**3B (MIN MAX) BACK TRACKING**

Consider a game which has 4 final states and paths to reach final state are from root to 4 leaves of a perfect binary tree as shown below. Assume you are the maximizing player and you get the first chance to move, i.e., you are at the root and your opponent at next level. **Which move you would make as a maximizing player considering that your opponent also plays optimally?**

**IMPLEMENTATION IN PYTHON  
  
PROGRAM:**  
import math

def minimax (curDepth, nodeIndex,

             maxTurn, scores,

             targetDepth):

    # base case : targetDepth reached

    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))

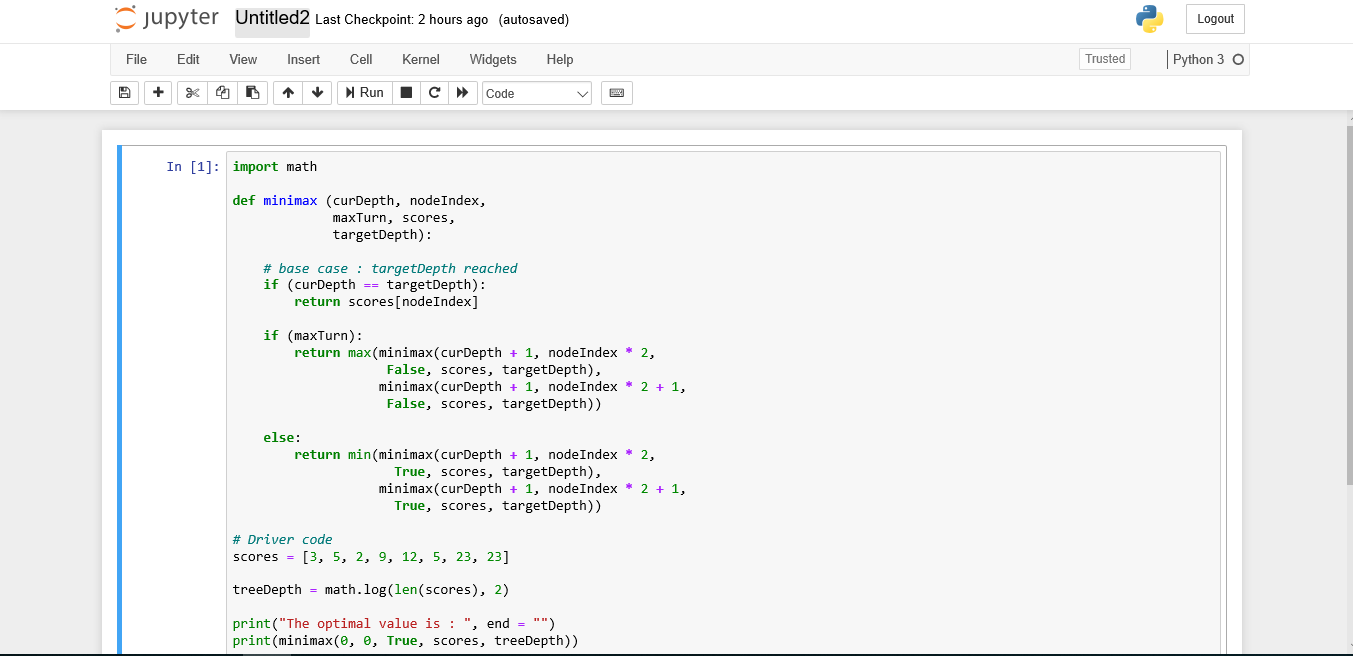
# Driver code

scores = [3, 5, 2, 9, 12, 5, 23, 23]

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

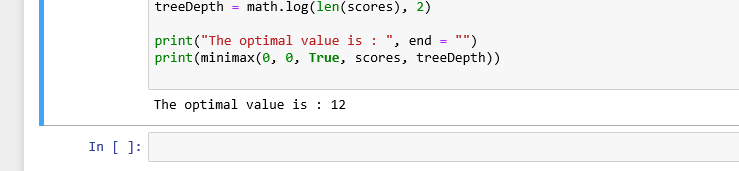
print("The optimal value is : ", end = "")

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

**OUTPUT:**  
The optimal value is : 12

**PYTHON CODE SCREENSHOT:**

**OUTPUT SCREENSHOT:**

****