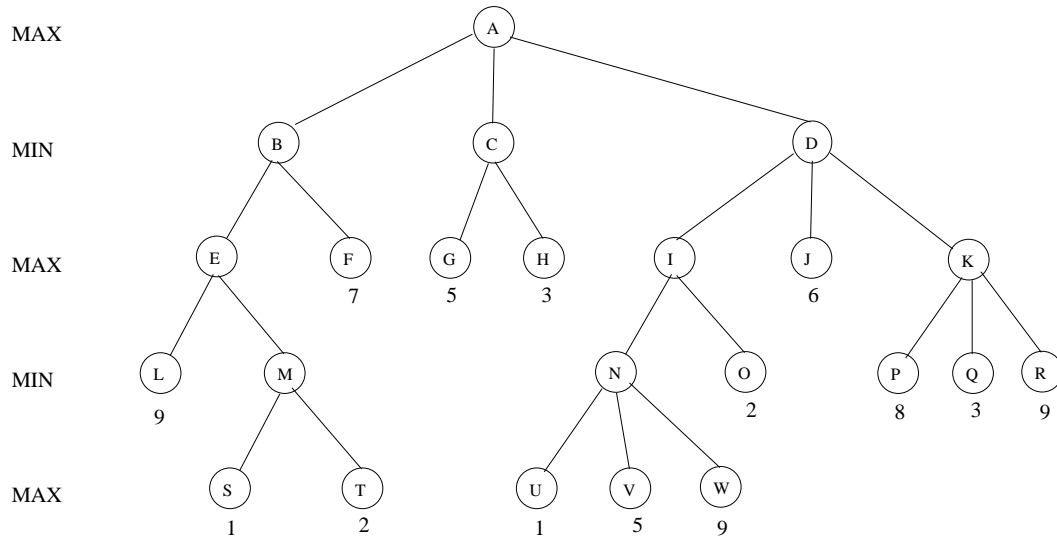


Practice Problems on Search in Games

Original date: October 8, 2005; Last revised: October 22, 2006

For all the problems in this document, MINIMAX refers to the minimax algorithm, while ALPHA-BETA refers to minimax with alpha-beta pruning.

1. Consider the following game tree.



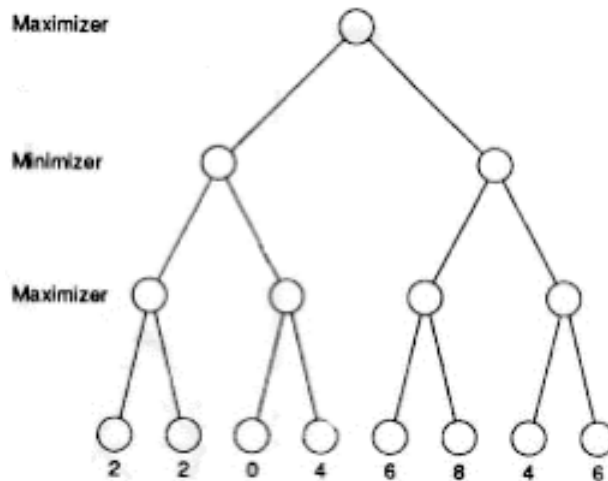
Assume that the children of each node are evaluated from left to right.

- (a) Use the MINIMAX to compute the value of each node in the game tree.
- (b) List the nodes that are evaluated by ALPHA-BETA.
- (c) Redraw the game tree such that ALPHA-BETA performs the *minimum* number of evaluations (i.e., the *best case* for ALPHA-BETA). How many nodes are evaluated by ALPHA-BETA in the resulting game tree?
- (d) Redraw the game tree such that ALPHA-BETA performs the *maximum* number of evaluations (i.e., the *worst case* for ALPHA-BETA). How many nodes are evaluated by ALPHA-BETA in the resulting game tree?

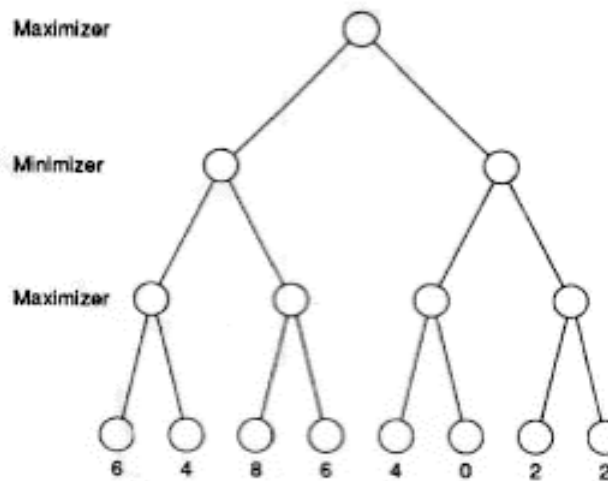
2. (Exercises 6.3 and 6.4 from the book) ¹

(a) Consider the game tree shown below.

Explore the tree using ALPHA-BETA. Indicate all parts of the tree that are cut off, and indicate the winning path or paths. Strike out all static evaluation values that do not need to be computed.



(b) Now consider the tree shown below, which is a mirror image of the tree shown above. Explore the tree using the ALPHA-BETA. Indicate all parts of the tree that are cut off. Indicate the winning path or paths. Strike out all static evaluation values that do not need to be computed.



(c) Compare the amount of cutoff in the above two trees. What do you notice about how the order of static evaluation nodes affects the amount of ALPHA-BETA cutoff?

¹The first two parts of this search in games problem are from the book *Artificial Intelligence* by Patrick Winston. The last part is due to Kimberle Koile.

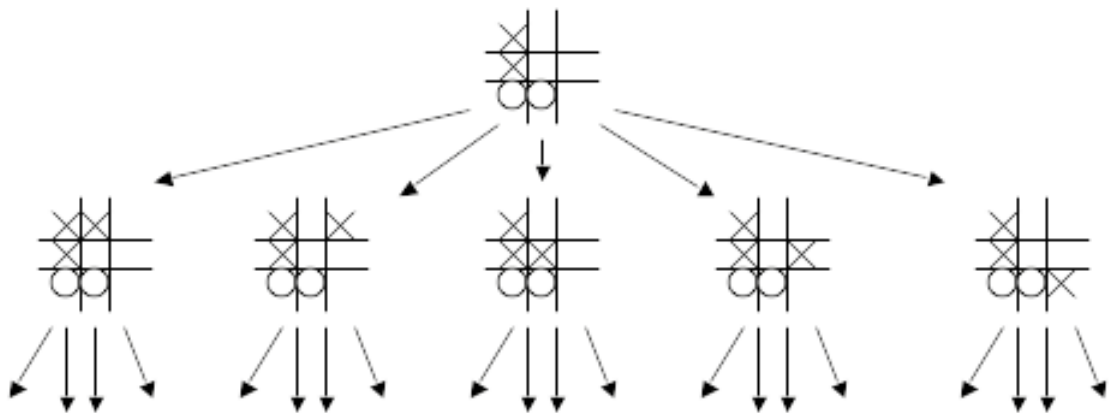
3. Tic-Tac-Toe ²

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:

```

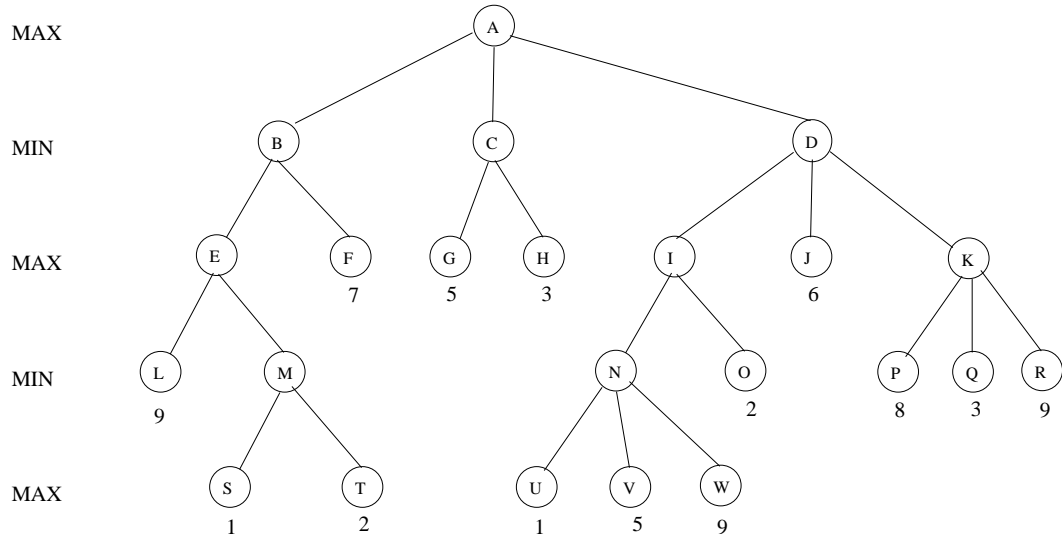
value  $V = 0$ 
for all rows, columns, diagonals  $R$  do
  if  $R$  contains three Xs then
     $V = 1000$ 
  else if  $R$  contains three Os then
     $V = -1000$ 
  else if  $R$  contains only two Xs then
     $V = V + 100$ 
  else if  $R$  contains only one X then
     $V = V + 10$ 
  else if  $R$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

Draw the four configurations possible from the leftmost and rightmost board configurations below. Use the above static evaluation function to rate the 8 board configurations and choose X's best move. (A reminder: The board configurations that you draw will show possibilities for O's next move.)



²This search in games problem is due to Kimberle Koile.

4. Consider the following game tree.



Use ALPHA-BETA with progressive deepening to compute the best move for the maximizer at node A (*i.e.*, move to B? C? Or D?). When performing ALPHA-BETA to depth d , use the evaluations obtained from performing ALPHA-BETA to depth $d - 1$ to order the way in which the children/successors of a node are evaluated (whenever possible). If an internal node of the game tree above needs to be statically evaluated when considering only up to a given depth, then their values are B 2, C 5, D 4, E 1, I 3, K 7, M 8, N 6. For every depth $d = 1, 2, 3, 4$,

- determine the best move for the maximizer at A;
- list the nodes in the order in which ALPHA-BETA evaluates them;
- calculate the number of static evaluations used as well as the total number of nodes evaluated.