α–β Pruning

Process:

- At every leaf node the MinMax value (utility at leaf node) is calculated,
- For every **MAX Player** node, the current **LARGEST** child MinMax value is saved in α,
- For every **MIN Player** node, the current **SMALLEST** child MinMax value is saved in β,
- If at a **MIN Player** node m the current value $\beta \leqslant \alpha$, then the search at node m can end. Here α is the **LARGEST** value of a **MAX Player** node in the path from the root to node m,
- If at a **MAX Player** node n the current value $\alpha \geqslant \beta$, then the search at node n can end. Here β is the **SMALLEST** value of a **MIN Player** node in the path from the root to node n.

If the root node is **MAX Player** node, start with: **AlphaBetaMax**(Node, $-\infty$, ∞)
If the root node is **MIN Player** node, start with: **AlphaBetaMin**(Node, $-\infty$, ∞)

```
AlphaBetaMax(Node, α, β)
IF isLeafNode(Node) is true THEN Return MinMaxValue(Node)
childrenNodes = getChildren(Node)
WHILE childrenNodes is NOT empty
    α = max(α, AlphaBetaMin(first(childrenNodes), α, β))
    IF α ⩾ β THEN Return β
    childrenNodes = rest(childrenNodes)
Return α
```

```
AlphaBetaMin(Node, α, β)
IF isLeafNode(Node) is true THEN Return MinMaxValue(Node)
childrenNodes = getChildren(Node)
WHILE childrenNodes is NOT empty
    β = min(β, AlphaBetaMax(first(childrenNodes), α, β))
    IF β ⩽ α THEN Return α
    childrenNodes = rest(childrenNodes)
Return β
```