Algorithm 1 MaxN-MC(Node *node*, int *depth*, int *numMCRollOuts*)

```
if node.isTerminal() then
    return evaluationFunction(node)
end if
if $depth > 0$ then
    for all child of node do
        child.value ← MaxN-MC(child, $depth - 1$)
    end for
    Find child of node with greatest child.value(node.getPlayerIndex())
    return child.value
else
    for $i = 1$ to numMCRollOuts do
        values(i) ← MonteCarloRollOut(node)
    end for
    return component-wise average of values
end if
```

Algorithm 2 MonteCarloRollOut(Node *node*)

```
if node.isTerminalNode() then
    return evaluationFunction(node)
end if
Pick child from node at random {Could replace with a bias given some heuristic}
return MonteCarloRollOut(child)
```