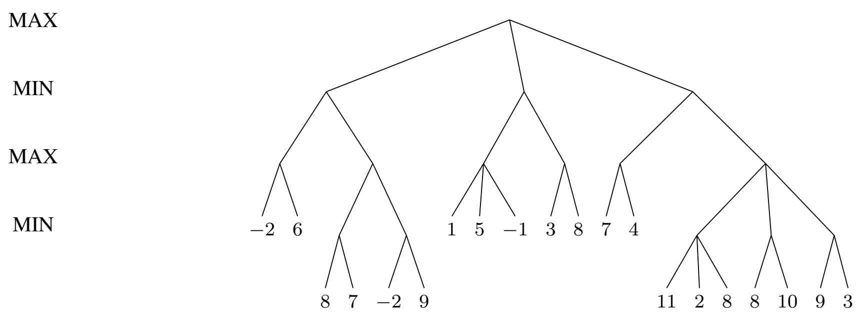
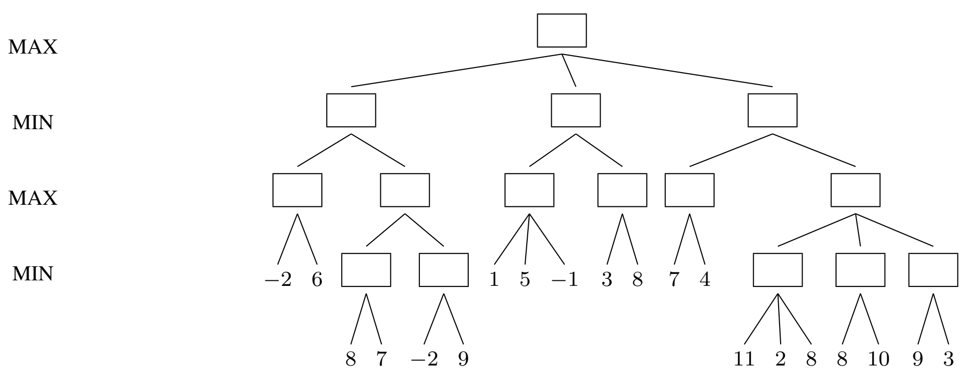
Exercise 1 :

Consider the following game tree corresponding to a two-player zero-sum game as specified in the lecture. As usual, **Max is to start in the initial state (i.e., the root of the tree)**. For the following algorithms, the expansion order is **from left to right, i.e., in each node the left-most branch is expanded first.**



1. In the following tree, perform Minimax search, i.e., annotate all internal nodes with the correct Minimax value. Which move does Max choose?



2. Consider, again, our game tree given above. Max chooses the action with the highest utility in the root of the tree. What can you say about the utility he will receive against:
   - An optimally playing opponent
   - A non-optimal playing opponent

   Note: Max always plays optimally.

Exercise 2 :

Max and Minnie are playing a classic children's game called Tic-tac-toe in which players take turns marking the cells in a 3×3 grid. Max marks Xs and Minnie marks Os. Minnie wins with utility -1 if **any line (horizontal, vertical, or diagonal)** fills up with three Os, whereas Max wins with utility $+1$ if **any** line fills up with three Xs. If there are no empty cells left and no one has won so far, the game ends in a draw with utility 0.

(a) Consider the state depicted below. Here, it is Max's turn to play.

x		o
x	o	
		o

Draw the **full** Minimax tree and annotate every node with its utility.

(b) Consider the evaluation function $f(x) :=$ the number of (horizontal) rows which contain at most one O. For example, in the initial state $f(x) = 3$.
Draw the Minimax tree with this evaluation function and a depth of 2. Annotate every node with its utility.

(c) Assuming a perfectly playing opponent, Minimax search without a depth limit will always guarantee a draw. However, it is not obvious whether this guarantee can still be made when imposing certain depth limits in combination with certain evaluation functions. As an example, consider the simple evaluation function g aimed to detect situations in which the opponent can win in one step (if it is their turn).

$$g(x) := \left\{ \begin{array}{rl} -1, & \text{if there is a line (horizontal, vertical, or diagonal) with} \\ & \text{two opponent marks and an empty field} \\ 0, & \text{otherwise} \end{array} \right\}$$

Now prove or disprove the following claim:
Running Minimax search with a depth limit of 3 and evaluation function g is sufficient to guarantee a draw against a perfectly playing opponent in a 3×3 Tic-tac-toe game. You may **not** assume that you are allowed to start the game.

Exercise 3 :

Please decide for each of the following statements whether it is true or false and justify your answer (1-3 sentences per statement).

1. A two-player zero-sum game has exactly three possible outcomes in terms of its utility function.
2. Full Minimax search always yields the best possible outcome in terms of its utility, no matter how the opponent plays.
3. Zero-sum games always have a Nash Equilibria
4. Zero-sum games always have a Nash Equilibria consisting of pure strategies
5. Non-Zero-sum games always have a Nash Equilibria

Exercise 4 :

Consider the following game representation in normal form:

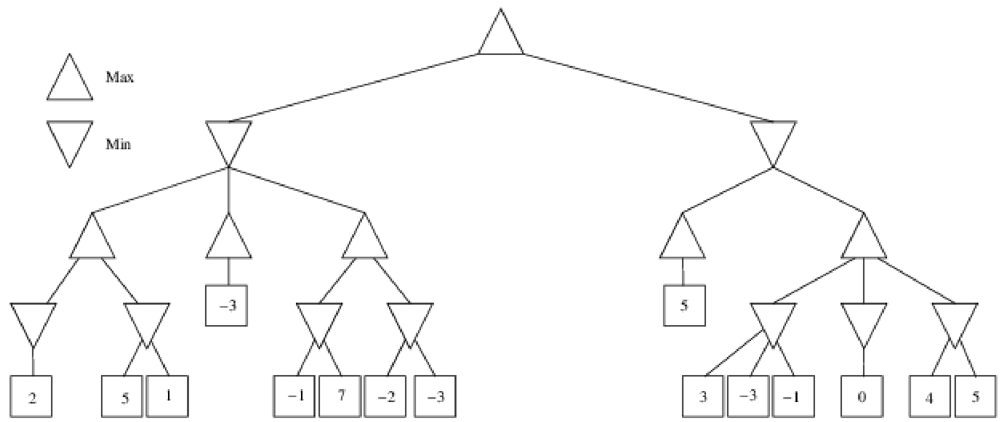
	Andy		
Barb	a_1	a_2	a_3
b_1	2 0	1 0	2 2
b_2	2 0	1 1	0 0
b_3	2 1	0 0	0 2
b_4	2 0	0 0	0 2
b_5	0 0	1 1	0 2
b_6	0 0	1 1	0 0

The matrix shows the utilities for Andy (red numbers) and Barb (green numbers) for each combinations of strategies they can choose (Andy has 3 strategies to choose from, Barb has 6).

- Determine at least two Nash equilibria consisting of pure strategies for this game.
- Show that there is no Nash equilibrium where Barb plays b_4, and Andy plays any (possibly mixed) strategy.

Exercise 5 :

Consider the following game tree:



a. Compute the utility values for all nodes.

b. If the utility values are computed in a depth-first order that always considers branches in left-to-right order, which nodes can be pruned, i.e. for which nodes is it not required to compute the utility value in order to determine the optimal strategy for both players?

c. For each node in the game tree, determine the ordering of the outgoing branches that is optimal in the following sense: if utility values are computed for nodes in that order, then a maximal number of nodes can be pruned in the utility computation.
