|  |  |
| --- | --- |
| **EX.N0 : 8** | **MIN MAX ALGORITHM** |
| **DATE:17.04.2024** |

**AIM:**

To implement the MIN MAX Algorithm using python Program.

**ALGORITHM:**

Step 1: Start with the current game state, player, and depth.

Step 2:  If the game is over, return a utility value (positive for a win, negative for a loss, 0 for a draw).

Step 3: For the maximizing player, choose the move that maximizes the utility value by recursively exploring possible moves.

Step 4: For the minimizing player, choose the move that minimizes the utility value by doing the same.

Step 5: Make an initial call with the current player and depth 0 to find the best move.

Step 6: Recursively explore all possible moves and counter-moves, considering rational opponents, to determine the best move for the current player.

**PROGRAM:**

import math

def minimax(curDepth, nodeIndex, maxTurn, scores, targetDepth):

    if curDepth == targetDepth:

        return scores[nodeIndex]

    if maxTurn:

        return max(minimax(curDepth + 1, nodeIndex \* 2, False, scores, targetDepth),

                   minimax(curDepth + 1, nodeIndex \* 2 + 1, False, scores, targetDepth))

    else:

        return min(minimax(curDepth + 1, nodeIndex \* 2, True, scores, targetDepth),

                   minimax(curDepth + 1, nodeIndex \* 2 + 1, True, scores, targetDepth))

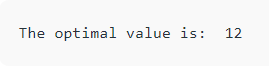
scores = [3, 5, 2, 9, 12, 5, 23, 23]

treeDepth = math.log(len(scores), 2)

print("The optimal value is:", end=" ")

print(minimax(0, 0, True, scores, treeDepth))

**OUTPUT:**



**RESULT:**

Thus the program to implement the MIN MAX Algorithm using python has been done and executed successfully.