

Scout

Alpha-beta wants to know is next child better than best so far if ^{updates as better} ~~child~~ ^{and show much} and asks 1) is next child better and if so, by how much (α, β)
result $> \alpha \rightarrow$ answer is YES, and result is value

Scout wants to know is next child better than best so far if so ^{null} ~~by how much~~ and asks 1) is the next child better (is value $\geq \alpha$) $(\alpha, \alpha+1)$
result $> \alpha \rightarrow$ answer is YES
(but don't know how much better)
result $\leq \alpha \rightarrow$ answer is NO
2) OK, how much better (result, β)

(principal variation search)
Scout ($p, \alpha, \beta, h, \text{depth}$)

if p is terminal then return value(p)

if $\text{depth} = 0$ then return $h(p)$

if p is a max position

for each reachable position p' ^{in rough order from best to worst} and while $\alpha < \beta$

if p' is first child

score \leftarrow Scout($p', \alpha, \beta, h, \text{depth}-1$)

else

score \leftarrow AB($p', \alpha, \alpha+1, h, \text{depth}-1$)

if $\alpha < \text{score} < \beta$

score \leftarrow Scout(score, score, $\beta, h, \text{depth}-1$)

$\alpha \leftarrow \max(\alpha, \text{score})$

return α

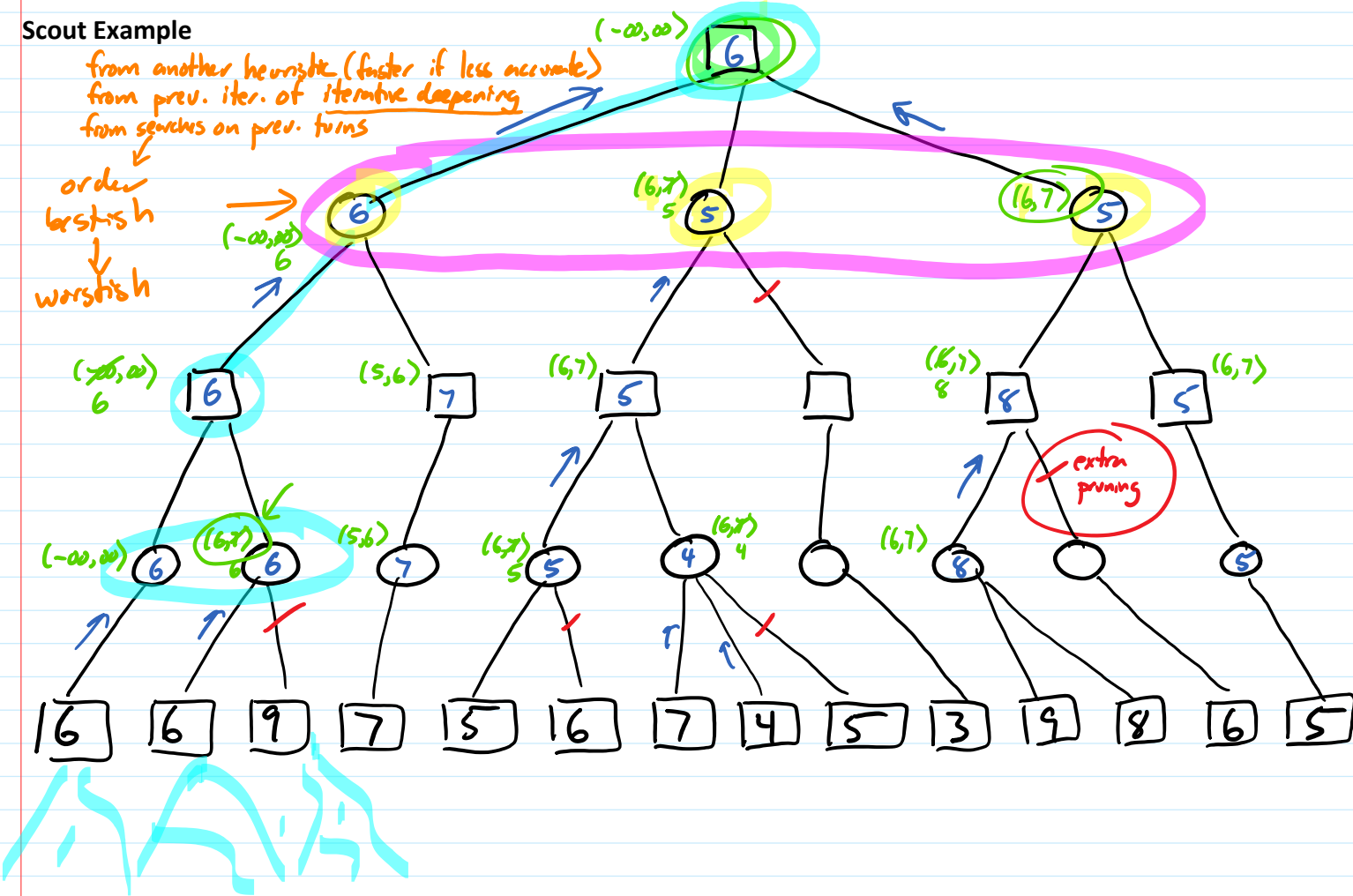
else

⋮

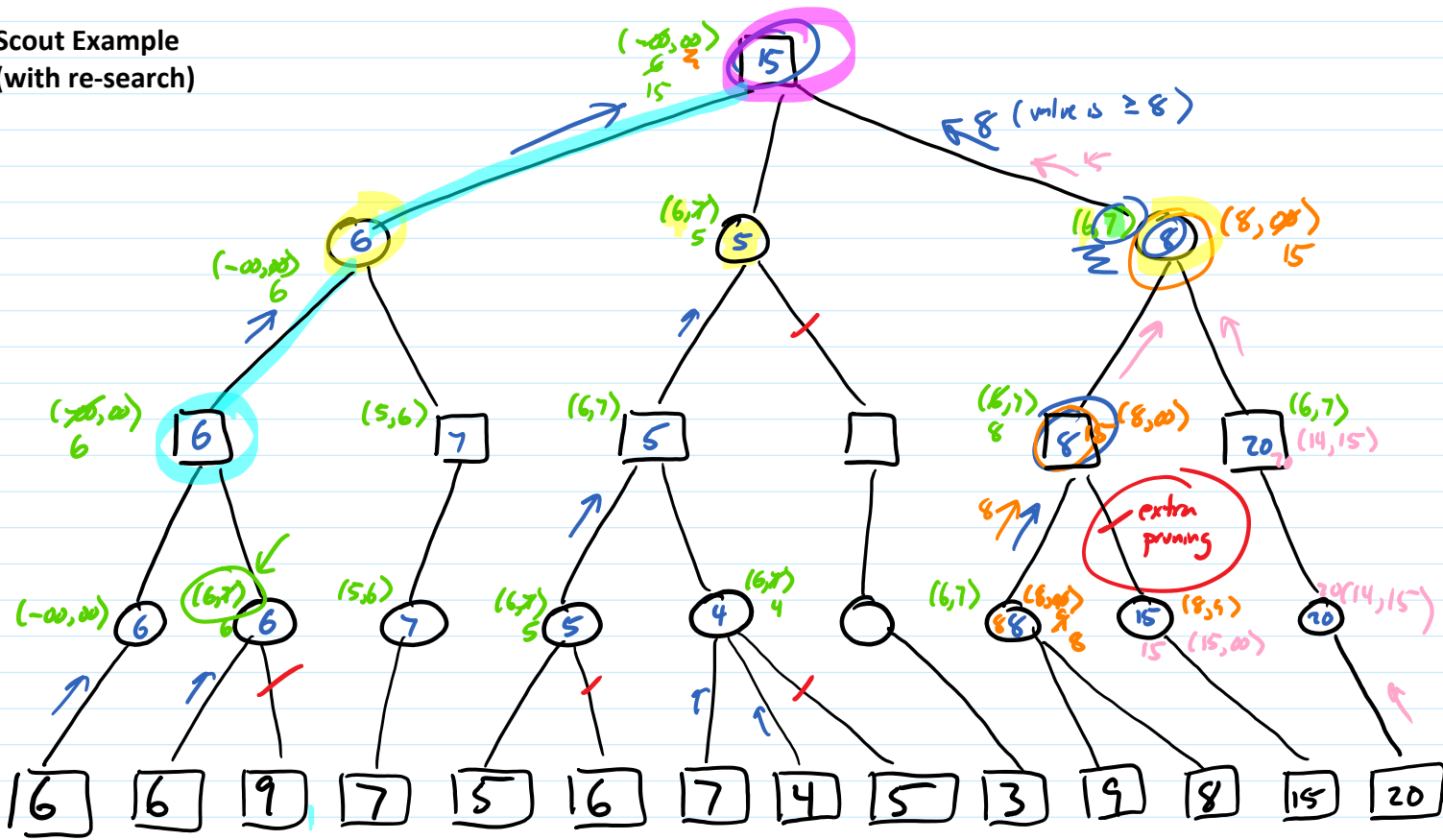
Scout Example

from another heuristic (faster if less accurate)
from prev. iter. of iterative deepening
from searches on prev. turns

order
bestish
↓
worstish



Scout Example
(with re-search)



MTD-f

integer-valued

node to evaluate

depth bound
heuristic

first guess

the better, the fewer iterations
correct → only 2 iterations of while
get guess from results of prev. searches
(prev. turn or lower depth)

Memory-enhanced
test driver

MTD-f (n, f, d, h)

lowerBound ← $-\infty$

upperBound ← ∞

$g \leftarrow f$

current guess
of value of
 n

while lowerBound < upperBound

$\beta \leftarrow \max(\text{lowerBound} + 1, g)$

$g \leftarrow A\text{-}B\text{-with-transposition-table}(n, \beta - 1, \beta, d, h)$

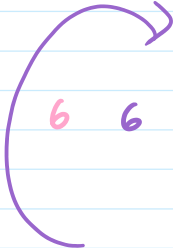
if $g < \beta$ then upperBound ← g

else then lowerBound ← g

return g

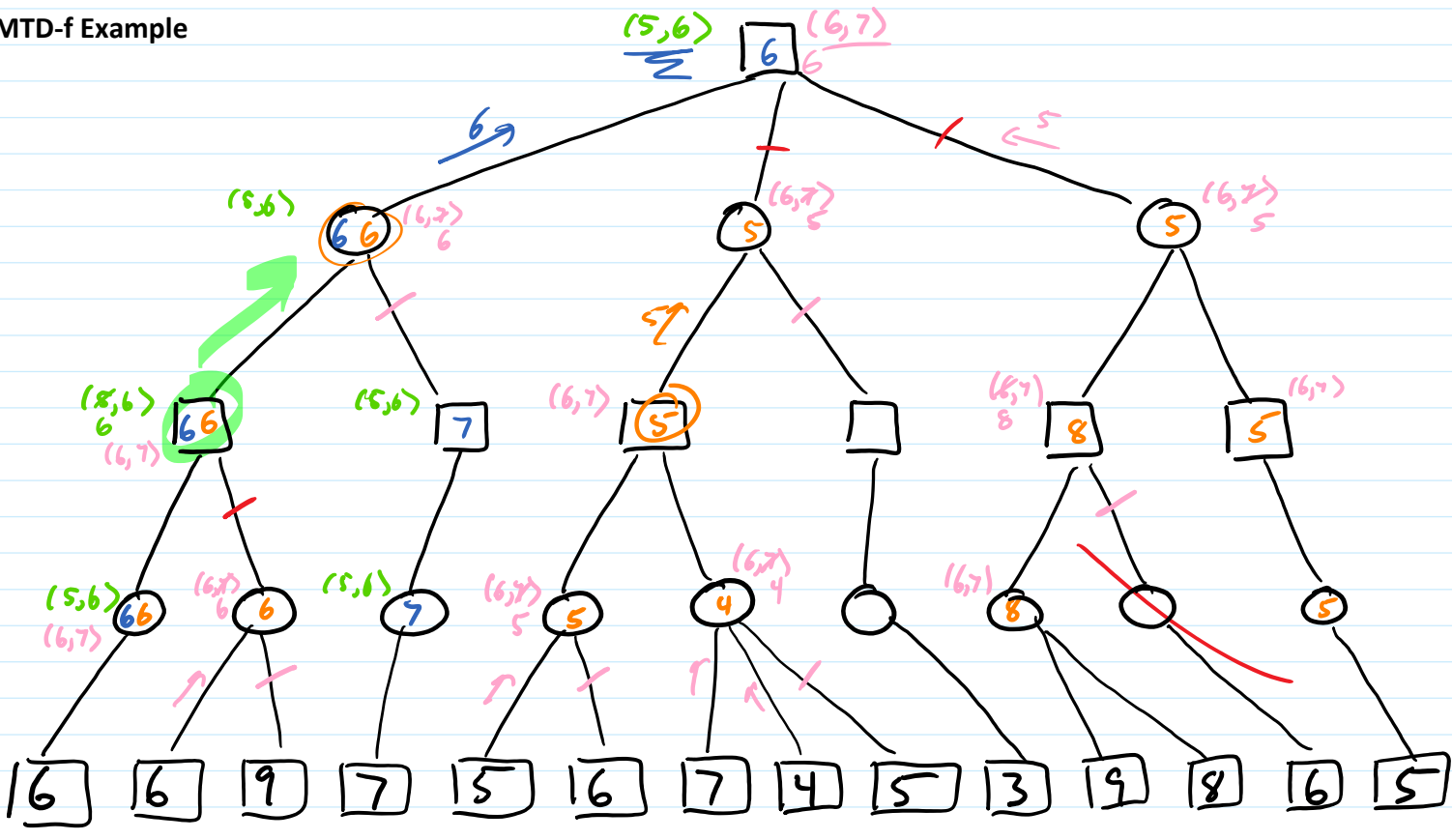
lb
(
g
s s+1
g+g
up
)

] (> {



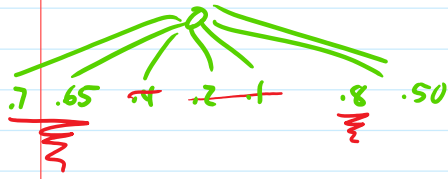
6 6

MTD-f Example



Monte Carlo Techniques

Flat Monte Carlo: for each action simulate to terminal using random play
choose action with highest observed average



worked well in Scrabble, Bridge

Combine with UCB:

Combine with tree search:
(Flat UCB)

Grow Tree asymmetrically: Monte Carlo Tree Search