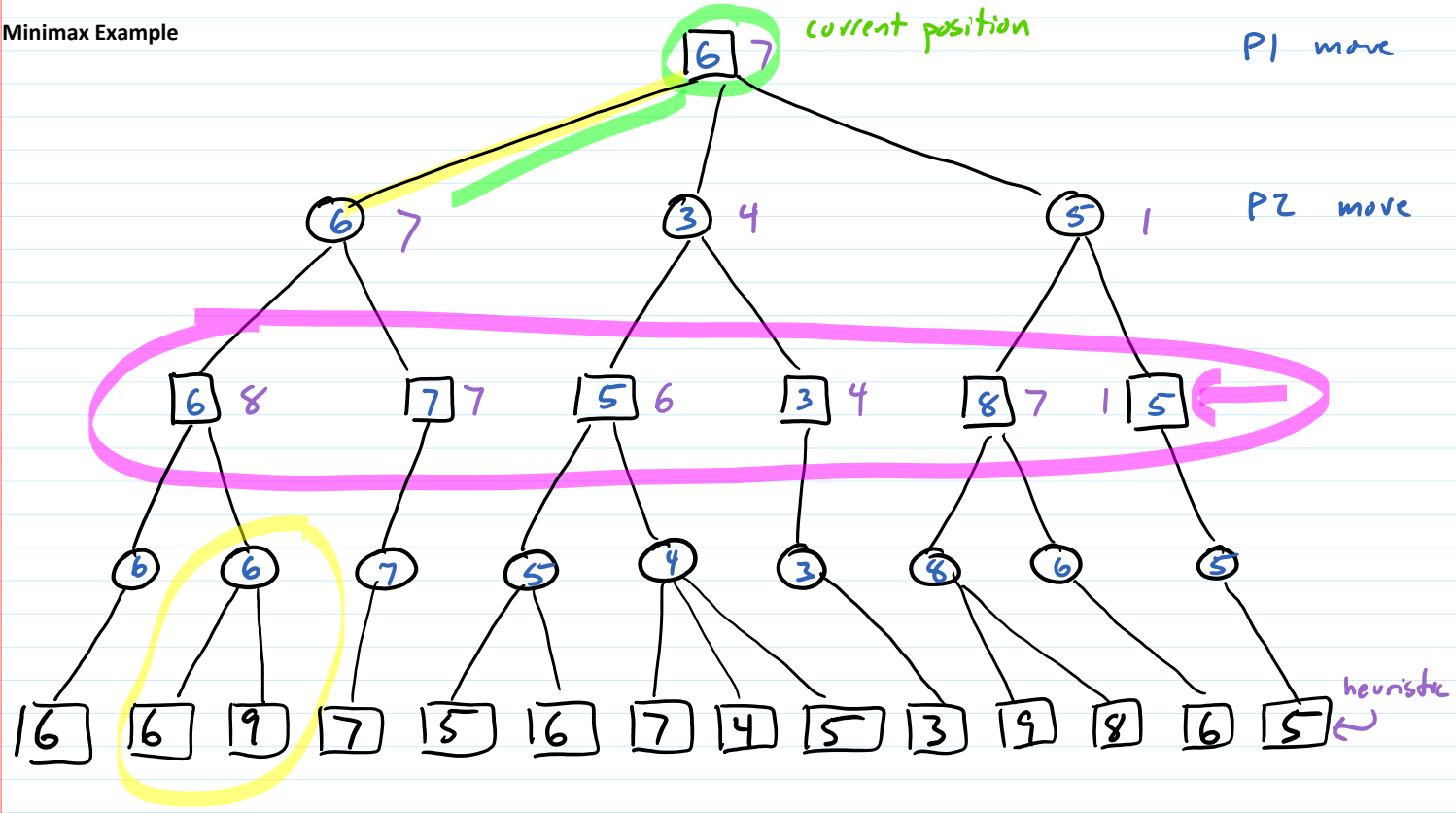
Calendar

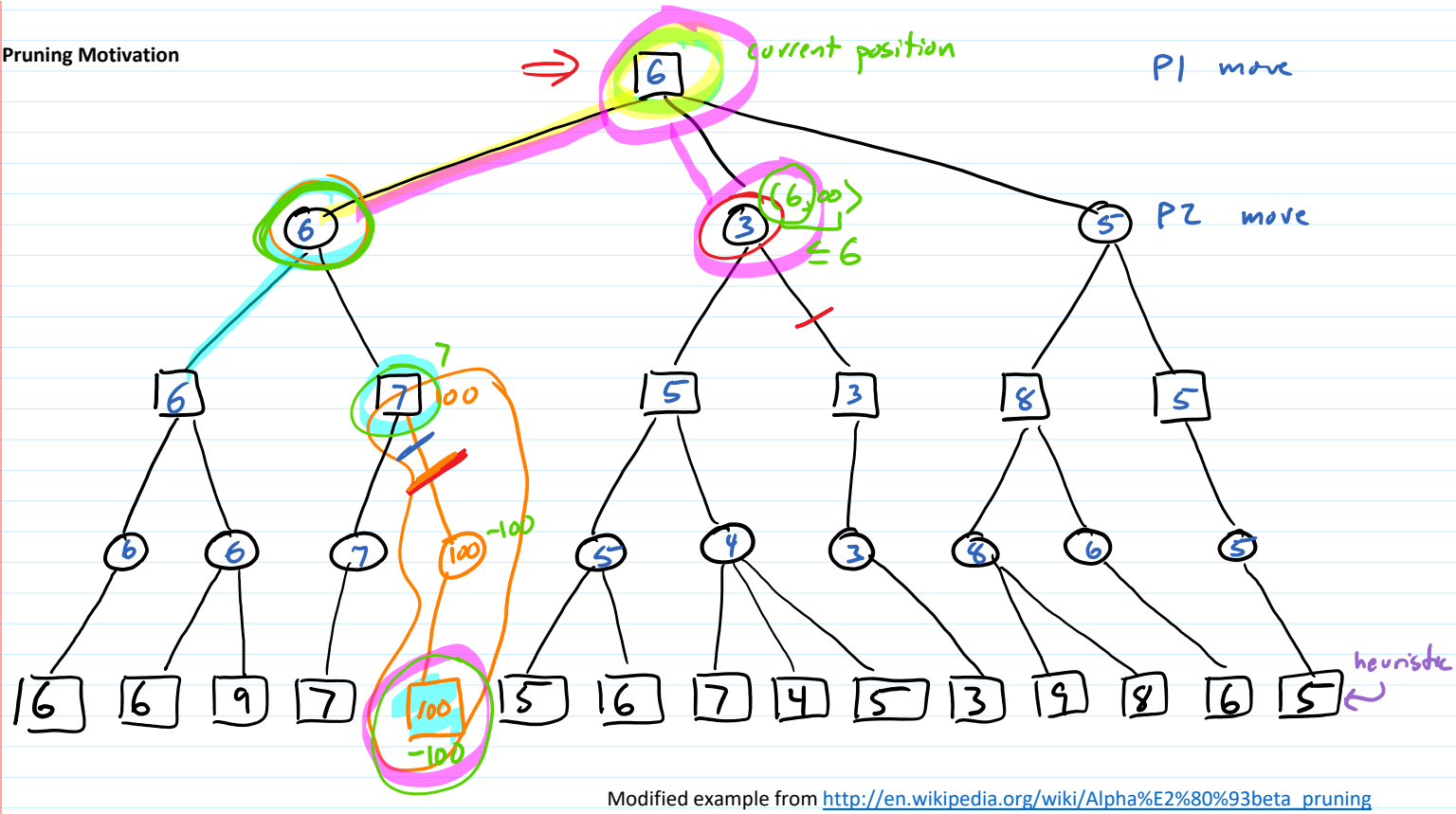
Oct	14	P3 (Blotto) due
	21	Q2 Simultaneous games
	28	P4 (search) due
Nov	6 Fri	Q3 Search quiz
	11	
	18	P5 (RL) due
	23-27	
Nov	30/Dec 4	videos due / responses due
Dec	10	Final projects due

Minimax Example



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

Pruning Motivation



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

Minimax w/ Transposition Table

depth bound

Minimax (pos, h, d)

determined by rules: +1 P1 win, 0 draw, -1 P2 win

If pos is terminal, return value(pos)

If depth == 0, return h(pos)

If (pos, d') for some $d' \geq d$ then return tt[pos, d']

Else if pos is P1's turn then

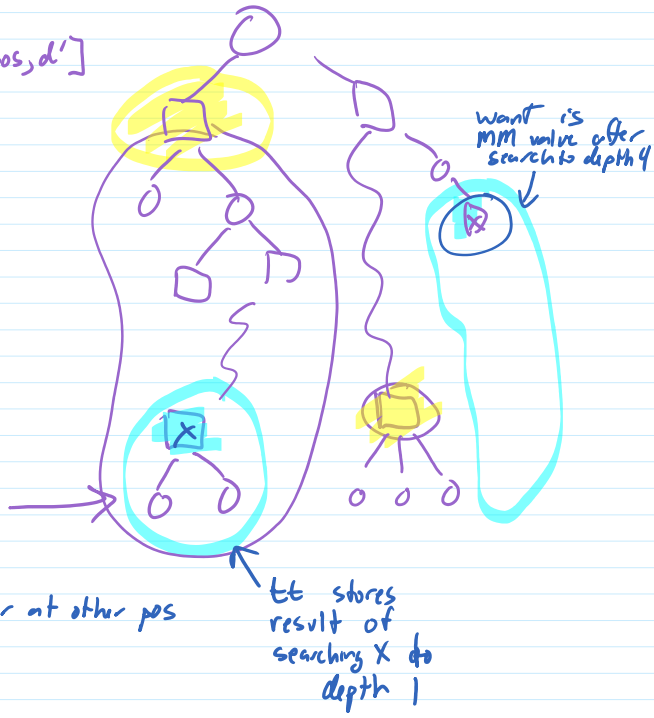
tt[pos, d] =

return $\max_{pos \to pos'} MM(pos', h, d-1)$

Else

tt[pos, d] =

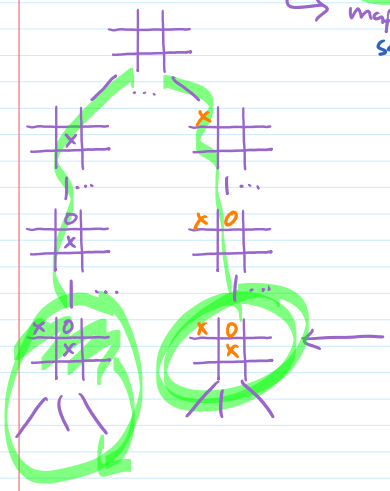
return $\min_{pos \to pos'} MM(pos', h, d-1)$



Transposition Table - (like memo in memoization)

↳ map from positions → values

save values for positions in case they reappear at other pos in tree



need replacement policy to avoid t.t. getting too big

stop storing new things?

replace least recently used item?

(cache replacement policies)

Alpha-Beta Pruning

range of values we care about

Alpha-Beta (p, α, β, h, depth) returns

postconditions

value (p) if p is terminal

h (p) if depth = 0

MM (p, h, depth) if $\alpha \le MM(p,d,h) \le \beta$

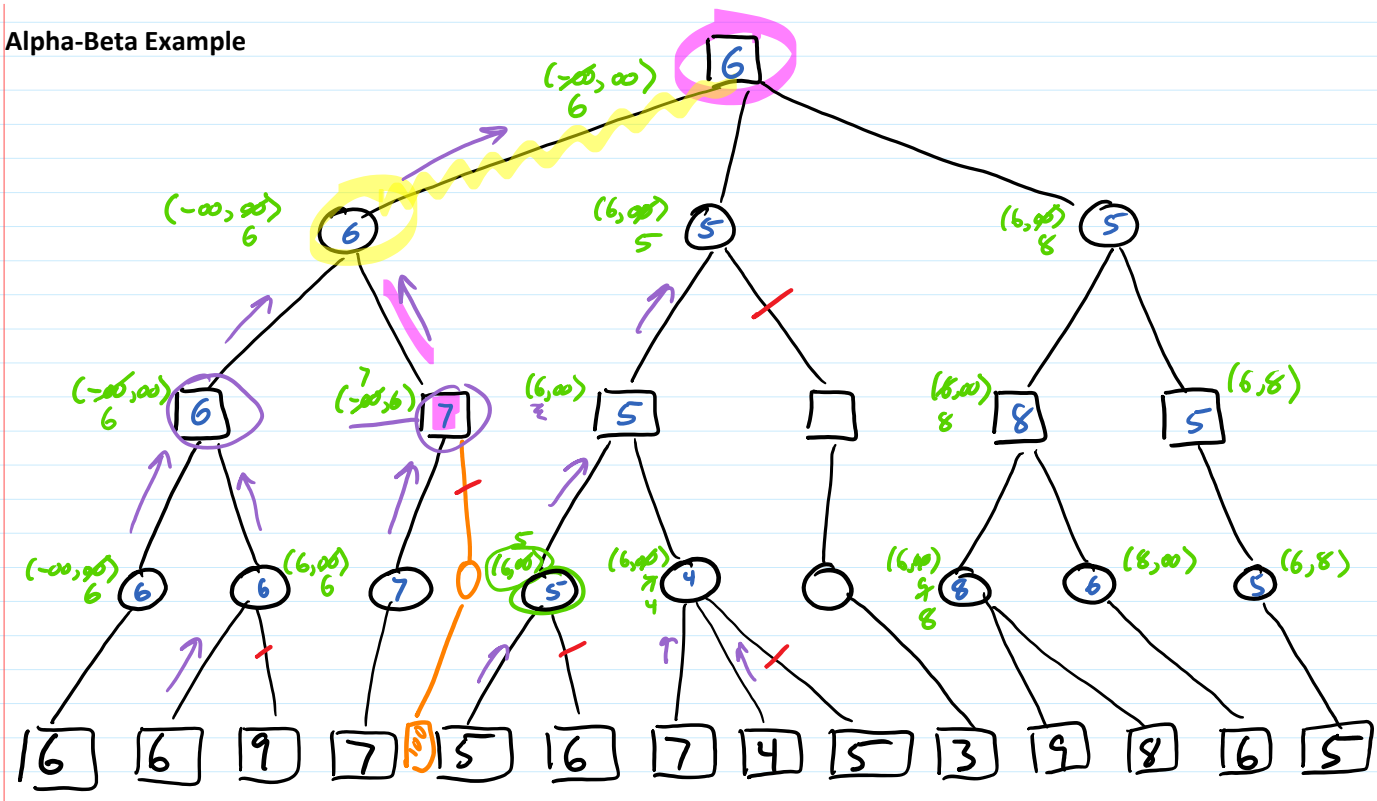
lower bound $\ge \beta$ on MM(p,d,h) if $MM(p,d,h) \ge \beta$

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

```
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, h, depth-1))
                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, h, depth-1)
                β ← min (β, b)
        return b
```

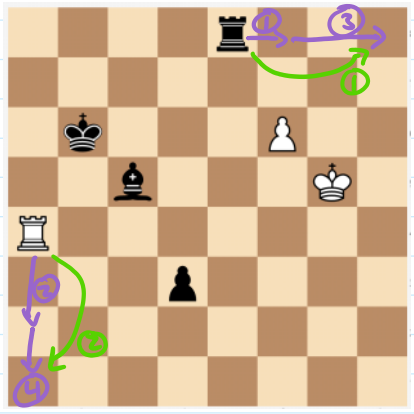
Alpha-Beta Example



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

Transposition Table

Positions may be reachable by multiple sequences of moves



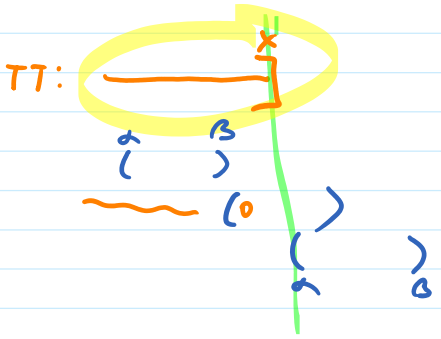
Keep table of values for all positions examined in tree

Keys: positions

Values: (value/bound, move, depth)

Add check at start of A-B

if pos present and searched depth ≥ desired depth
if value is exact, return value
else if upper bound ≤ α return X
≤ X
else if lower bound ≥ β return X
≥ X



Save returned values in table

if value ≥ β store lower bound
else if value ≤ α upper bound
else store exact value

Scout Example