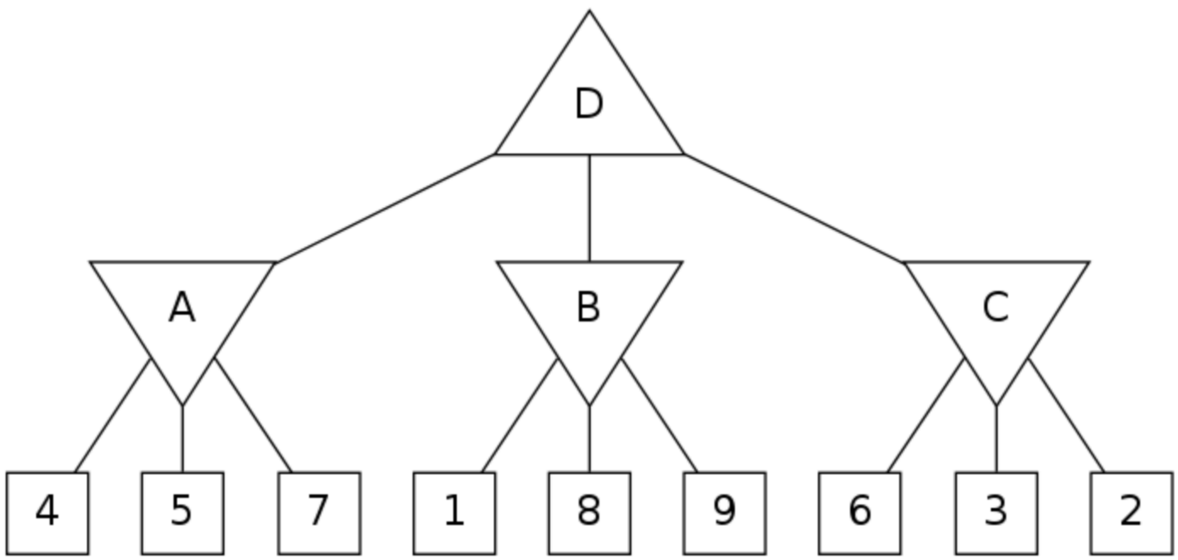
CS 4499/5599 HW 3

1 Minimax

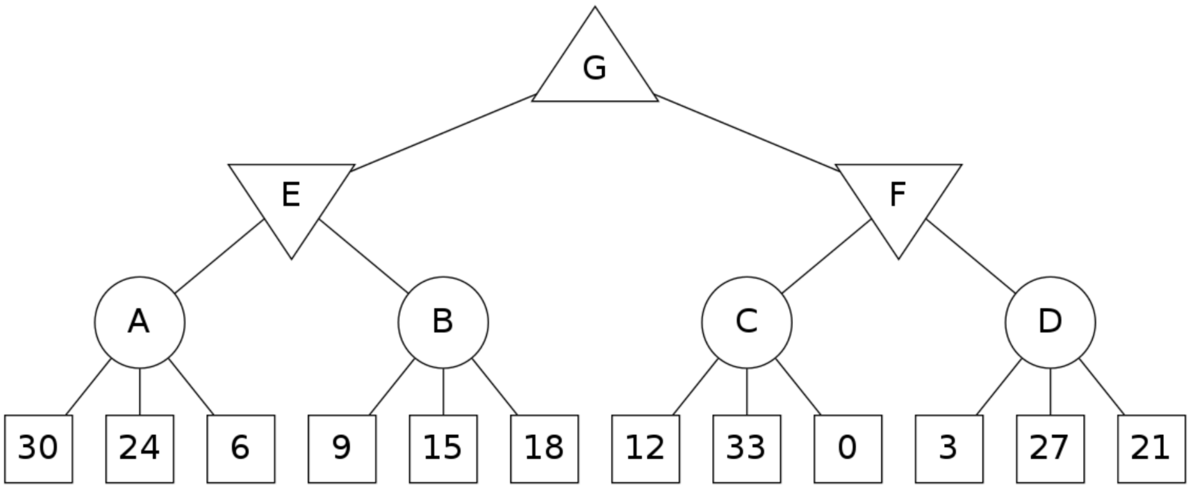
Consider the zero-sum game tree shown below. Triangles that point up, such as at the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player. Outcome values for the maximizing player are listed for each leaf node, represented by the values in squares at the bottom of the tree. Assuming both players act optimally, carry out the minimax search algorithm. Give the values for the letter nodes.



A	B	C	D

2 Expectimax

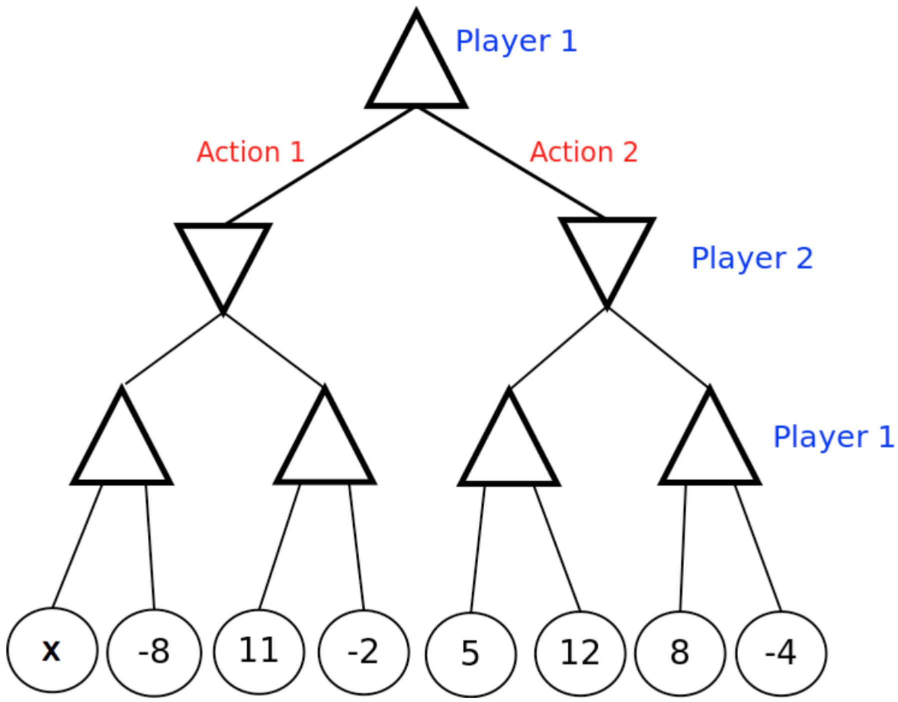
Consider the game tree shown below. As in the previous problem, triangles that point up, such as the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player. The circular nodes represent chance nodes in which each of the possible actions may be taken with equal probability. The square nodes at the bottom represent leaf nodes. Assuming both players act optimally, carry out the expectimax search algorithm. Enter the values for the letter nodes in the boxes below the tree.



A	B	C	D	E	F	G

3 Unknown Leaf Value

Consider the following game tree, where one of the leaves has an unknown payoff, x. Player 1 moves first, and attempts to maximize the value of the game.



Each of the next 3 questions asks you to write a constraint on x specifying the set of values it can take. In your constraints, you can use the letter x, integers, and the symbols > and <. If x has no possible values, write 'None'. If x can take on all values, write 'All'. As an example, if you think x can take on all values larger than 16, you should enter x > 16.

Assume Player 2 is a minimizing agent (and Player 1 knows this). For what values of x is Player 1 guaranteed to choose Action 1?

Assume Player 2 chooses actions at random with each action having equal probability (and Player 1 knows this). For what values of x is Player 1 guaranteed to choose Action 1?

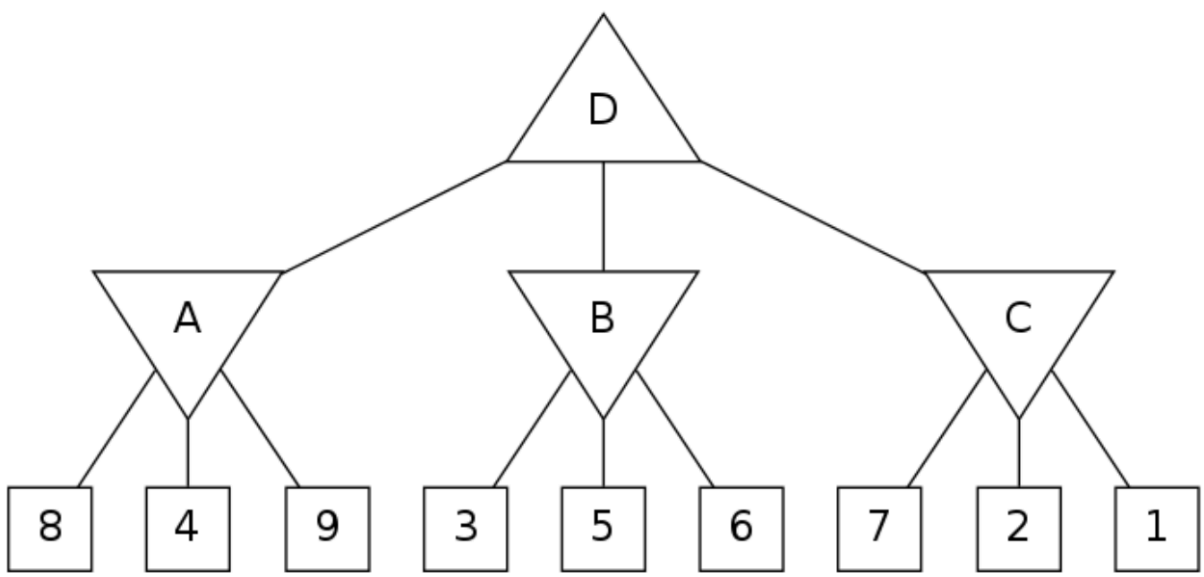
Denote the minimax value of the tree as the value of the root when Player 1 is the maximizer and Player 2 is the minimizer. Denote the expectimax value of the tree as the value of the root when Player 1 is the maximizer and Player 2 chooses actions at random (with equal probability). For what values of x is the minimax value of the tree worth more than the expectimax value of the tree?

Is it possible to have a game, where the minimax value is strictly larger than the expectimax value?

4 Alpha-Beta Pruning

Consider the game tree shown below. Triangles that point up, such as at the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player. Assuming both players act optimally, use alpha-beta pruning to find the value of the root node. The search goes from left to right; when choosing which child to visit first, choose the left-most unvisited child. In the first set of boxes below, enter the values of the labeled nodes. Then, in the second set of boxes, enter a 'x' for the leaf nodes that don't get visited due to pruning. For leaf nodes that do get visited, leave the corresponding entry blank.

Hint: Note that the value of a node where pruning occurs is not necessarily the maximum or minimum (depending on which node) of its children. When you prune on conditions V > β or V < α, assume that the value of the node is V.



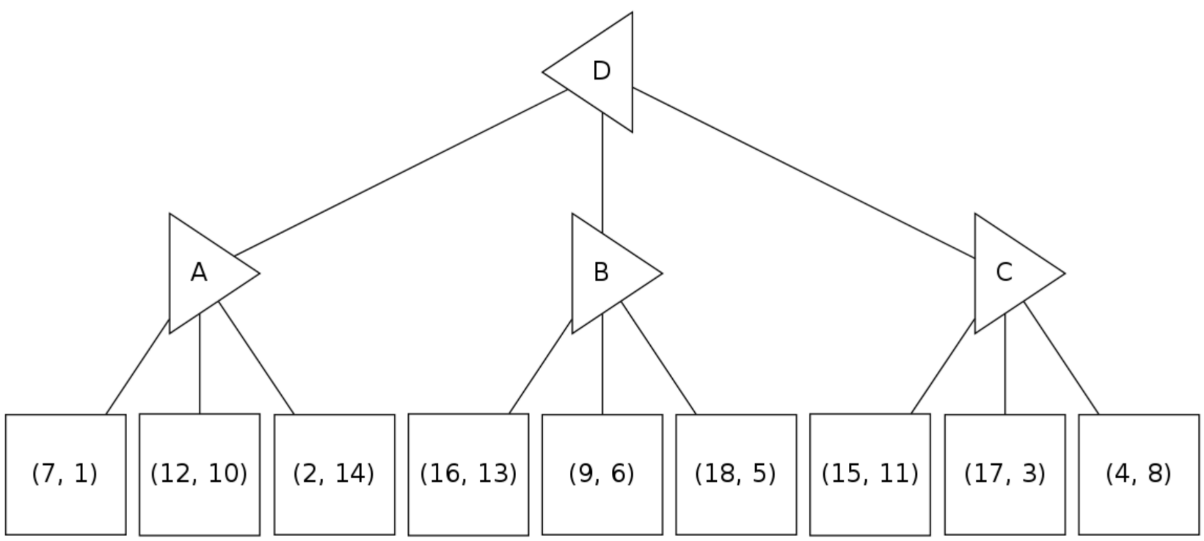
A	B	C	D

8	4	9	3	5	6	7	2	1

5 Non-Zero Sum Games

The standard minimax algorithm calculates worst-case values in a zero-sum two player game, i.e. a game for which in all terminal states *s*, the utilities for players A (MAX) and B (MIN) obey $U_A(s) + U_B(s) = 0$. In this zero-sum setting, we know that $U_A(s) = -U_B(s)$, so we can think of player B as simply minimizing U_A. In this problem, you will consider the non-zero-sum generalization, in which the sum of the two players' utilities are not necessarily zero. The leaf utilities are now written as pairs (U_A, U_B). In this generalized setting, A seeks to maximize U_A, the first component, while B seeks to **maximize** U_B, the second component.

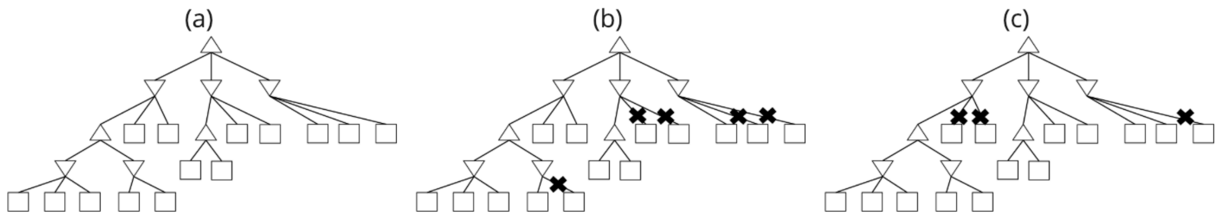
Consider the non-zero-sum game tree below. Note that left-pointing triangles (such as the root of the tree) correspond to player A, who maximizes the first component of the utility pair, whereas right-pointing triangles (nodes on the second layer) correspond to player B, who maximizes the second component of the utility pair. Propagate the terminal utility pairs up the tree using the appropriate generalization of the minimax algorithm on this game tree. In case of ties, choose the leftmost child. Select the correct values for the letter nodes below the tree. Your answer should be in the format (X, Y), where X is the value of Player A and Y is the value of Player B at a node.



A	B	C	D

6 Possible Pruning

Assume we run Alpha-Beta pruning, expanding successors from left to right, on a game with tree as shown in Figure (a) below.



For each of the following statements, indicate whether the statement is True or False:

A. There exists an assignment of utilities to the terminal nodes such that no pruning will be achieved (shown in Figure (a)).

B. There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (b) will be achieved.

C. There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (c) will be achieved.

Hint: Perhaps the simplest check is as follows: pruning of children of a minimizer node *m* is possible (for some assignment to the terminal nodes), when both of the following conditions are met: (i) the value of another child of *m* has already been determined, (ii) somewhere on the path from *m* to the root node, there is a maximizer node *M* for which an alternative option has already been explored. The pruning will then happen if any such alternative option for the maximizer had a higher value than the value of the "another child" of *m* for which the value was already determined.

7 Suboptimal Strategies

Player MAX and player MIN are playing a zero-sum game with a finite number of possible moves. MAX calculates the minimax value of the root to be *M*. You may assume that at every turn, each player has at least 2 possible actions. You may also assume that a different sequence of moves will always lead to a different score (i.e., no two terminal nodes have the same score).

For each of the following statements, indicate whether the statement is True or False:

A. Assume MIN is playing sub-optimally at every turn, but MAX does not know this. The outcome of the game could be larger than *M*.

B. Assume MIN is playing sub-optimally at every turn. If MAX plays according to the minimax strategy, the outcome of the game could be less than *M*.

C. Assume MIN is playing sub-optimally at every turn. MAX following the minimax policy will guarantee a better outcome than *M*.

D. Assume MIN is playing sub-optimally at every turn, and MAX knows exactly how MIN will play. There exists a policy for MAX to guarantee a better outcome than *M*.

8 Rationality of Utilities

Consider a lottery *L* = [0.2, A; 0.3, B; 0.4, C; 0.1, D], where the utility values of each of the outcomes are *U(A) = 1*, *U(B) = 3*, *U(C) = 5*, and *U(D) = 2* . What is the utility of this lottery, *U(L)*?

Consider a lottery *L1* = [0.5, *A*; 0.5, *L2*], where *U(A) = 4*, and *L2* = [0.5, *X*; 0.5, *Y*] is a lottery, and *U(X) = 4*, *U(Y) = 8*. What is the utility of the the first lottery, *U(L1)*?

Assume A > B, B > L, where L = [0.5, C; 0.5, D], and D > A. Assuming rational preferences, which of the following statements are guaranteed to be true?

A. A > L
B. A > C
C. A > D
D. B > C
E. B > D