Part IV

Two player games, Minimax and Alpha-beta pruning

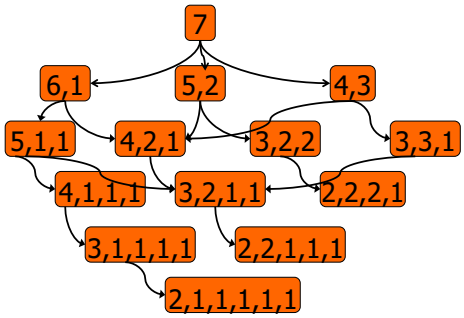
The Game of Jianshizi (Nim)

- There are two opponents
- A number of stones are placed on a table
- At each move the player must divide a pile of stones into two non-empty piles of different sizes
- Nim is a so-called misère (povery) game in that the first person who cannot move looses

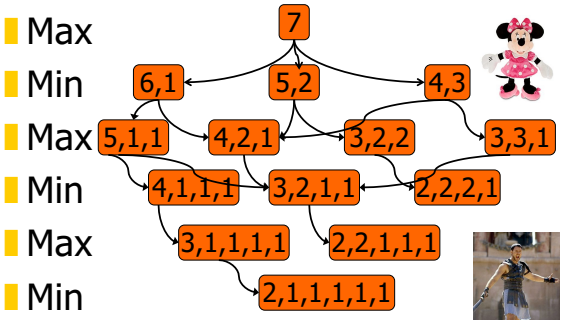
Example game play for 8 stones

- 1 ● ● ● ● ● ● ● ●
- 2 ● ● ● ● ● ● ● ●
- 1 ● ● ● ● ● ● ● ●
- 2 ● ● ● ● ● ● ● ●
- 1 ● ● ● ● ● ● ● ●
- Player 2 then wins

Every game played with 7 stones



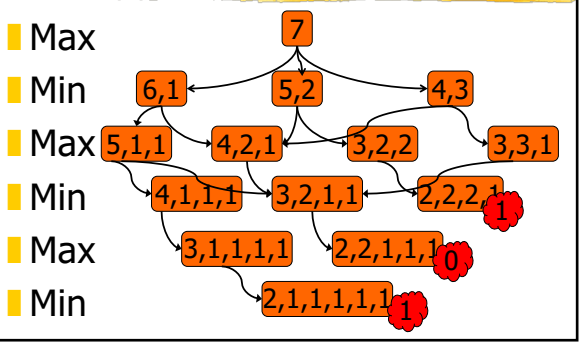
Max = Maximus (player) vs Min = Minny (opponent)



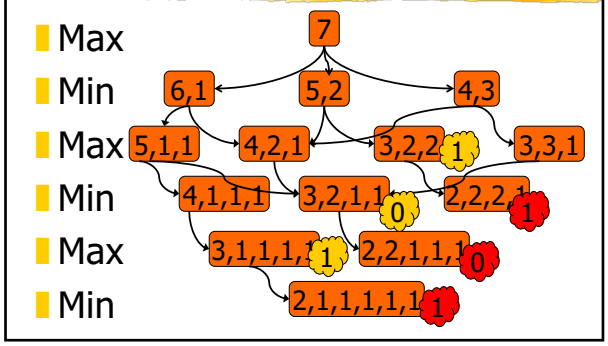
Minimax [von Neumann, 1928] for reasoning about who wins

- Max wants to maximise Max's score
- Min wants to minimise Max' s score
- Decorate the leaf nodes with a value that indicates who wins:
  - ie. 1 if Max wins
  - ie. 0 if Max loses
- Decorate the nodes of the tree by propagating values upwards

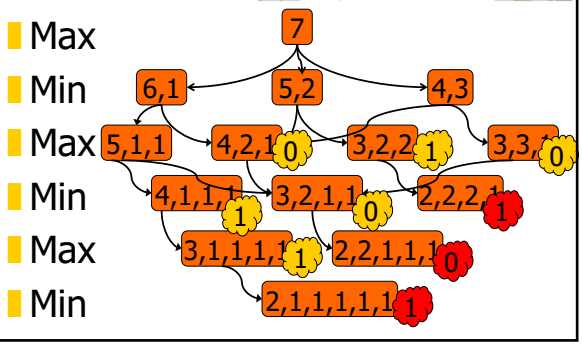
Decorating terminal nodes



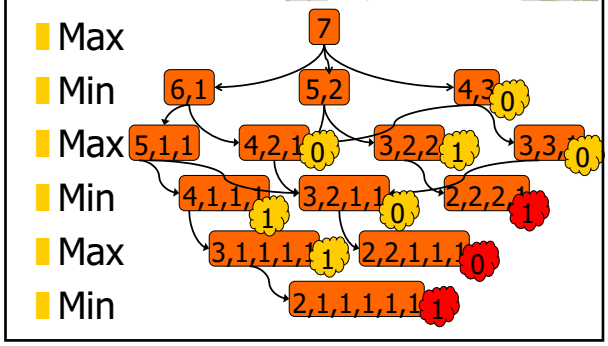
Decorate nodes with just one successor



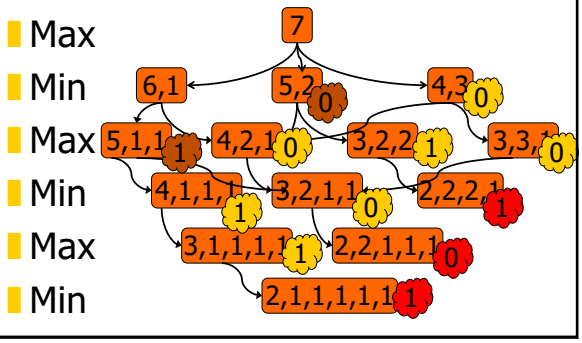
Decorate nodes with just one successor



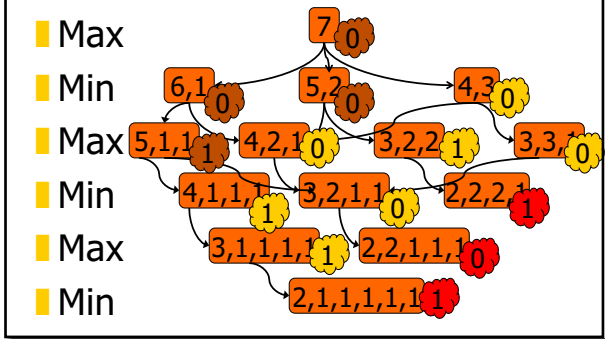
Decorate nodes with just one successor



Can a player win *if* they make good choices? (Minimax)



Can a player win *if* they make good choices? (Minimax)



Reviewing the complete search tree

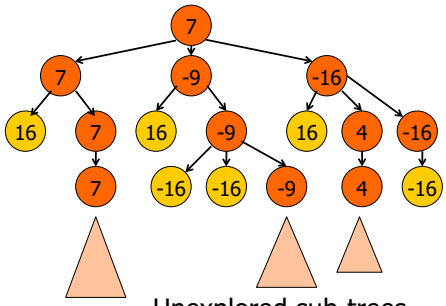
- What are Max's prospects for winning?
- What move should Min do if she has piles of 5, 2 stones?
- Suppose Min makes the wrong move into 5, 1, 1 stones. What should Max do next?
- The decoration represents the best that the player (in that state) can achieve

Other games

- Theoretically perfect: 0-1 games where one player has as much information as their opponent (Go, Othello)
- Theoretically imperfect: poker
- Practically imperfect: cannot see to game end
- At a Max ply in Chess:
  - ie. if Max checkmates Min, then value of +32
  - ie. if Min checkmates Max, value of -32
  - ie. else heuristic strength of -32 ≤ value ≤ 32

Heuristic Minimax for Chess with 4-ply look ahead

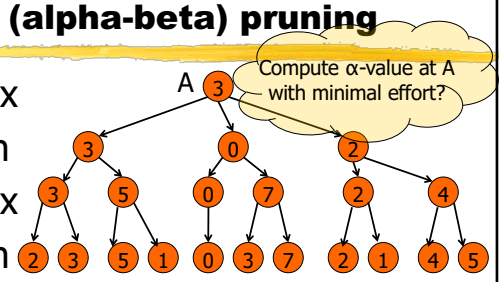
- Max
- Min
- Max
- Min



Unexplored sub-trees

α–β (alpha-beta) pruning

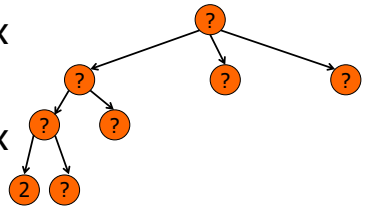
- Max
- Min
- Max
- Min



- α-values in Max ply never decrease when another child is known
- β-values in Min ply never increase

α–β pruning (step 1)

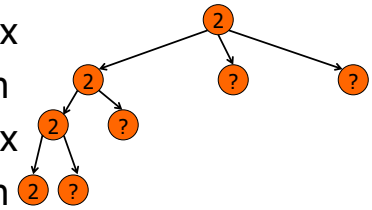
- Max
- Min
- Max
- Min



Descend to depth of 4-ply and calculate heuristic of 2, say, for leaf

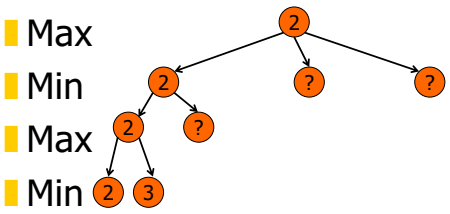
α–β pruning (step 2)

- Max
- Min
- Max
- Min



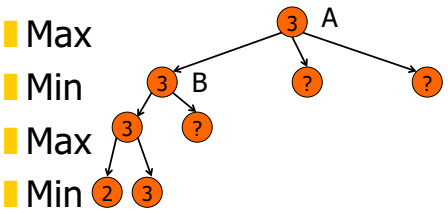
Propagate heuristic value up to root

α–β pruning (step 3)



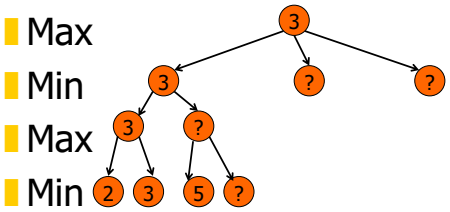
Descend and calculate next heuristic value, say, 3

α–β pruning (step 4)



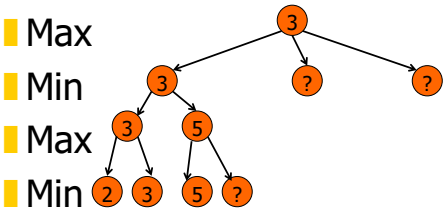
Propagate 3 up to root
Update A and B since they only have one evaluated successor

α–β pruning (step 5)



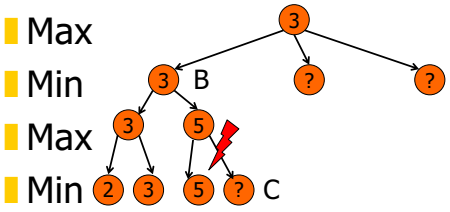
Descend to depth of 4-ply and assign heuristic of 5, say, to next leaf

α–β pruning (step 6)



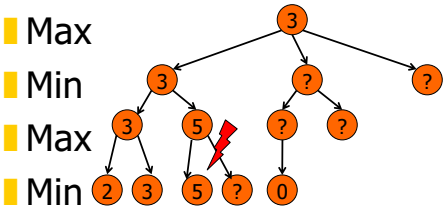
Propagate 5 back up to root

α–β pruning (step 7)



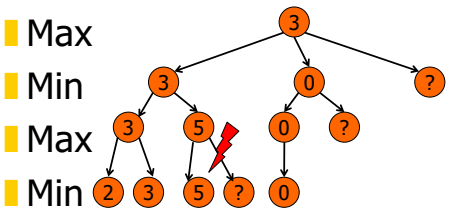
No need to compute heuristic for C since this cannot effect B

α–β pruning (step 8)



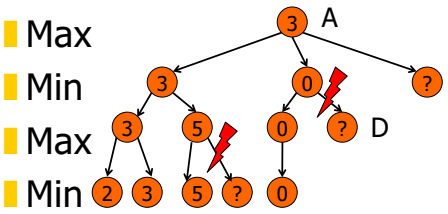
Descend to depth of 4-ply and assign heuristic of 0, say, to next leaf

α–β pruning (step 9)



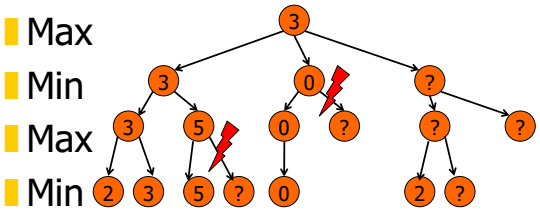
Propagate 0 up the tree

α–β pruning (step 10)



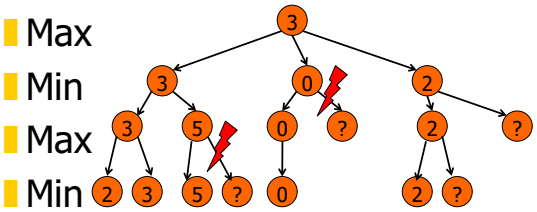
No need to compute heuristic for D
since this cannot effect A

α–β pruning (step 11)



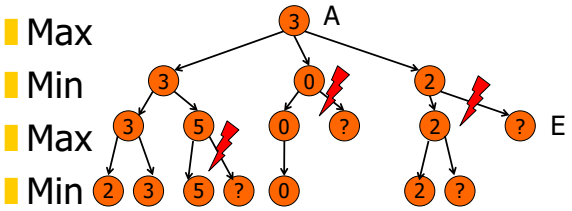
Descend to depth of 4-ply and assign
heuristic of 2, say, to next leaf

α–β pruning (step 12)



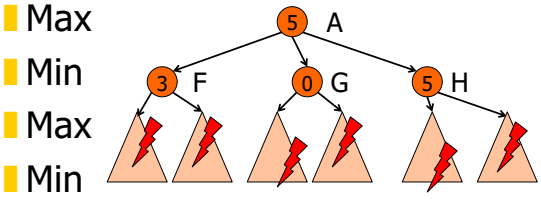
Propagate 2 up the tree

α–β pruning (step 13)



No need to compute heuristic for E
since this cannot effect A

How can Max make use of α–β pruning when at A?



Find β-values at F, G and H
Move to H as best chance of winning

Summary

- Focus is on two player games
- Use min or max to decorate alternate plys
- Use heuristics when trees are deep
- Do not need to decorate every sub-trees