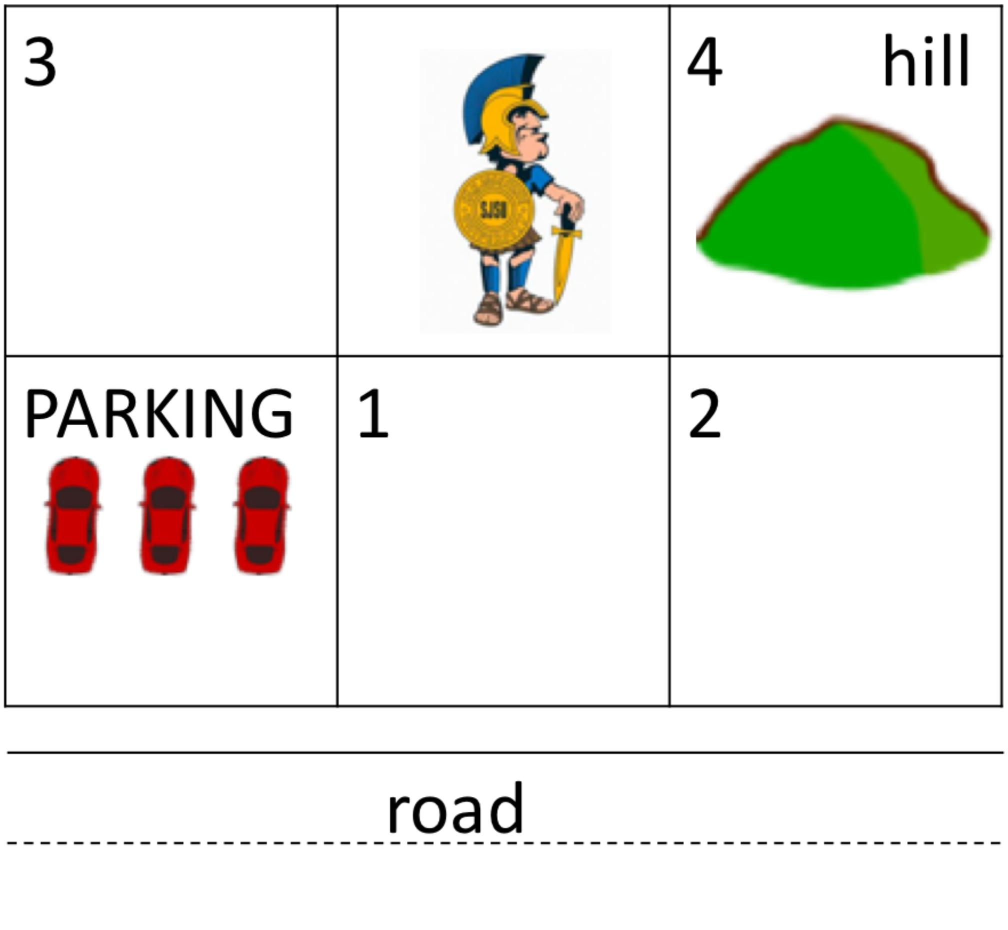
Score for this quiz: **12.5** out of 15
Submitted Apr 8 at 11:34am
This attempt took 60 minutes.

Question 1

3.5 / 4 pts

Our task is to design the layout of a small college.

The campus will have 4 structures: an administration building (A), a bus stop (B), classrooms (C) and dorms (D). We'll also have a gigantic statue of the college mascot, a blue and yellow Spartan and a parking lot. Each structure (including the bus stop) must be placed in one of 4 available spaces on the grid shown below. These available spaces are numbered 1 through 4.



Adjacent spaces are spaces that share a side, so space 1 is adjacent to space 2 but not to space 3 or 4.

The layout must satisfy the following constraints:

- The administration structure (A) must be adjacent to the bus stop (B).
- The bus stop (B) must be adjacent to the road.
- The dorms must not be adjacent to the road.
- The administration structure (A) must not be on a hill.
- All structures must be in different grid squares.

Our goal is to solve this design problem as a CSP.

We first enforce node consistency.

What are the domains of the variables, A, B, C and D after enforcing node consistency?

Please list each domain as a set: for example {1,2,3,4}

Domain of variable A after enforcing node consistency:

{1,2,3}

Domain of variable B after enforcing node consistency:

{1,2}

Domain of variable C after enforcing node consistency:

{1,2,3,4}

Domain of variable D after enforcing node consistency:

{3,4}

The next step is to apply the AC-3 algorithm and enforce the consistency of the whole CSP.

We start with arc A->B.

What is the domain of the variable A after the consistency of the A->B is enforced? {1,2}

We then enforce the consistency of arc B->A

What is the domain of the variable B after the consistency of the arc B->A is enforced? {1,2}

What is the domain of the variable C after running the AC-3 algorithm to completion? {3,4}

What is the domain of the variable D after running the AC-3 algorithm to completion? {3,4}

Answer 1:

Correct! {1,2,3}

orrect Answer {1, 2, 3}

Answer 2:

Correct! {1,2}

orrect Answer {1, 2}

Answer 3:

Correct! {1,2,3,4}

orrect Answer {1, 2, 3, 4}

Answer 4:

Correct! {3,4}

orrect Answer {3, 4}

Answer 5:

Correct! {1,2}

orrect Answer {1, 2}

Answer 6:

Correct! {1,2}

orrect Answer {1, 2}

Answer 7:

ou Answered {3,4}

orrect Answer {1, 2, 3, 4}

orrect Answer {1,2,3,4}

Answer 8:

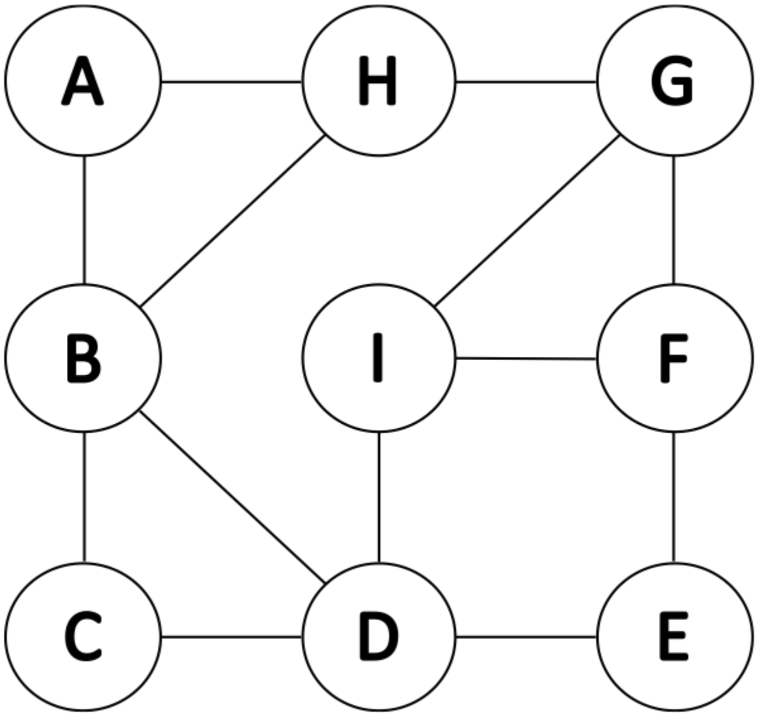
Correct! {3,4}

orrect Answer {3, 4}

Question 2

1 / 1 pts

Consider the binary constraint graph shown below.



We could like to apply cutset conditioning to solve the problem more efficiently.

What is the smallest cutset for this graph?

If the smallest cutset includes more than one node/variable, please list them separated by a comma.

Correct!

B, I

orrect Answers

I, B

B,I

I,B

B, I

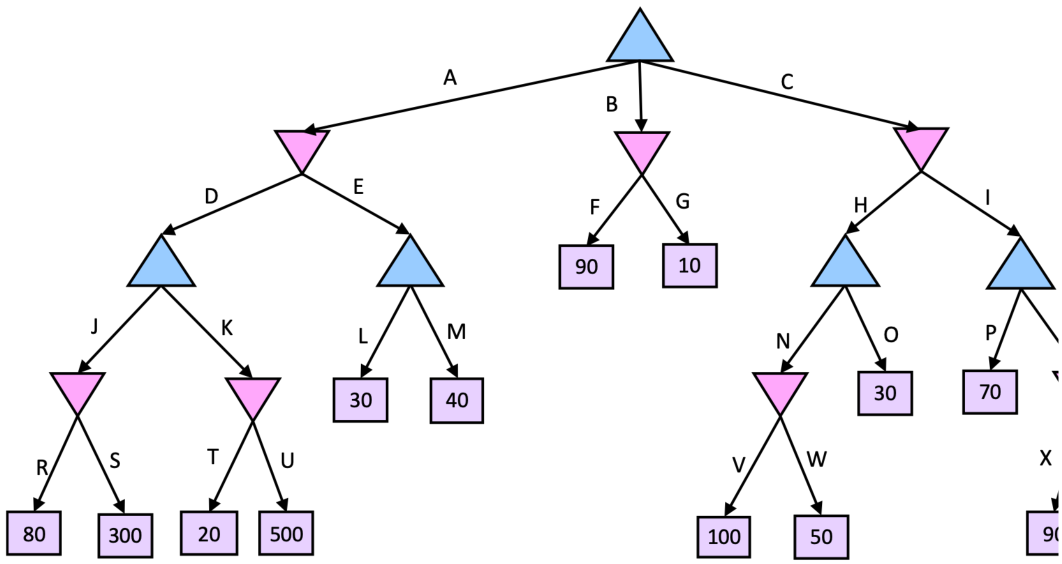
Question 3

2 / 2 pts

Consider the zero-sum game tree shown below.

Triangles that point up, such as at the top node (root), represent choices for the maximizing player (MAX); triangles that point down represent choices for the minimizing player (MIN).

Outcome values for the maximizing player are listed for each leaf node, represented by the values in the rectangles.



Which branch(es) will be pruned when we apply alpha-beta pruning?

If there is more than one branch, please enter them separated by a comma. If no branch is pruned, please write none. Note that if a major branch is pruned, all its sub-branches are also pruned; there is no need to list the sub-branches separately.

Correct!

U,Q

orrect Answers

U,Q

U, Q

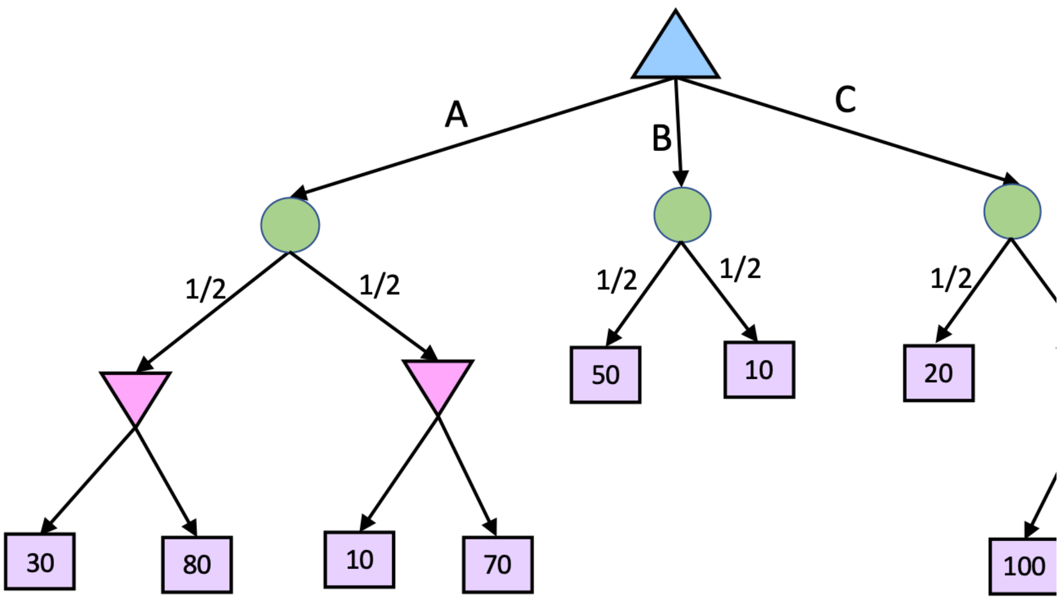
Question 4

2 / 2 pts

Consider the zero-sum game tree shown below.

Triangles that point up, such as at the top node (root), represent choices for the maximizing player (MAX); triangles that point down represent choices for the minimizing player (MIN). Circles represent chance nodes.

Outcome values for the maximizing player are listed for each leaf node, represented by the values in rectangles. The outcomes of the chance nodes all have equal probabilities.



What is the value at the root? 30

Which of the following actions should the maximizing player (MAX) take:

A, B or C? B

Answer 1:

Correct!

30

Answer 2:

Correct!

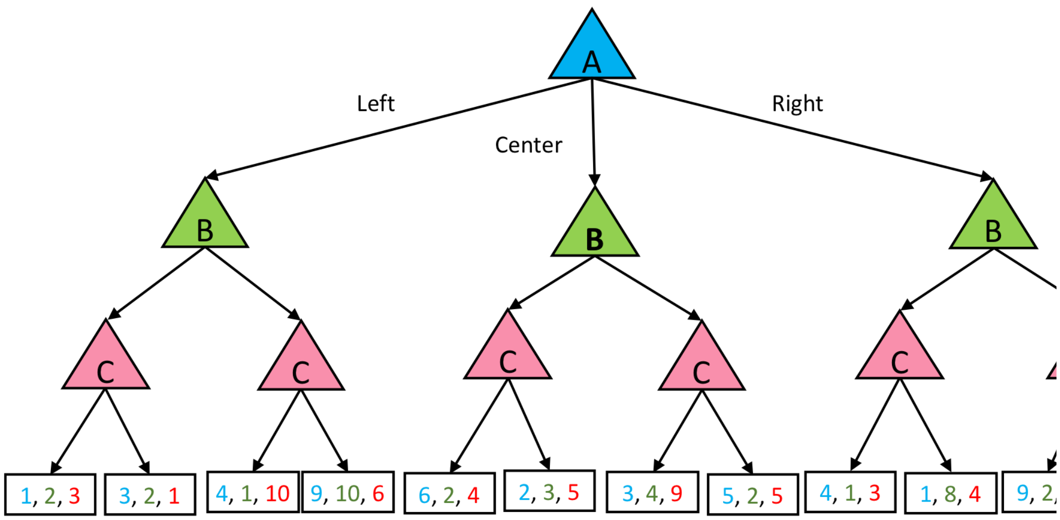
B

Question 5

0 / 1 pts

Consider the non zero-sum game tree with 3 players A, B, and C.

The leaf utilities are written as tuples (U_A, U_B, $U_{C,}$), with the first component U_A denoting the utility of that leaf to A, the second component U_B denoting the utility of that leaf to B and the third component U_C denoting the utility of that leaf to C. A seeks to maximize U_A, B seeks to maximize U_B, and C seeks to maximize U_C.



Assuming that all the players act optimally, what is the best utility that player A can achieve? 10

Which of the following actions should player A take: Left, Center or Right? Left

Answer 1:

ou Answered 10

orrect Answer 3

Answer 2:

ou Answered | Left

orrect Answer | Center

orrect Answer | center

Question 6

1 / 1 pts

Skolemize the following first-order sentence:

∀ y (∃ x Even(x) ∧ Greater(x, y))

You may copy and paste the symbols you need:

∧ ∨ ¬ ⇒ ⇔ ∀ ∃ = ≠

Your Answer:

subst({x/F(y), Even(x) ∧ Greater(x,y))

we get: ∀ y Even(F(y) ∧ Greater(F(y),y) when F(y) could be greater function

Question 7

2 / 2 pts

Convert the following sentence to the Conjunctive Normal Form.

∀x ((Baby(x) ∧ Hungry(x)) => Crying(x))

Please show all 6 steps for full credit.

You may copy and paste the symbols you need:

∧ ∨ ¬ ⇒ ⇔ ∀ ∃

The standard logical equivalence table is included below for your convenience:

$(\alpha \land \beta)$	$\equiv$	$(\beta \land \alpha)$	commutativity of $\land$
$(\alpha \lor \beta)$	$\equiv$	$(\beta \lor \alpha)$	commutativity of $\lor$
$((\alpha \land \beta) \land \gamma)$	$\equiv$	$(\alpha \land (\beta \land \gamma))$	associativity of $\land$
$((\alpha \lor \beta) \lor \gamma)$	$\equiv$	$(\alpha \lor (\beta \lor \gamma))$	associativity of $\lor$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \lor \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha))$	biconditional eli
$\neg(\alpha \land \beta)$	$\equiv$	$(\neg\alpha \lor \neg\beta)$	De Morgan
$\neg(\alpha \lor \beta)$	$\equiv$	$(\neg\alpha \land \neg\beta)$	De Morgan
$(\alpha \land (\beta \lor \gamma))$	$\equiv$	$((\alpha \land \beta) \lor (\alpha \land \gamma))$	distributivity of $\land$ o
$(\alpha \lor (\beta \land \gamma))$	$\equiv$	$((\alpha \lor \beta) \land (\alpha \lor \gamma))$	distributivity of $\lor$ o

Your Answer:

∀x ((Baby(x) ∧ Hungry(x)) => Crying(x))

1. ∀x - (Baby(x) ∧ Hungry(x)) ∨ Crying(x)

2. ∀x - Baby(x) ∨ - Hungry(x) ∨ Crying(x)

3. standalize variable (no applicable)

4. skolize(not applicable)

5. Drop universal: - Baby(x) ∨ - Hungry(x) ∨ Crying(x)

6. Apply distributivity (no applicable)

CNF : - Baby(x) ∨ - Hungry(x) ∨ Crying(x)

Question 8

1 / 2 pts

Consider the following Knowledge Base that has already been converted to CNF.

¬Sleeping(x) V ¬InOffice(x) (Clause 1)
Busy(Anna) V Sleeping(Anna) (Clause 2)

InOffice(Anna) (Clause 3)

Your task is to prove **by resolution** that Anna is busy:

Busy(Anna)

Show all the steps for full credit.

Make sure you identify all the clauses (number them) and the unifiers used in the resolution algorithm. List the resulting clause (the resolvent) of each resolution step.

You may copy and paste the symbols you need:

∧ ∨ ¬ ⇒ ⇔ ∀ ∃

Your Answer:

Clause 3 and clause 1:

InOffice(Anna) ∧ ¬Sleeping(x) V ¬InOffice(x)

subst (x/ Anna, InOffice(Anna) ∧ ¬Sleeping(x) V ¬InOffice(x))

apply resolution, we have : -Sleeping(Anna) (clause 4)

Clause 4 and clause 2:

Busy(Anna) V Sleeping(Anna) ∧ -Sleeping(Anna)

apply resolution, we have Busy(Anna) ==> we proved Busy(Anna)

We need to add not Busy(Anna) and derive an empty clause.

Quiz Score: **12.5** out of 15