**Minimax Algorithm**

The Minimax Implementation is actually used for decision-making in games which  
uses log-based depth calculation for binary trees and finds the optimal move by alternating Min & Max choices.

In this code it is implemented as below:

* If is-Max (Maximizing Player): Takes the maximum of the two child nodes.
* If not is-Max (Minimizing Player): Takes the minimum of the two child nodes.
* Represents the leaf nodes of the decision tree (terminal values).
* Uses log2(len(score)) to determine the depth of a **perfect binary tree**.
* Starts the minimax process from the **root** (depth 0, index 0) assuming the Minimizing Player goes first.
* Outputs the optimal minimax value for the given tree.