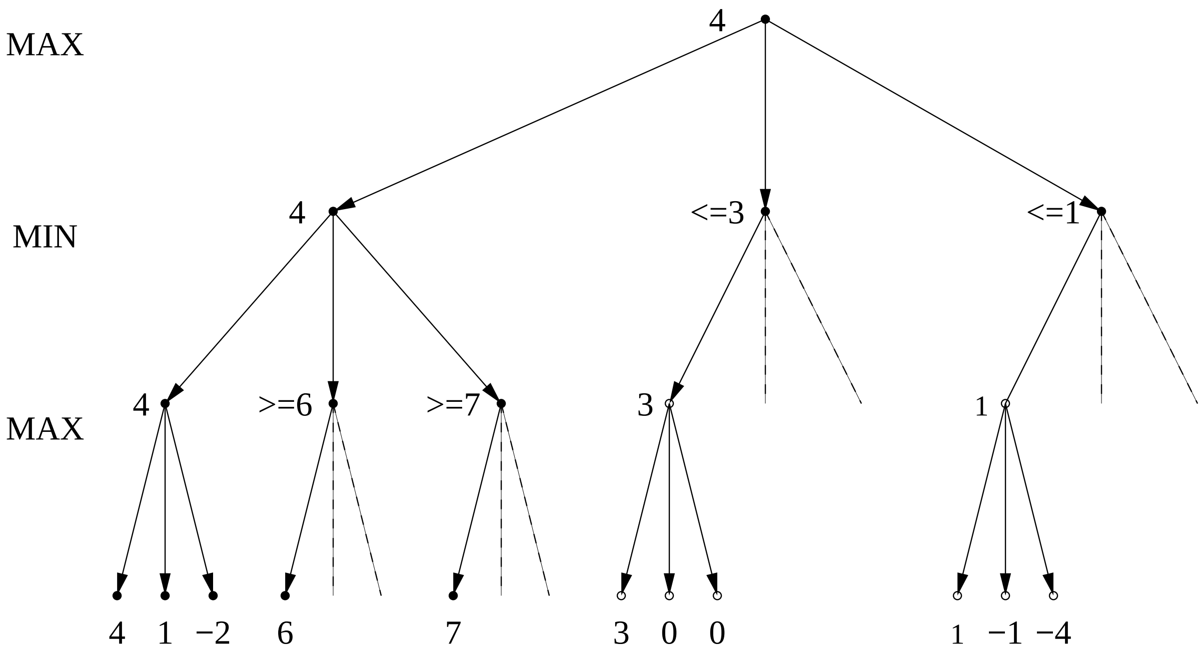
Best-Case Analysis of Alpha-Beta Pruning

In this analysis, we consider the *best-case behavior* of alpha-beta pruning. Although optimistic, it turns out that this corresponds reasonably well to its usual behavior. An example of best-case behavior is illustrated in the following figure. Dotted lines indicate pruned branches.



In this figure, the branching factor is 3, and the depth is 3. Notice that when states are considered for MAX's move, they are considered from highest to lowest evaluation. For MIN's move, they are considered from lowest to highest. A good way to approximate this ordering is to simply apply the evaluation function and sort the states accordingly. Generally, the maximum/minimum state will be among the first few. [How well does this work for the Tic-Tac-Toe evaluation function?]

Now assuming that best-case analysis is a reasonable thing to do in this case, let's first consider recursive equations which give the number of states to be considered. From the equations, we will determine a rough bound on the branching factor.

Note that in the figure, we either know the value of a state exactly, or we know a bound on its value. To determine the exact value of a state, we need the exact value of one of its children, and bounds on the rest of its children. To determine a bound of a state's value, we need the exact value of one of its children.

Based on these observations, let $S(k)$ be the minimum number of states to be considered k ply from a given state when we need to know the exact value of the state. Similarly, let $R(k)$ be the minimum number of states to be considered k ply from a given state when we need to know a bound on the state's value. As usual, let b be the branching factor. Thus, we have:

$$S(k) = S(k-1) + (b-1)\, R(k-1)$$

i.e., the exact value of one child and bounds on the rest, and

$$R(k) = S(k-1)$$

i.e., the exact value of one child. The base case is $S(0) = R(0) = 1$. Note, for the above figure, this gives $S(3) = b^2 + b - 1 = 11$ for $b = 3$.

When we expand the recursive equation, we get:

$$\begin{aligned} S(k) &= S(k-1) + (b-1)\, R(k-1) \\ &= (S(k-2) + (b-1)\, R(k-2)) + (b-1)\, S(k-2) \\ &= b\, S(k-2) + (b-1)\, R(k-2) \\ &= b\, S(k-2) + (b-1)\, S(k-3) \end{aligned}$$

It is obvious that $S(k-3) < S(k-2)$, so:

$$\begin{aligned} S(k) &< (2b-1)\, S(k-2) \\ &< 2b\, S(k-2) \end{aligned}$$

That is, the branching factor every two levels is less than $2b$, which means the effective branching factor is less than $\sqrt{2b}$.

So, for even k, we derive $S(k) \le \left(\sqrt{2b}\right)^k$, which is not too far off the asymptotic upper bound of $\left(\sqrt{b} + 1/2\right)^{k+1}$. In effect, alpha-beta pruning can nearly double the depth that a game tree can be searched in comparison to straightforward minimax.