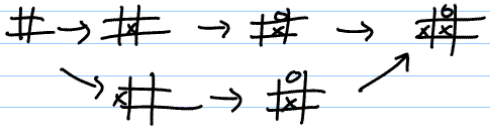
MINIMAX(n) returns game theoretic value of node n
(can also be used w/ heuristics)

if n is a leaf
    find winner, return value (ex: +1 for P1 won, 0 for draw, -1 for P2 won)
else if n is a max node (P1's move)
    return $\max_{c \text{ is a child of } n}$ MINIMAX(c)
else
    return $\min_{c \text{ is a child of } n}$ MINIMAX(c)

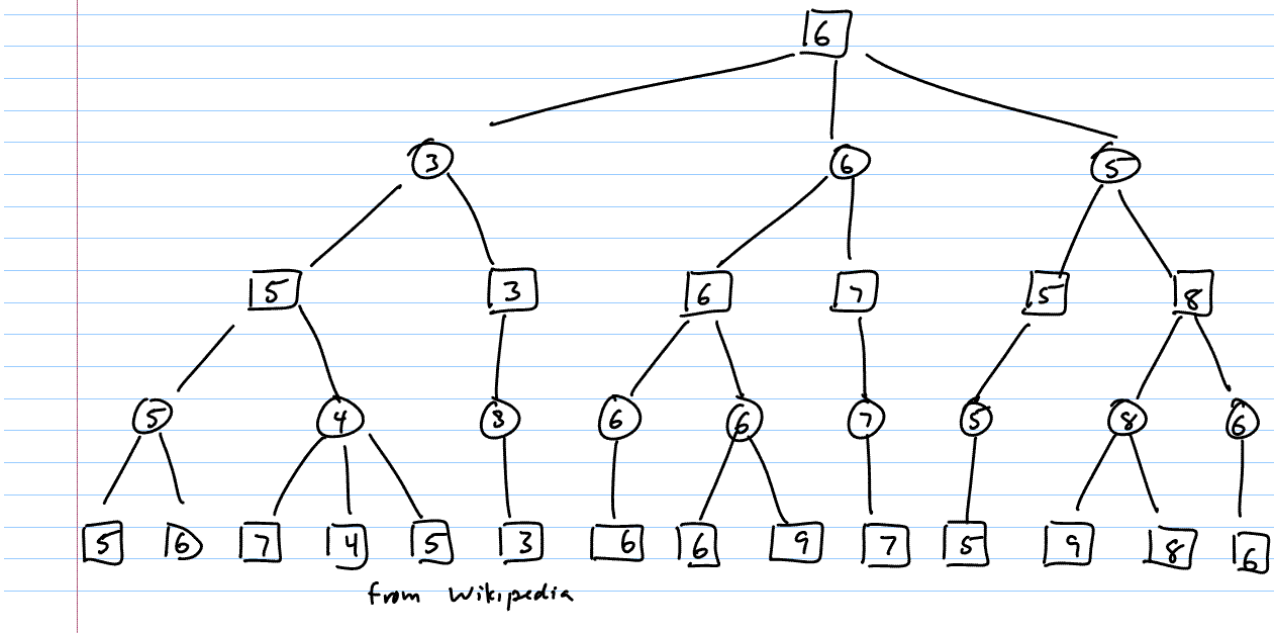
Problem: same node may be reachable > 1 way



Solution: maintain map (hash table) of previously seen positions, their values

New Problem: space

New Solution: some sort of cache


□ - Max
○ - Min
from Wikipedia

α/β Pruning: keep track

α: lower bound on value of (not necessarily proper) MAX ancestor

β: upper bound on value of (not necessarily proper) MIN ancestor

return value might not be actual game theoretic value unless $(\alpha, \beta) = (-\infty, \infty)$, but the returned value, if a lie, will not be a hurtful lie

```
ALPHABETA(n, α, β)
{
    if n is leaf
        return its value
    else if n MAX node
        for each child c
            v = ALPHABETA(c, α, β)
            α = max(α, v)
            if (α ≥ β)                    // MIN player won't let us get here
                return α
        return α
    else if n is MIN
        for each child c
            v = ALPHABETA(c, α, β)
            β = min(β, v)
            if (α ≥ β)
                return β
        return β
```