Exercises Lecture 4
Intelligent System Programming (ISP)

Exercise 1

“Osero” is a Japanese board game similar to “go”. There are two players, one with black pieces and one with white pieces. The players take turns and in each turn, a player places a single piece on a square of a quadratic board. A piece can only be placed next to a piece that is vertically or horizontally adjacent to a piece already on the board. If the placement of a piece causes some of the opponent’s pieces to lie between opposite colored pieces either in the vertical or horizontal direction, they are “captured” and shift color. The game can be terminated at any time, and the utility of a game state for a player equals the number of pieces of the player’s color.

As an example, in the game state shown below to the left, white can place a piece in b4, c5, d5, e4, d3, c2, b1, a2, and a3. If white places a piece in b4, the resulting game state is the one shown below to the right.

5					
4			●	○	
3		●	●		
2		○			
1					
	a	b	c	d	e

5					
4		○	○	○	
3		○	●		
2		○			
1					
	a	b	c	d	e

The utility for black is 3 for the left state and 1 for the right.

Assume that MAX plays black and consider the initial state shown below

5	●				
4	○				
3					
2					
1					
	a	b	c	d	e

a) Draw the game tree from the initial state to a depth of 2 ply (i.e., a single move of MAX and a single move of MIN).

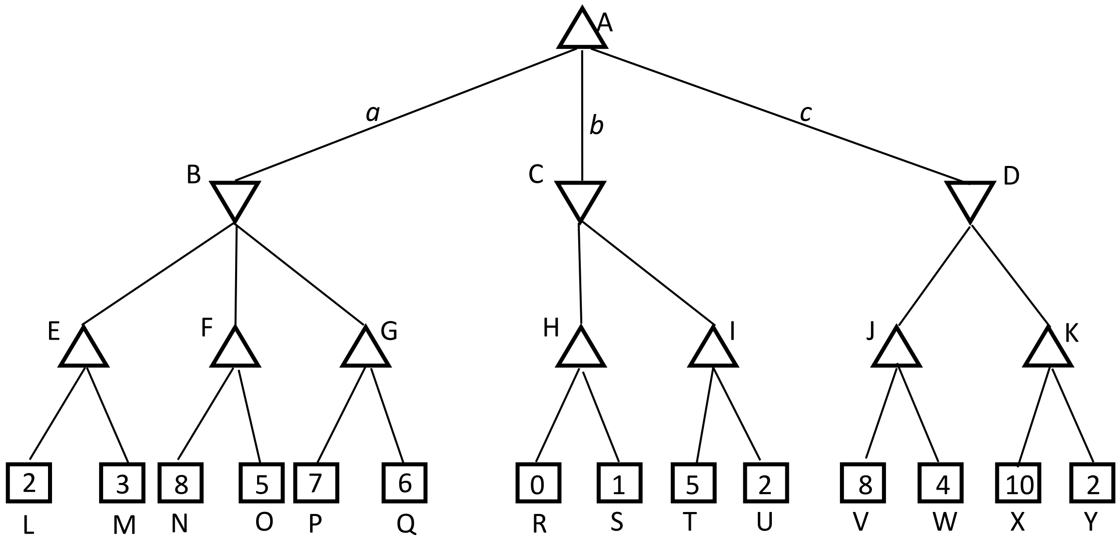
b) What is the minimax value of the initial state?

c) What is the minimax decision of MAX?

Exercise 2

(Adapted from exercise 12.2, Nils J. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann, 1998).

Consider the following game tree in which the static scores (numbers in leaf boxes) are all from the MAX player's point of view.



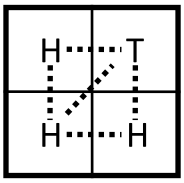
a) What action should the MAX player choose?
b) What nodes would not need to be examined using MiniMax with α-β pruning – assuming that nodes are examined in the left-to-right order?

Exercise 3 (adapted from RN10 exercise 5.7)

Argue that the utility obtained by MAX using MINIMAX decisions against a suboptimal MIN will never be lower than the utility obtained playing against an optimal MIN. Can you come up with a game tree in which MAX can do still better using a *suboptimal* strategy against a suboptimal MIN?

Mandatory assignment

Consider a simplified tic-tac-toe game with four positions as shown below.



Assume that Max and Min play with circle and cross tokens, respectively. The game is played using a coin with 50% chance of landing on each side. The rules are as follows:

1. Initially the board is empty.
2. A player starts his turn by flipping the coin.
3. If the coin shows heads, the player can place a token on positions marked H, otherwise the player must place a token on the position marked T.
4. The game terminates when
    a. a player is unable to place a token, because there are no empty positions left with the right mark, or
    b. a player wins the game.
5. A player wins the game, if the player has two tokens on any of the 5 pair of positions connected by a dashed line.
6. The utility of the game is 1if Max wins, -1 if Min wins, and 0 otherwise.

1) Modify the algorithm MiniMax-Decision to the ExpectiMiniMax algorithm, and write its complete pseudo code.
2) Assume as usual that Max starts the game and that Max just flipped the coin and got heads.
    a. Carefully draw the complete game tree on an A3 paper.
    b. What is the Expectiminimax decision for Max?