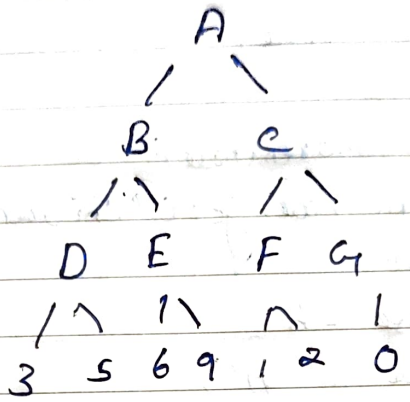
Exp No: 5
Date:

MINMAX Algorithm

Aim:

Implement the Minimax algorithm in python

Algorithm:



1. The function recursively evaluates a tree
2. It takes more depth, depth of tree and a boolean if player is Maximising
3. If it is a terminal node return node value.
4. The function gets childs nodes asking the get child nodes function.
5. computes best score for Maximising A.

Program:

```
import Math
def Minimax (node, depth, is_Maximizing)
    if depth == 0;
        return node

    if is_Maximizing:
    best_value = - Math.inf
     for child in get_children (node)
         value = Minimax (child, depth - 1, False)
```

```
        best_value = max(best_value, value)
        return best_value
    else:
        best_value = math.inf
        for child in get_children(node):
            value = minimax(child, depth-1, True)
            best_value = min(best_value, value)
        return best_value

def get_children(node):
    return node.get("children", [])

game_tree = {
    "value": "A",
    "children": [{
        "value": "B",
        "children": [{"value": "D", "children": [],
                      "terminal_value": 3},
                     {"value": "E",
                      "children": [], "terminal_value": 5}
                     ]},
    {
        "value": "C", "children": [
        {"value": "F", "children": [], "terminal_value": 8},
        {"value": "G", "children": [], "terminal_value": 9}
        ]}]
}

if __name__ == "__main__":
    best_score = minimax(game_tree, 2, True)
    print("Best score for Maximizer(A):",
          best_score)
```