Game playing

- We will focus on two player, perfect information zero-sum games:
  - perfect information — no hidden state
  - zero-sum — what is good for one player is bad for the other
- We will use search to analyze these games...
- But this search is different because the two players alternate and have different objectives.
- Assume both players will always make the best move
- Key question: What is the best move?
MINIMAX search

MINIMAX(n)
1. return MAX-PLAYER(n)

MAX-PLAYER(n)
1. if game corresponding to node n is over, return value
2. find the children C of node n
3. for each child $c_i \in C$, let $v_i = \text{MIN-PLAYER}(c_i)$
4. return maximum $v_i$

MIN-PLAYER(n)
1. if game corresponding to node n is over, return value
2. find the children C of node n
3. for each child $c_i \in C$, let $v_i = \text{MAX-PLAYER}(c_i)$
4. return minimum $v_i$
MINIMAX search with alpha-beta pruning

\[
\text{AB/MINIMAX}(n)
\]
1. return \(\text{AB/MAX-PLAYER}(n, -\infty, \infty)\)

\[
\text{AB/MAX-PLAYER}(n, \alpha, \beta)
\]
1. if game is over or depth cutoff reached, 
   evaluate game state and return value.
2. find the children \(C\) of node \(n\)
3. for each child \(c_i \in C\)
   - \(\alpha \leftarrow \max(\alpha, \text{AB/MIN-PLAYER}(c_i, \alpha, \beta))\)
   - if \(\alpha \geq \beta\), return \(\beta\)
4. return \(\alpha\)

\[
\text{AB/MIN-PLAYER}(n, \alpha, \beta)
\]
1. if game is over or depth cutoff reached, 
   evaluate game state and return value.
2. find the children \(C\) of node \(n\)
3. for each child \(c_i \in C\)
   - \(\beta \leftarrow \min(\beta, \text{AB/MIN-PLAYER}(c_i, \alpha, \beta))\)
   - if \(\alpha \geq \beta\), return \(\alpha\)
4. return \(\beta\)