# Explanation of Minimax Algorithm Code

The given Python code implements the Minimax algorithm, a fundamental concept in Artificial Intelligence used for decision-making in two-player games (like Chess, Tic Tac Toe, etc.). The algorithm assumes that both players play optimally — one tries to maximize the score, and the other tries to minimize it.

## Code Explanation

1. \*\*Importing math module\*\*   
 The math module is imported to use the log function to calculate the depth of the game tree based on the number of leaf nodes.

2. \*\*Function Definition – minimax()\*\*   
 The function `minimax(curDepth, nodeIndex, maxTurn, scores, targetDepth)` is defined with the following parameters:  
 - `curDepth`: Current depth of the node in the tree.  
 - `nodeIndex`: Index of the current node.  
 - `maxTurn`: Boolean value that determines if it’s the maximizing player’s turn (`True`) or the minimizing player’s turn (`False`).  
 - `scores`: List of all possible scores at the leaf nodes.  
 - `targetDepth`: The maximum depth of the game tree.

3. \*\*Base Case\*\*   
 If the current depth equals the target depth, the function returns the corresponding score from the scores list. This represents reaching a leaf node in the game tree.

4. \*\*Recursive Case – Maximizer’s Turn\*\*   
 If `maxTurn` is True, the algorithm calculates both possible outcomes (left and right child) and returns the \*\*maximum\*\* value. This simulates the player trying to get the highest possible score.

5. \*\*Recursive Case – Minimizer’s Turn\*\*   
 If `maxTurn` is False, the algorithm calculates both possible outcomes and returns the \*\*minimum\*\* value. This simulates the opponent trying to minimize the player’s score.

6. \*\*Driver Code\*\*   
 - The `scores` list contains the leaf node values: `[3, 5, 2, 9, 3, 5, 2, 9]`.  
 - The tree depth is calculated using `math.log(len(scores), 2)`.  
 - The `minimax()` function is called starting from depth 0 and node index 0, assuming the first move is the maximizer’s turn.  
 - Finally, the optimal value is printed, which represents the best guaranteed outcome if both players play optimally.

## Output

The optimal value is: 9