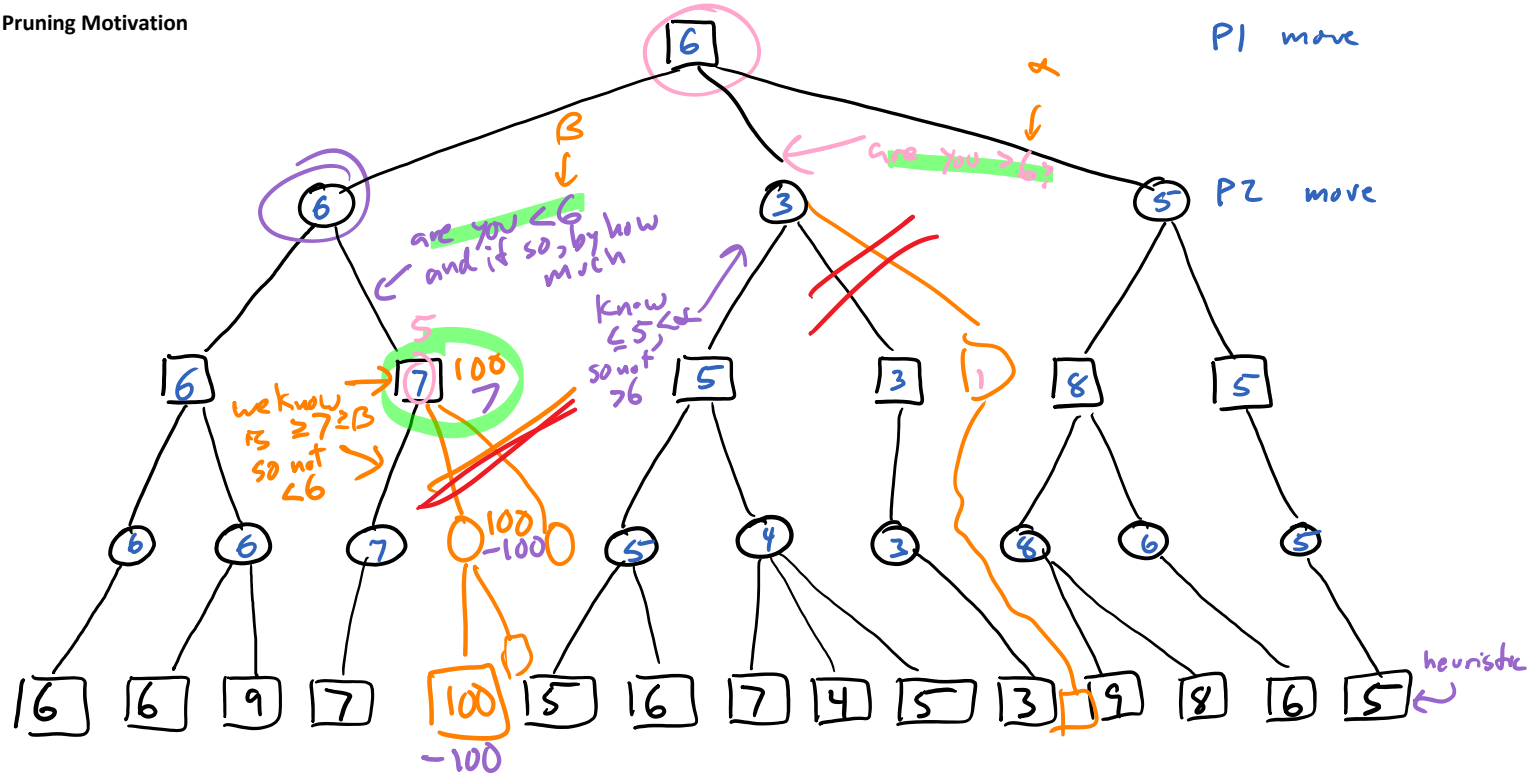
Pruning Motivation



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

Alpha-Beta Pruning

range of values to distinguish between

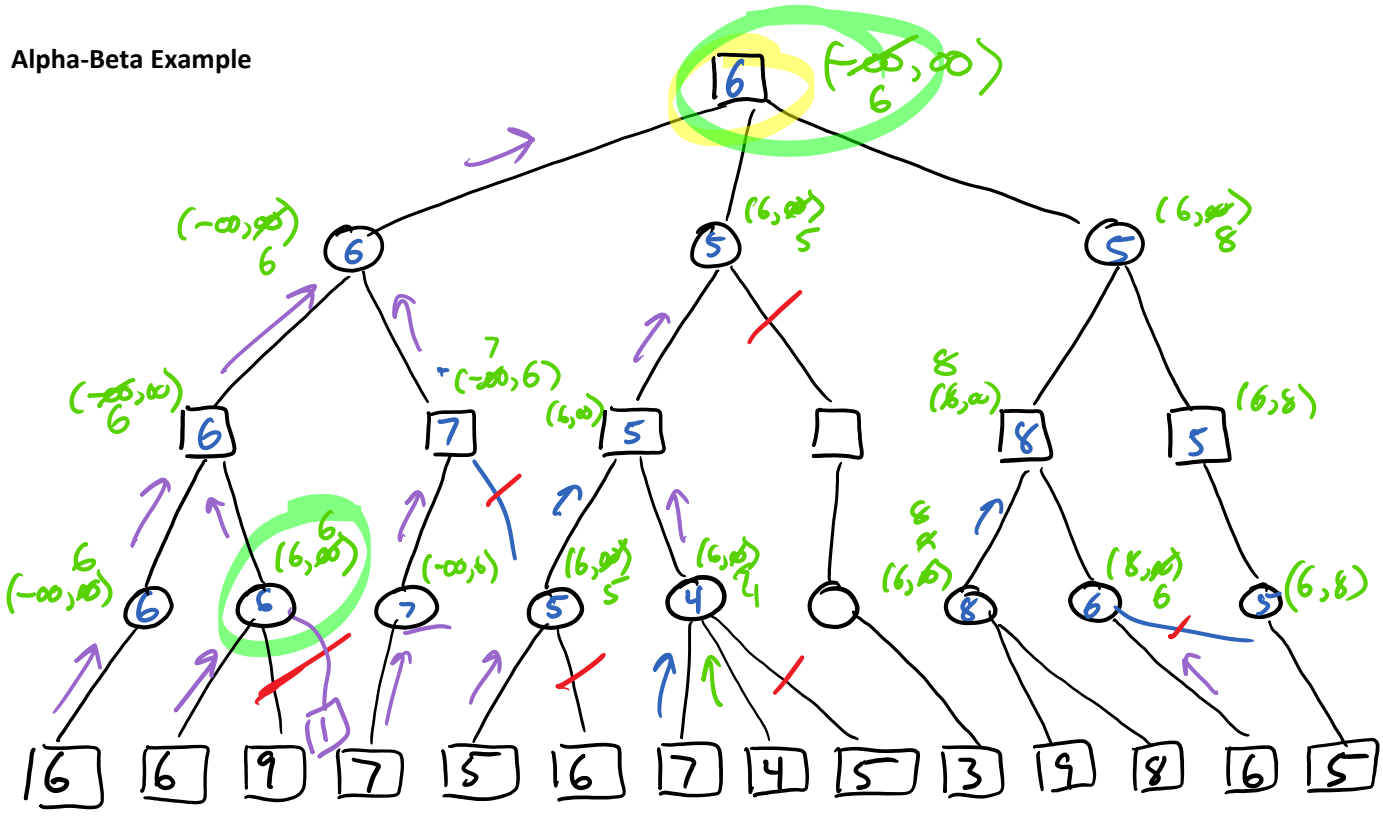
Alpha-Beta $(p, \alpha, \beta, h, depth)$ returns

postconditions:
- value (p) if p is terminal
- $h(p)$ if $depth = 0$
- $MM(p,h,d)$ if $\alpha \leq MM(p,h,d) < \beta$
- lower bound $\geq \beta$ on $MM(p,h,d)$ if $MM(p,h,d) \geq \beta$
- upper bound $\leq \alpha$ on $MM(p,h,d)$ if $MM(p,h,d) \leq \alpha$

```
if depth = 0 then return heuristic(p)        ← can use transposition table too → keys include α,β values specify ≤, ≤, or ≥
if p is terminal then return value(p)
if p is a max position                       (P1)
    a ← -∞   (value of best child so far — a lower bound on value of p)
    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                                         (P2)
    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
```

α = -∞
1st child makes a = 6
? to 2nd child is value ≥ 6
α

Alpha-Beta Example



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

Multi-Armed Bandit

Given unknown probability distributions $R_1, \ldots, R_K$ with means $\mu_1, \ldots, \mu_K$

Choose indices $i_1, i_2, \ldots$ to optimize payout

Regret = diff between observed reward and (play arm w/ highest μ_i) best possible expected reward

$$\rho_T = T \cdot \mu^* - \sum_{t=1}^{T} \hat{r}_t$$

μ^*: highest μ_i; $\hat{r}_t$: reward obtained at time t

"optimal" means zero average regret

$$P\left(\lim_{T\to\infty} \frac{\rho_T}{T} = 0\right) = 1$$

Ex:

	Arm 1		Arm 2		Arm 3	
	prob	payout	prob	payout	prob	payout
	$\frac{1}{3}$	2	$\frac{1}{4}$	5	$\frac{1}{100}$	200
	$\frac{2}{3}$	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{99}{100}$	0
			$\frac{1}{2}$	0		

$\mu_1 = \frac{2}{3}$ $\mu_2 = \frac{7}{8}$ $\mu_3 = 2$

uniform rotation: cycle through each arm 1 2 3 1 2 3 . . . .

$\frac{1}{3}$ of time average $(2 - \frac{1}{3})$ regret

$$\lim_{T\to\infty} \frac{\rho_T}{T} \geq \frac{4}{9}$$

greedy : play each machine once, then always play one [illegible] from

greedy : play each machine once, then always play one w/ highest reward from 1st round

if pick best machine after 1 round, $\lim_{T\to\infty} \frac{\rho_T}{T} = 0$

but P(pick best machine) < 1

ε-greedy : play one round, then last observed mean reward w/p $1-\varepsilon$ and randomly w/p ε

balances exploration/exploitation

still has $\lim_{T\to\infty} \frac{\rho_T}{T} \geq \underbrace{\frac{\varepsilon}{2}}_{\text{p(choose arm 1)}} \cdot \overbrace{\frac{4}{3}}^{\text{mean regret from arm 1}}$

zero regret

UCB

Choose arm j that maximizes $\bar{r}_j + \sqrt{\frac{2 \ln T}{n_j}}$

exploitation: $\bar{r}_j$ = mean reward for arm j

exploration: $\sqrt{\frac{2 \ln T}{n_j}}$; n_j = # plays of arm j

Monte Carlo Tree Search

tree ← root

asymmetric tree growth

no heuristic needed

UCB (choosing a child w/0 visits when one exists)

Until out of time

traverse tree root ⇝ leaf

UCT = MCTS + UCB

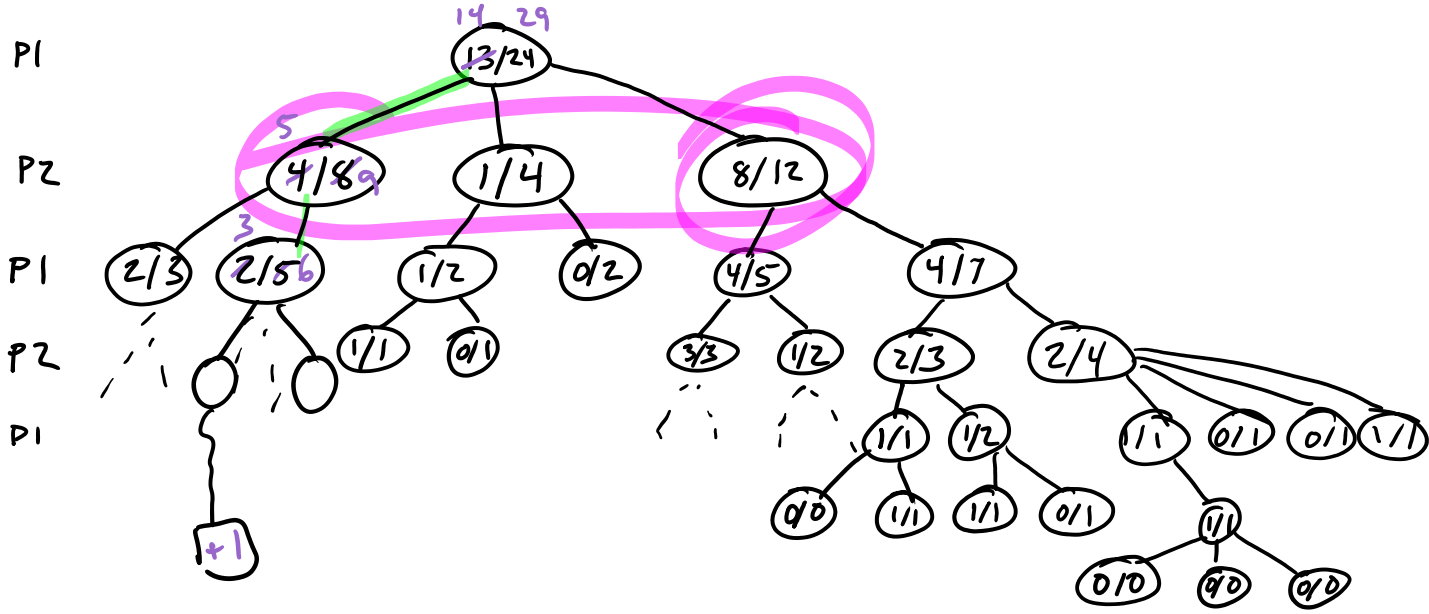
expand if leaf (non-terminal and non-zero visits) expandable, add its children

simulate play to terminal pos (from arbi. child if newly expanded)

can be random

update backprop result up tree from start of simulation ⇝ root

return move to child w/ best stats (highest mean reward or highest visit count)



UCT = MCTS

advantages: convergent

anytime

no domain knowledge

easily parallelized

disadvantages : no domain knowledge

some games not amenable

MCTS adapted for games that aren't trees