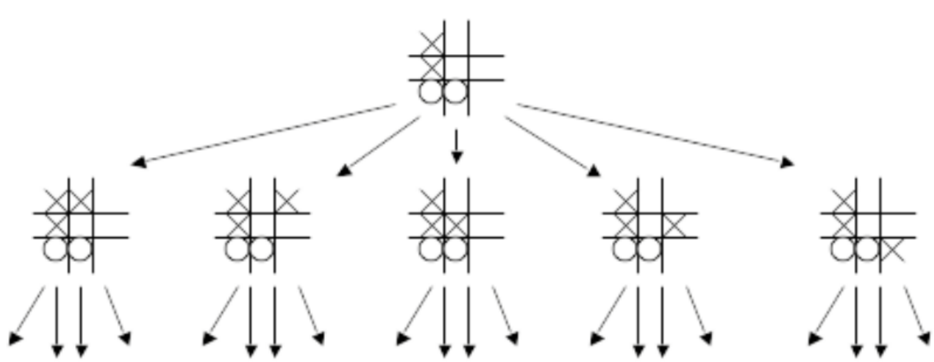
COMP 6721 - Artificial Intelligence
Minimax and Alpha-Beta Pruning

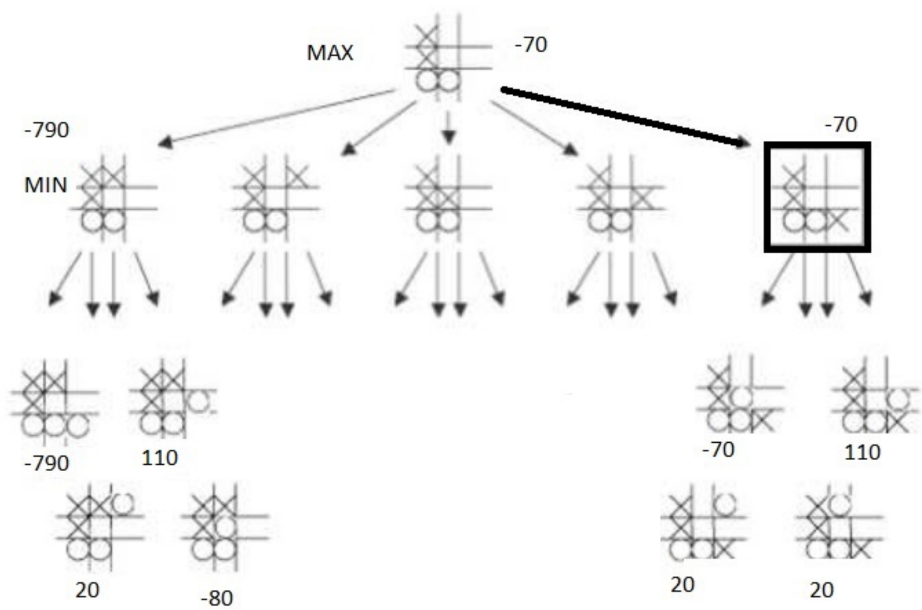
Solutions

Question 1 Consider state space search for the game of Tic-Tac-Toe. You are the X player, looking at the board shown below, with five possible moves. You want to look ahead to find your best move and decide to use the following evaluation function for rating board configurations:

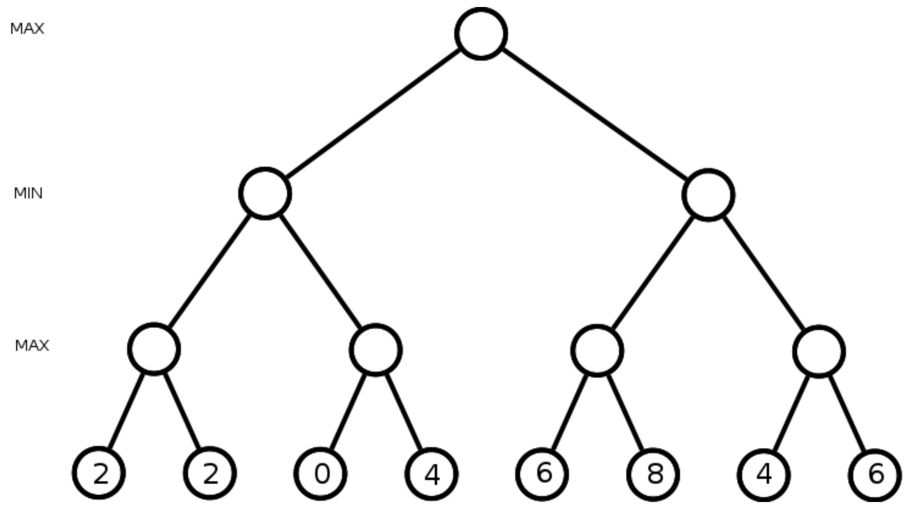
value $V = 0$
for all rows, columns, diagonals R **do**:
    if R contains three Xs **then**:
        $V = V + 1000$
    else if R contains three 0s **then**:
        $V = V - 1000$
    else if R contains two Xs **then**:
        $V = V + 100$
    else if R contains two Os **then**:
        $V = V - 100$
    else if R contains one X **then**:
        $V = V + 10$
    else if R contains one O **then**:
        $V = V - 10$
    end if
end for
return V

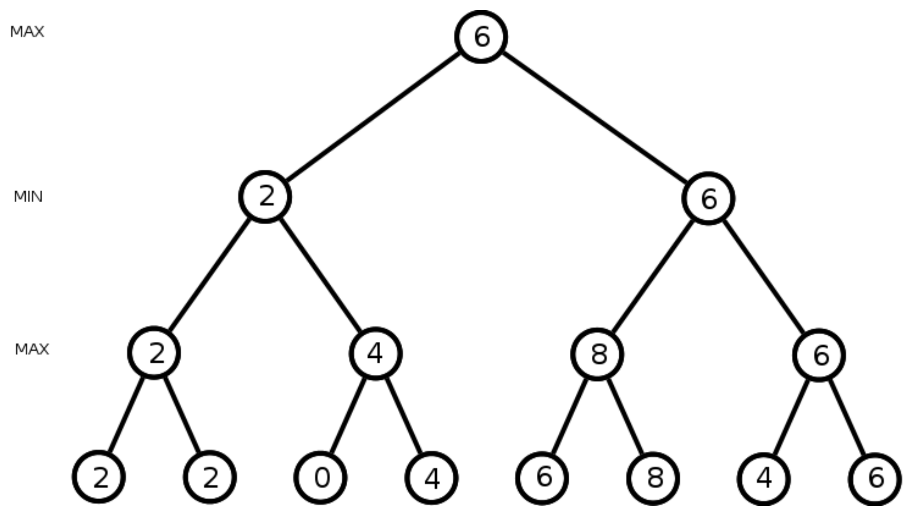
Draw the four possible configurations for the leftmost and the rightmost board configurations below. Use the evaluation function above to rate these 8 board configurations and choose X's best move.



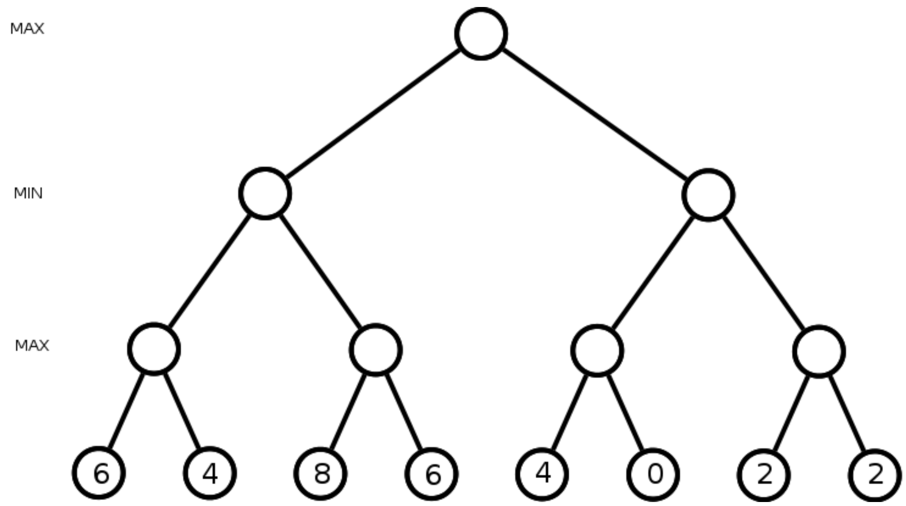

MAX
-70
-790
MIN
-70
-790
110
-70
110
20
-80
20
20

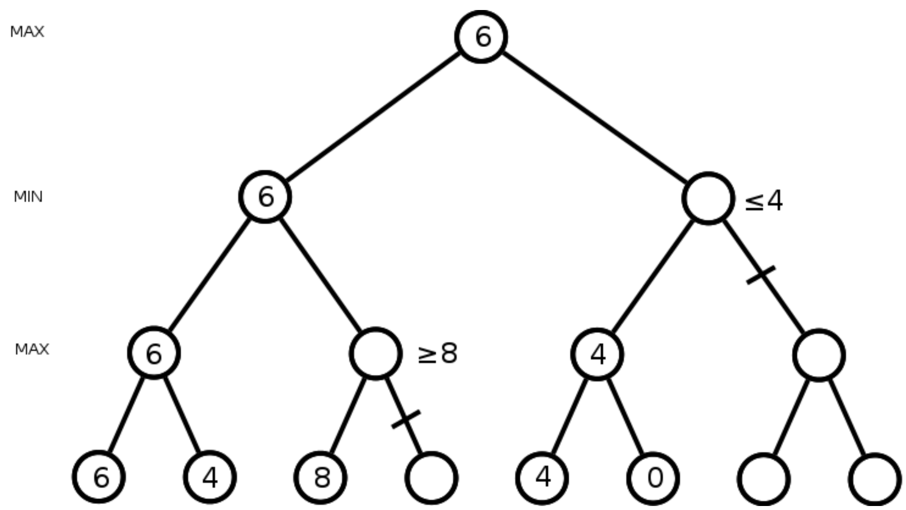
Question 2 (a) Consider the game tree shown below. Explore the tree using Alpha-Beta. Indicate all parts of the tree that are pruned, and indicate the winning path or paths.





(b) Now do the same for the tree below, which is a mirror image of the tree shown above.

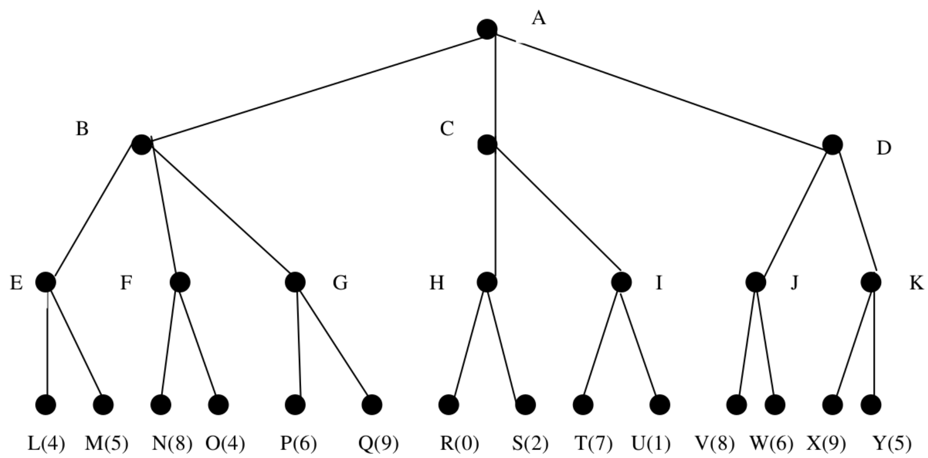




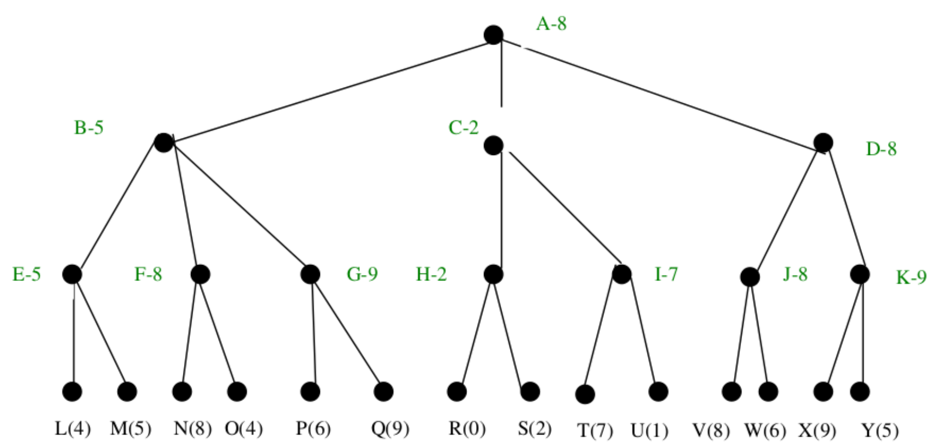
(c) Compare the amount of pruning in the above two trees. What do you notice about how the order of evaluation nodes affects the amount of Alpha-Beta pruning? *If the evaluated nodes are ordered in the manner described below, then alpha-beta gets maximal pruning.*

You get maximal cutoff if the left-most descendent of a MAX node has the largest $e(n)$ *value compared to its siblings. For a MIN level, you get maximal pruning if the left-most descendent has the lowest* $e(n)$ *value compared to its siblings.*

Question 3 Consider the game tree below. Each node is labelled with a letter, and the evaluation function for each leaf is indicated in parentheses. Assume that the MAX player goes first.

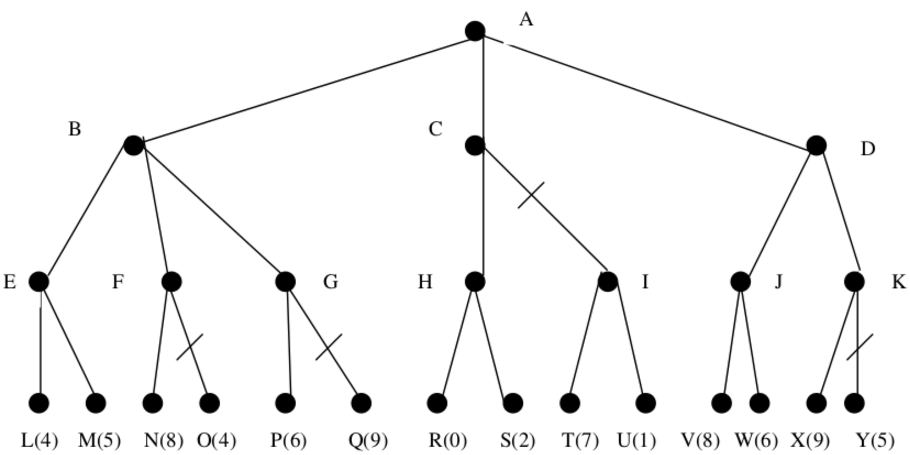


(a) Compute the minimax game value of nodes A, B, C, and D using the minimax algorithm. Show all values that are brought up to the internal nodes. What should MAX do?

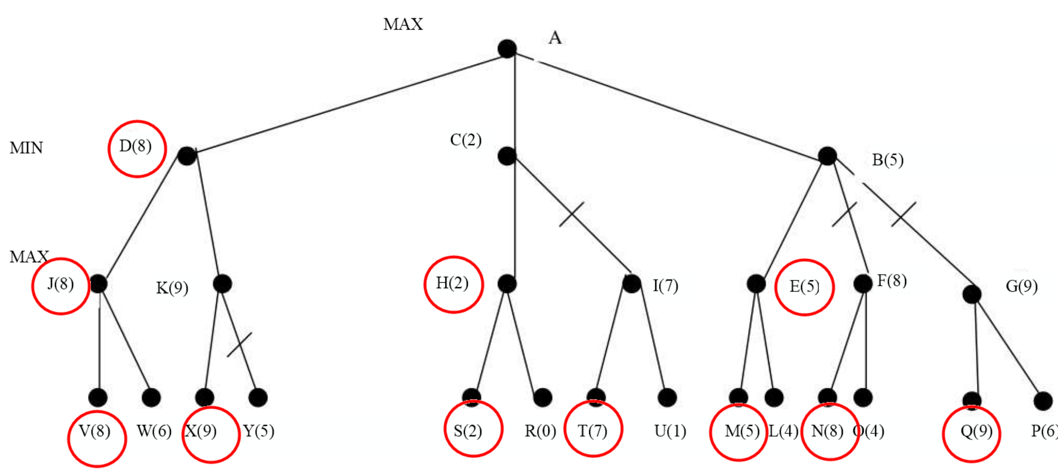


MAX should go right towards state D.

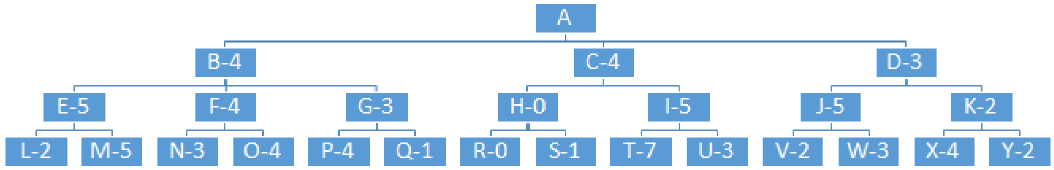
(b) Cross out the branches of all the nodes that are *not* visited by alpha-beta pruning. Show all your work.



(c) Draw a new game tree by re-ordering the children of each internal nodes (B to M), such that the new game tree is equivalent to the tree above, but alpha-beta pruning will prune as many nodes as possible. *This is an optimal tree, other trees could also be optimal for pruning.*



Question 4 Consider the following game tree.



The value of the evaluation function at each node is shown next to its name. For example, B-4 indicates that node B has an evaluation function of 4. All evaluations are from the point of view of the first player.

(a) Assume that the first player is the maximizing player MAX and she looks that all levels (ie, to the level labeled L, M, N, O, . . .). List in order the states that will **NOT** be examined when using alpha-beta pruning.

(b) What move should MAX choose?

(c) Suppose that instead of looking down all levels, MAX can only afford to look at level 2 (ie, the level with E, F, G, H, . . . instead of the level with L, M, N, O. . .). In theory, could that change MAX's move? Explain.

(a) *Nodes that will not be visited are: Q I T U K Y X*

(b) *MAX should choose to move left (node B)*

(c) *Yes. MAX's best move could change. The deeper a player can afford to look ahead, the more informed will be his decision.*

Question 5 Consider the classical game of Tetris. The objective of the game is to move and rotate each falling block so as to create an entire row of block pieces without any gap. If such a row is created, then that row disappears, and all rows on top of it fall down.

For example, in the figure above, a 1x4 block is falling from the top. The player can move this block left and right and rotate it by 90 degree to have it upright. As the figure shows, the player has at least 5 choices to position the block. Out of these 5 choices, position 5 is preferable, because then the 4 bottom rows would be complete. These rows would then disappear; all rows above would fall down, leaving more space on top to place the next random block to fall. The game ends when too many blocks are stacked up (no new complete rows can be created) and no more blocks can be placed on the board.

(a) Formulate a simple heuristic to determine how to place a falling (random) block on an existing board.

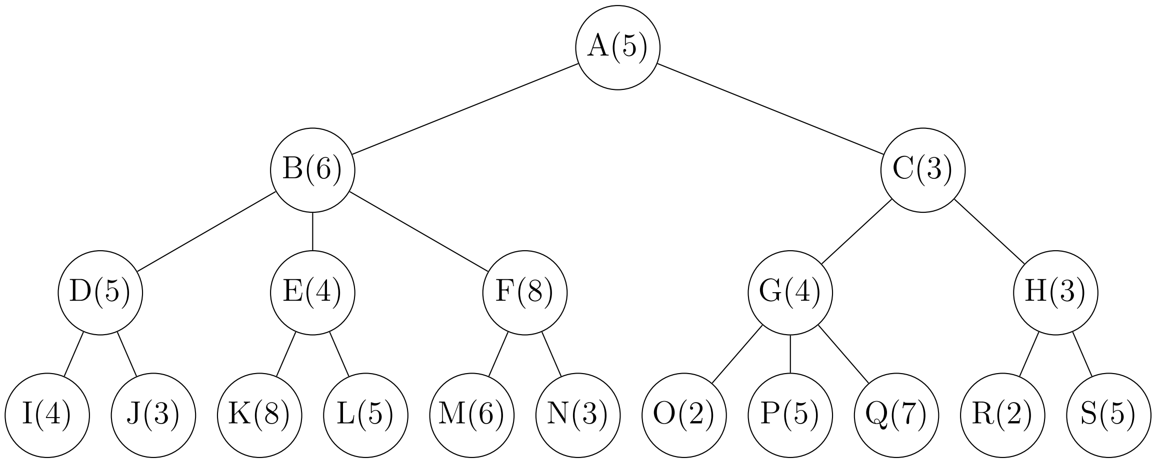
A possible heuristic is to count the number of complete rows created (that will be deleted). In that case, h(option2) = 0 and h(option4) = 0.

Another heuristic is to count the number of "touching sides" i.e. how many sides of the falling block will be in contact with the Tetris pile. In that case, h(option2) = 5 (2 edges on the left + the bottom + 2 edges on the right) h(option4) = 8 (4 edges on the left + the bottom + 3 edges on the right)

There are plenty of other possible heuristics.

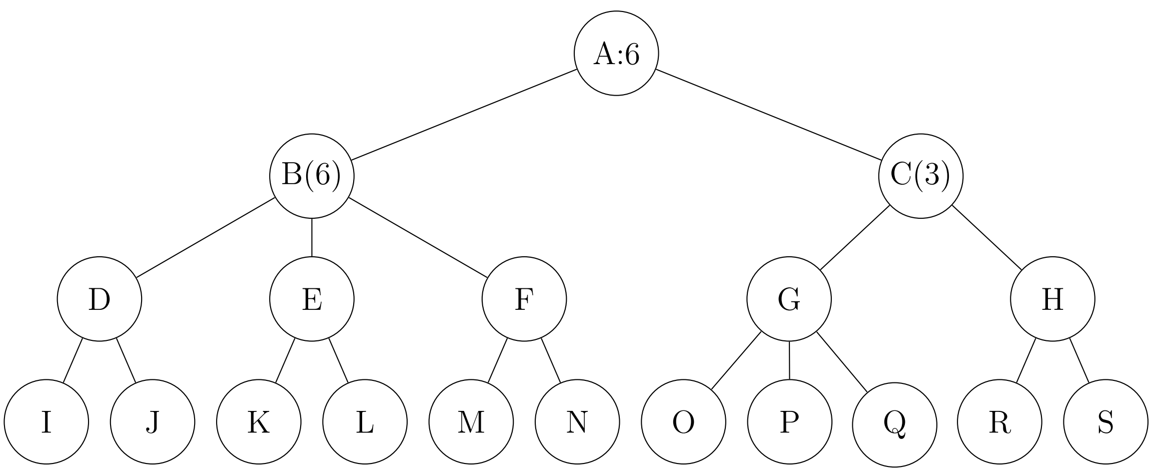
(b) Now apply your heuristic to evaluate option 2 and option 4 of the figure above.

Question 6 Consider tree below, in each node the value of the heuristic function is indicated inside parentheses. Assume Max plays first.



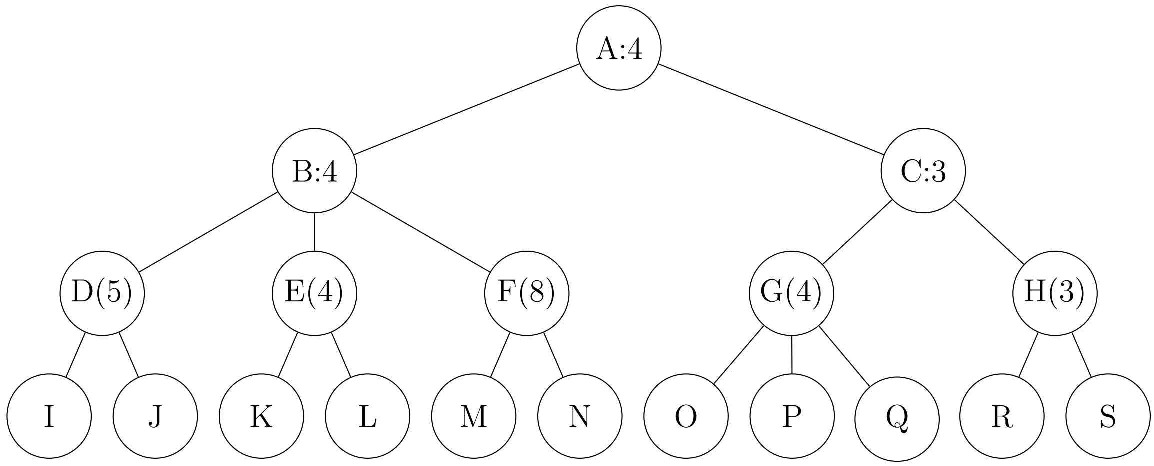
What should Max do first if:

(a) if Max can explore only one level.



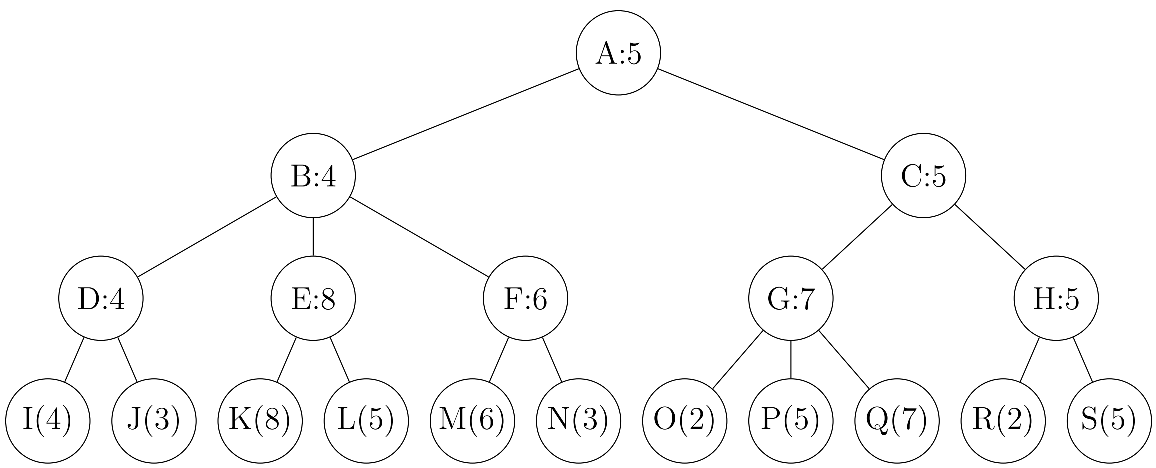
Max should go to B

(b) if Max can explore two levels.



Max should go to B

(c) if Max can explore three levels.



Max should go to C