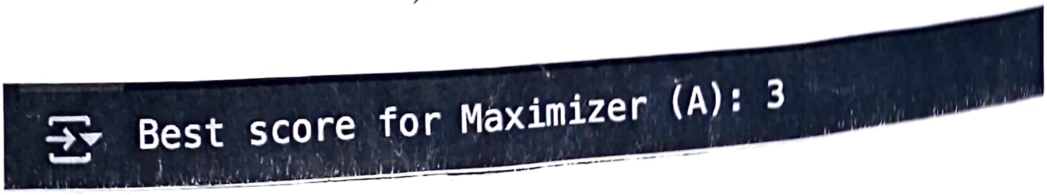
PROGRAM:

```
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 getchildren(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':6}
          ]},
          {value: 'C', 'children':[
                {value: 'F', children [] , 'terminal value': 1}
                {value: 'G', children[] , terminal_value: 8}
    ]}]}
 }
if_name_ == _'main_':
    best_score= minimax(gametree, 2, True)
    print(f"Best score for Maximizer (A): {best_score}")
```



```
Best score for Maximizer (A): 3
```

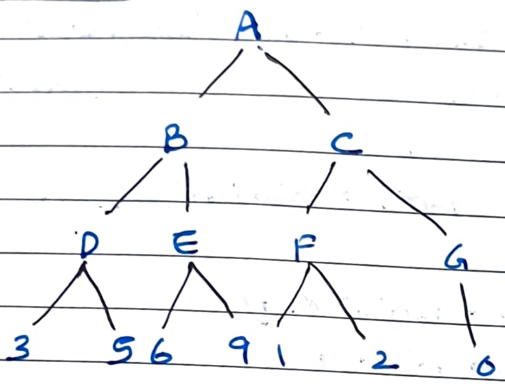
classmate
Date
Page

28/10/24

EXP NO: 7

AIM: Implement MINIMAX algorithm in python.

ALGORITHM:



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

RESULT:

Thus the minimax algorithm has been implemented in python.