30. Alpha-Beta Pruning for Tic Tac Toe

def alpha\_beta(board, depth, alpha, beta, is\_max):

if check\_win(board, 'O'):

return 1

if check\_win(board, 'X'):

return -1

if is\_full(board):

return 0

if is\_max:

max\_eval = -float('inf')

for i in range(3):

for j in range(3):

if board[i][j] == ' ':

board[i][j] = 'O'

eval = alpha\_beta(board, depth + 1, alpha, beta, False)

board[i][j] = ' '

max\_eval = max(max\_eval, eval)

alpha = max(alpha, eval)

if beta <= alpha:

break

return max\_eval

else:

min\_eval = float('inf')

for i in range(3):

for j in range(3):

if board[i][j] == ' ':

board[i][j] = 'X'

eval = alpha\_beta(board, depth + 1, alpha, beta, True)

board[i][j] = ' '

min\_eval = min(min\_eval, eval)

beta = min(beta, eval)

if beta <= alpha:

break

return min\_eval