

## Evaluation Strategy in Minimax

The evaluation strategy is implemented in the alphabeta function, but an abstracted version of the strategy is implemented in the evaluate function definition even though it isn't used for the grader's convenience. Essentially, the alphabeta function where the heavy lifting of the move calculation is done returns the move done and the player2 score(me)-player1 score if that move is taken. When deciding which move to take, it takes the move with the highest score on player2's turn, and the lowest score on player1's turn.

## Alpha Beta Pruning Parameter

Beta prunings occur when the beta values are 0, -1, -2, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, and 21 and the alpha values are 0, -1, 1, -2, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, and 21. No alpha prunings were made. These values are also printed out in the terminal.

## Time Taken By Algorithm to Make Move

Since my algorithm is only using depth 2, which I found to be the best performing depth since the moves are relatively fast and only take about 2-3 seconds and the performance against the AI is acceptable. Increasing depth increases the time exponentially, and by a depth of 4, the program takes 45 seconds per turn, and wins with the same score as depth 2.