Game Evaluation

$$V_{eval}(s) = \begin{cases} Utility(s) & \text{if } isEnd(s) \\ \sum_{a \in Actions(s)} \pi_{agent}(s,a)\, V_{eval}(Succ(s,a)) & \text{if } Player(s) = agent \\ \sum \pi_{opp}(s,a) \cdot V_{eval}(Succ(s,a)) & \text{if } Player(s) = opp \end{cases}$$

Exptmax

$$V_{exptmax}(s) = \begin{cases} Utility(s) & \text{if } isEnd(s) \\ \max_{a \in Actions(s)} V_{exptmax}(Succ(s,a)) & \text{if } Player(s) = agent \\ \sum \pi_{opp}(s,a) \cdot V_{exptmax}(Succ(s,a)) & \text{if } Player(s) = opp \end{cases}$$

minimax

$$V_{minimax}(s) = \begin{cases} Utility(s) & \text{if } isEnd(s) \\ \max_{a \in Actions(s)} V_{minimax}(Succ(s,a)) & \text{if } Player(s) = agent \\ \min_{a \in Actions(s)} V_{minimax}(Succ(s,a)) & \text{if } Player(s) = opp \end{cases}$$

Policies

Deterministic policies: $\pi_p(s) \in \text{Actions}(s)$

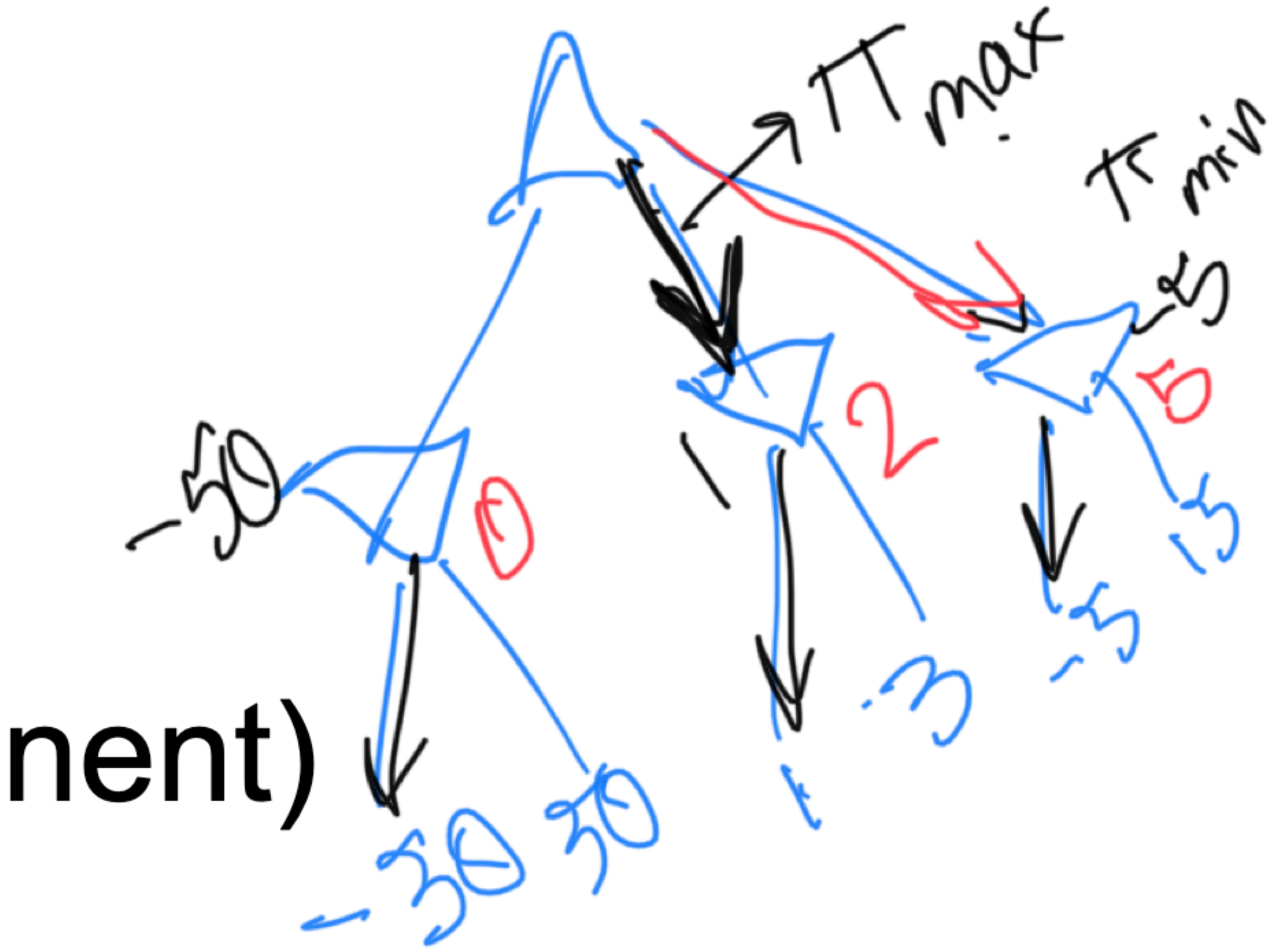
action that player p takes in state s

Stochastic policies $\pi_p(s, a) \in [0, 1]$: $\sum_{a \in \text{Actions}(s)} \pi_p(s, a) = 1$

probability of player p taking action a in state s

[semi-live solution: `humanPolicy`]

Face off



Recurrences produces policies:

$$V_{\text{exptmax}} \Rightarrow \pi_{\text{exptmax}(7)}, \pi_7 \text{ (some opponent)}$$

$$V_{\text{minmax}} \Rightarrow \pi_{\max}, \pi_{\min}$$

Play policies against each other:

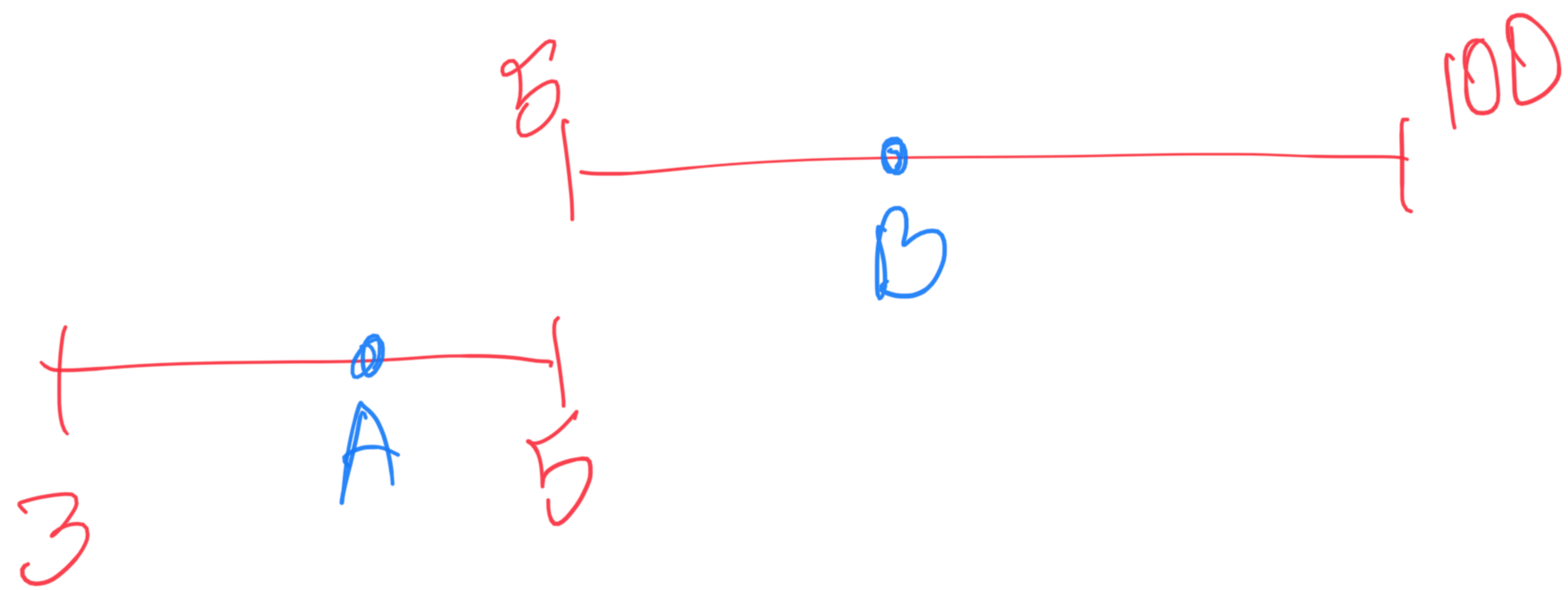
	$\pi_{\min}$	π_7
$\pi_{\max}$	$V(\pi_{\max}, \pi_{\min})$	$V(\pi_{\max}, \pi_7)$
$\pi_{\text{exptmax}(7)}$	$V(\pi_{\text{exptmax}(7)}, \pi_{\min})$	$V(\pi_{\text{exptmax}(7)}, \pi_7)$

What's the relationship between these values?

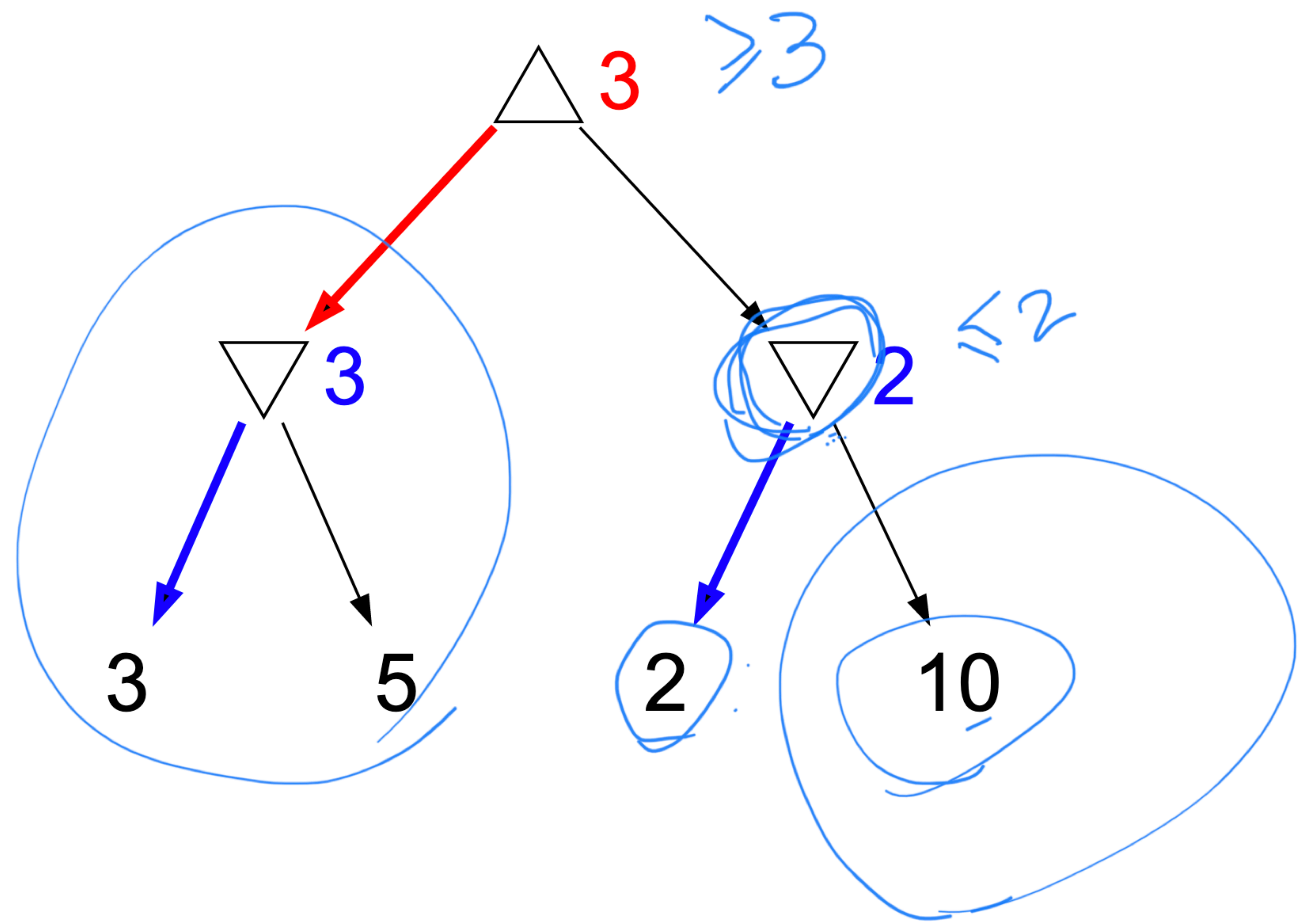
Pruning principle

Choose A or B with maximum value:

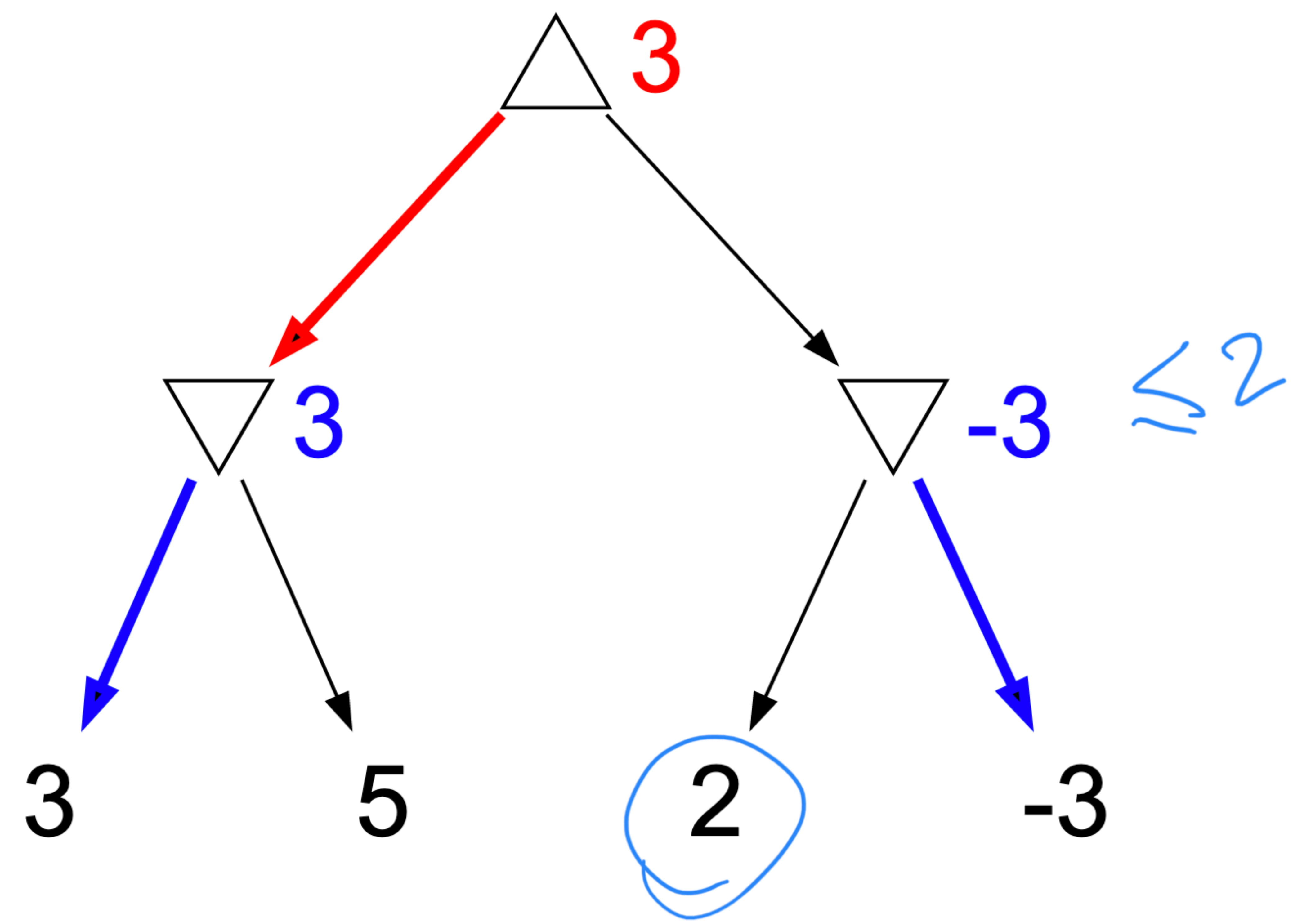
A: [3, **5**]　　　　　　B: [**5**, 100]



Pruning game trees



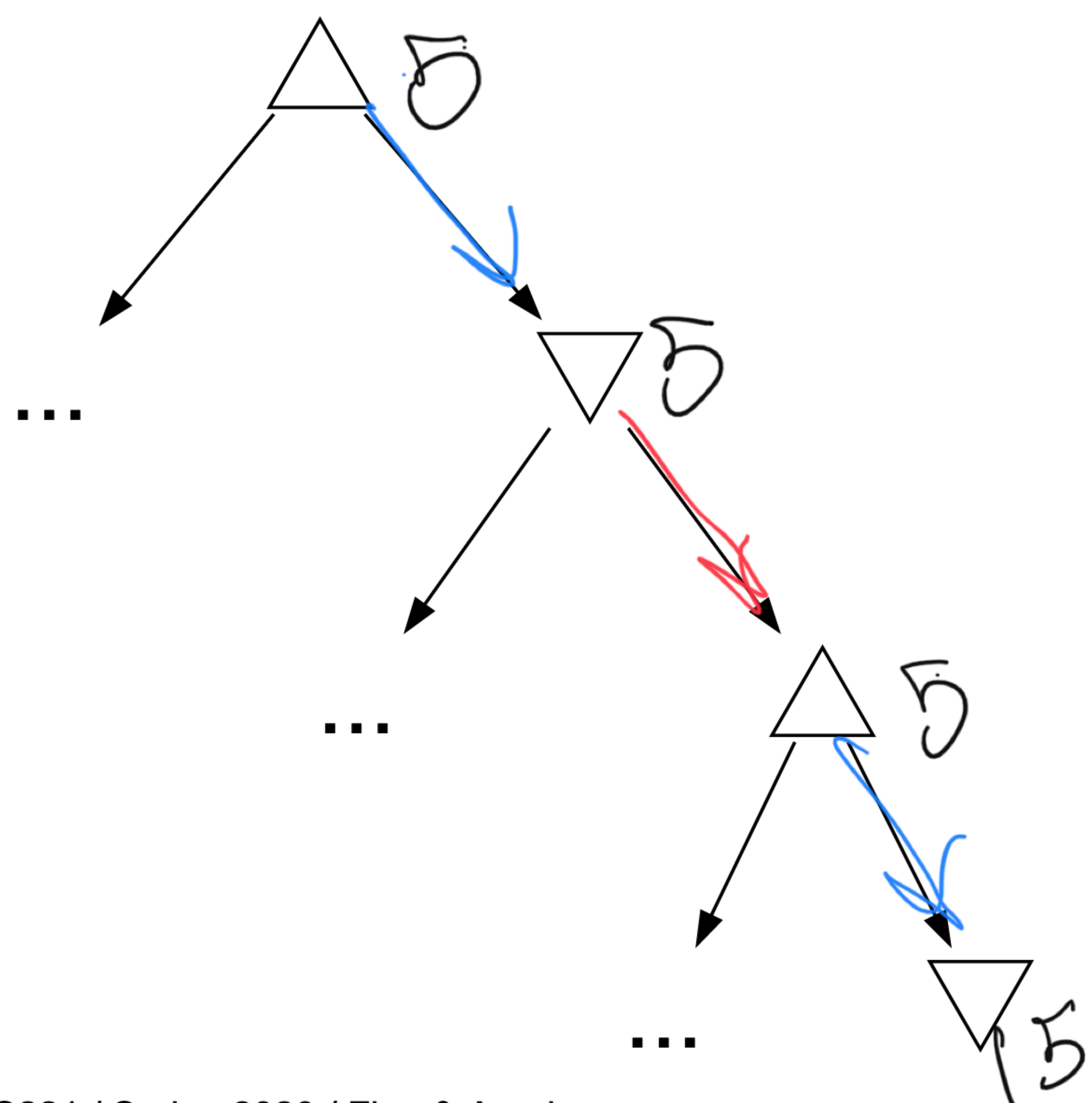
Pruning game trees



Alpha-beta pruning

Key idea: optimal path

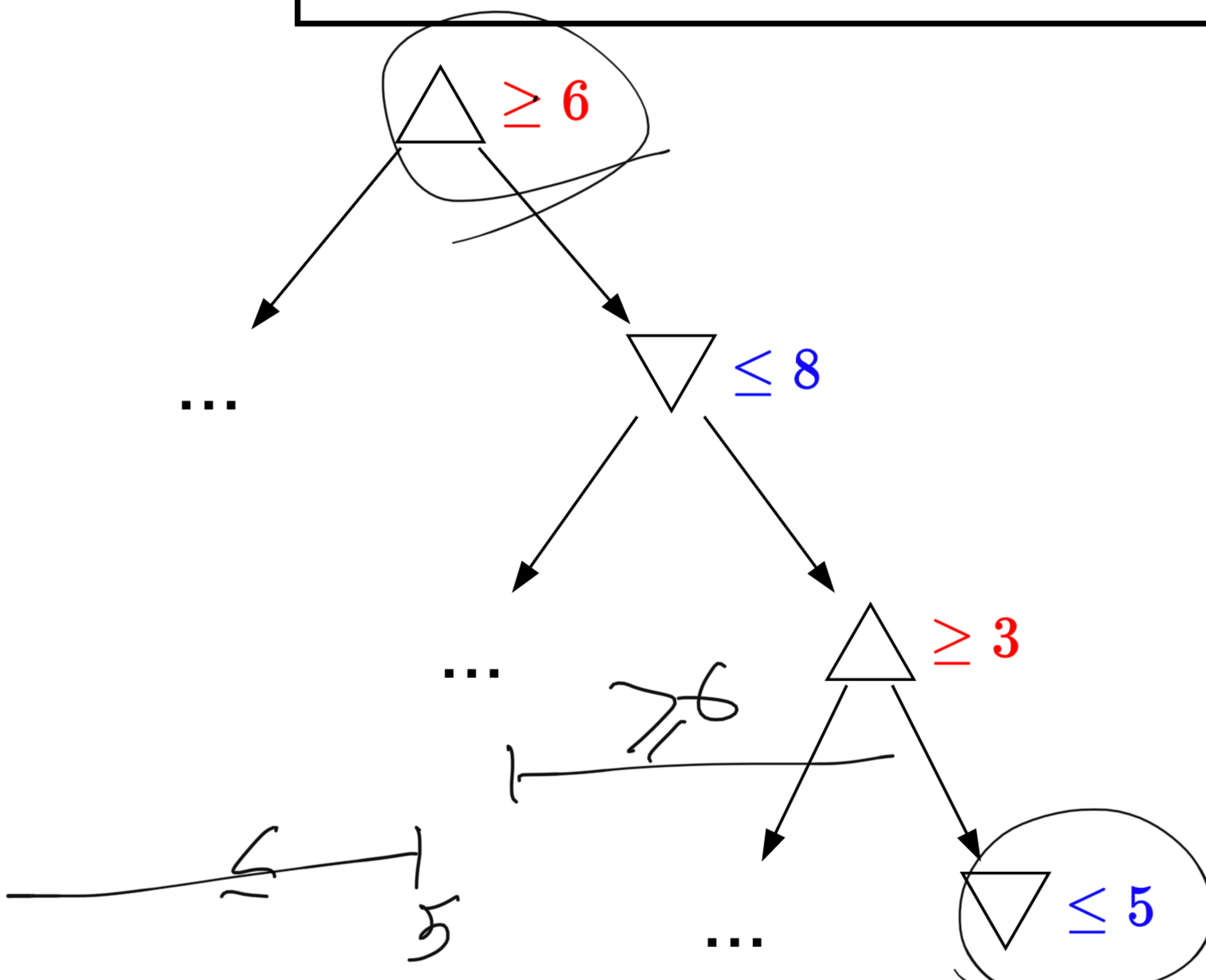
The optimal path is path that minimax policies take.
Values of all nodes on path are the same.



Alpha-beta pruning

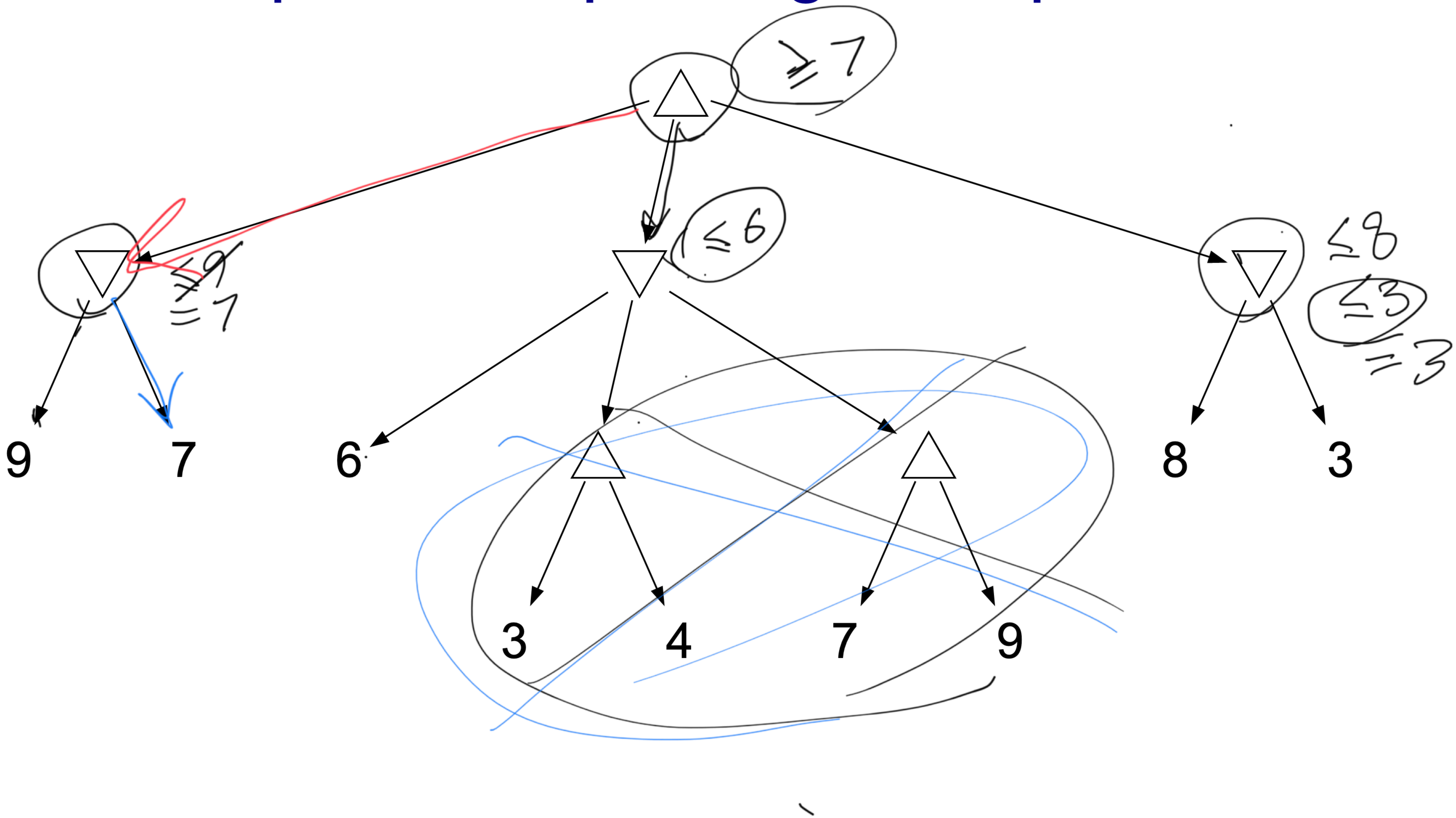
Key idea: optimal path

The optimal path is path that minimax policies take.
Values of all nodes on path are the same.

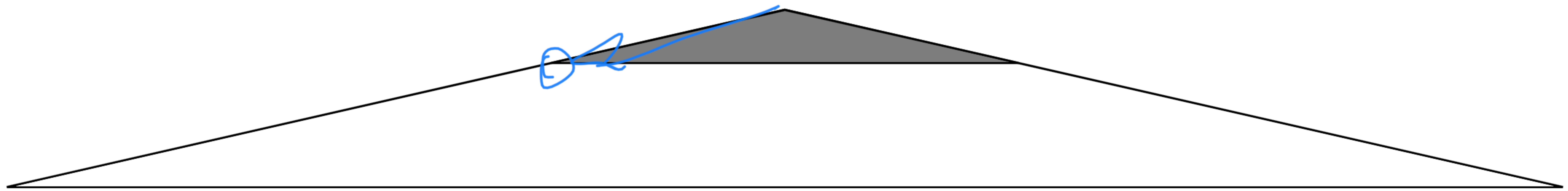


- a_s: lower bound on value of max node s
- b_s: upper bound on value of min node s
- Prune a node if its interval doesn't have non-trivial overlap with every ancestor (store $\alpha_s = \max_{s' \preceq s} a_s$ and $\beta_s = \min_{s' \preceq s} b_s$)

Alpha-beta pruning example



Depth-limited search



Limited depth tree search (stop at maximum depth $d_{\max}$):

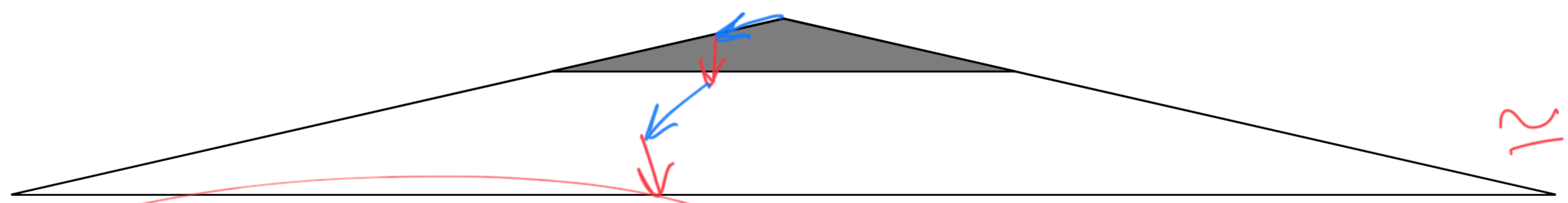
$$V_{\text{minmax}}(s, d) = \begin{cases} \text{Utility}(s) & \text{IsEnd}(s) \\ \text{Eval}(s) = V_{\text{minimax}}(s) & d = 0 \\ \max_{a \in \text{Actions}(s)} V_{\text{minmax}}(\text{Succ}(s, a), d) & \text{Player}(s) = \text{agent} \\ \min_{a \in \text{Actions}(s)} V_{\text{minmax}}(\text{Succ}(s, a), d - 1) & \text{Player}(s) = \text{opp} \end{cases}$$

Use: at state s, call $V_{\text{minmax}}(s, d_{\max})$

Convention: decrement depth at last player's turn

Depth-limited search

Game Eval $(\pi_{\text{approx}}, \pi_{\text{simulator}}) \simeq V_{\text{minmax}}(s,d)$



Limited depth tree search (stop at maximum depth $d_{\max}$):

$$V_{\text{minmax}}(s, d) = \begin{cases} \text{Utility}(s) & \text{IsEnd}(s) \\ \text{Eval}(s) & d = 0 \\ \max_{a \in \text{Actions}(s)} V_{\text{minmax}}(\text{Succ}(s,a), d) & \text{Player}(s) = \text{agent} \\ \min_{a \in \text{Actions}(s)} V_{\text{minmax}}(\text{Succ}(s,a), d-1) & \text{Player}(s) = \text{opp} \end{cases}$$

Use: at state s, call $V_{\text{minmax}}(s, d_{\max})$

Convention: decrement depth at last player's turn