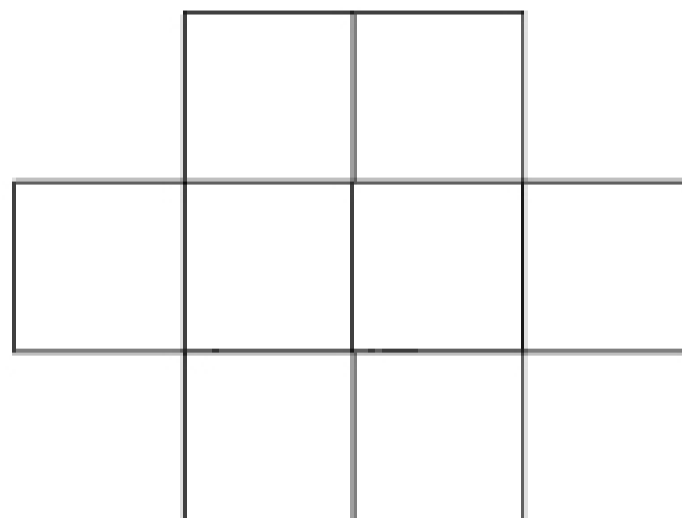


Homework #4

(Due Th. May 10)

CSP

- 1) Consider the following binary constraint network: There are 4 variables: X_1 , X_2 , X_3 , X_4 with the domains: $D_1=\{1,2,3,4\}$, $D_2=\{3,4,5,8,9\}$, $D_3=\{2,3,5,6,7,9\}$, $D_4=\{3,5,7,8,9\}$. The constraints are $X_1 \geq X_2$, $X_2 > X_3$ or $X_3 - X_2 = 2$, $X_3 \neq X_4$.
 - a. Draw the constraint graph.
 - b. Is the network arc-consistent? If not, compute the arc-consistent network. (show the whole process of enforcing arc-consistency and not just the final network)
 - c. Is the network consistent? If yes, give a solution.
- 2) Consider the 8 squares positioned as follows:



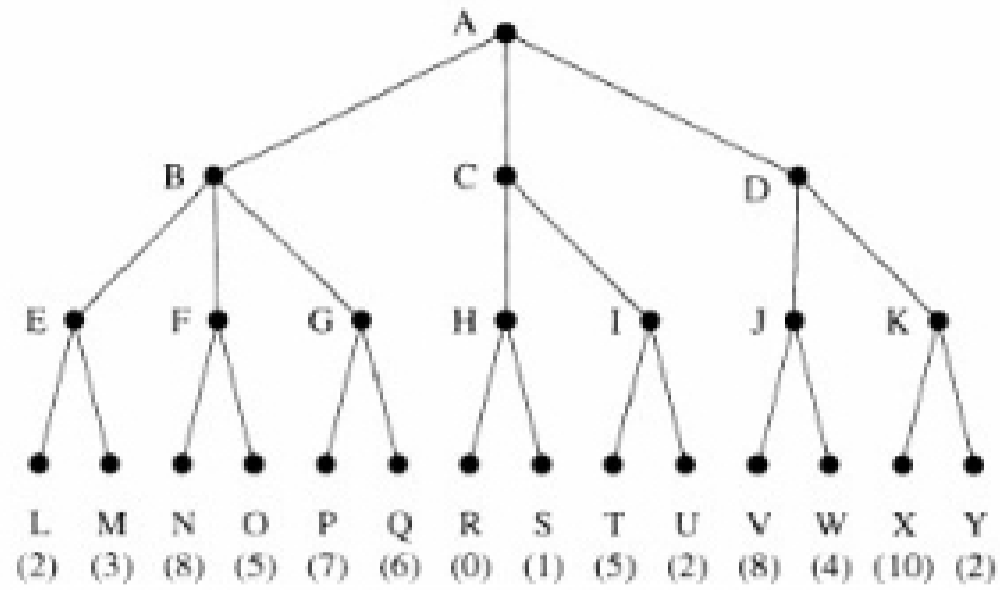
The task is to label the boxes above with the numbers 1-8 such that the labels of any pair of adjacent squares (i.e. horizontal, vertical or diagonal) differ by at least 2 (i.e. 2 or more).

- a. Write all constraints and draw the constraint graph.
- b. Is the network arc-consistent? If not, compute the arc-consistent network. (show the whole process of enforcing arc-consistency and not just the final arc-consistent network)

- c. Is the network consistent? If yes, give a solution.

Games

- 1) (Based on question 6.1 in Russel and Norvig) Consider a Tic-Tac-Toe game. Let X_n be the number of rows, columns or diagonals containing exactly n X 's and no O 's. Similarly, let O_n be the number of rows, columns or diagonals containing exactly n O 's and no X 's. We propose a utility function which assigns +1 to any position with $X_3 = 1$ (i.e. winning position) and assigns -1 to any position with $O_3 = 1$ (i.e. losing position). The linear evaluation function we suggest is $3X_2 + X_1 - (3O_2 + O_1)$:
- How many states (i.e. board positions) are there in a Tic-Tac-Toe game (including symmetric board positions)
 - What is the depth of the complete game tree? Does the complete game tree contain all the board positions you counted in (a)?
 - Show the game tree down to depth 2, namely starting from an empty board to all position in which there is one X and one O on the board.
 - Mark on your tree the evaluation of the positions at level 2 (i.e. the value of the utility function at each position). Thereafter, mark on your tree the min-max values.
 - Apply alpha-beta search on your tree, and mark the pruned subtrees when traversing from left to right, from right to left. What is the optimal order?
 - What property should the leaf values of a tree have so that pruning will be maximized (resp. minimized) when the tree below is traversed from left to right.
- 2) Consider the following game tree in which the static scores (in parentheses at the tip nodes) are all from the first player's point of view. Assume that the first player is the maximizing player.



- What move should the first player choose?
- What nodes would not need be examined using the alpha-beta algorithm – assuming that the nodes are examined left-to-right order?