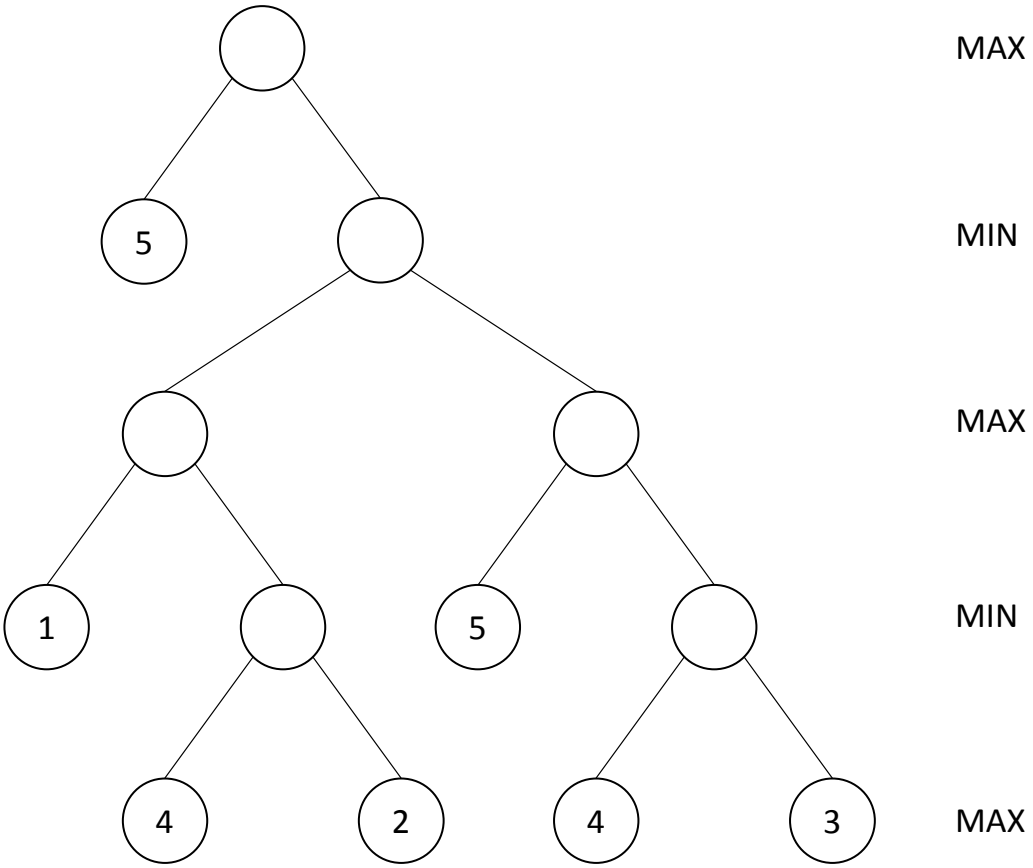
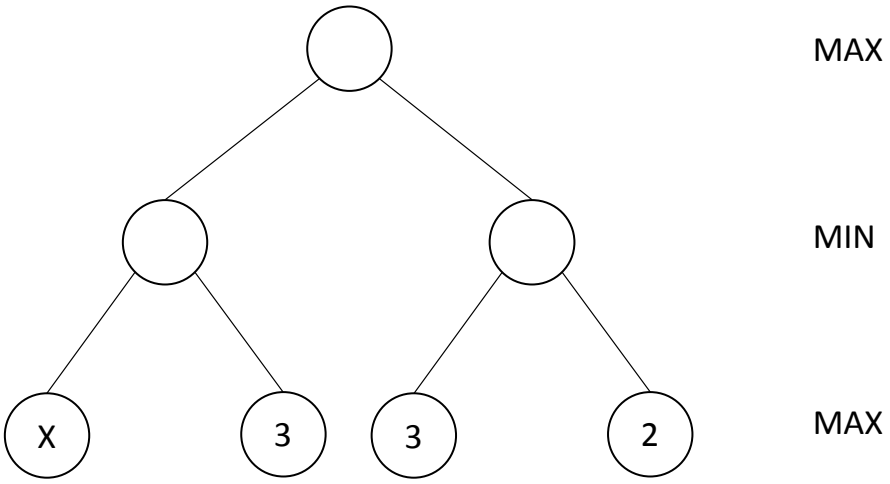
Adversarial Search

1) What is the minimax value of the root node for the game tree below? Cross out the node(s) whose value(s) the alpha-beta method never determines, assuming that it performs a depth-first search that always generates the leftmost child node first and a loss (and win) of MAX (and MIN) corresponds to a value of $-\infty$ (and ∞, respectively). Determine the alpha and beta values of the remaining node(s).



2) Assume that you are given a version of the alpha-beta method that is able to take advantage of the information that all node values are integers that are at least 1 and at most 6. Determine ALL values for X that require the algorithm to determine the values of ALL nodes of the following game tree, assuming that the alpha-beta method performs a depth-first search that always generates the leftmost child node first.



3) The minimax algorithm returns the best move for MAX under the assumption that MIN plays optimally. What happens if MIN plays suboptimally? Is it still a good idea to use the minimax algorithm?

4) In order to perform a move in the game a player has to roll a die. If the outcome is odd (1,3 or 5) the player has three possible moves that will reach states with utilities: 4, 5 and 10. If the outcome of the die is 2 or 4, the player has two possible moves that will reach states with utilities: 4, and 8. Finally, if the outcome of the die is 6, there are three moves that reach states with utilities 5, 7 and 9. Draw the expectiminimax tree and calculate the expectiminimax value of the root node given that:

a) The player is a maximum player.

b) The player is a minimum player.