CS-171, Intro to A.I., SS-1, 2018 — Quiz # 3 — 20 minutes

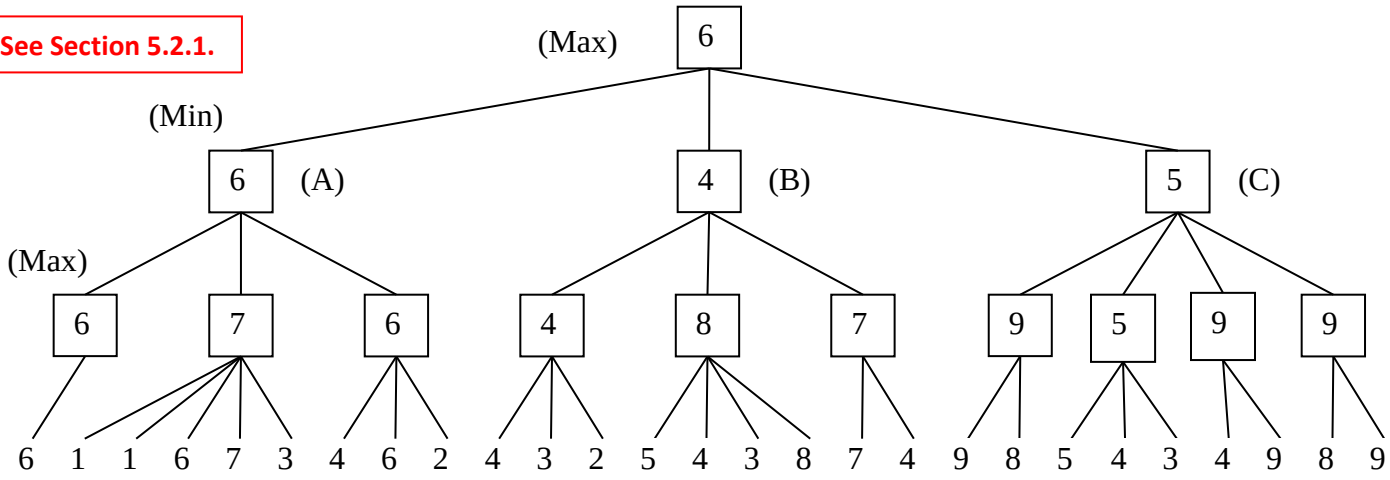
NAME: _____

YOUR ID: _____ ID TO YOUR RIGHT: _____ ROW NO.: _____ SEAT NO.: _____

1. (15 pts total, -5 pts for each error, but not negative) MINI-MAX SEARCH IN GAME TREES.
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 (number at bottom) is the estimated score returned by the heuristic static evaluator.

1.a. Fill in each blank square with the proper mini-max search value.
1.b. What is the best move for Max? (write A, B, or C) A
1.c. What score does Max expect to achieve? 6

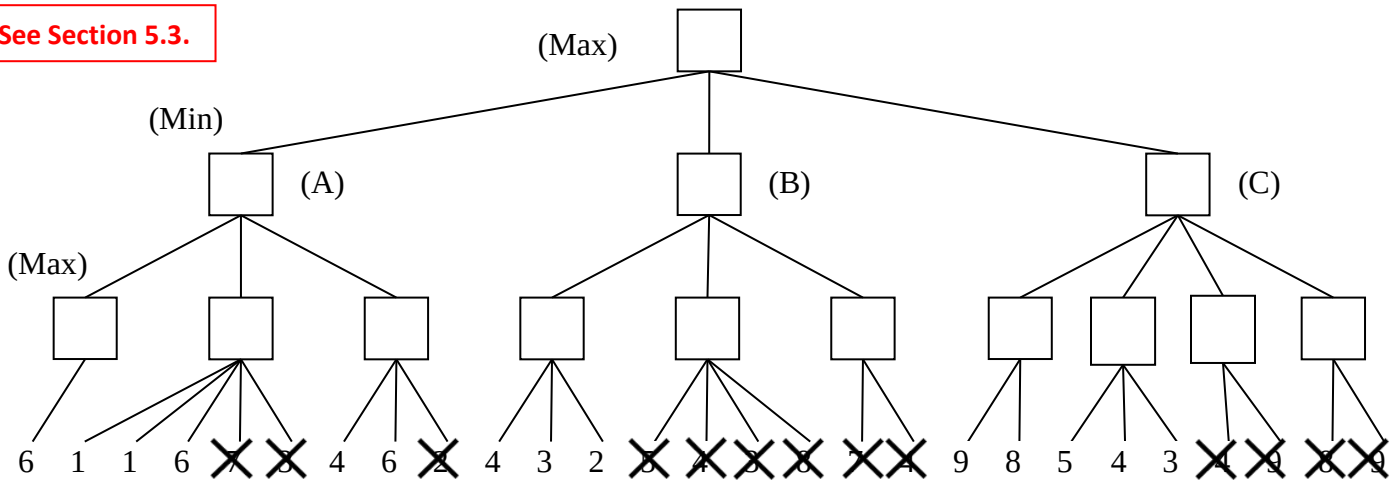
See Section 5.2.1.



2. (35 pts total, -5 for each error, but not negative) ALPHA-BETA PRUNING. Process the tree left-to-right. This is the same tree as above (1.a). You do not need to indicate the branch node values again.

Draw X over each leaf node (number at bottom) that will be pruned by Alpha-Beta Pruning.

See Section 5.3.



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

3. (25 pts total, 5 pts each) Unifiers and Unification.
Write the **most general unifier** (or MGU) of the two terms given, or "None" if no unification is possible. Write your answer in the form of a substitution as given in your book, e.g., the substitution {*x / John, y / Mary, z / Bill*} means substitute *x* by *John,* substitute *y* by *Mary*, and substitute *z* by *Bill.*
The first one is done for you as an example.

3.a. **(example)** UNIFY(*Knows(John, x), Knows(John, Jane)*) { *x / Jane* }

3.b. **(5 pts)** UNIFY(*Knows(John, x), Knows(y, Jane)*) { *x / Jane, y / John* }

3.c. (5 pts) UNIFY(*Knows(John, x), Knows(y, Father (y))*) { y / John, x / Father (John) }

3.d. (5 pts) UNIFY(*Knows(John, F(x)), Knows(y, F(F(z)))*) { *y / John, x / F (z)* }

3.e. (5 pts) UNIFY(*Knows(John, F(x)), Knows(y, G(z))*) None

3.f. (5 pts) UNIFY(*Knows(John, F(x)), Knows(y, F(G(y)))*) { *y / John, x / G (John)* }

4. (25 pts total, 5 pts each) Quantifiers.
In this problem, Likes(A, B) means A likes B, and Sister(A, B) means A is a sister of B. Single-argument predicates have their intended meaning; e.g., Cat(A) means A is a cat, etc. Fill in each blank below with Y (= Yes) or N (= No) depending on whether the first order logic sentence correctly expresses the English sentence.
The first one is done for you as an example.

4.a. **(example)** N "All cats are mammals."
$\forall x\ Cat(x) \wedge Mammal(x)$
should be $\forall x\ Cat(x) \Rightarrow Mammal(x)$

4.b. **(5 pts)** Y "Spot has a sister who is a cat."
$\exists x\ Sister(x, Spot) \wedge Cat(x)$

4.c. (5 pts) N "Every person has someone that they like."
$\exists x\ Person(x) \wedge (\forall y\ Person(y) \Rightarrow Likes(x, y))$

4.d. (5 pts) N "There is someone who likes everyone."
$\forall x\ Person(x) \Rightarrow (\exists y\ Person(y) \wedge Likes(x, y))$

4.e. (5 pts) Y "Everyone likes ice cream."
$\forall x\ (Person(x) \Rightarrow Likes(x, IceCream)) \equiv \neg \exists x \neg (Person(x) \Rightarrow Likes(x, IceCream))$

4.f. (5 pts) Y "All men are mortal."
$\forall x\ Man(x) \Rightarrow Mortal(x)$