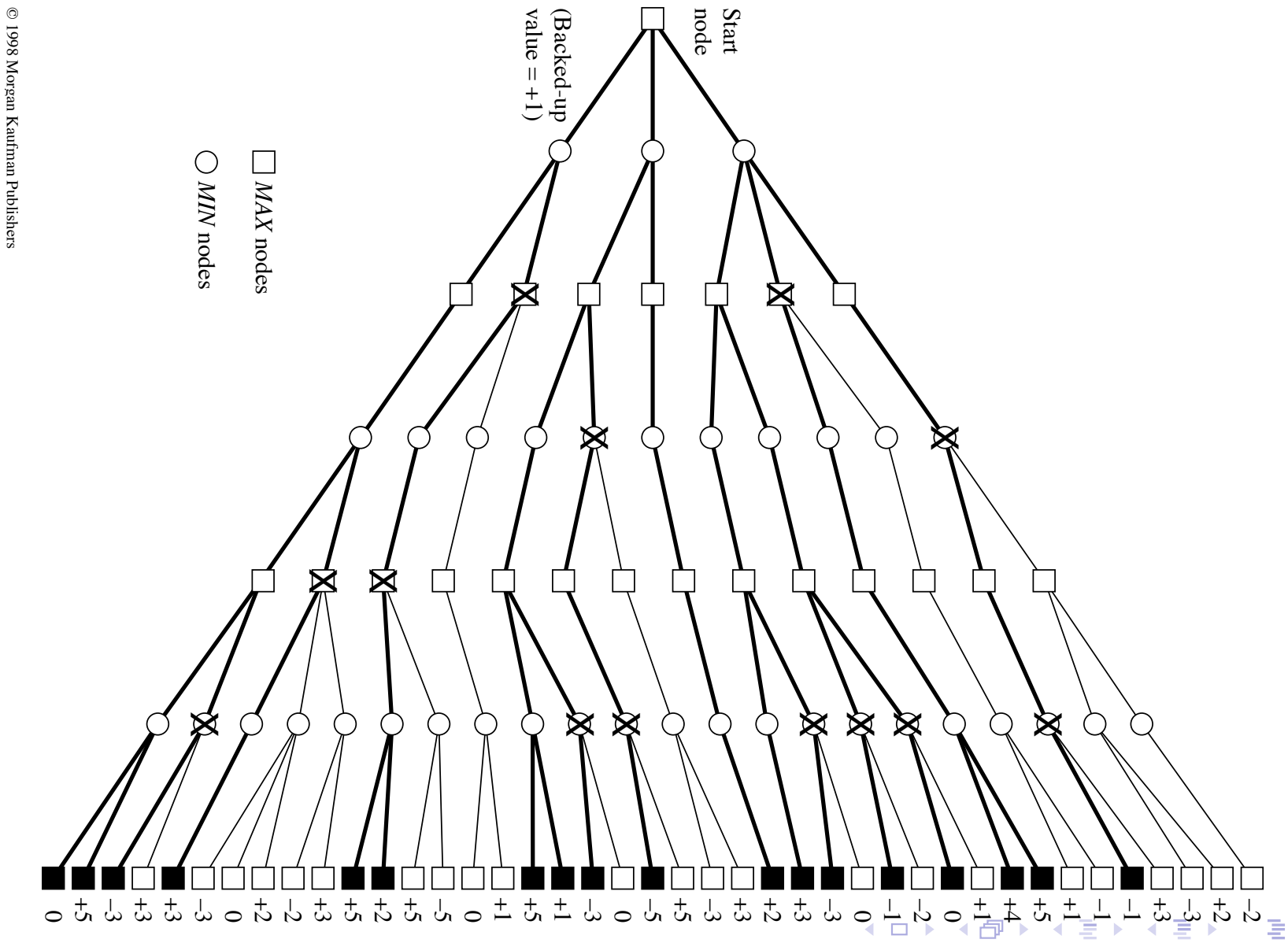
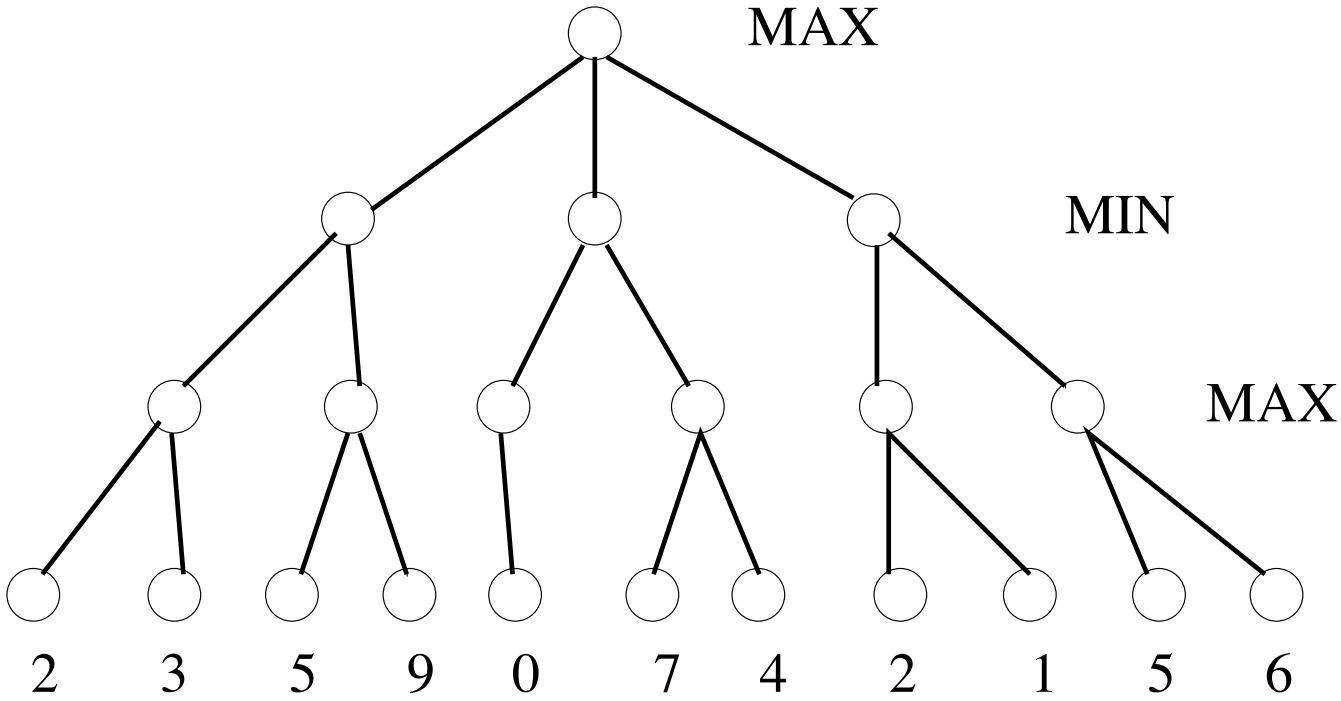
# Alpha-Beta Pruning Example



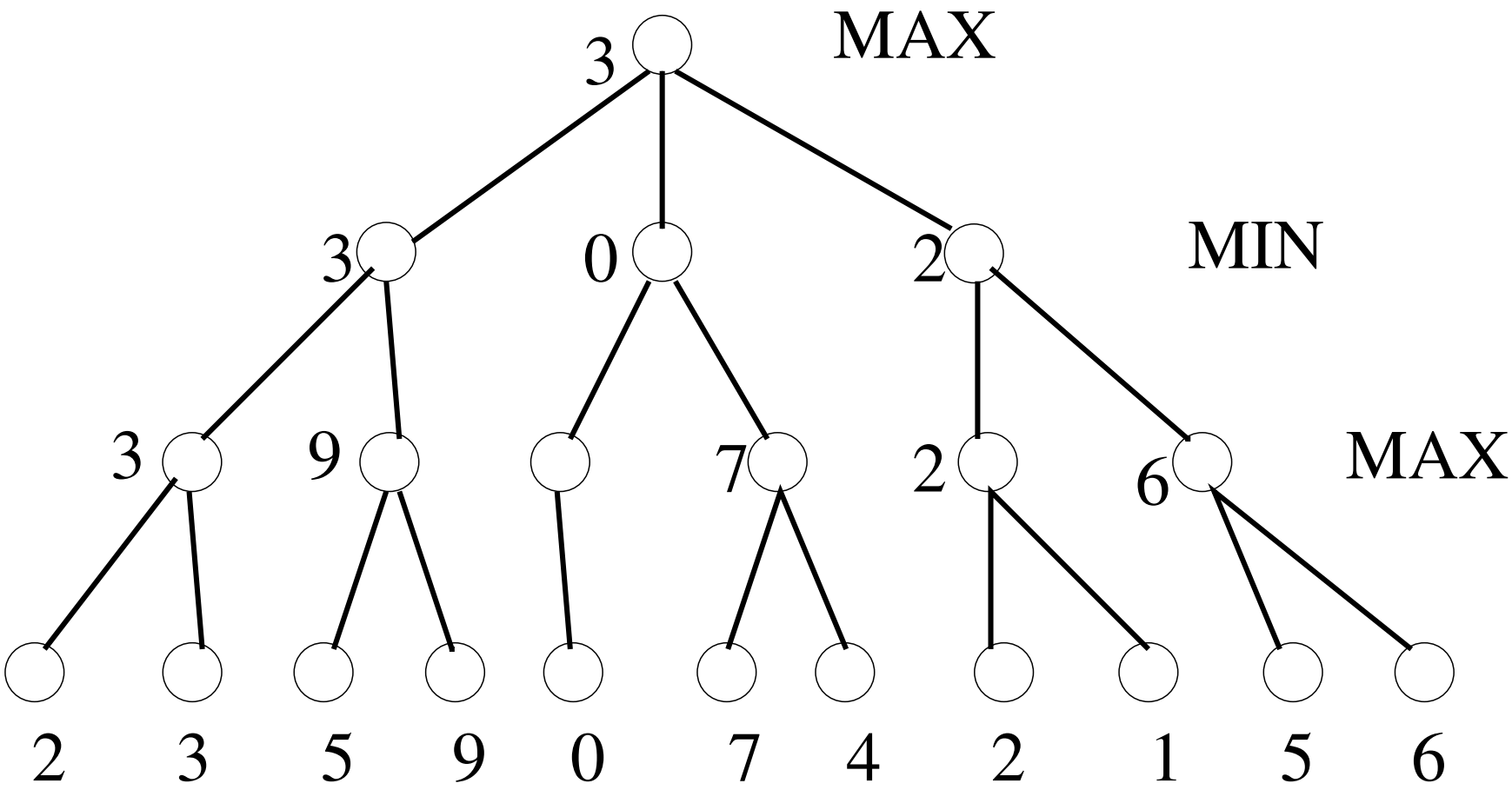
# Problem #7

A partial search tree for a two player game is given below.

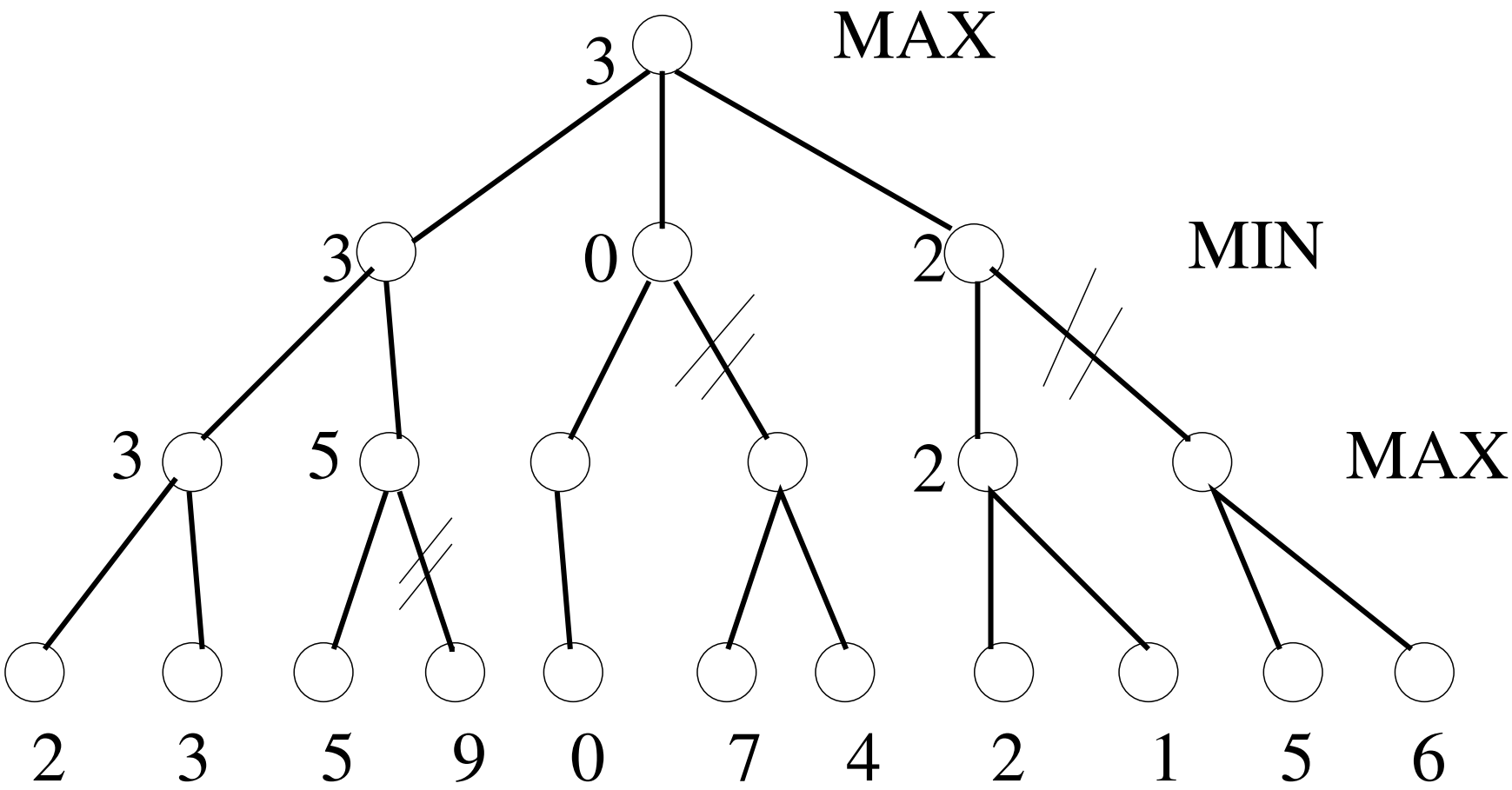
a) Find the best move for the MAX player using the minimax procedure.
b) Using alpha-beta pruning show which parts of the tree do not need to be searched. Indicate where the cutoffs occur.



# Problem #7: Minimax



# Problem #7: Alpha-beta Pruning



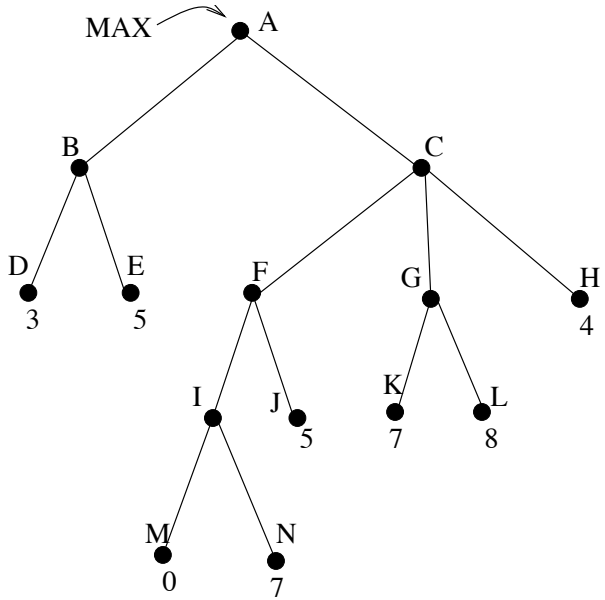
# Problem #22

Consider the following game tree.

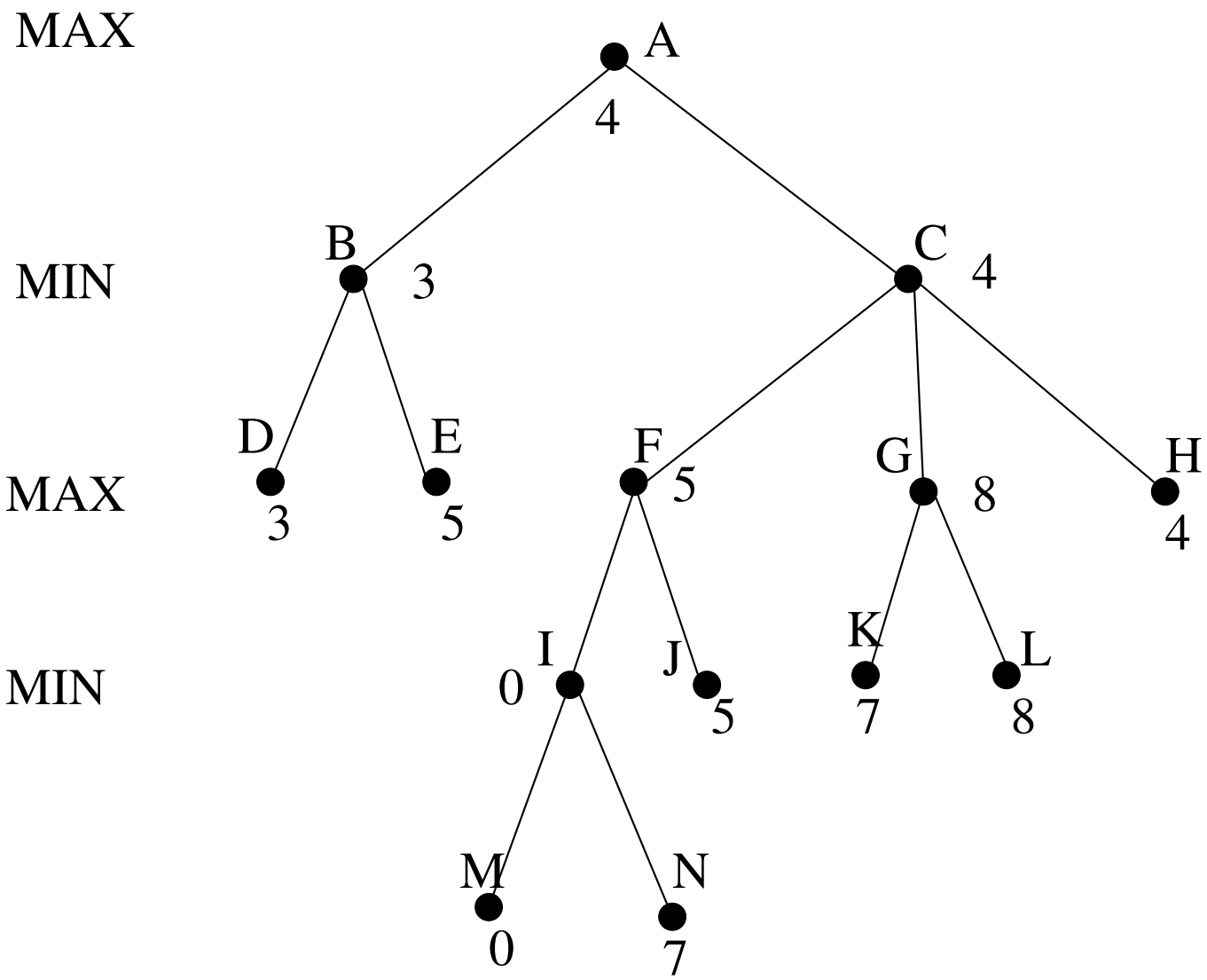
(a) Find the best move for the MAX player using the minimax procedure.

(b) Perform a left-to-right alpha-beta pruning on the tree. Indicate where the cutoffs occur.

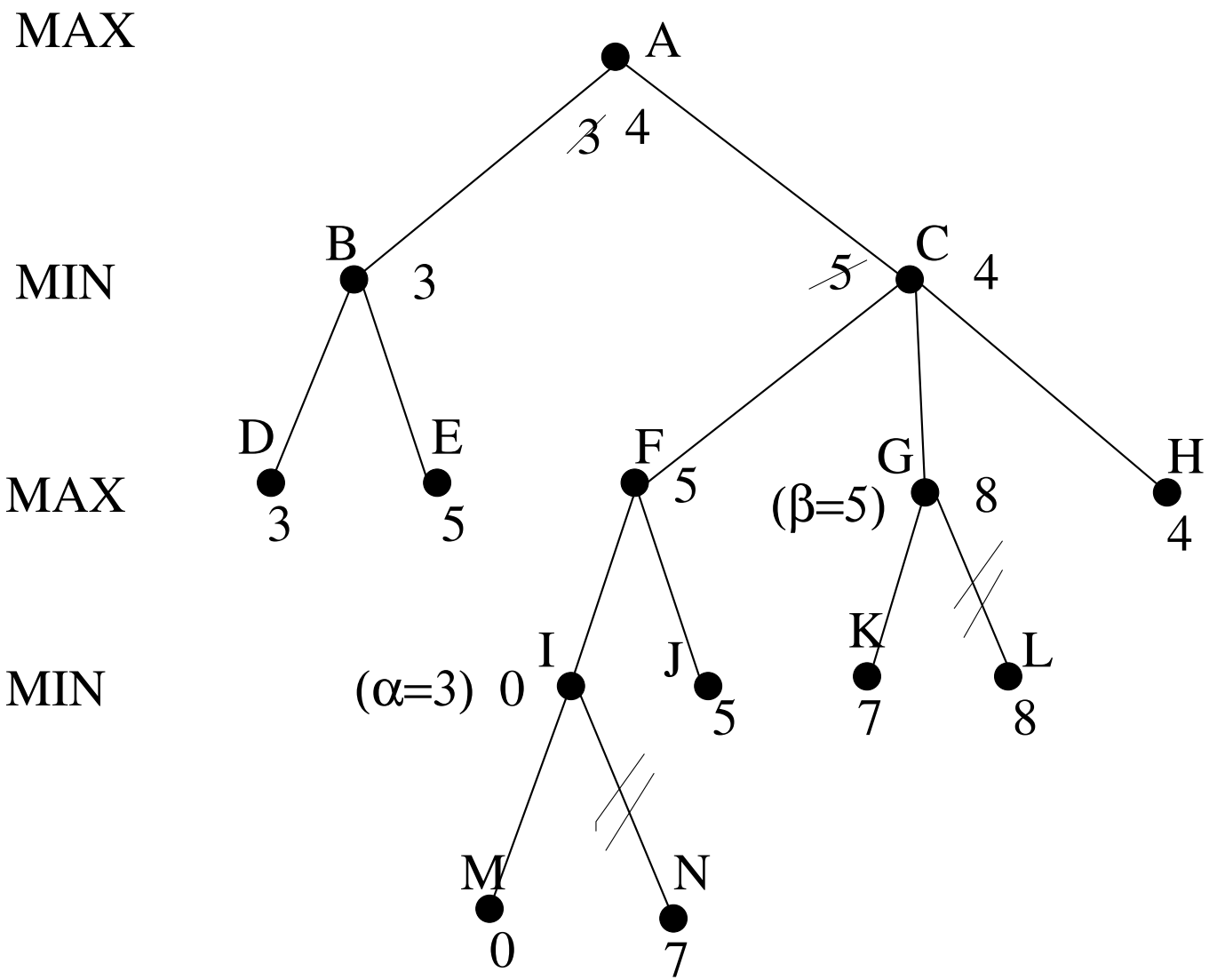
(c) Perform a right-to-left alpha-beta pruning on the tree. Discuss why different pruning occurs.



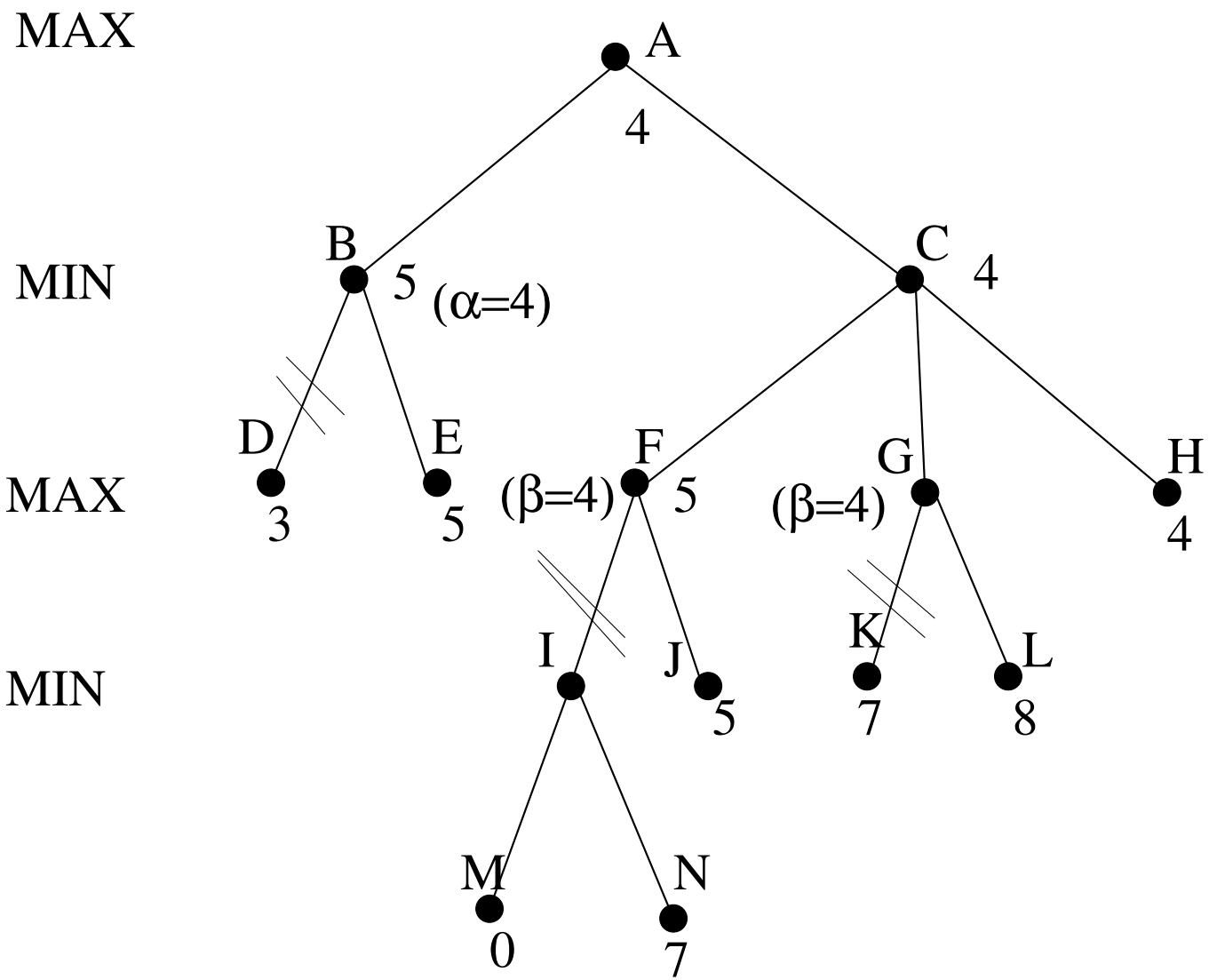
# Problem #22: Minimax



# Problem #22: Alpha-beta pruning left to right



# Problem #22: Alpha-beta pruning right to left



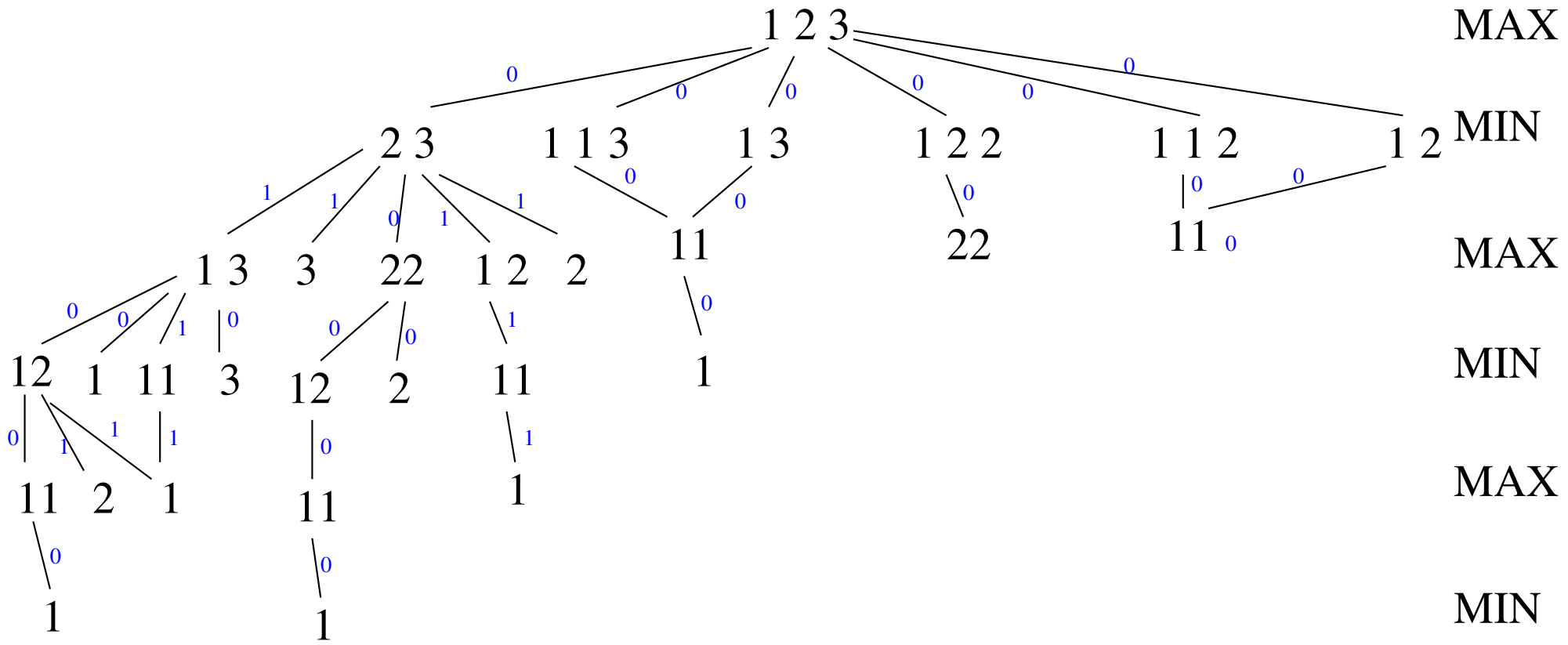
# Problem #30: A nim-game tree

A simple version of the *nim* game is played as follows: Two players alternate in removing stones from three piles initially containing two, two, and three stones, respectively. The player who picks up the last stone wins. At any given turn a player can pick one or more stones from a single pile; at least one stone has to be picked every time.

a) Show, by drawing a game tree, which player can always win.

b) Is it necessary to generate the whole tree to find a winning strategy? Explain why or why not.

# Problem #30: A nim-game tree solution




b) It is possible to use previous solutions and alpha-beta pruning (if appropriate).