ARTIFICIAL INTELLIGENCE LAB

B A Saran
22011102012
B.Tech CSE(IoT)-A

MIN-MAX ALGORITHM

AIM: To implement the Min Max Algorithm.

ALGORITHM:

1. **Define Function**: Create a function `minimax(curDepth, nodeIndex, maxTurn, scores, targetDepth)` to evaluate the game tree recursively.

2. **Base Case**: If `curDepth` equals `targetDepth`, return the score at `scores[nodeIndex]` (leaf node value).

3. **Maximizing Player**: If `maxTurn` is `True`, recursively calculate the maximum value between the left (`nodeIndex * 2`) and right (`nodeIndex * 2 + 1`) child nodes.

4. **Minimizing Player**: If `maxTurn` is `False`, recursively calculate the minimum value between the left and right child nodes.

5. **Execute**: Call `minimax(0, 0, True, scores, targetDepth)` to start from the root and print the optimal value.

CODE:

```
import math

def minimax (curDepth, nodeIndex,maxTurn, scores, targetDepth):

if (curDepth == targetDepth):

return scores[nodeIndex]

if (maxTurn):

return max(minimax(curDepth + 1, nodeIndex * 2, False, scores, targetDepth), minimax(curDepth + 1, nodeIndex * 2 + 1, False, scores, targetDepth))

else:

return min(minimax(curDepth + 1, nodeIndex * 2, True, scores, targetDepth), minimax(curDepth + 1, nodeIndex * 2 + 1, True, scores, targetDepth))

scores = [8, 12, 6, 14, 20, 18, 16, 22, 5, 10, 3, 9, 11, 15, 19, 25]

treeDepth = math.log(len(scores), 2)

print("The optimal value is : ", end = "")

print(minimax(0, 0, True, scores, treeDepth))
```

OUTPUT:

```
PS C:\Users\saran\Downloa
n/AI-main/CIA1/minmax.py
The optimal value is : 8
```

RESULT: The Min Max algorithm is implemented.