Tic Tac Toe:

The Minimax and Alpha-Beta pruning algorithm over-calculates all possibilities of one game.

When playing against minimax algorithm and human player place the first mark, let’s say ‘X’, then the algorithm will go through **255168** different scenarios which is the most scenarios tic-tac-toe can have. However, if we optimize it with Alpha-beta pruning, this number is reduced to about 8000 and decrease exponentially as game goes on.

For Alpha-Beta pruning algorithm,

Each side adopts either maximizer alpha-max or minimizer beta-min. One is trying to get highest score possible where another tries on minimum.

I think it's reasonable that we need points to calculate, compare, and prune in the min/max process as a Utility state. For example, if AI wins, we add 10 points to AI, and if player wins, -10 for the AI. So, while AI playing as maximizer, it generates score for each cell and makes decision which cell is best to place, which means that AI has chosen a branch in a muti-leaves tree.

The reason that scenarios are exponentially decrease is because when human player responses, there are many moves simulated by AI are omitted. For example, if human and AI both placed 1 mark, algorithm still assumes what if human player place on all other 7 empty cells and so on, but once human player place it somewhere, other 6 cell branches of possibilities are no longer hold.

There should be 2 function when apply in code, Alpha and Beta, that look for max() and min(), there are recursively called and bounded together in one function (can be called something like alpha-beta search)so it loops through all scenarios and returns the optimal action for the current player on the board. If we are doing one AI and one player game, it has if statement to detect whether AI is having mark ‘X’ or ‘O’ and call Alpha or Beta function accordingly, and human player will not need it.

Human player plays which mark, go first or second, are not making difference. If we let algorithm go first, since board is empty and it just going to place mark randomly.

Here is one example AB pruning code (include GUI of the game)

[CS50\_AI\_Tic/tictactoe.py at main · reshanf-iiitd/CS50\_AI\_Tic (github.com)](https://github.com/reshanf-iiitd/CS50_AI_Tic/blob/main/tictactoe.py)

Also, I would like to share this video which I found extremely helpful for me to understand these concepts.

[I made an unbeatable Tic Tac Toe AI (Minimax algorithm) - YouTube](https://www.youtube.com/watch?v=3nzupVMpZeA)