Exno: Minmax algorithm

Date:

Aim: To implement minmax algorithm using Python language.

Algorithm:

1. Define the Base case:
   If the current depth equals the target depth, return the score at the current node.
2. Recursive case:
   * If it's the maximizing players turn
     -> Recursively call the minmax function for the two child nodes of the current node.
     -> Take the maximum of the values returned by these calls.
   * If its the minimizing player's turn
     -> Recursively call the minmax function for the two child nodes of the current node
     -> Take the minimum of the values returned by these calls.
   * Repeat until Base case is met
     -> The function will continue to call itself for each level until it reaches the target dept
     -> At Each level it ~~reaches the target depth~~ alternates between maximizing and minimizing until it returns the optimal score at the root node.

Code:

```
import math
def minimax (curdepth, nodeIndex,
        maxturn, scores, target depth):
#base case: target Depth reached
    if (curdepth == target Depth):
    return scores [node Index]
    if (max turn):
        return max(minimax (curdepth +1, node Index * 2,
        False, scores, target Depth)
        minimax (curdepth + 1, node Index * 2 + 1,
        False, scores, target Depth))
#Driver code
    Scores = [3,5,2,9,12,5,23,23]
    tree depth = math.log (len (scores), 2)
    print ("The optimal value is :", end = " ")
    print (minimax (0,0, True, scores, tree Depth)).
```

output:

The optimal value is : 12

Result:

The program is successfully executed and output is verified.