**Group Members: Abdul Moiz Asif(263802)**

**Hassan Munir Ahmad(243211)**

**AI lab 9**

**Code:**

from binarytree import tree,bst

import time

import math

height\_tree = 3

tree = bst(height=height\_tree,is\_perfect=True)

def minimax(position, depth, maximizingPlayer):

if(depth == 0):

return position.value

elif(maximizingPlayer):

return max(-math.inf, minimax(position.left, (depth - 1), False), minimax(position.right, (depth - 1), False))

else:

return min(math.inf, minimax(position.left, (depth - 1), True), minimax(position.right, (depth - 1), True))

def minimax\_pruning(position, depth, alpha, beta, maximizingPlayer):

if (depth == 0):

return position.value

if (maximizingPlayer):

maxEval = -math.inf

for child in [position.left,position.right]:

eval = minimax\_pruning(child, (depth) - 1, alpha, beta,False)

maxEval=max(eval,maxEval)

alpha = max(alpha,eval)

if(beta <=alpha):

break

return maxEval

else:

minEval = math.inf

for child in [position.left, position.right]:

eval = minimax\_pruning(child, (depth) - 1, alpha, beta, True)

minEval = min(eval, minEval)

beta = min(beta, eval)

if (beta <= alpha):

break

return minEval

start = time.time()

print("MinMax:", minimax(tree, height\_tree, True))

print("Time using minimax:", time.time()-start)

start = time.time()

print("MinMaxPruned:", minimax\_pruning(tree, height\_tree, -math.inf, math.inf, True))

print("Time using minimaxPruned:", time.time()-start)

**Output:**

