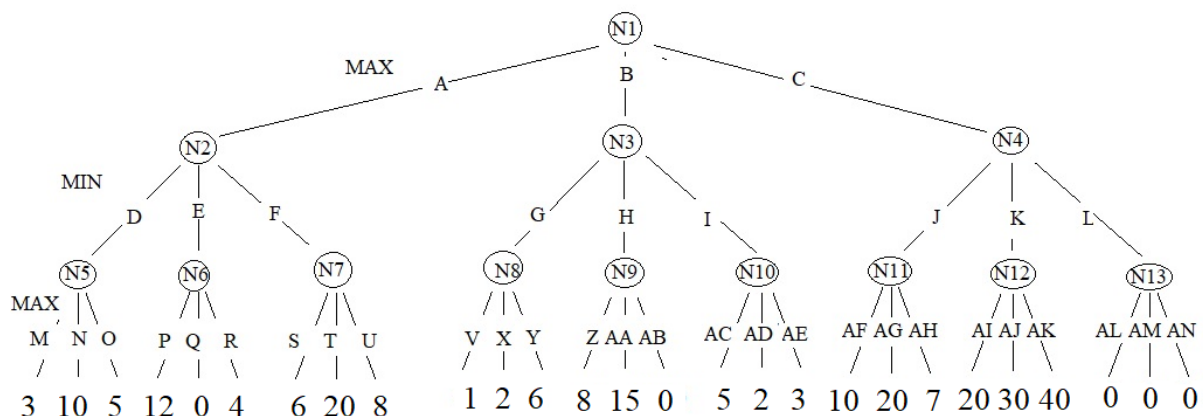
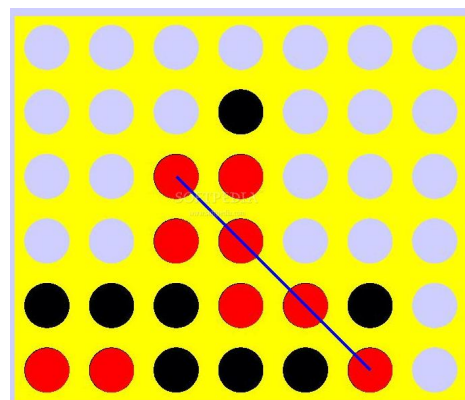


## Search with Opponents/Games

Apply the Minimax algorithm with alpha-beta cuts to the following tree that has a branch factor of 3 at the top level, 3 at the second level, and also 3 at the final level, and with the evaluation function values indicated for the final line. Indicate the final value of each node and which branches are cut by the Alpha-Beta cuts.



The game called "Connect Four" in the English language version ("4 em Linha" in the Portuguese version - [https://en.wikipedia.org/wiki/Connect\\_Four](https://en.wikipedia.org/wiki/Connect_Four)) is played on a vertical board of 7x6 squares (i.e., 7 squares wide and 6 squares high), by two players, to which are initially assigned 21 pieces to each (one of the players has white pieces and the other black pieces, or pieces "X" vs pieces "O").



The two players play alternately one of their pieces. The piece to be played is placed on the top of the board and slides either to the base of the board, or in a cell immediately above

another one already occupied (see previous figure). The winner will be the player who manages to obtain a line of 4 pieces of its color/symbol horizontally, vertically, or diagonally. If the 42 pieces are played without any player getting a line, the final result will be a draw.

- a) Formulate this game as a search problem with opponents, indicating the state representation, moves/operators (and respective names, preconditions, and effects), and the objective test.
- b) Implement a simple version of the “Connect-Four” game in a programming language of your choice.
- c) Implement the following functions:
  - c1) *int nlines4 (int Player)* that given the state of the board calculates the number of lines with 4 pieces (horizontal, vertical, diagonal) of a given player.
  - c2) *int nlines3 (int Player)*, similar to the previous function, but which calculates the number of sets of 4 consecutive spots that have three pieces of the player followed by an empty spot, i.e., that are possibilities to win the game.
  - c3) *int central (int Player)*, that assigns 2 points to each player piece in the center column of the board (column 4) and 1 point to each piece in the columns around it (columns 3 and 5).
- d) Implement an agent to play the game using the minimax algorithm with alpha-beta cuts.
- e) Compare the results of the implemented agents, playing 10 matches of this game with each other, using the minimax algorithm with alpha-beta cuts, with levels (2, 4, 6 and 8), and the following evaluation functions:  
*Agent1: EvalF1 = nlines4(1) - nlines4(2)*  
*Agent2: EvalF2 = 100\* EvalF1 + nlines3(1) - nlines3(2)*  
*Agent3: EvalF3 = 100\* EvalF1 + central(1) - central(2)*  
*Agent4: EvalF4 = 5\* EvalF2 + EvalF3*
- f) Conclude about the effectiveness of each of the evaluation functions/agents and the effect of the depth used in the Minimax Algorithm.
- g) How could you improve the evaluation function for this type of agent?