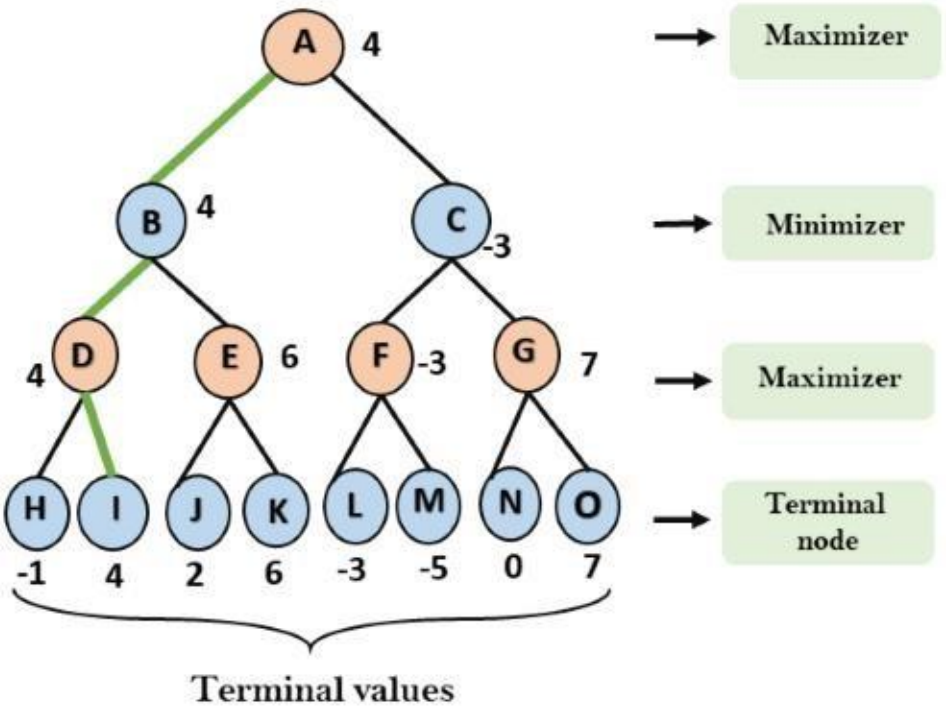
EX.NO: 04 **DATE:**

MINIMAX ALGORITHM

- A simple example can be used to explain how the minimax algorithm works. We've included an example of a game-tree below, which represents a two-player game.
- There are two players in this scenario, one named Maximizer and the other named Minimizer.
- Maximizer will strive for the highest possible score, while Minimizer will strive for the lowest possible score.
- Because this algorithm uses DFS, we must go all the way through the leaves to reach the terminal nodes in this game-tree.
- The terminal values are given at the terminal node, so we'll compare them and retrace the tree till we reach the original state.



CODE:

```
# Function to perform Minimax Algorithm
def minimax(depth, node_index, is_max, scores, height):
    if depth == height:
        return scores[node_index]

    if is_max:
        return max(
            minimax(depth + 1, node_index * 2, False, scores, height),
            minimax(depth + 1, node_index * 2 + 1, False, scores, height),
        )
    else:
        return min(
            minimax(depth + 1, node_index * 2, True, scores, height),
            minimax(depth + 1, node_index * 2 + 1, True, scores, height),
        )

def calculate_tree_height(scores):
    import math
    return math.ceil(math.log2(len(scores)))

scores = [3, 5, 6, 9, 1, 2, 0, -1]
height = calculate_tree_height(scores)

optimal_value = minimax(0, 0, True, scores, height)
print(f"The optimal value for the Maximizer is: {optimal_value}")
```

OUTPUT:

```
The optimal value for the Maximizer is: 5
```

RESULT:

Thus, the minmax algorithm program has been implemented successfully.