Transposition Table

Minimax (pos, h, depth) — h: heuristic; tt; depth: level below pos at which to apply h

If pos is terminal, return value(pos)

If depth == 0, return h(pos)

if (pos, d) in tt for d ≥ depth then return tt[pos, d]

Else if pos is P1's turn then return $\max_{pos \to pos'} MM(pos', h, depth-1)$

Else return $\min_{pos \to pos'} MM(pos', h, depth-1)$

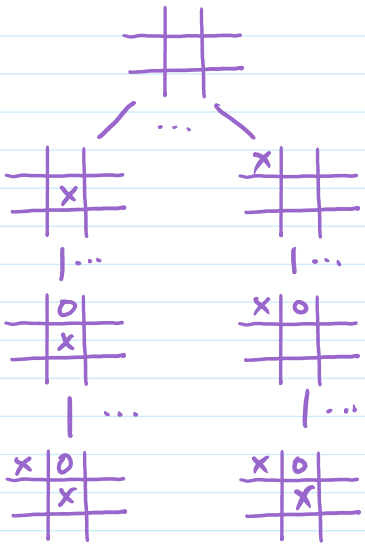
tt[pos, depth] = ...

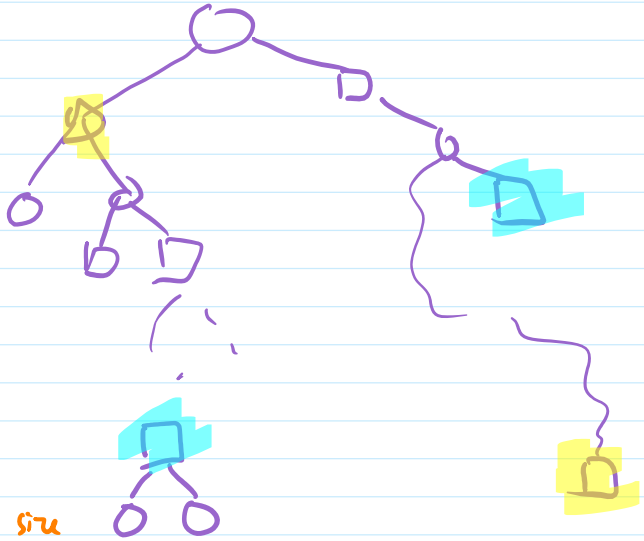
tt[pos, depth] =

Transposition Table

↳ memo (map from positions to position values)

ex. +1 for P1 winning strat
-1 for P2
or some estimate

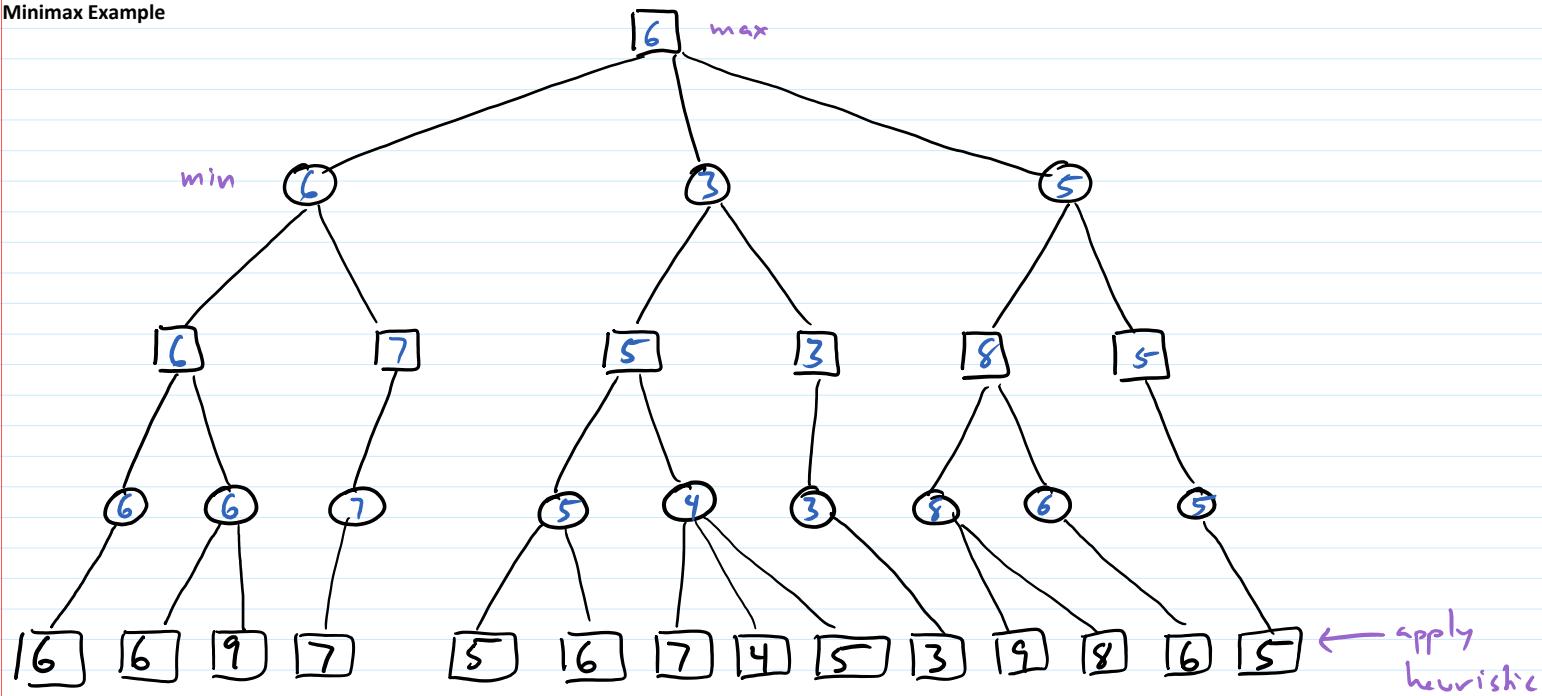




as trans. table grows,

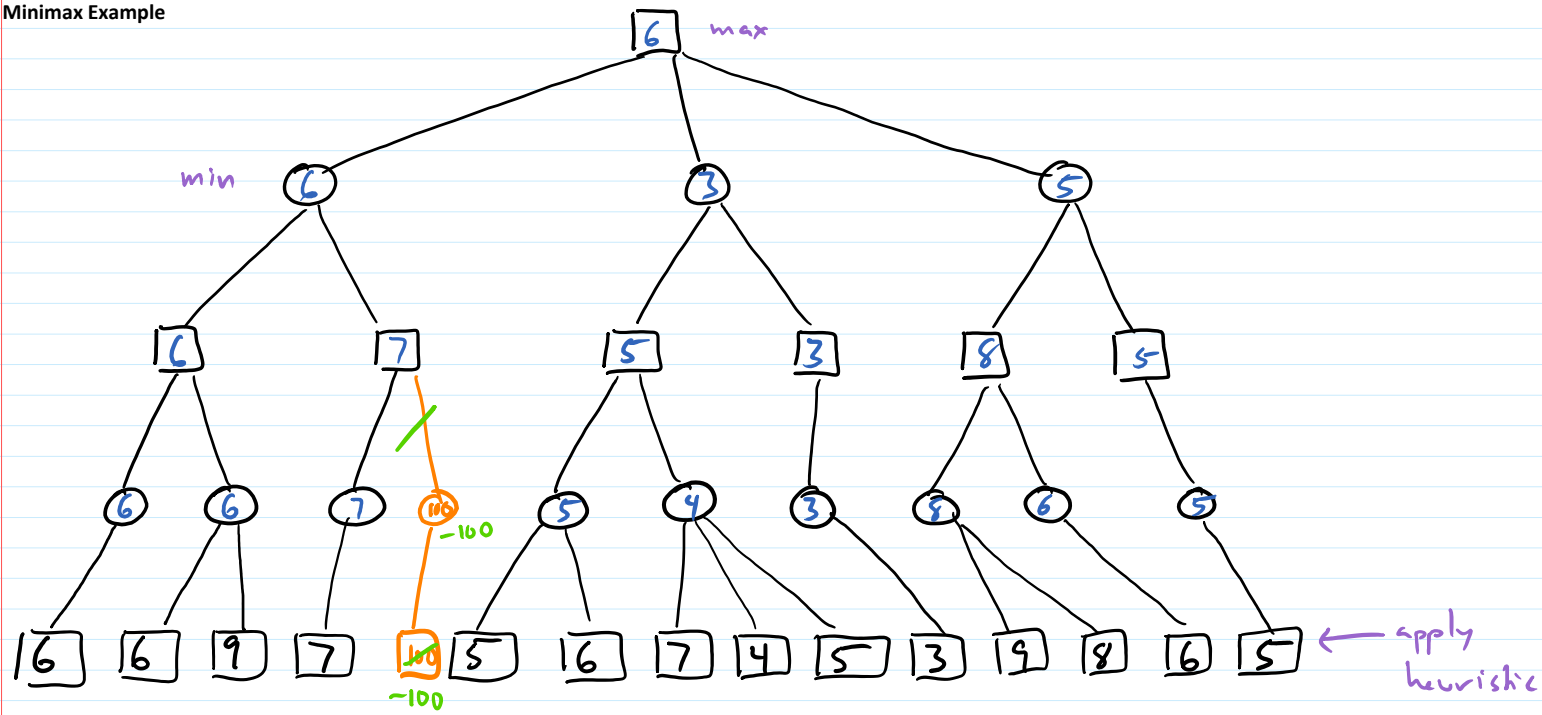
use replacement policy to manage size

Minimax Example



Modified example from http://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning

Minimax Example



Modified example from http://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning

Alpha-Beta Pruning

$-\infty, \infty$ for call to root (so returns same thing as $MM(root, depth, h)$)

Alpha-Beta $(p, \alpha, \beta, depth, h)$ returns

value (p) if p is terminal

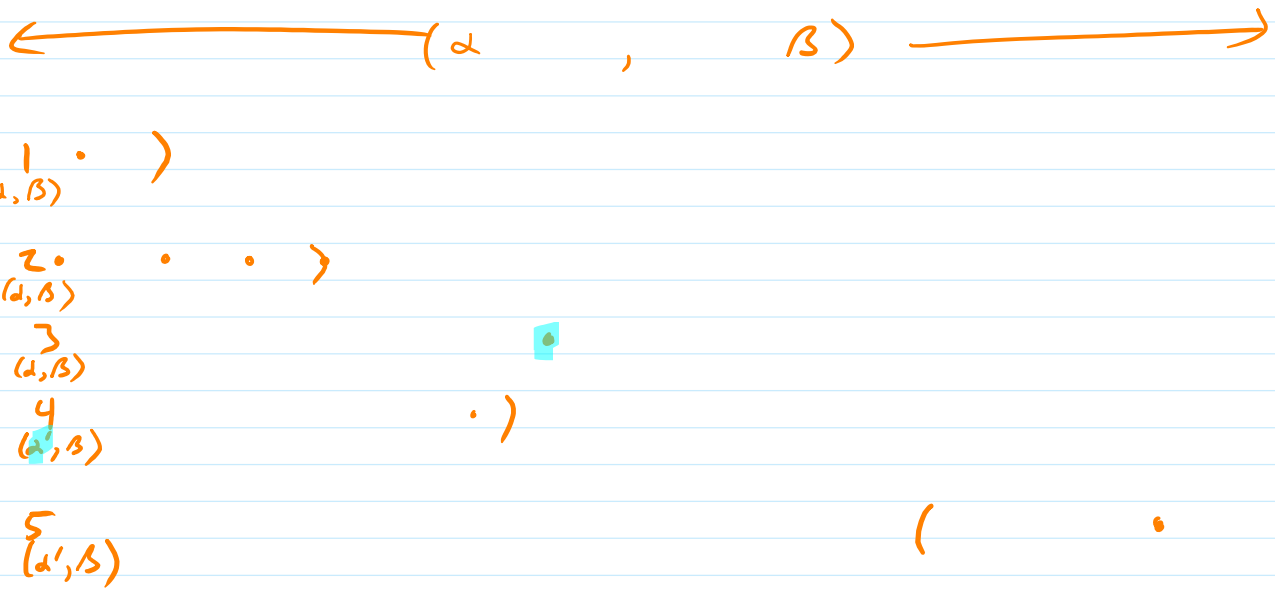
$h(p)$ if $depth = 0$

result of normal minimax (w/ same depth, heuristic) → $MM(p,d,h)$ if $\alpha \le MM(p,d,h) \le \beta$

lower bound $\geq \beta$ on $MM(p,d,h)$ if $MM(p,d,h) \geq \beta$

upper bound $\leq \alpha$ on $MM(p,d,h)$ if $MM(p,d,h) \leq \alpha$

```
if depth = 0 then return heuristic(p)
if p is terminal then return value(p)
if p is a max position
        a ← -∞
        for each position p' reachable in one move from p and while α<β
            a ← max(a, AB(p', α, β, depth-1, h))
            α ← max(α, a)
        return a
else
        b ← ∞
        for each position p' reachable in one move from p and while α<β
            b ← min(b, AB(p', α, β, depth-1, h))
            β ← min(b, β)
        return b
```


(α , β)
child 1
(α, β)
child 2
(α, β)
child 3
(α, β)
child 4
(α', β)
child 5
(α', β)

Alpha-Beta Example

Modified excerpt from https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pr