



Mini-Max Algorithm in Artificial Intelligence



- ▶ Mini-max algorithm is a recursive or backtracking algorithm which is used in decision-making and game theory.
- ▶ Mini-Max algorithm uses recursion to search through the game- tree.
- ▶ In this algorithm two players play the game, one is called MAX and other is called MIN.



- ▶ Both the players fight it as the opponent player gets the minimum benefit while they get the maximum benefit.
- ▶ The minimax algorithm performs a depth-first search algorithm for the exploration of the complete game tree.
- ▶ The minimax algorithm proceeds all the way down to the terminal node of the tree, then backtrack the tree as the recursion.



Working

- ▶ An example of game-tree which is representing the two-player game.
- ▶ In this example, there are two players one is called Maximizer and other is called Minimizer.
- ▶ Maximizer will try to get the Maximum possible score, and Minimizer will try to get the minimum possible score.



- ▶ This algorithm applies DFS, so in this game-tree, we have to go all the way through the leaves to reach the terminal nodes.
- ▶ At the terminal node, the terminal values are given so we will compare those value and backtrack the tree until the initial state occurs.



Consider a game which has 4 final states and paths to reach final state are from root to 4 leaves of a perfect binary tree as shown below. Assume you are the maximizing player and you get the first chance to move, i.e., you are at the root and your opponent at next level.

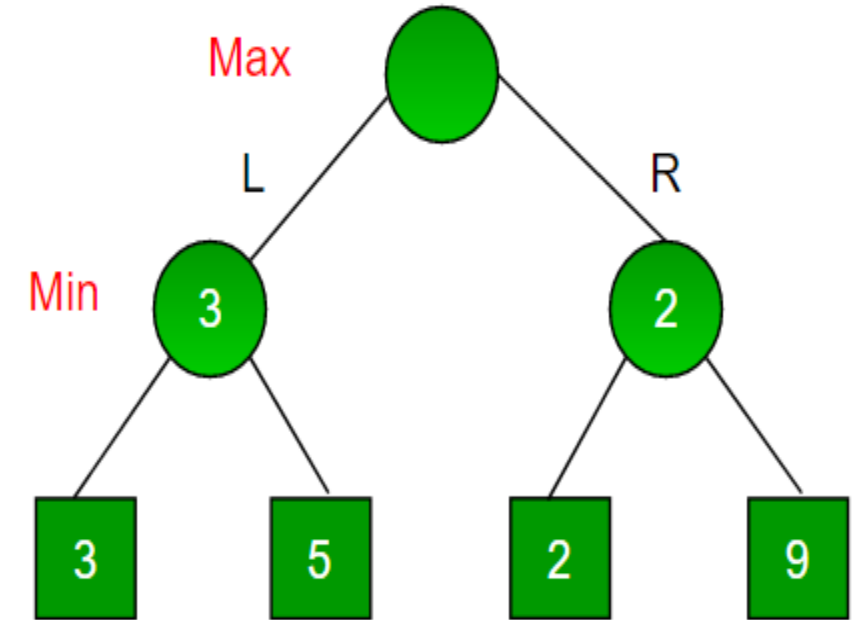
Which move you would make as a maximizing player considering that your opponent also plays optimally?

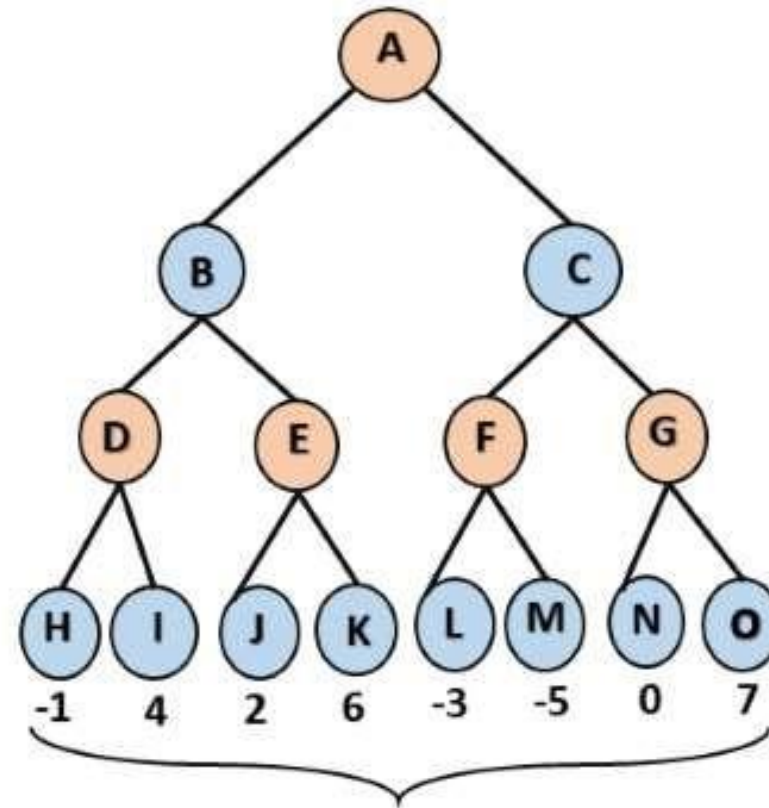
Since this is a backtracking based algorithm, it tries all possible moves, then backtracks and makes a decision.

Maximizer goes LEFT: It is now the minimizers turn. The minimizer now has a choice between 3 and 5. Being the minimizer it will definitely choose the least among both, that is 3

- Maximizer goes RIGHT: It is now the minimizers turn. The minimizer now has a choice between 2 and 9. He will choose 2 as it is the least among the two values.

Being the maximizer you would choose the larger value that is 3. Hence the optimal move for the maximizer is to go LEFT and the optimal value is 3.





Terminal values



Maximizer



Minimizer

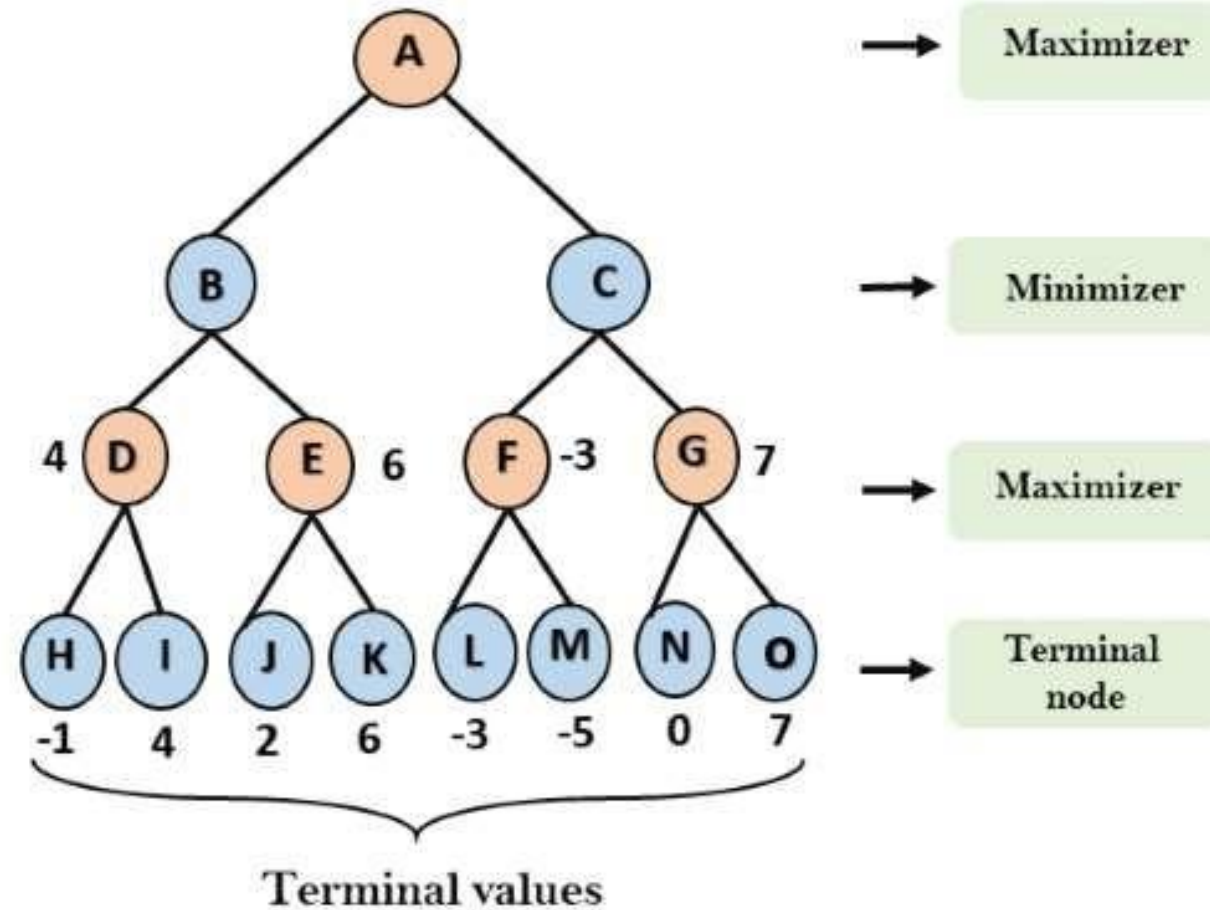


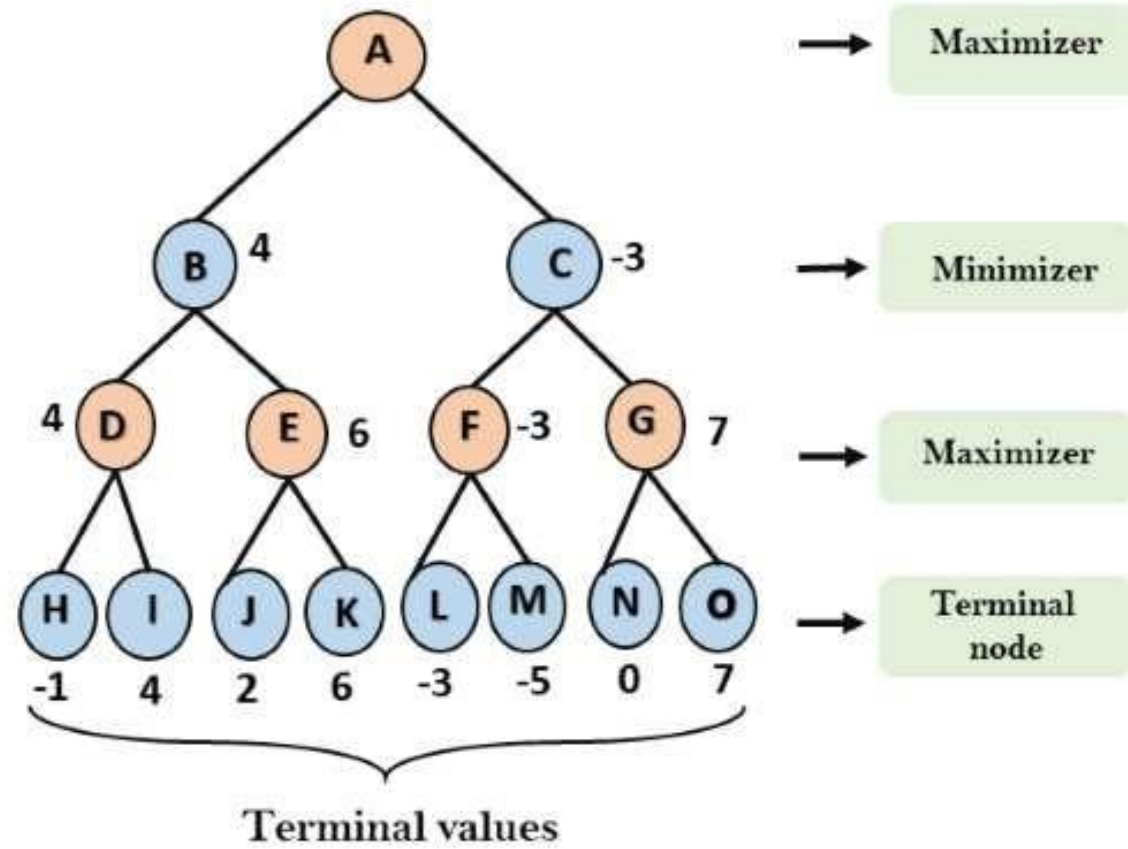
Maximizer

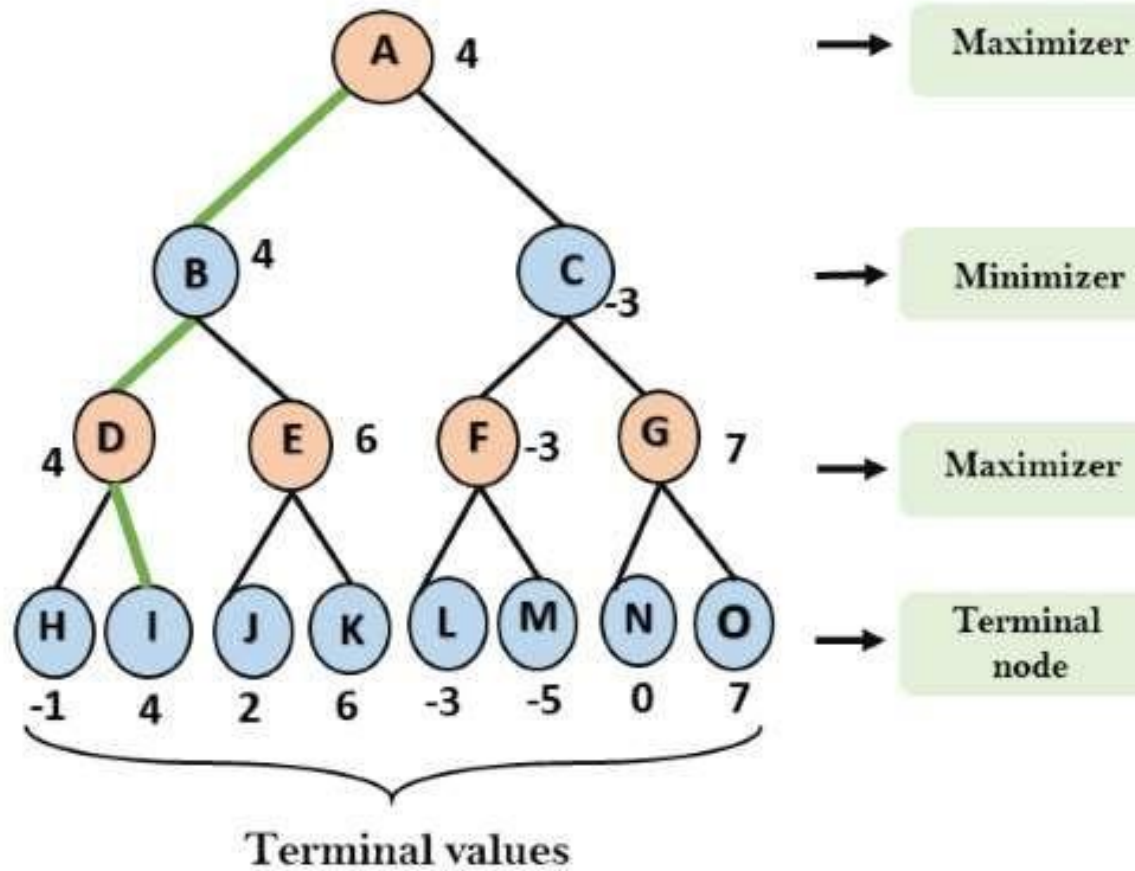


Terminal
node



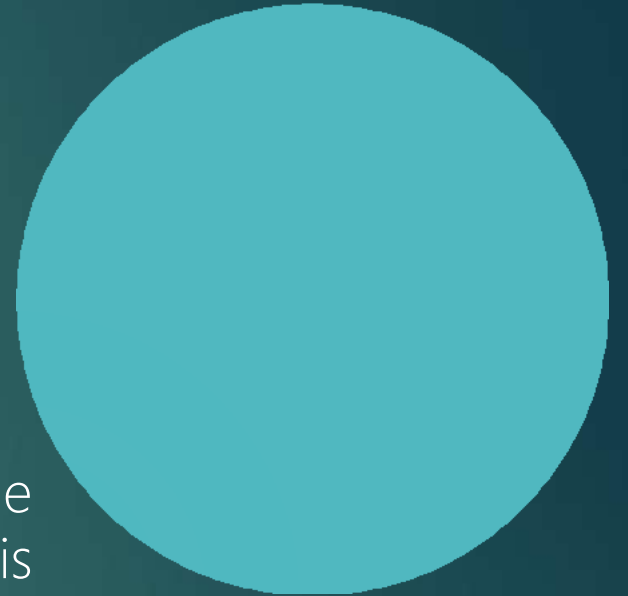






Properties

- ▶ Complete- Min-Max algorithm is Complete. It will definitely find a solution (if exist), in the finite search tree.
- ▶ Optimal- Min-Max algorithm is optimal if both opponents are playing optimally.
- ▶ Time complexity- As it performs DFS for the game-tree, so the time complexity of Min-Max algorithm is $O(b^m)$, where b is branching factor of the game-tree, and m is the maximum depth of the tree.
- ▶ Space Complexity- Space complexity of Mini-max algorithm is also similar to DFS which is $O(bm)$.



Limitation

- ▶ The main drawback of the minimax algorithm is that it gets really slow for complex games such as Chess, go, etc. This type of games has a huge branching factor, and the player has lots of choices to decide. This limitation of the minimax algorithm can be improved from alpha-beta pruning.

