**Games as Trees**

Have you ever played a game against someone and felt like they were always two steps ahead? No matter what clever move you tried, they had somehow envisioned it and had the perfect counterattack. This concept of thinking ahead is the central idea behind the *minimax algorithm*.

The minimax algorithm is a decision-making algorithm that is used for finding the best move in a two player game. It's a recursive algorithm — it calls itself. In order for us to determine if making move A is a good idea, we need to think about what our opponent would do if we made that move.

We'd guess what our opponent would do by running the minimax algorithm from our opponent's point of view. In the hypothetical world where we made move A, what would they do? Surely they want to win as badly as we do, so they'd evaluate the strength of their move by thinking about what we would do if they made move B.

As this process repeats, we can start to make a tree of these hypothetical game states. We'll eventually reach a point where the game is over — we'll reach a leaf of the tree. Either we won, our opponent won, or it was a tie. At this point, the recursion can stop. Because the game is over, we no longer need to think about how our opponent would react if we reached this point of the game.

Instructions

On this page, you'll see the game tree of a Tic-Tac-Toe game that is almost complete. At the root of the node, it is "X"'s turn.

Some of the leaves of the tree still have squares that can be filled in. Why are those boards leaves?

