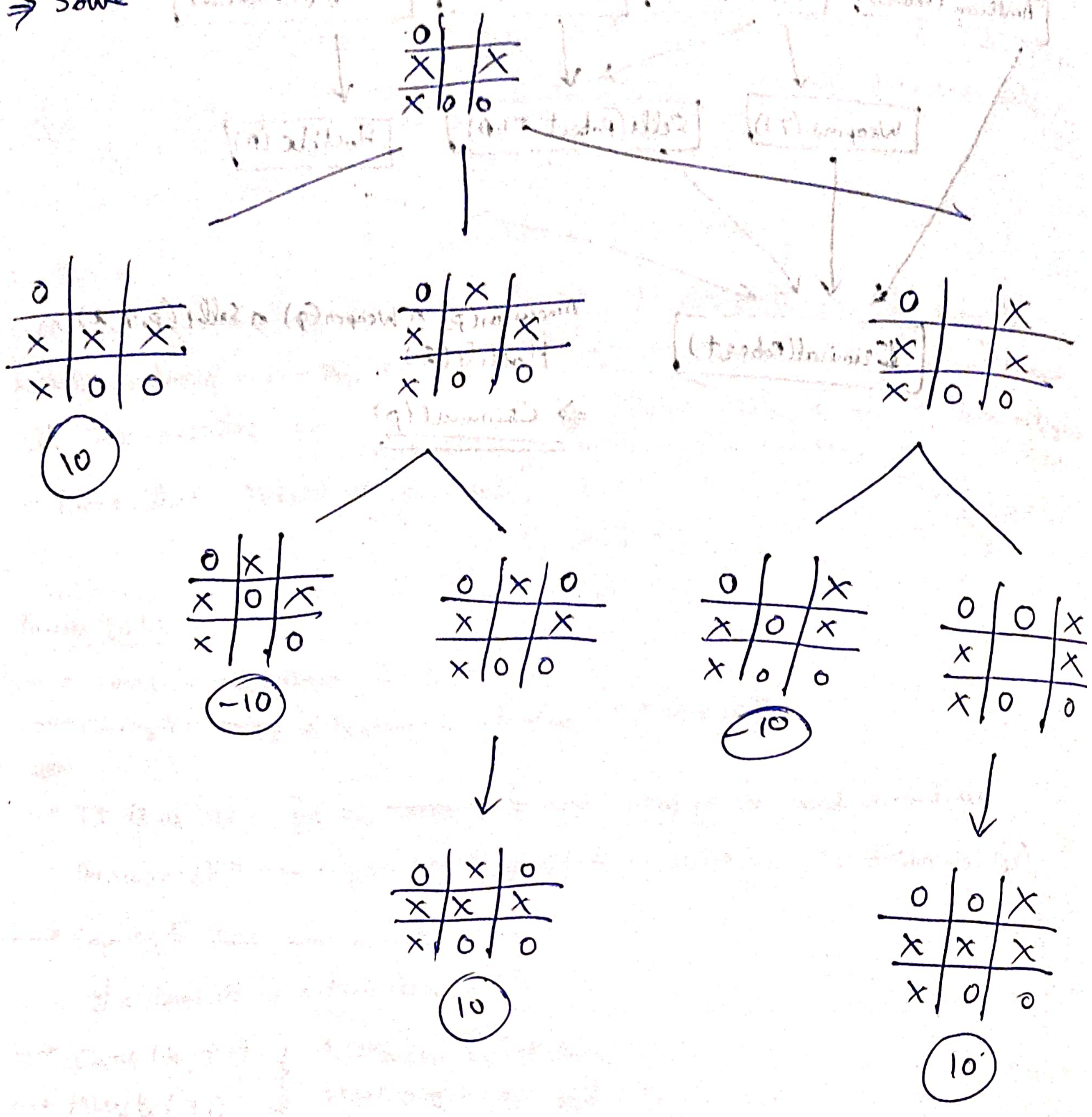
Lab-10

⇒ Solve Tic Tac Toe game using minimax algorithm



Algorithm:-

```
def score(game)
      if game.win?(@player)
            return 10
    else if game.win?(@opponent)
            return -10
    else
            return 0
    end
end
```

```
~~def minimax (game)~~
    ~~return score (game) if game.over ?~~
    ~~scores = []~~
    ~~moves = []~~

function minimax (board, depth, isMaximizing):
    if game_over (board):
        return evaluate (board)

    if isMaximizing :
        best_score = -∞
        for each empty cell (row, col) in board:
            simulate_move (board, row, col, 'X')
            score = minimax (board, depth + 1, False)
            undo_move (board, row, col)
            best_score = max (best_score, score)
        return best_score

    else :
        best_score = +∞
        for each empty cell (row, col) in board:
            simulate_move (board, row, col, 'O')
            score = minimax (board, depth + 1, True)
            undo_move (board, row, col)
            best_score = min (best_score, score)
        return best_score


function find_best_move (board, isMaximizing):
    best_move = (-1, -1)
    best_score = -∞ if isMaximizing else +∞
    for each empty cell (row, col) in board:
        simulate_move (board, row, col, 'X' if isMaximizing else 'O')
        move_score = minimax (board, 0, not isMaximizing)
        undo_move (board, row, col)
        if isMaximizing and move_score > best_score:
            best_score = move_score
            best_move = (row, col)
        elif not isMaximizing and move_score < best_score:
            best_score = move_score
            best_move = (row, col)
return best_move
```