

Ishmal Nadeem

Roll No:101

AI LAB TASK

Documentation for min_max.ipynb

This notebook contains the implementation of the **Minimax Algorithm**, which was part of **Lab Task 8**.

Purpose and Algorithm Logic

The Minimax algorithm is a decision-making rule used in game theory and artificial intelligence to minimize the potential loss in a worst-case scenario. It is a recursive function designed to find the optimal move for a player, assuming the opponent is also playing optimally.

The core function, `minimax(node, isMinTurn)`, operates as follows:

- **Base Case:** If the node is a terminal state (an integer value), the function returns that value.
- **Maximizing Player's Turn (`isMinTurn = False`):** The algorithm compares the values of the child nodes and returns the **larger** value.
- **Minimizing Player's Turn (`isMinTurn = True`):** The algorithm compares the values of the child nodes and returns the **smaller** value.

Implementation Details

- A sample game **tree** is defined as a nested list structure: `[[[3, 2], [5, 4]], [[7, 2], [7, 7]]]`.
- The code demonstrates the results for both potential starting turns:
 - **Minimizer's Turn (Root):** `minimax(tree, True)` returns **4**.
 - **Maximizer's Turn (Root):** `minimax(tree, False)` returns **7**.