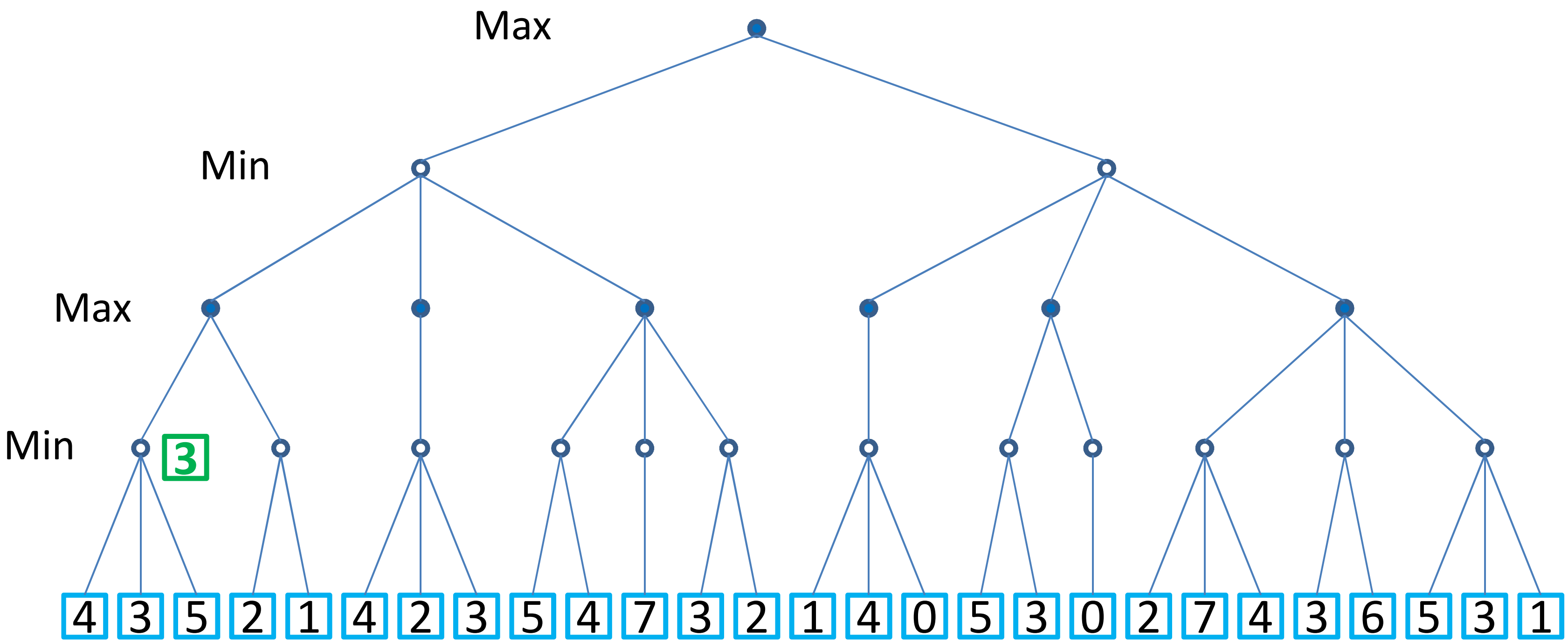
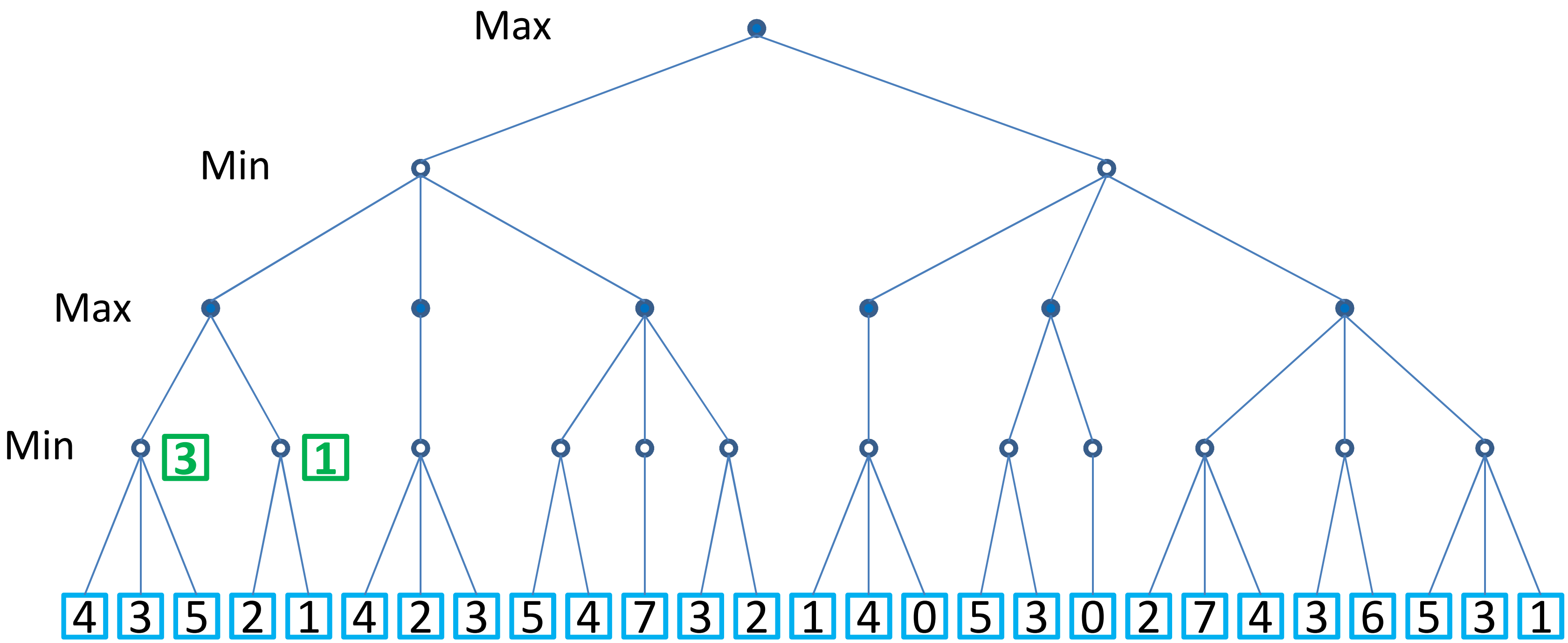
MiniMax & Constraint Processing: MiniMax Algorithm

MINIMAX WITHOUT $\alpha\beta$-PRUNING

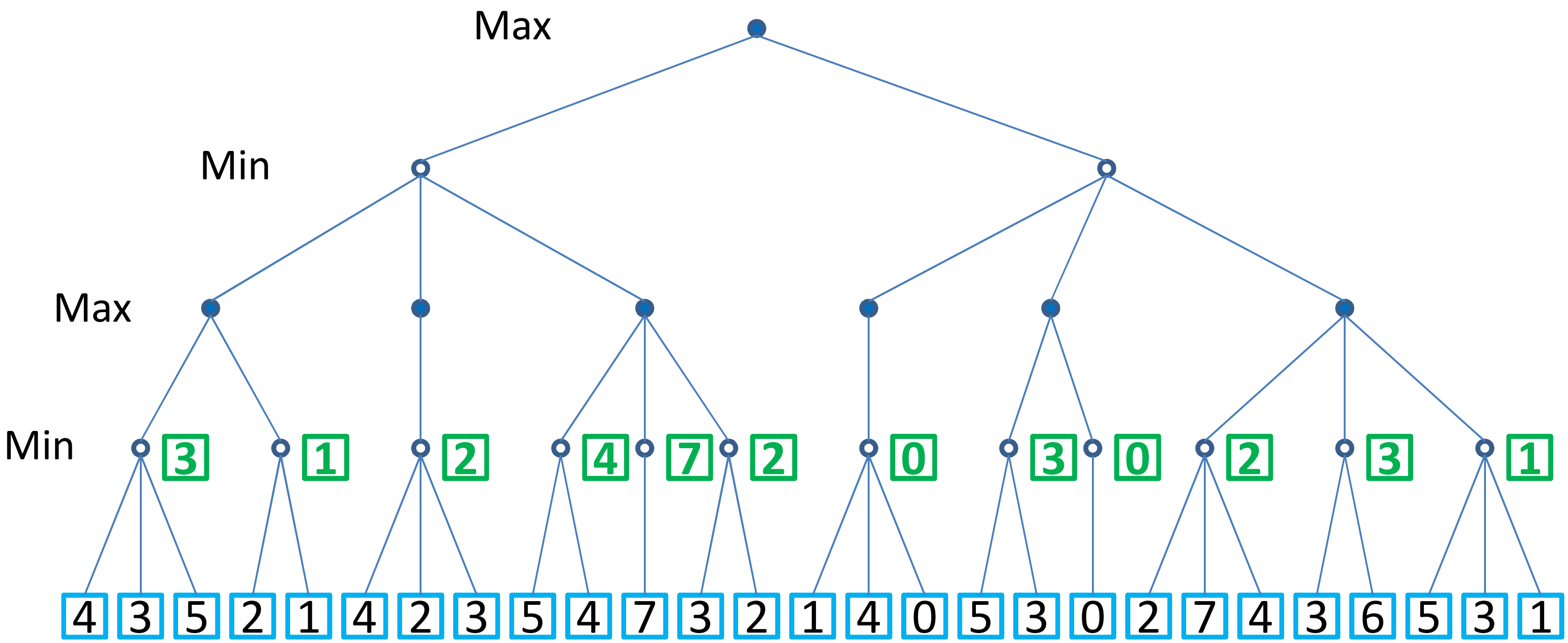
MiniMax without $\alpha\beta$-pruning



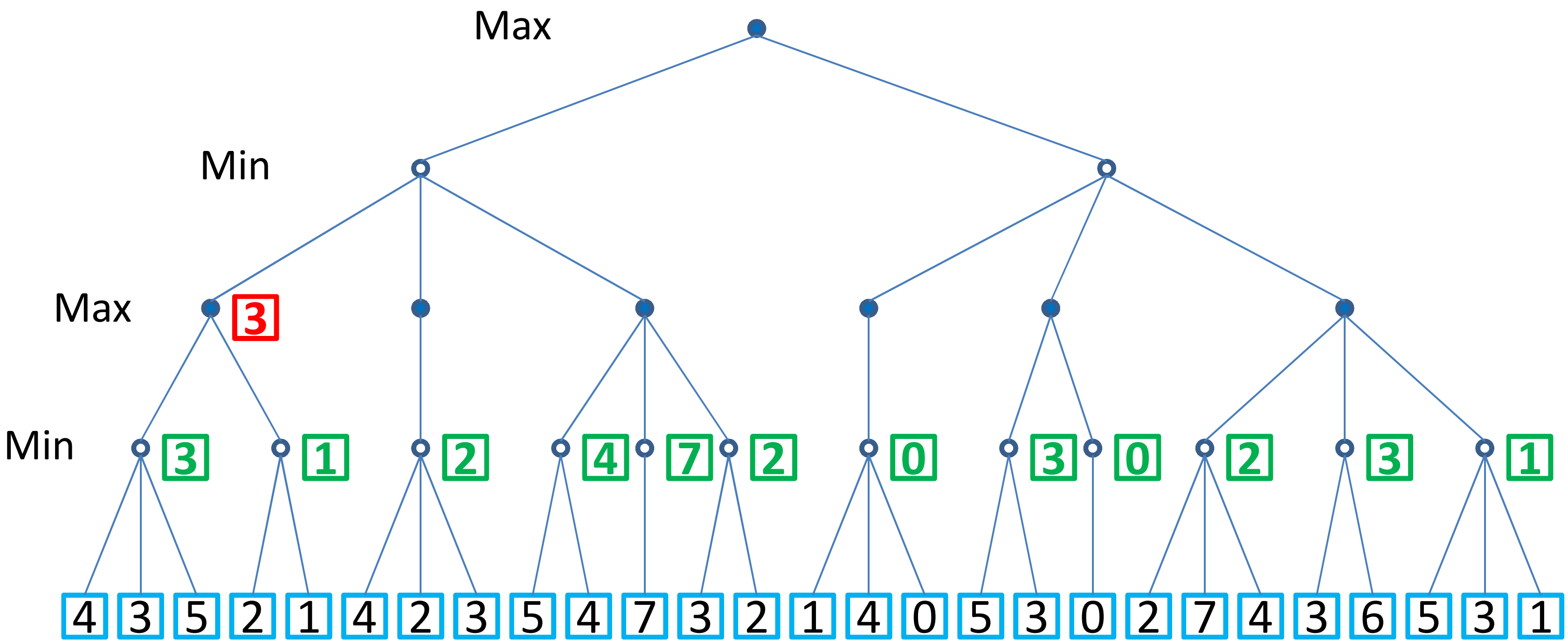
MiniMax without $\alpha\beta$-pruning



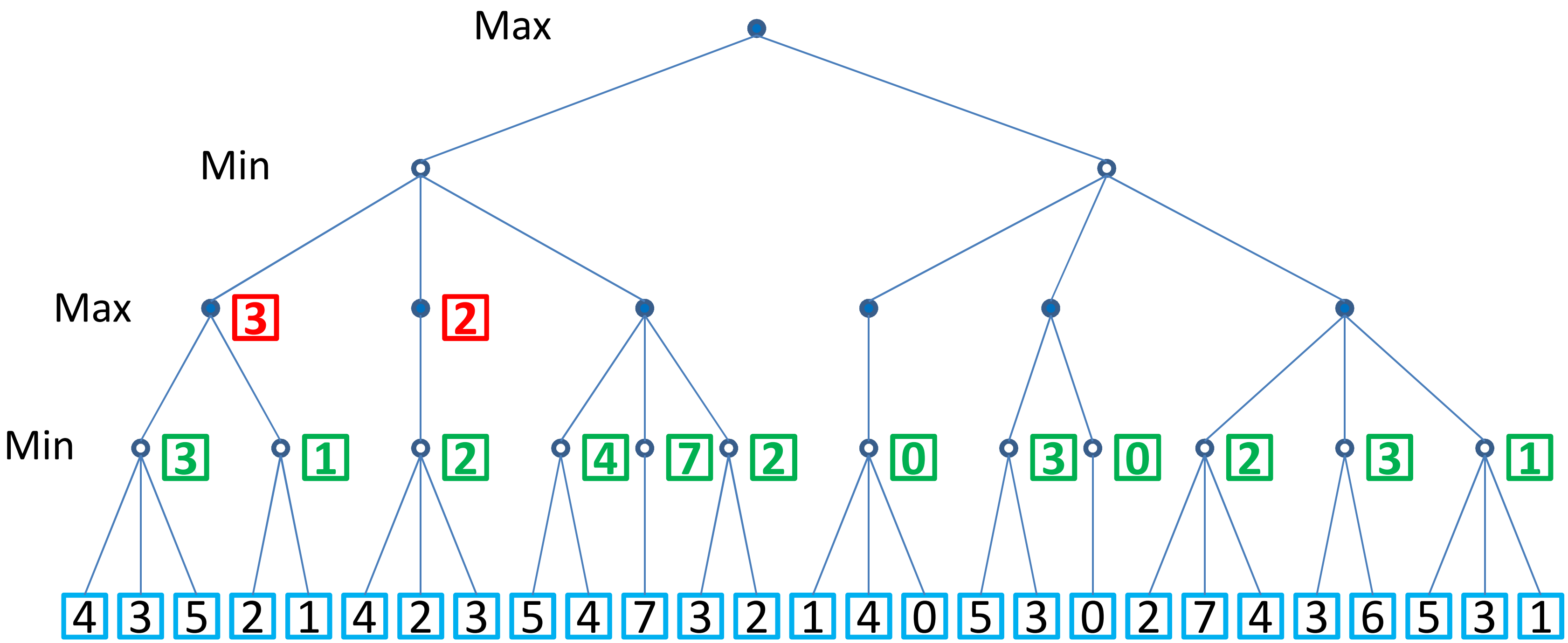
MiniMax without $\alpha\beta$-pruning



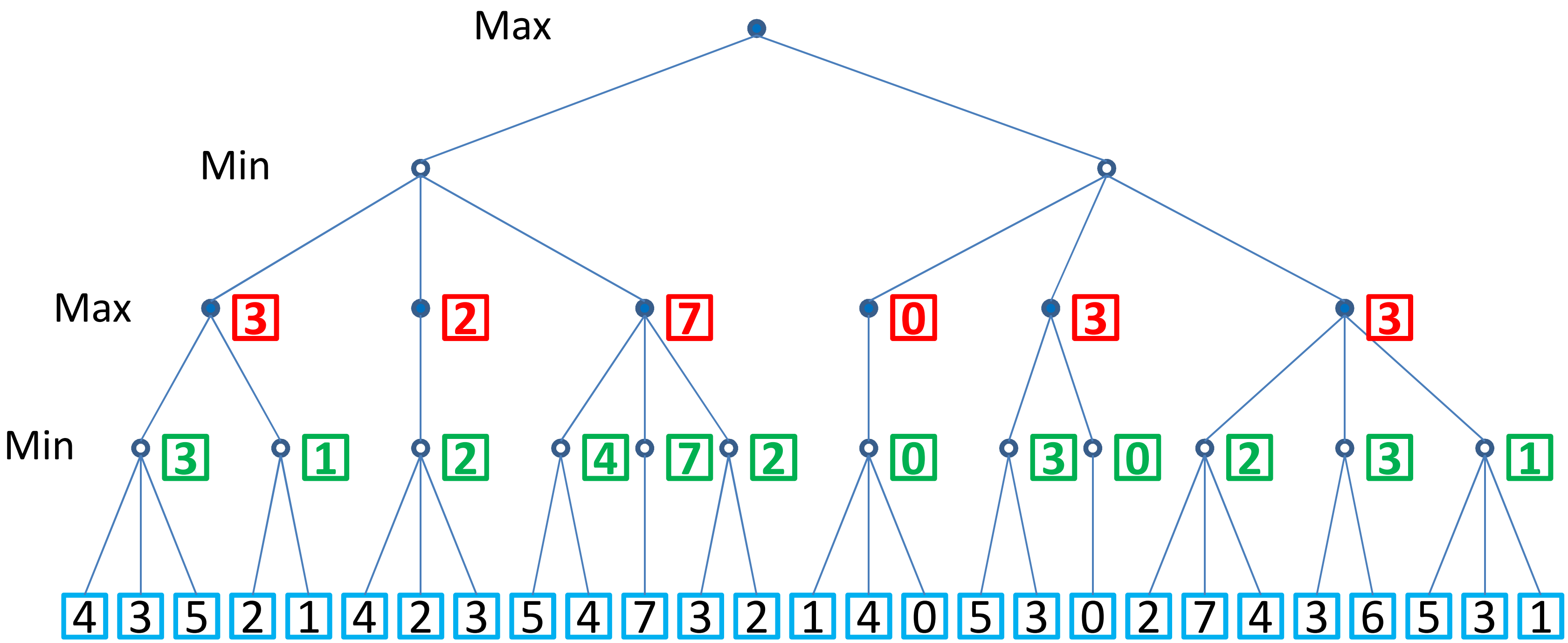
MiniMax without $\alpha\beta$-pruning



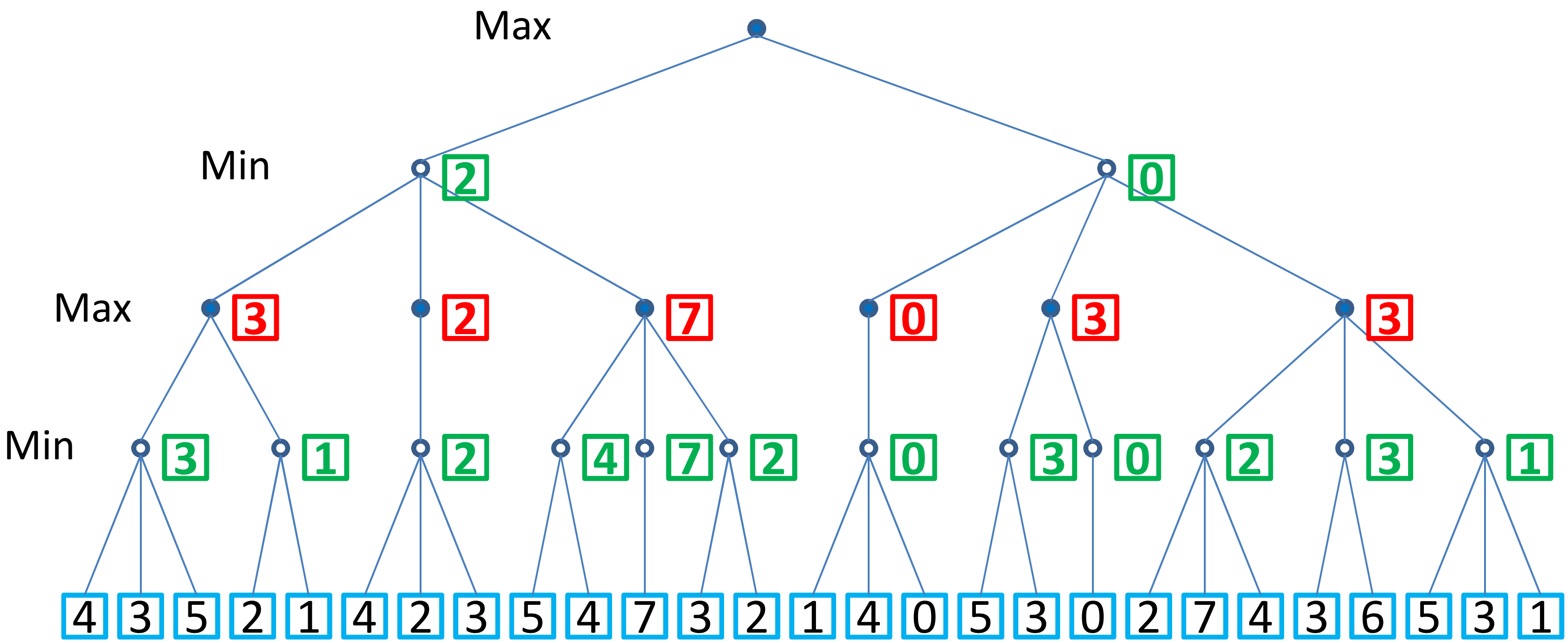
MiniMax without $\alpha\beta$-pruning



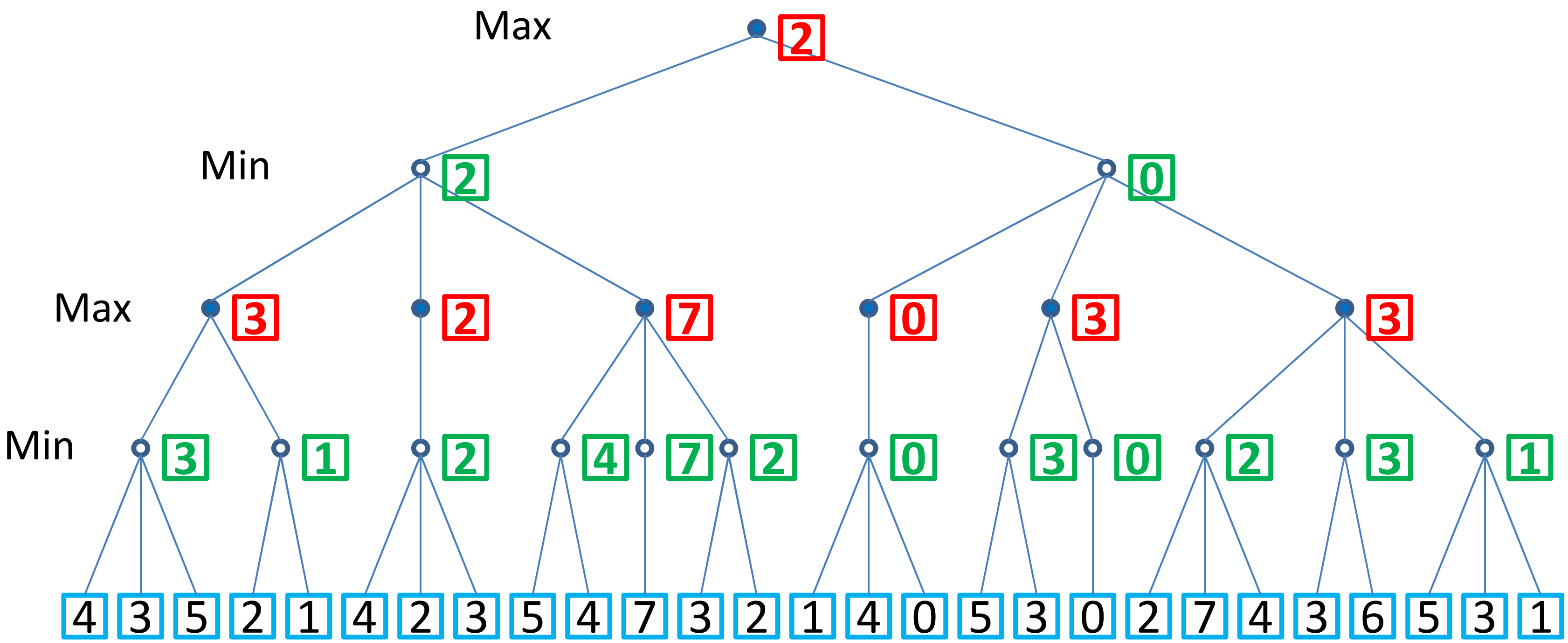
MiniMax without $\alpha\beta$-pruning



MiniMax without $\alpha\beta$-pruning



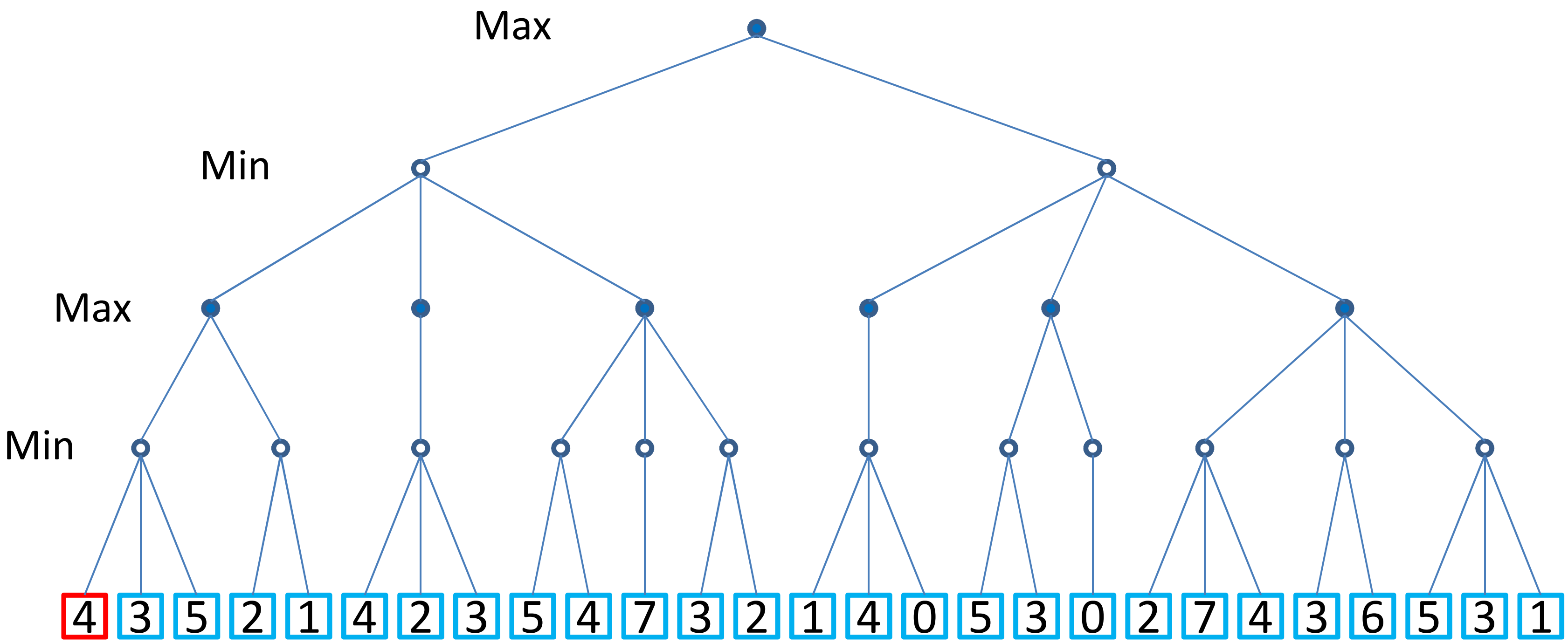
MiniMax without $\alpha\beta$-pruning



MiniMax & Constraint Processing: MiniMax Algorithm

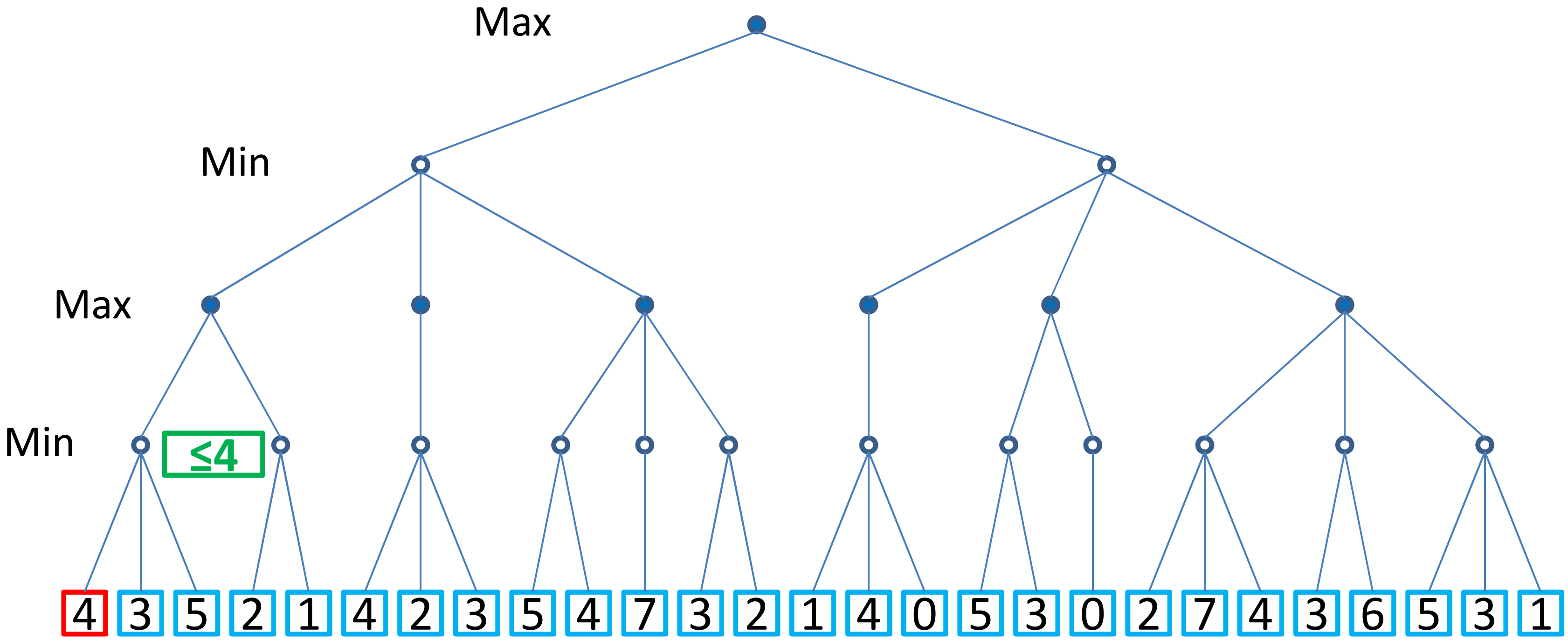
MINIMAX WITH $\alpha\beta$-PRUNING

MiniMax with $\alpha\beta$-pruning

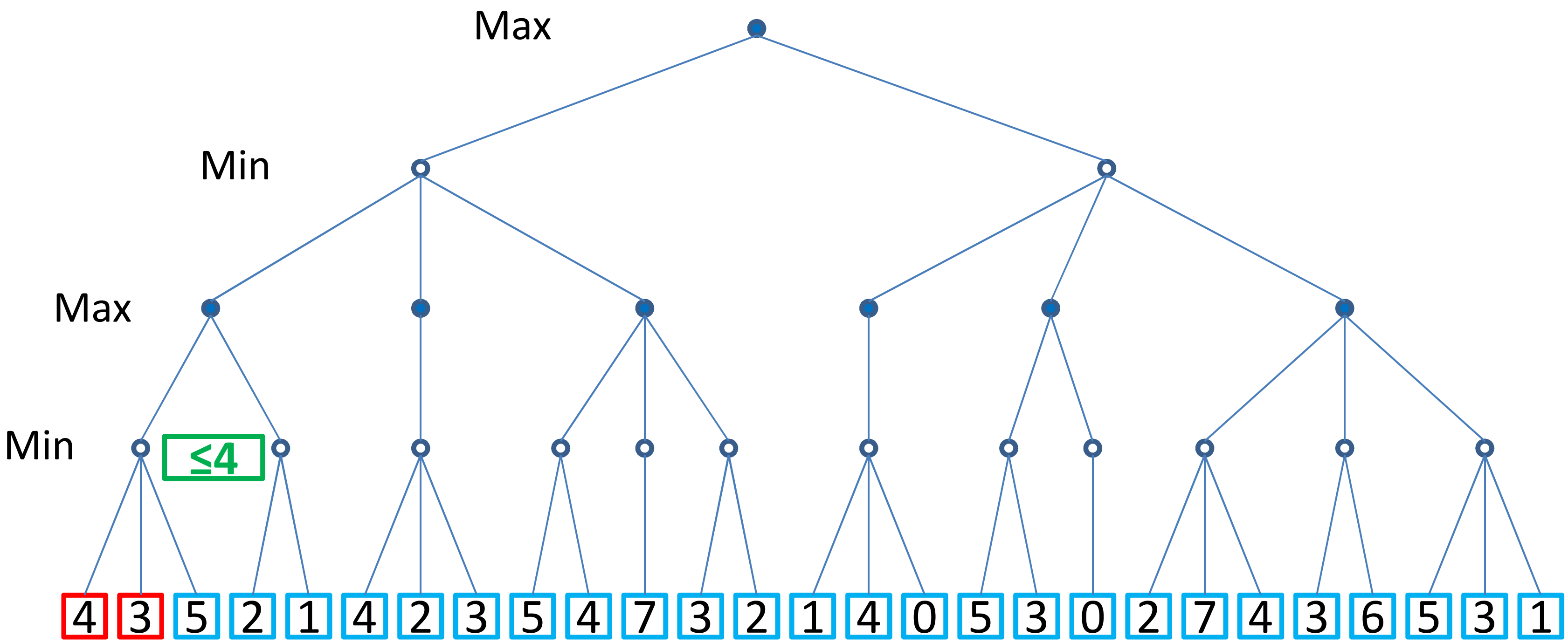


MiniMax with $\alpha\beta$-pruning

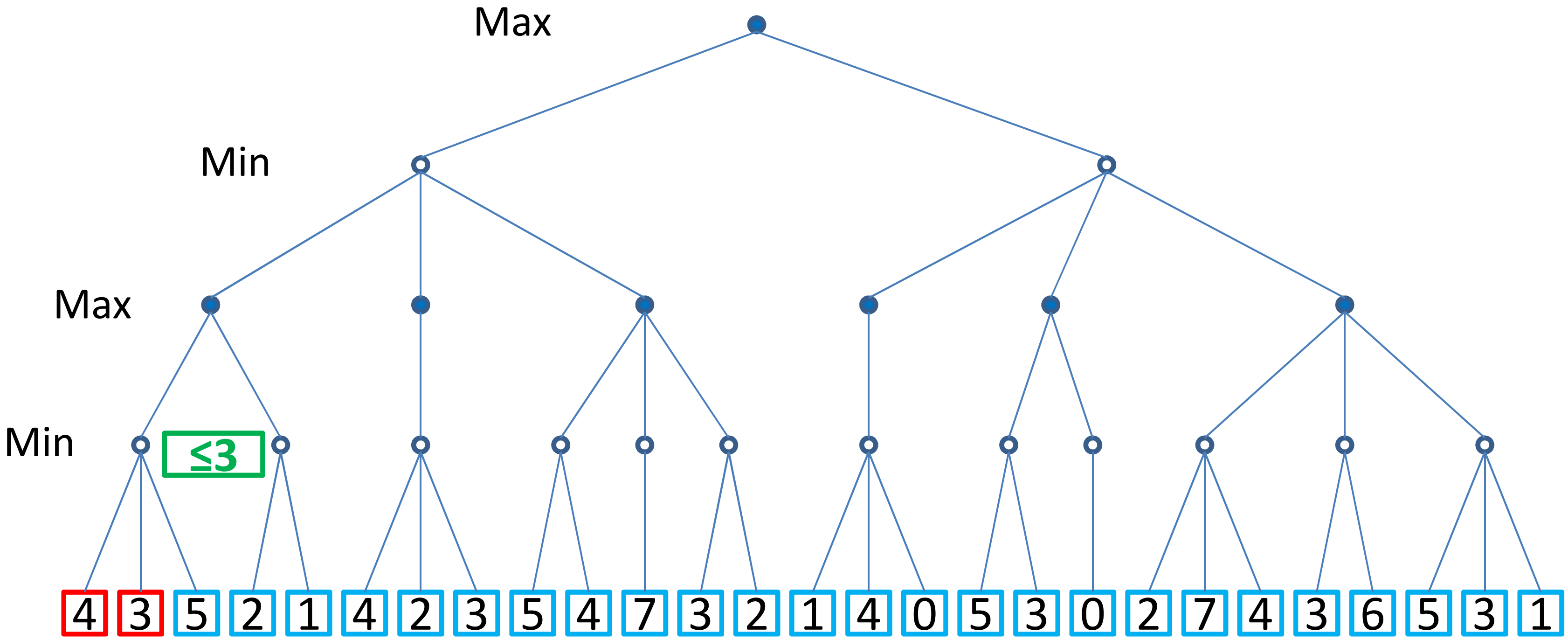
- **α-nodes**: Temporary values at MIN-nodes



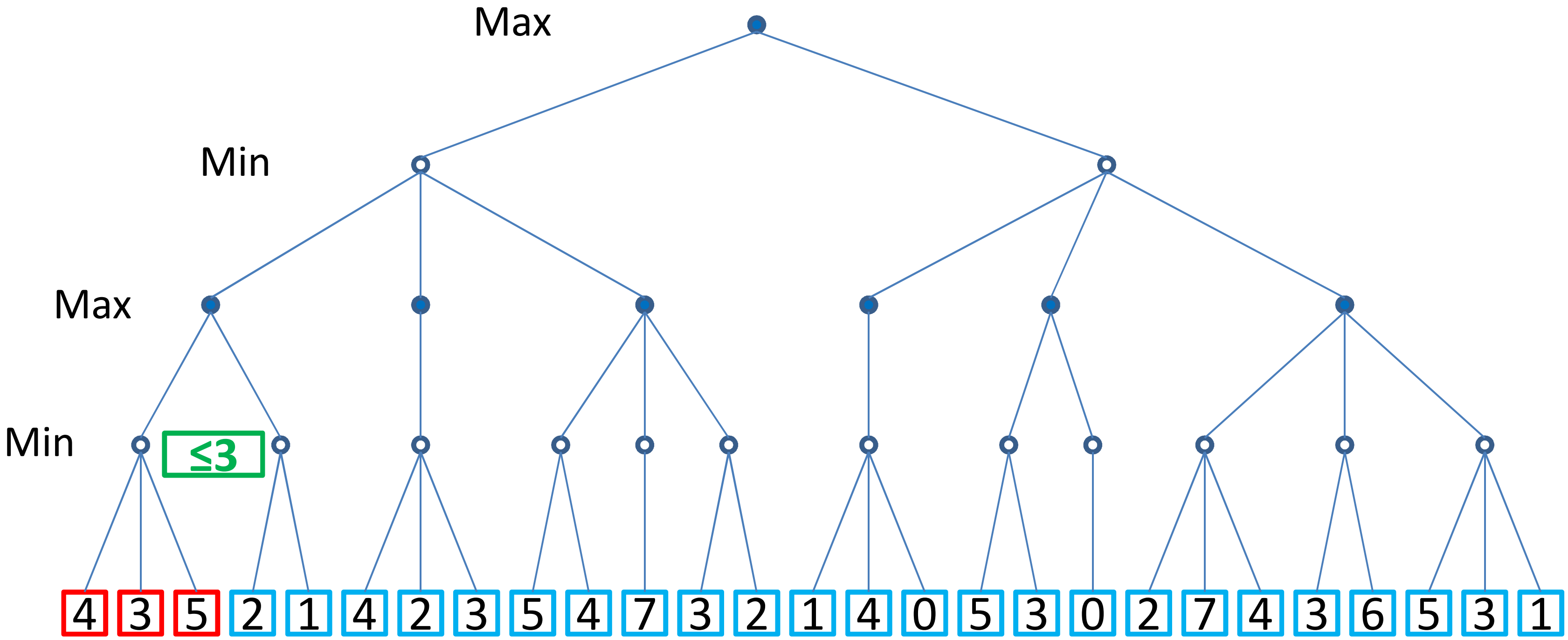
MiniMax with $\alpha\beta$-pruning



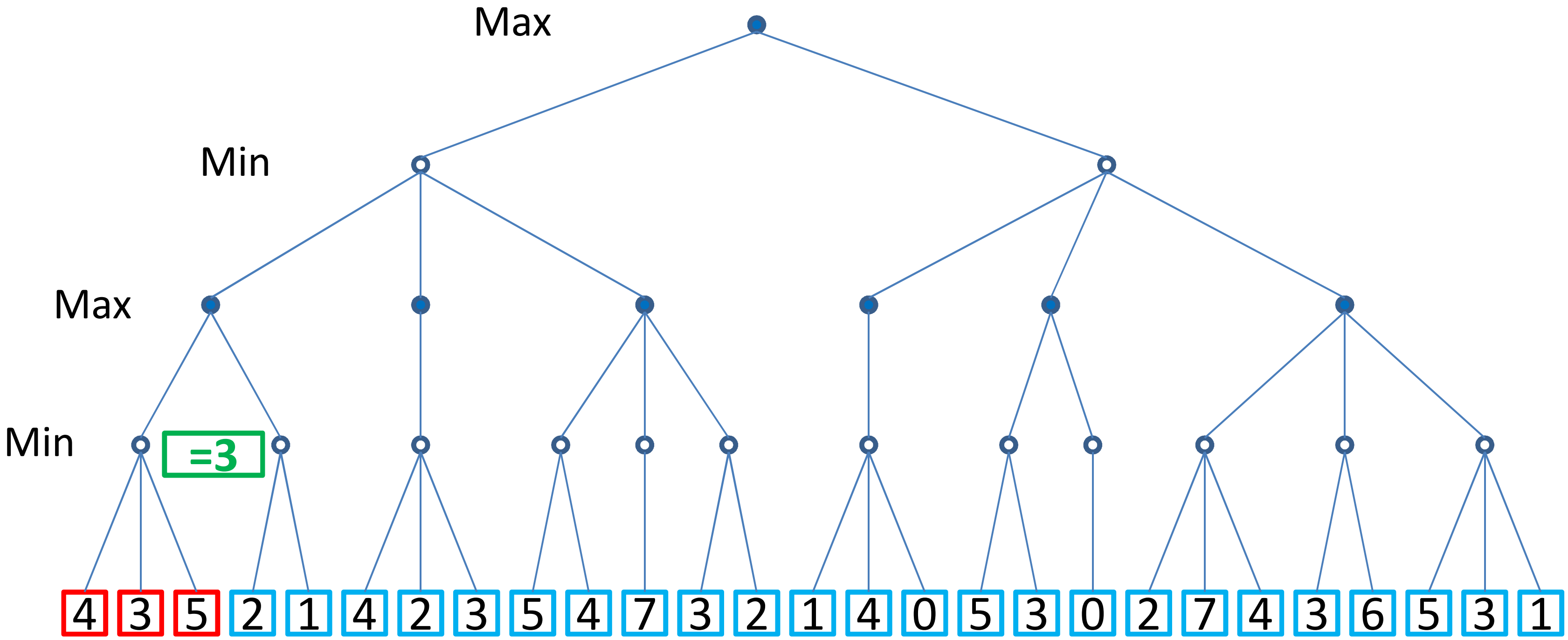
MiniMax with $\alpha\beta$-pruning



MiniMax with $\alpha\beta$-pruning

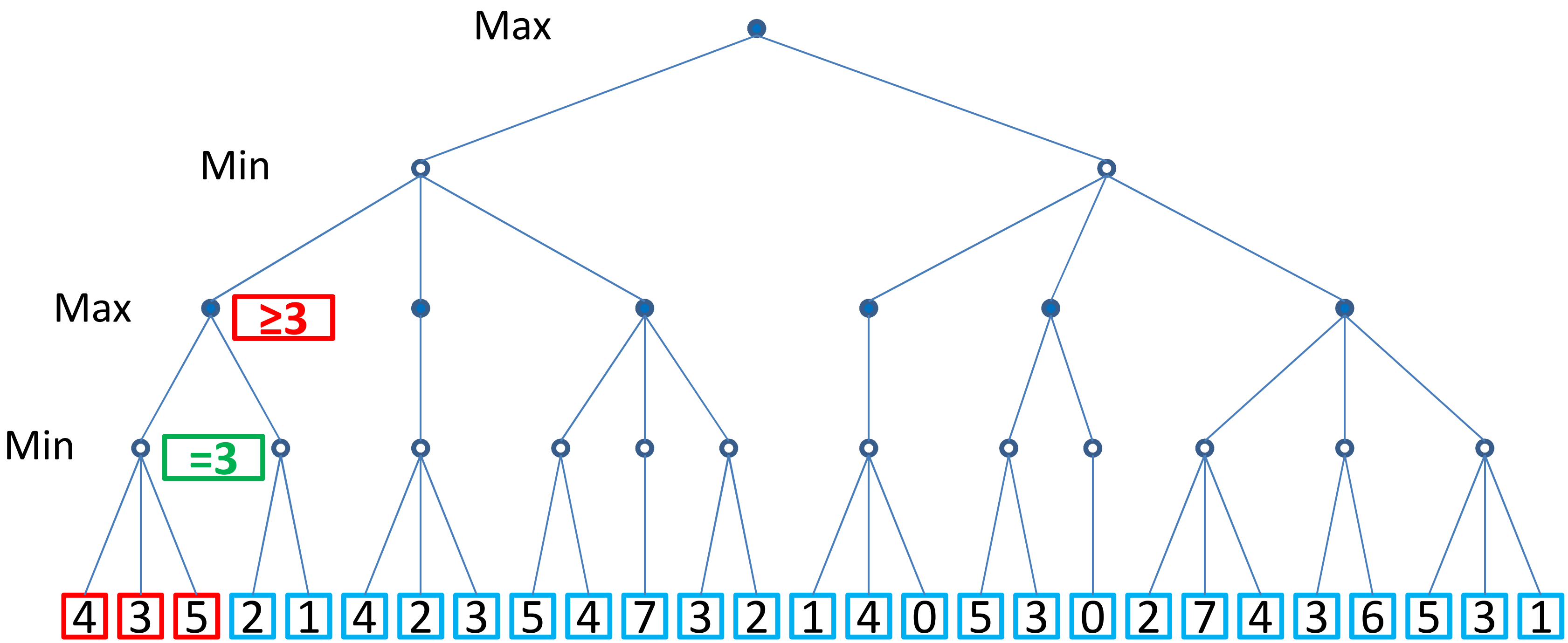


MiniMax with $\alpha\beta$-pruning

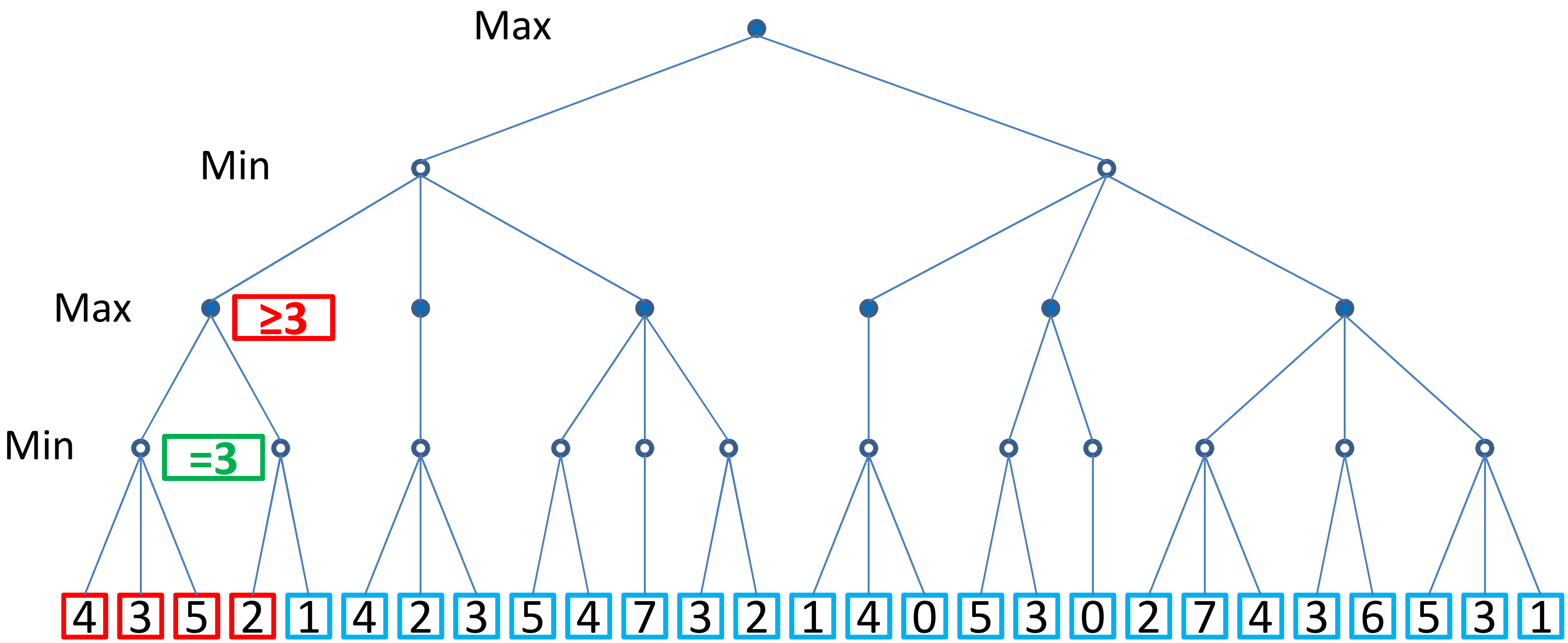


MiniMax with $\alpha\beta$-pruning

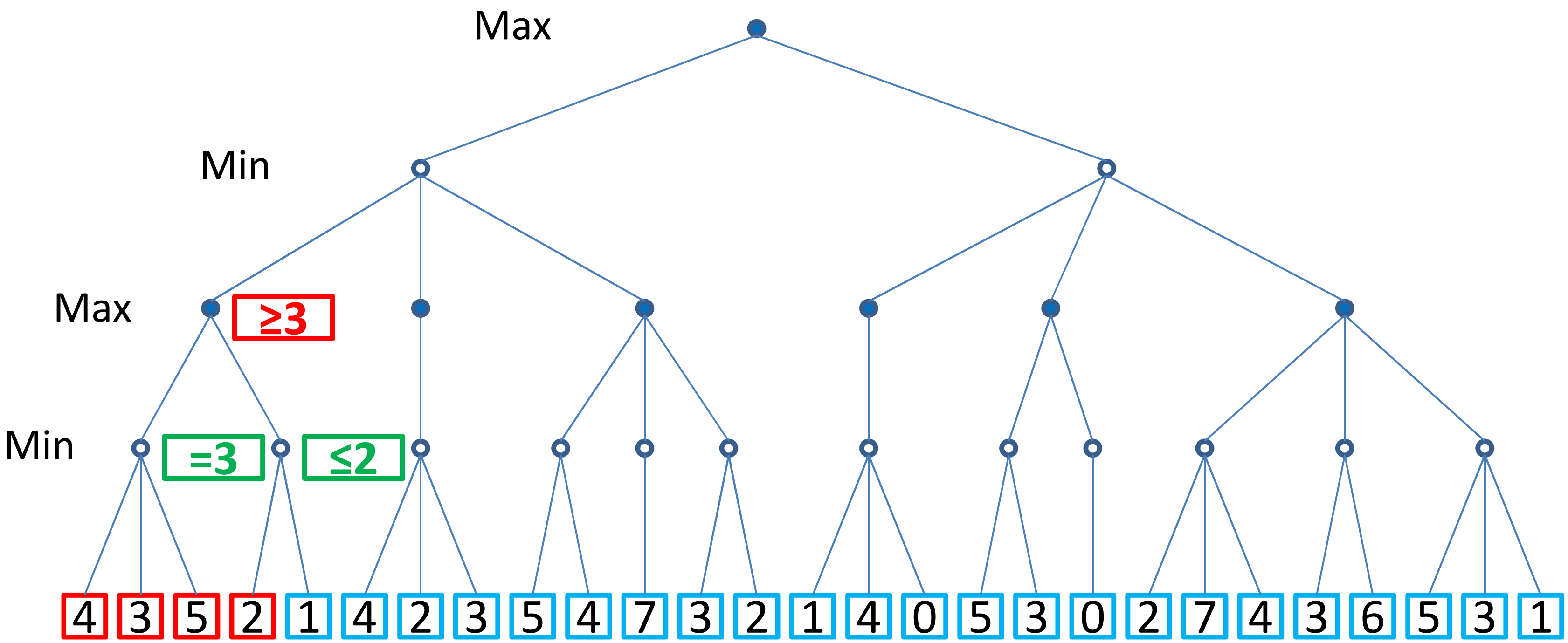
- ***β*-nodes**: Temporary values at MAX-nodes



MiniMax with $\alpha\beta$-pruning

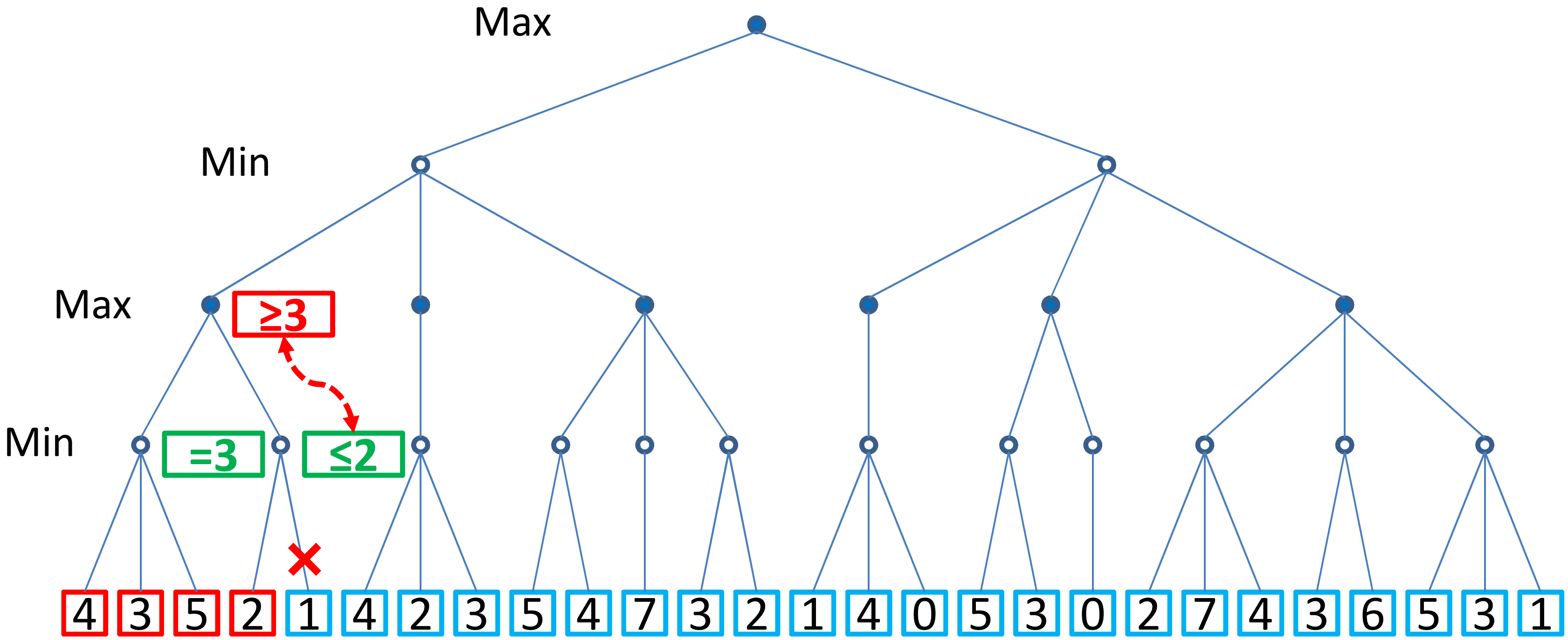


MiniMax with $\alpha\beta$-pruning

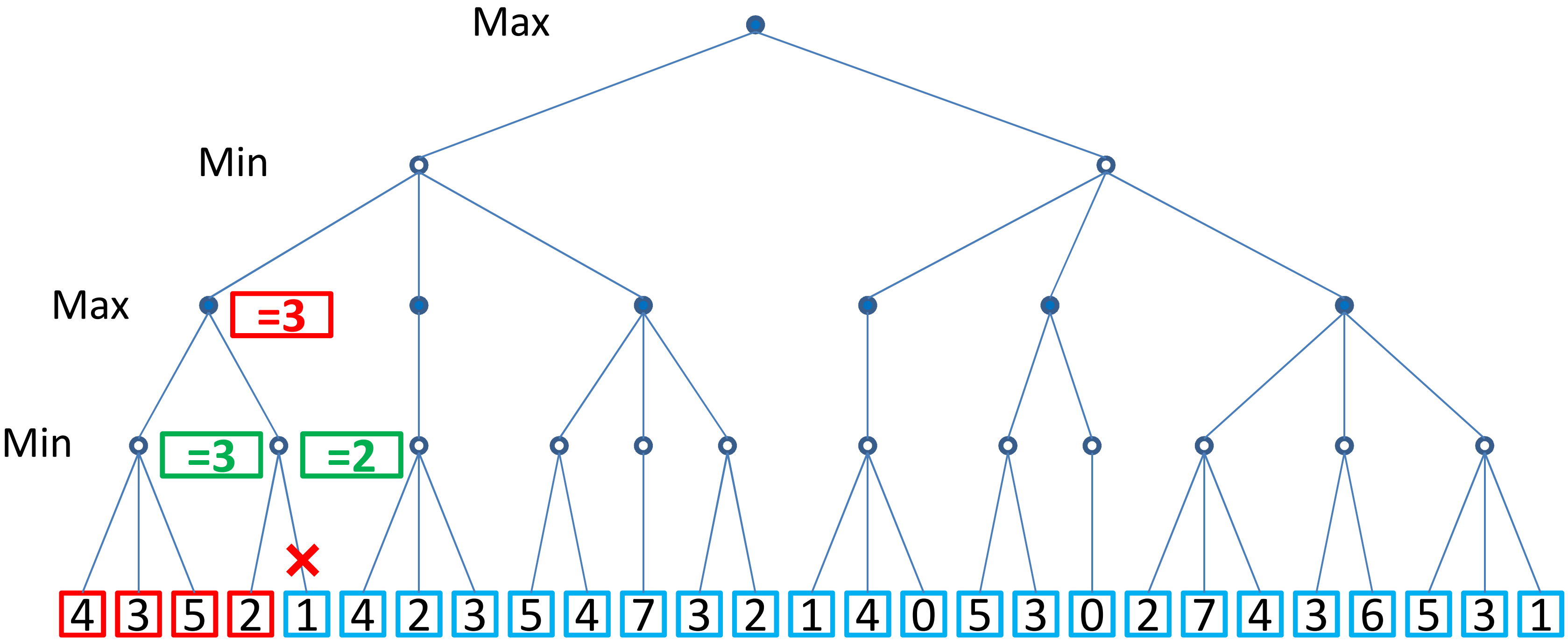


MiniMax with $\alpha\beta$-pruning

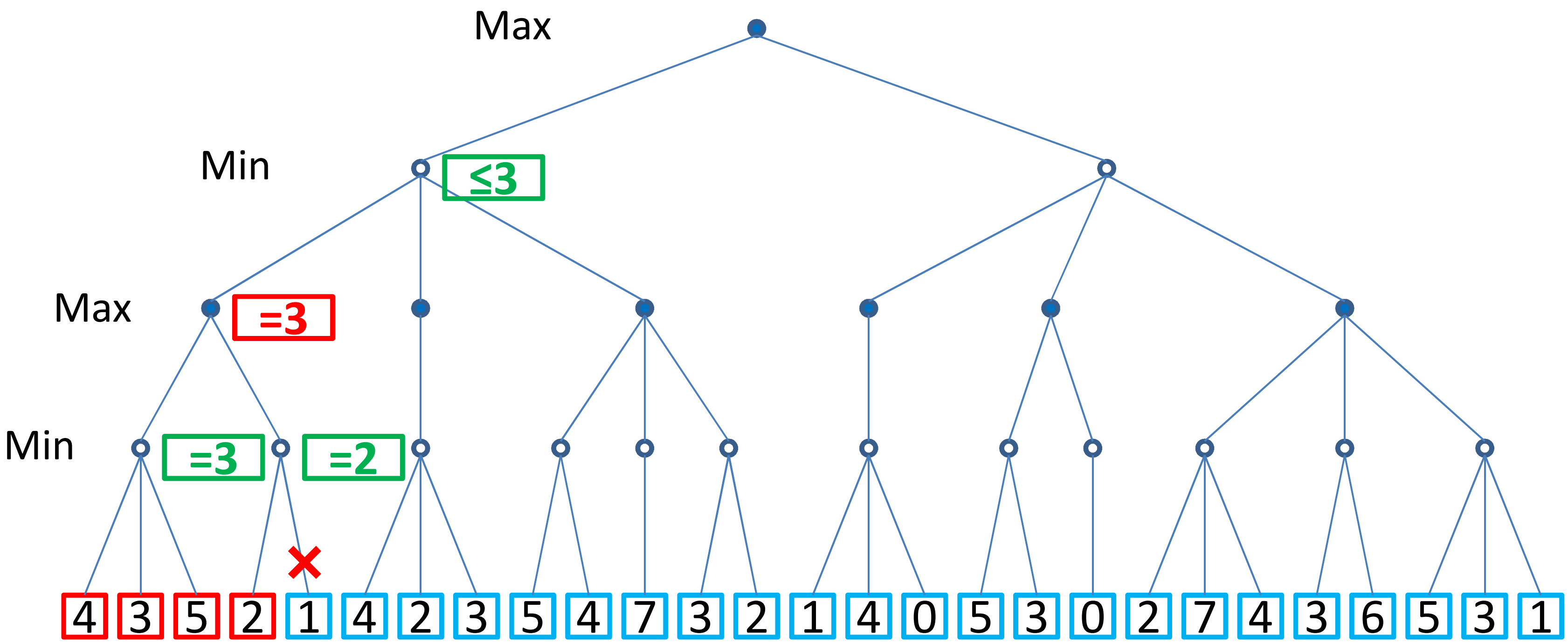
- **Prune: Parent β-node ≥ Child α-node**



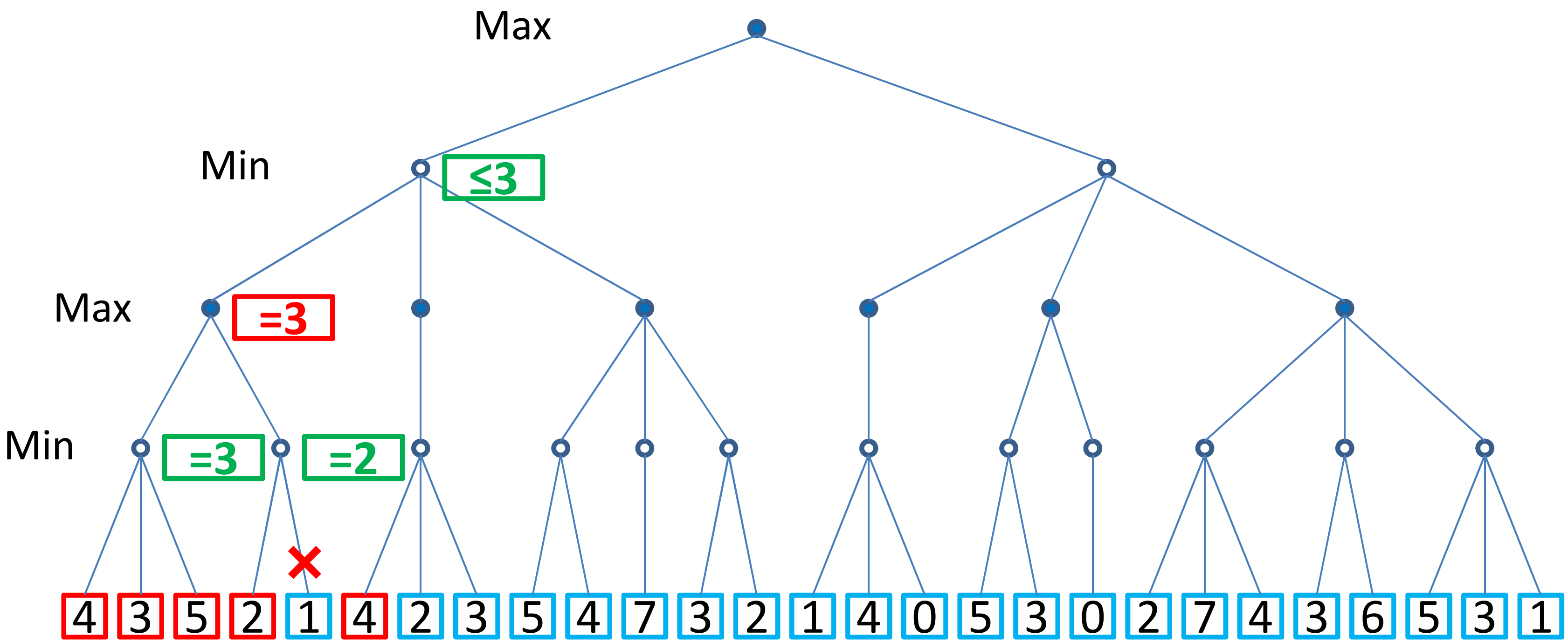
MiniMax with $\alpha\beta$-pruning



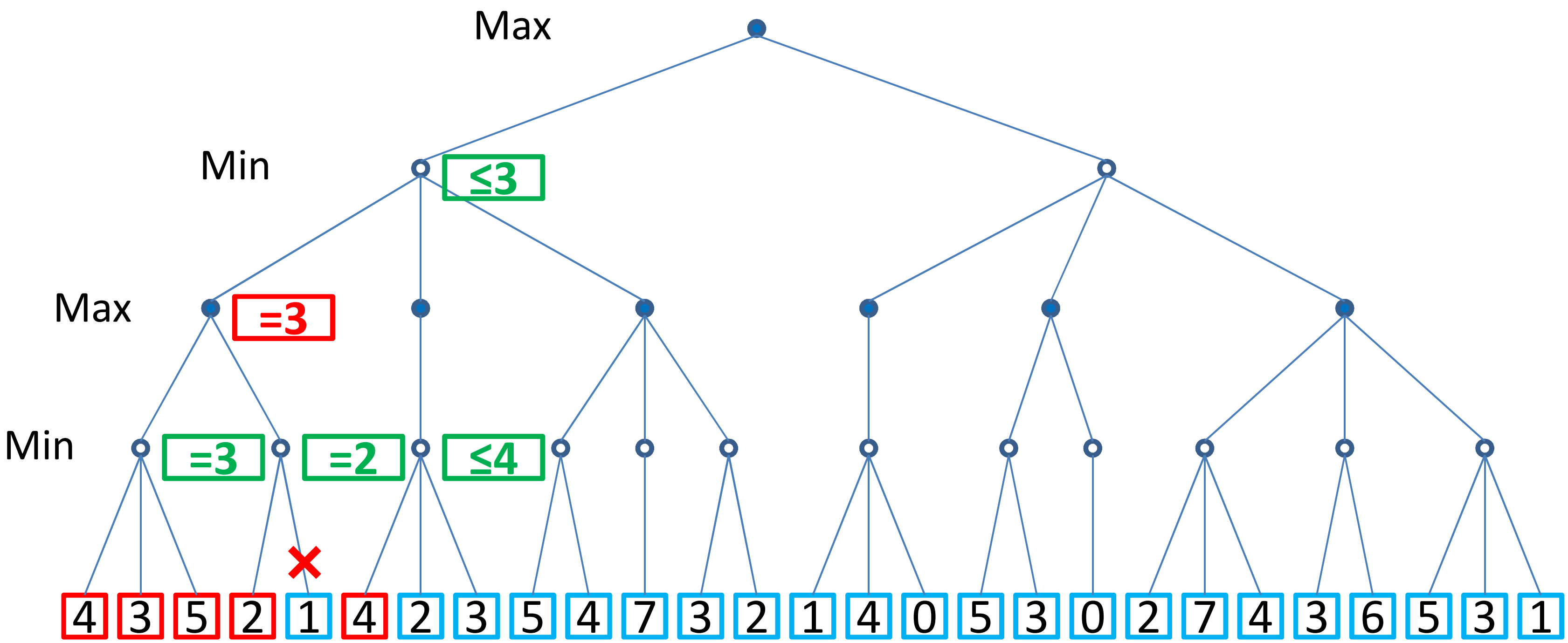
MiniMax with $\alpha\beta$-pruning



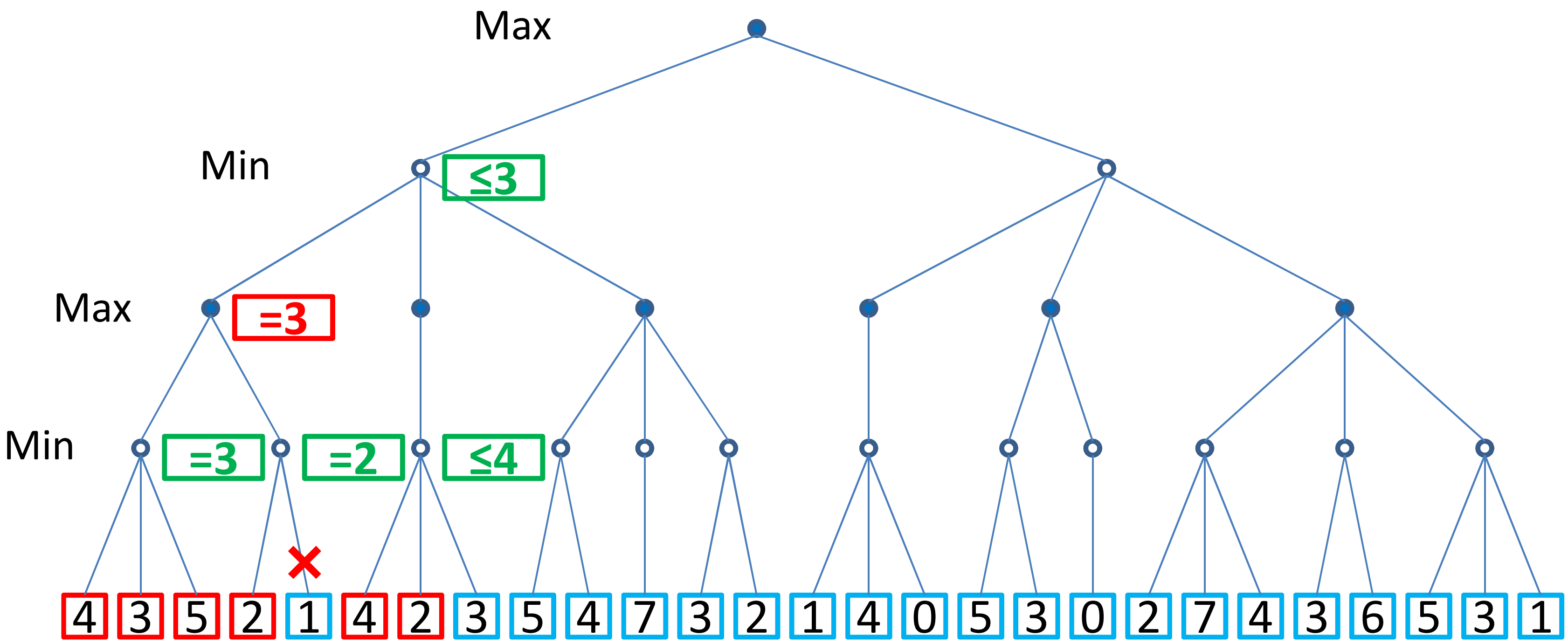
MiniMax with $\alpha\beta$-pruning



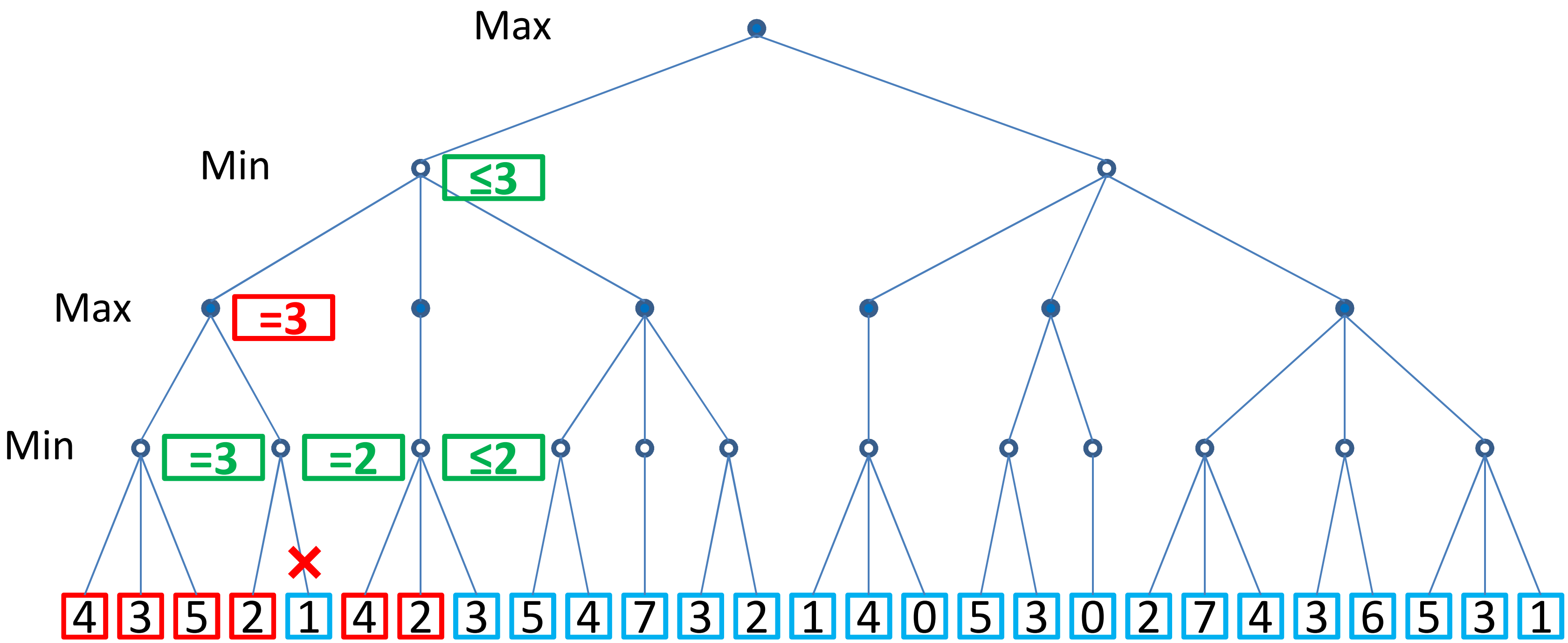
MiniMax with $\alpha\beta$-pruning



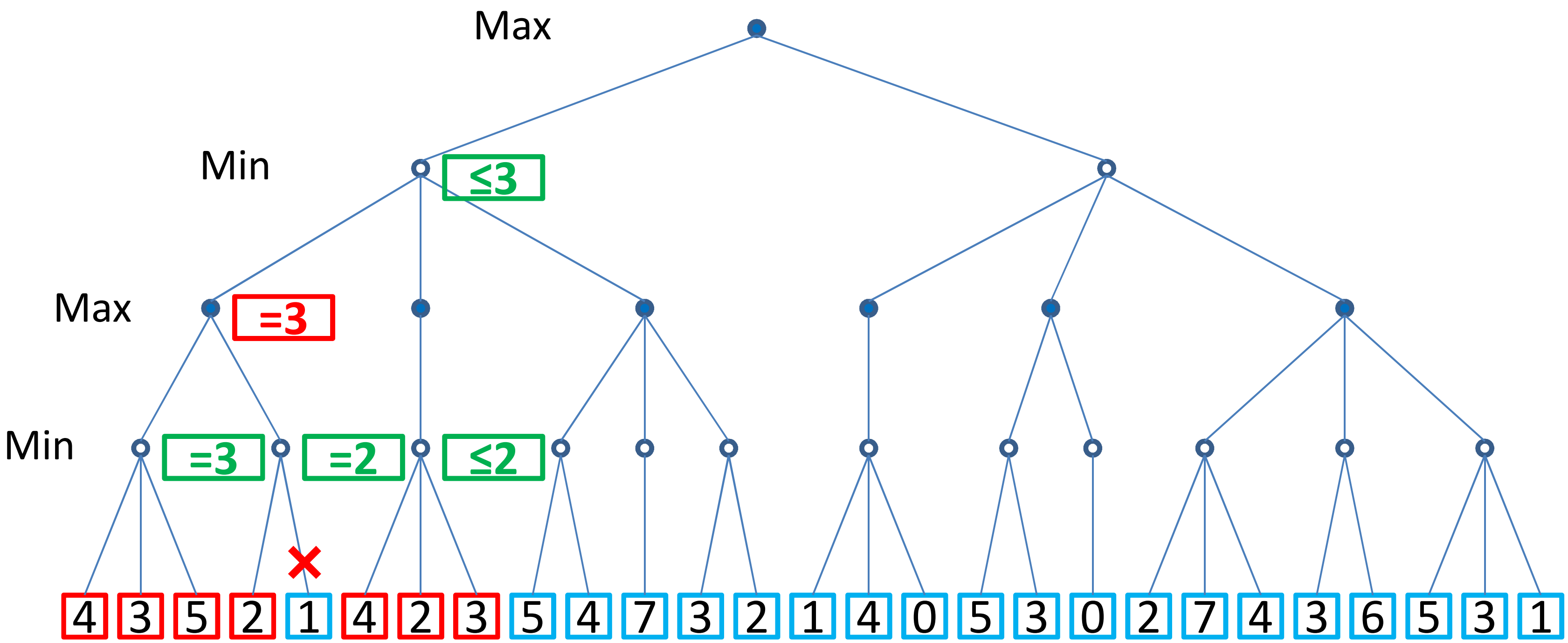
MiniMax with $\alpha\beta$-pruning



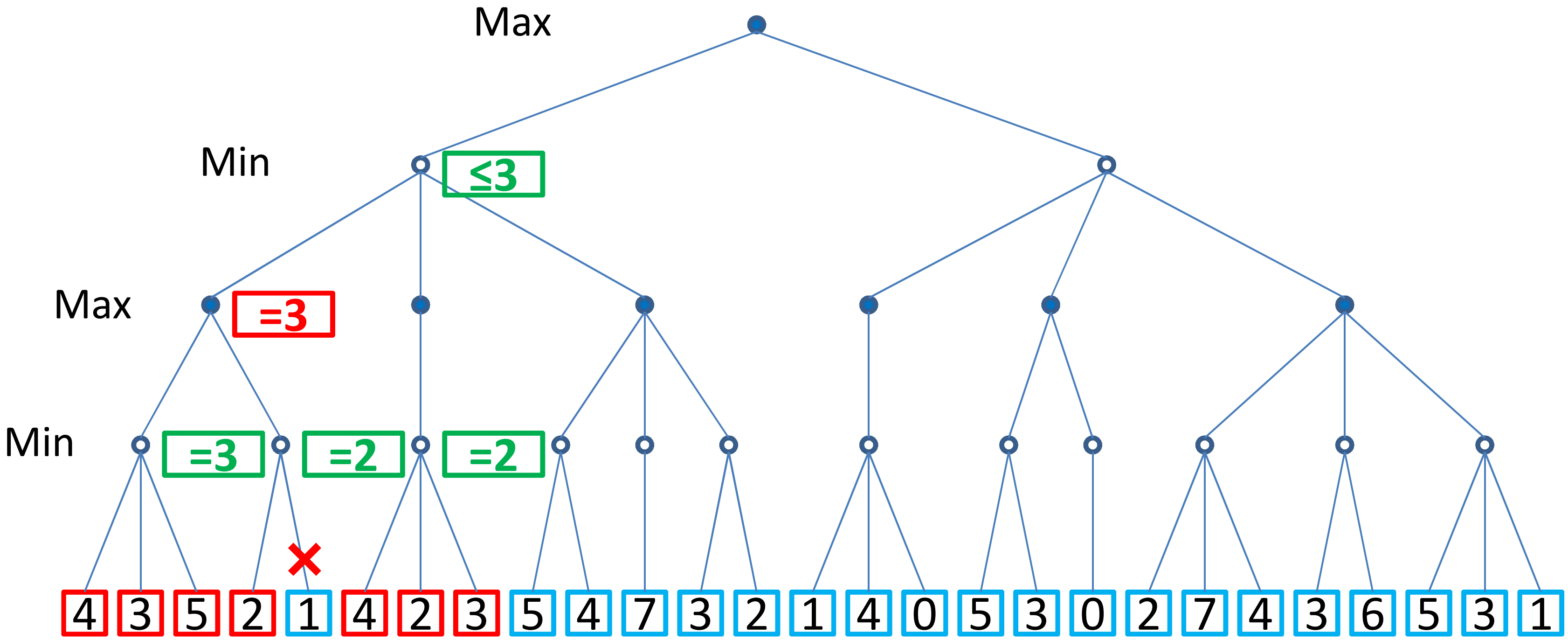
MiniMax with $\alpha\beta$-pruning



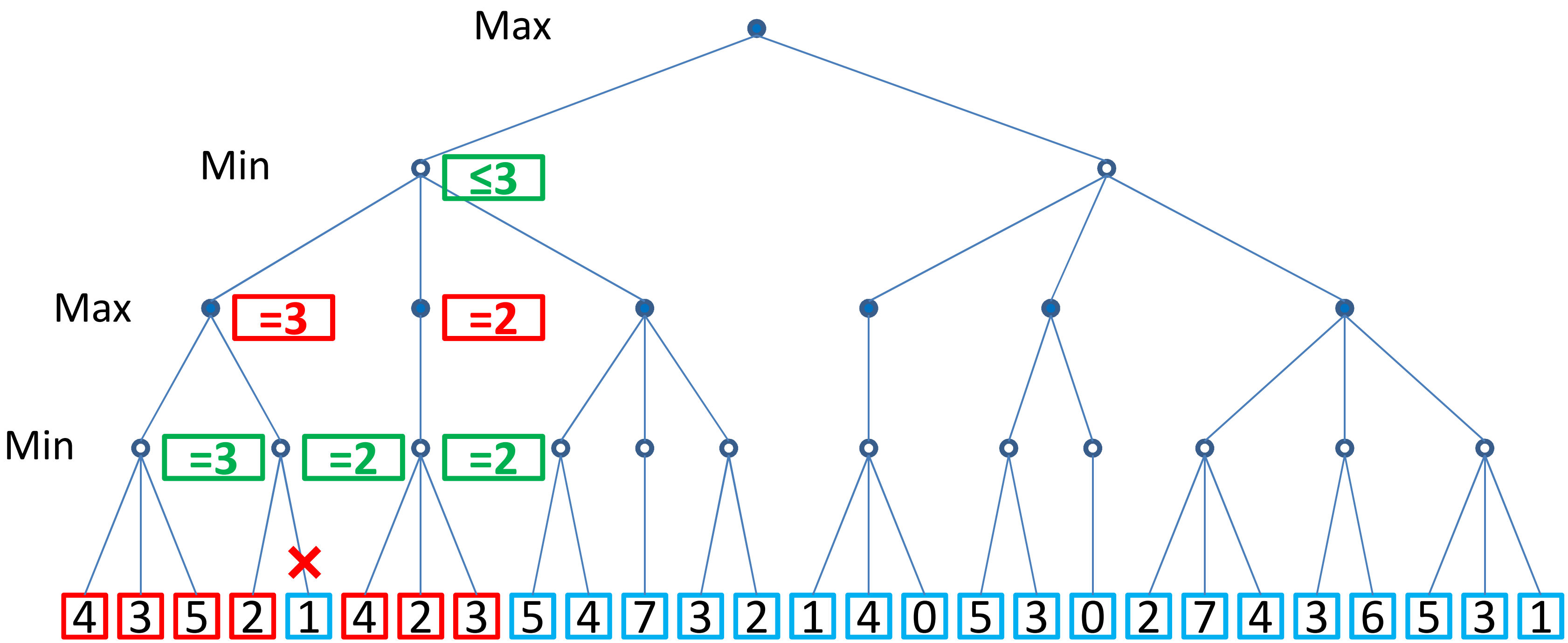
MiniMax with $\alpha\beta$-pruning



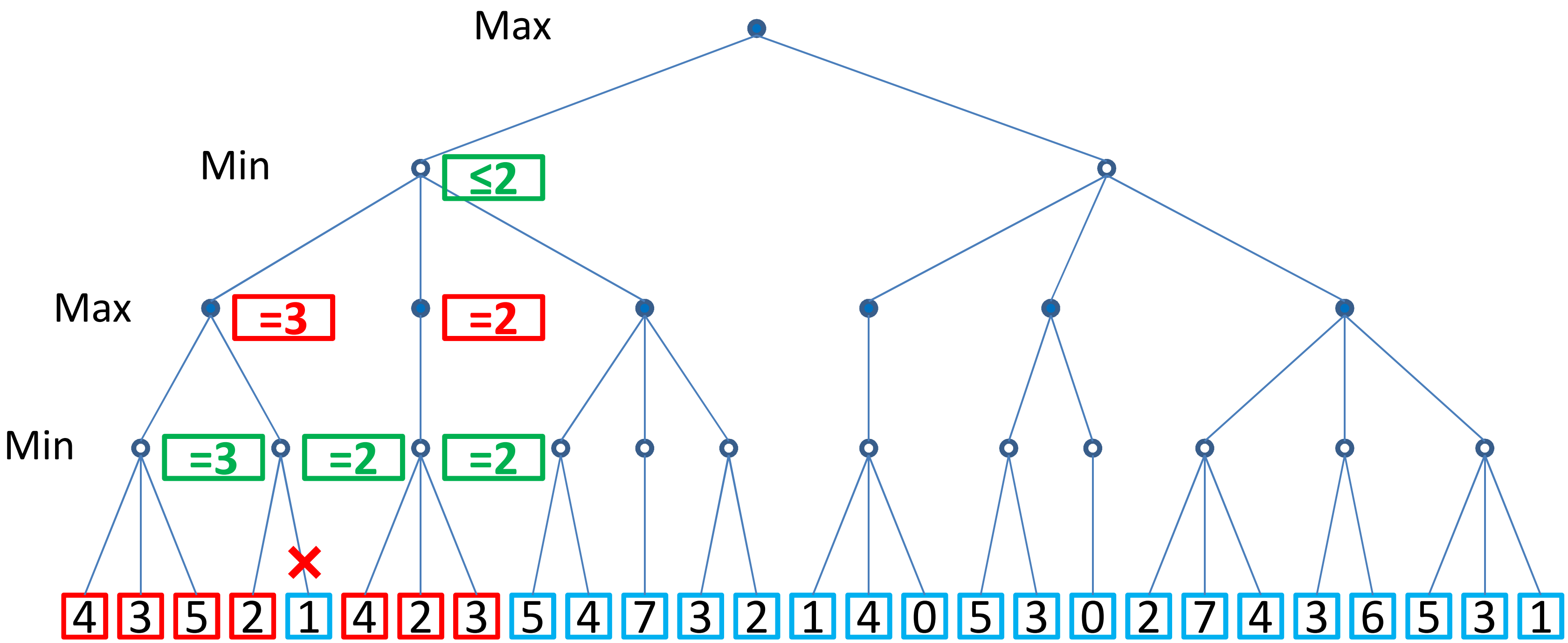
MiniMax with $\alpha\beta$-pruning



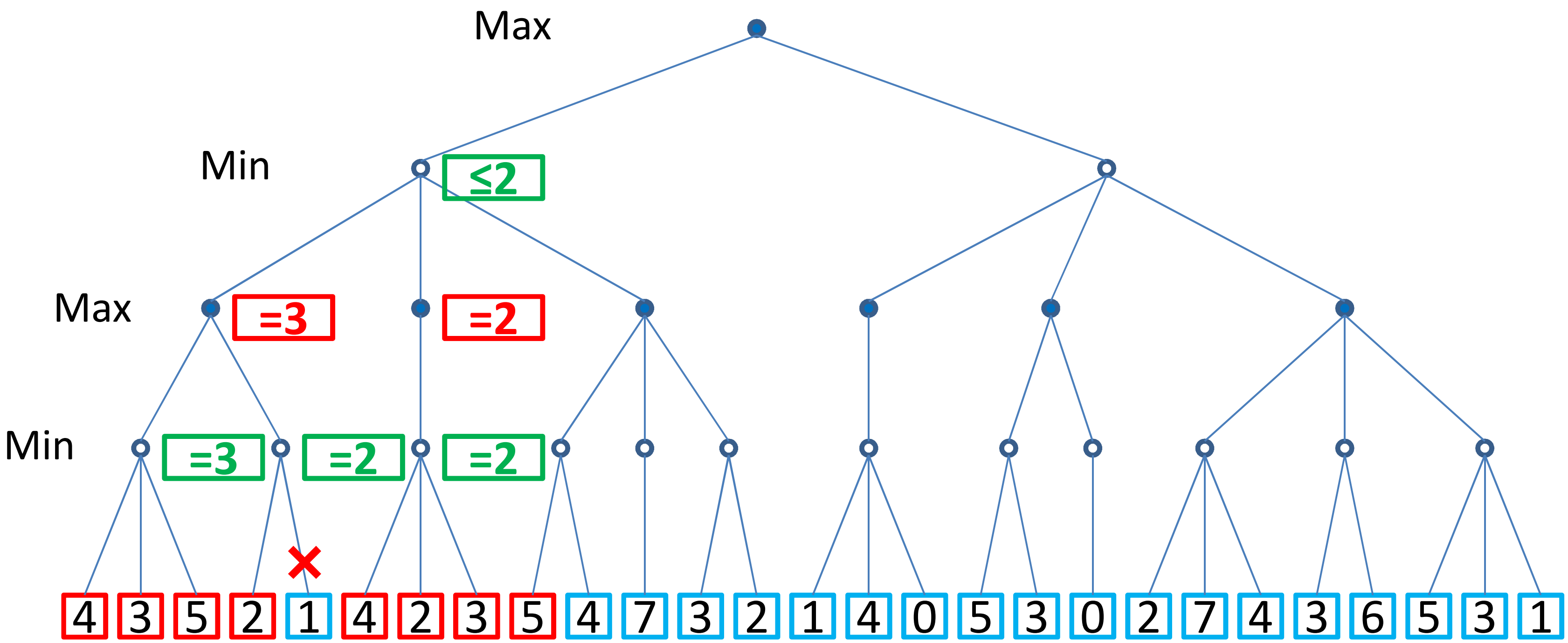
MiniMax with $\alpha\beta$-pruning



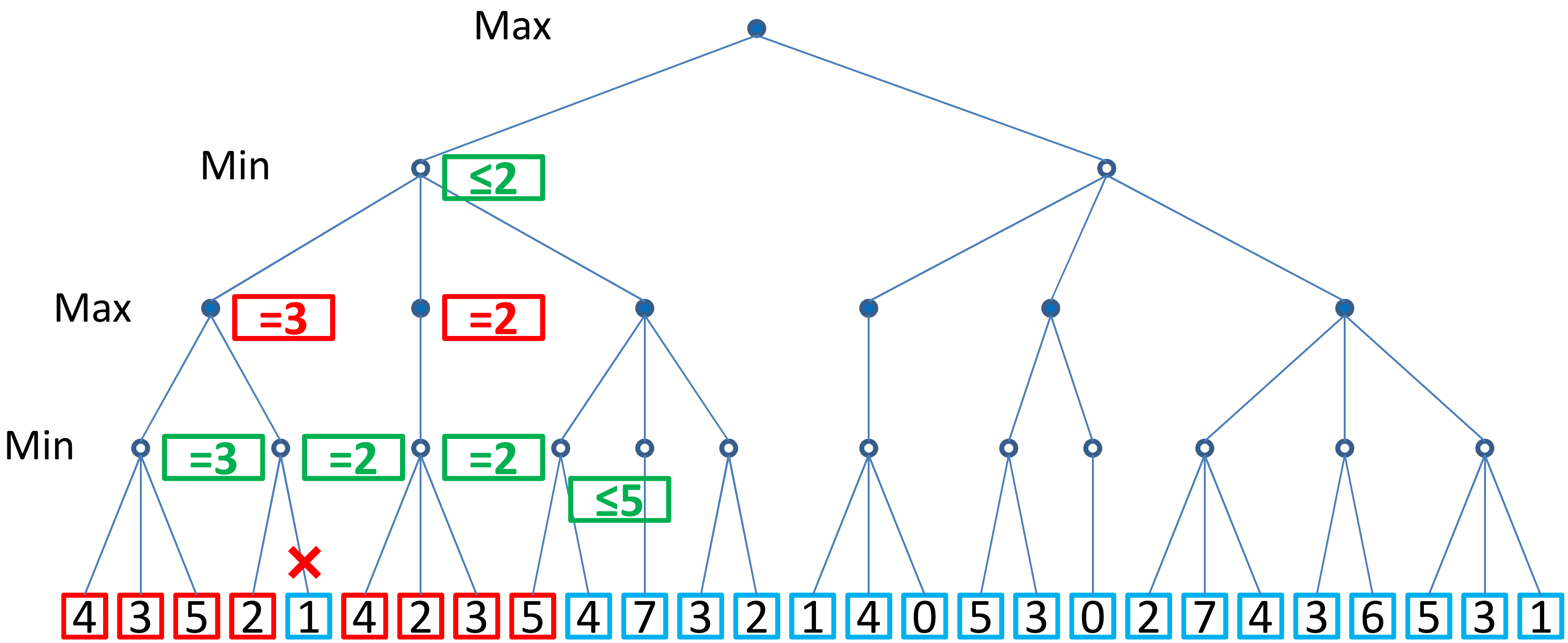
MiniMax with $\alpha\beta$-pruning



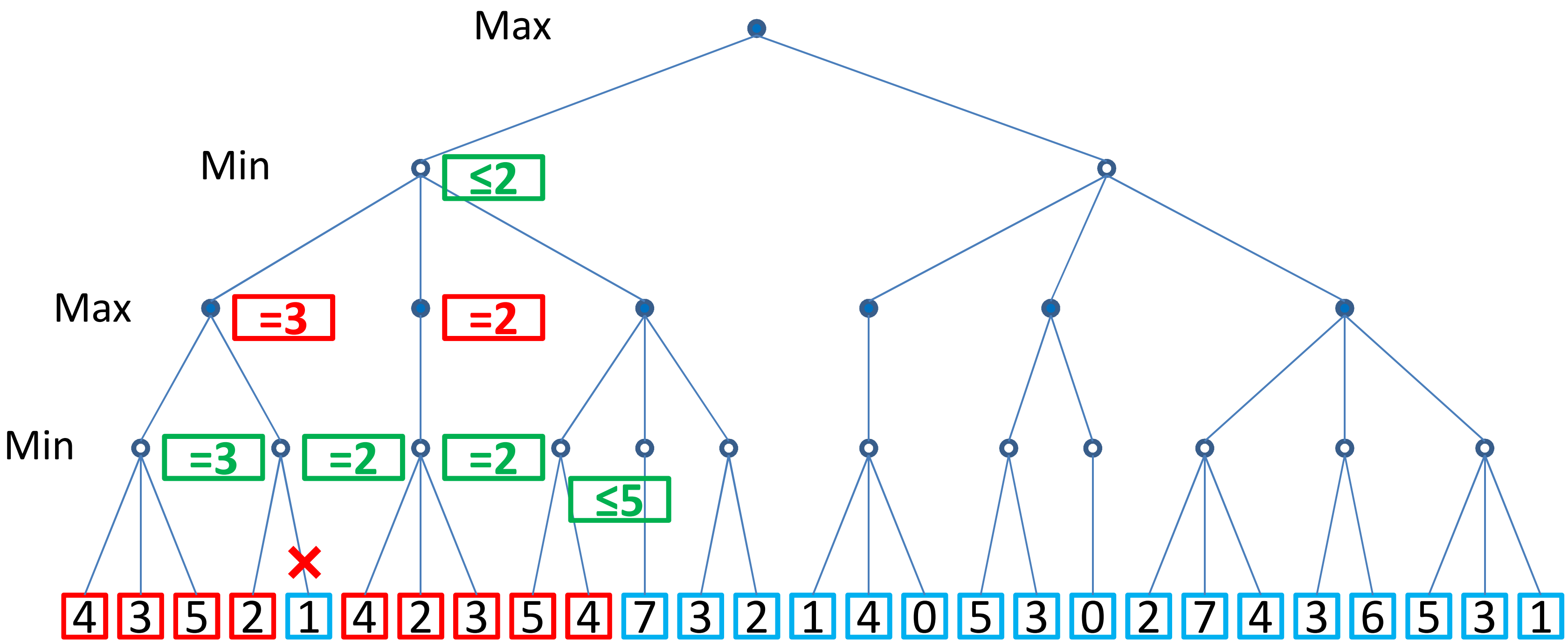
MiniMax with $\alpha\beta$-pruning



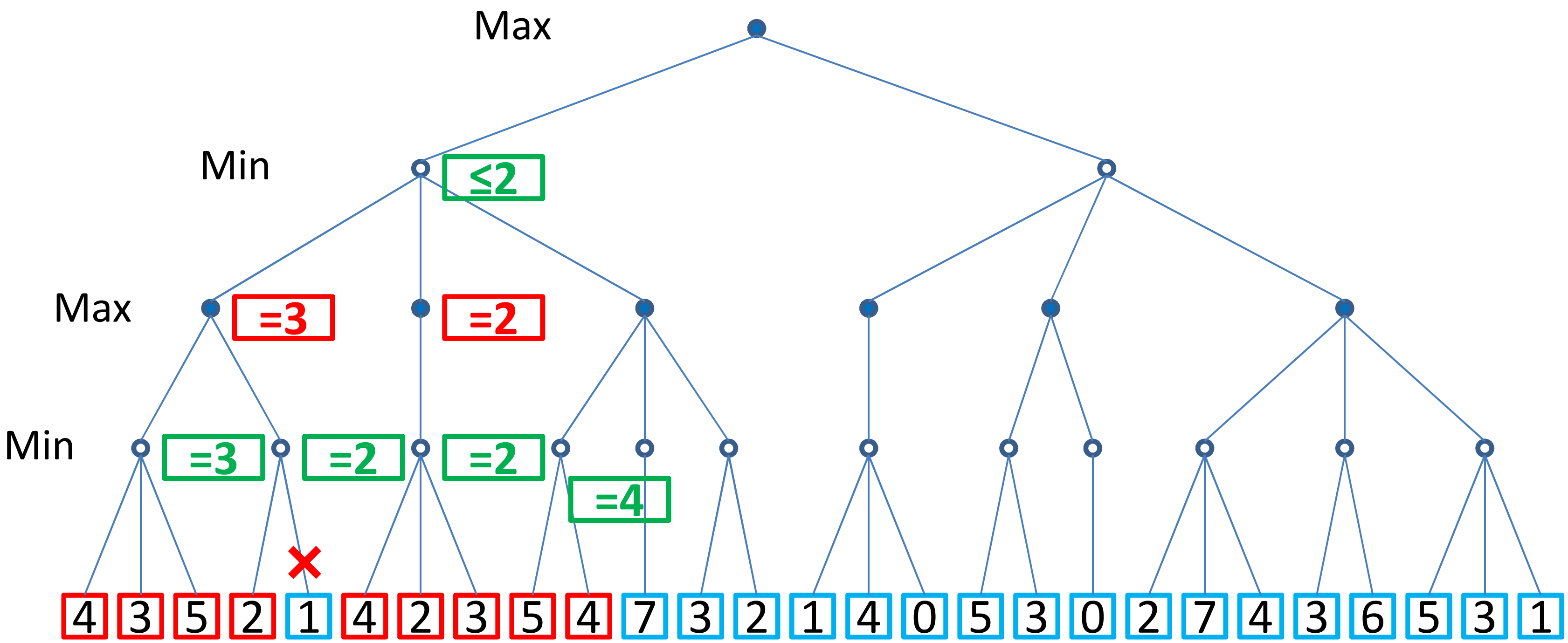
MiniMax with $\alpha\beta$-pruning



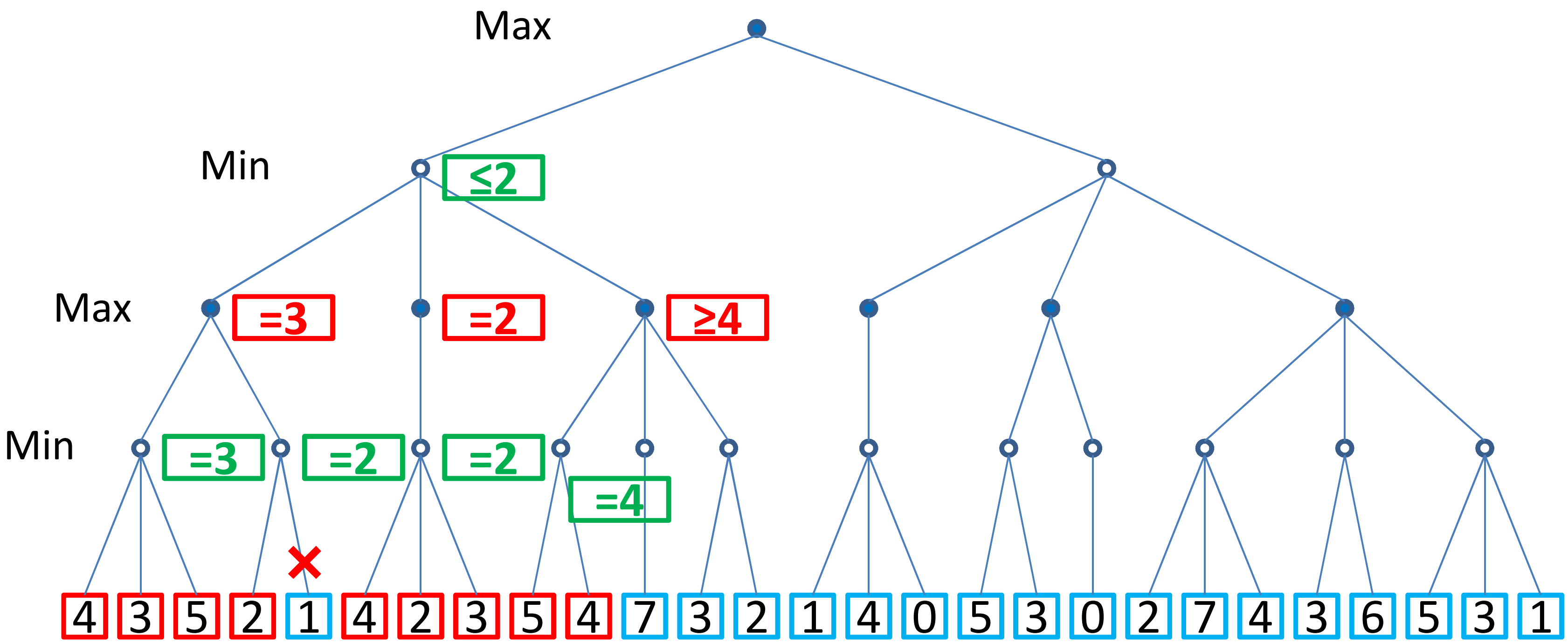
MiniMax with $\alpha\beta$-pruning



MiniMax with $\alpha\beta$-pruning

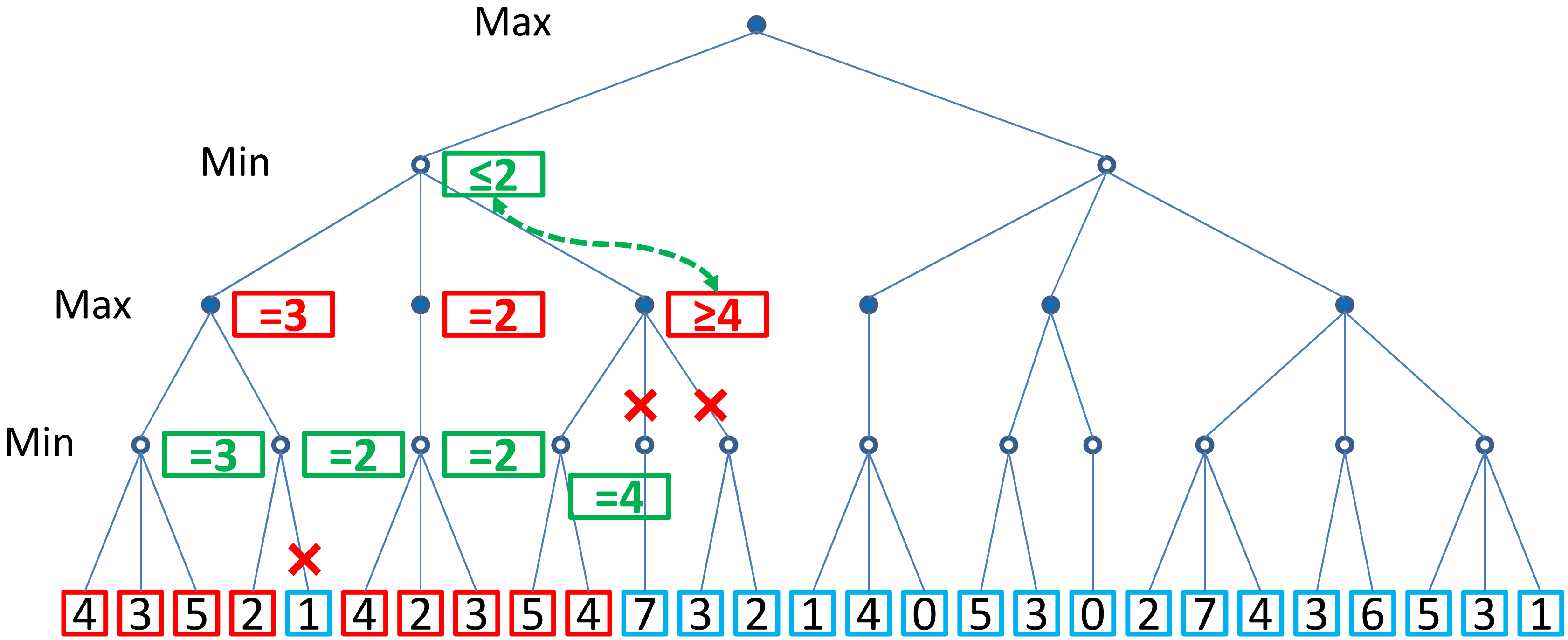


MiniMax with $\alpha\beta$-pruning

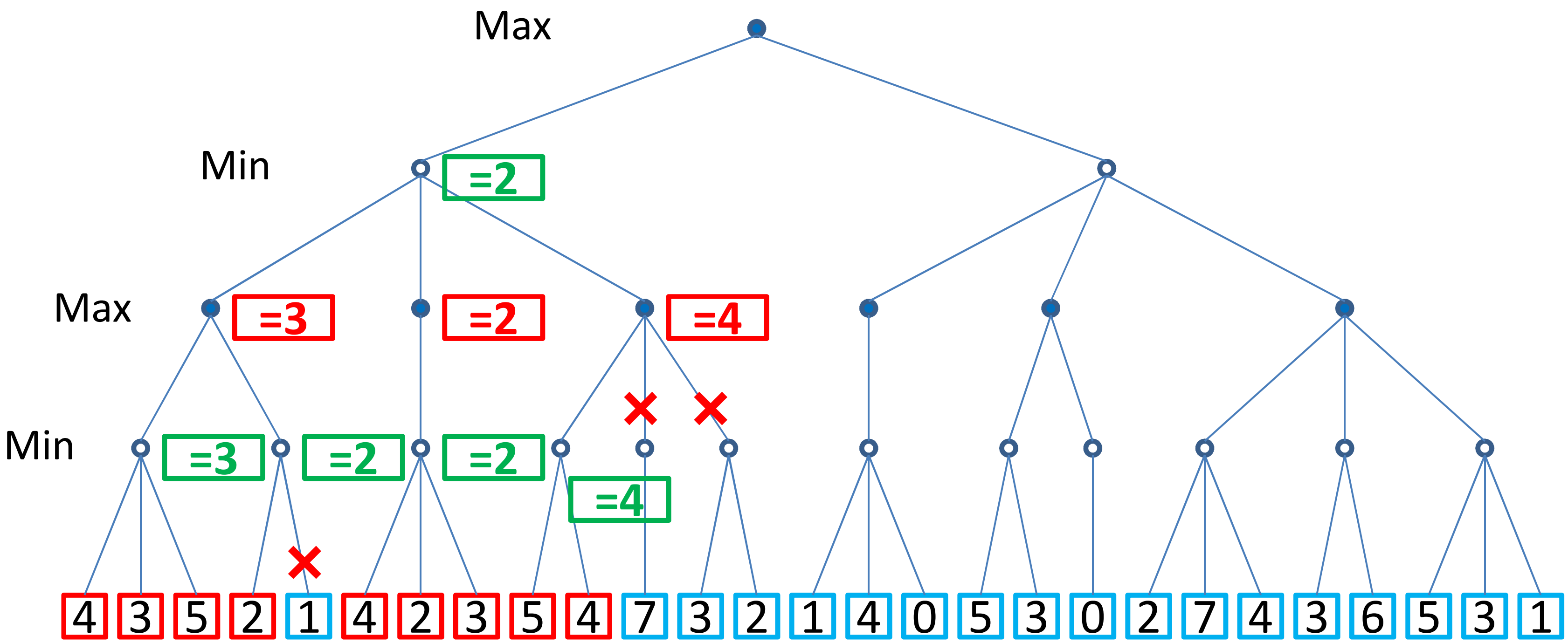


MiniMax with $\alpha\beta$-pruning

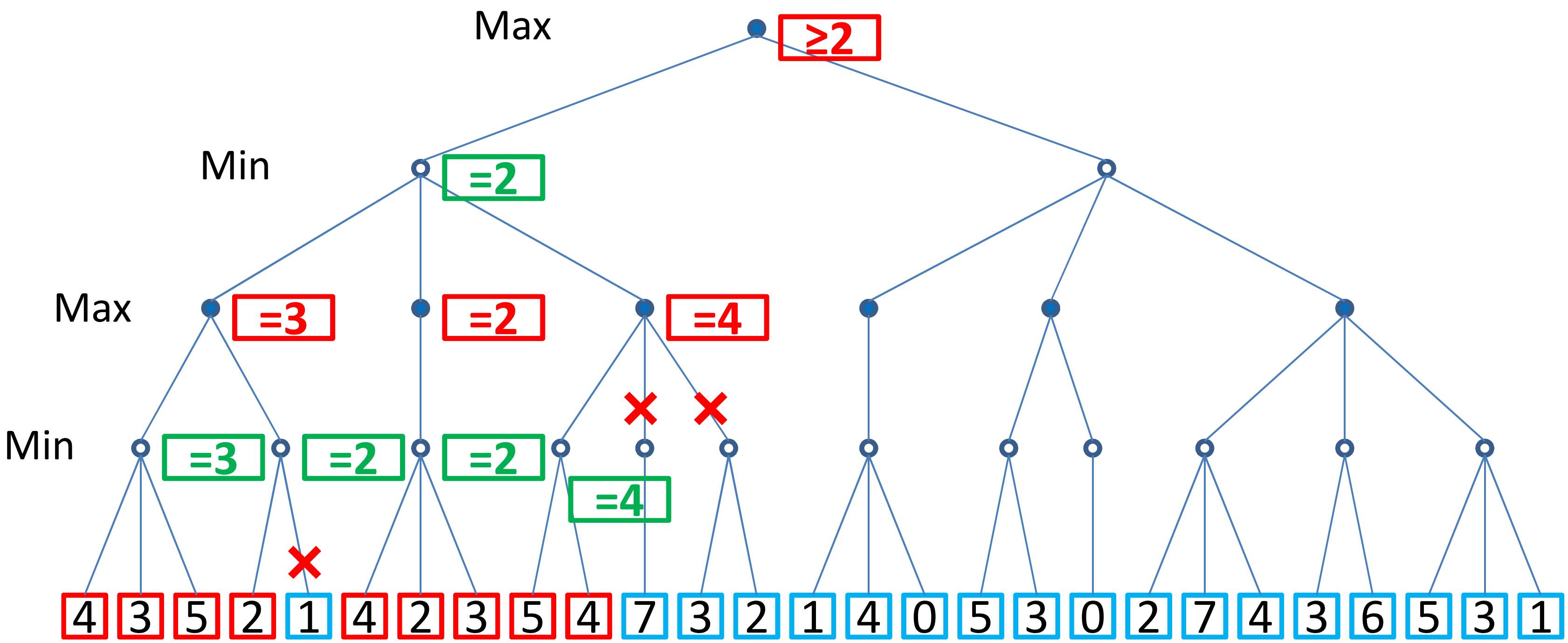
- **Prune: Parent α-node ≤ Child β-node**



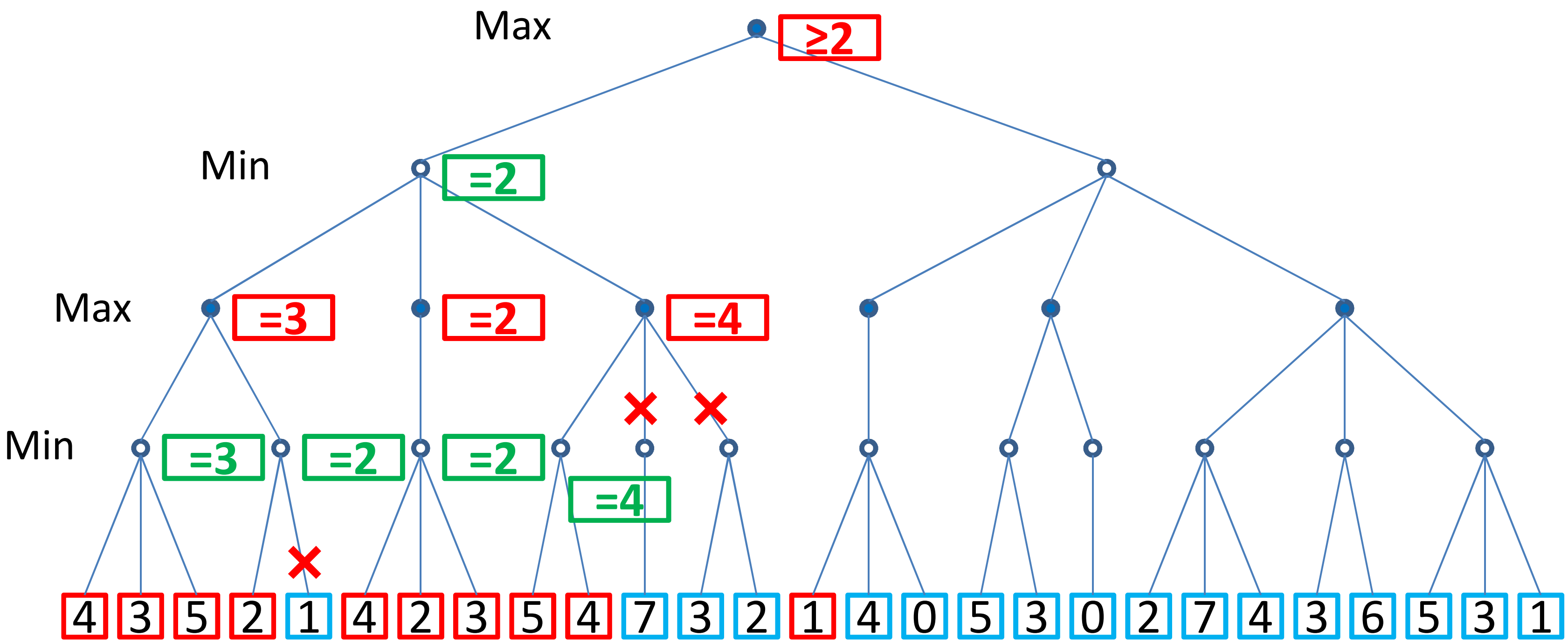
MiniMax with $\alpha\beta$-pruning



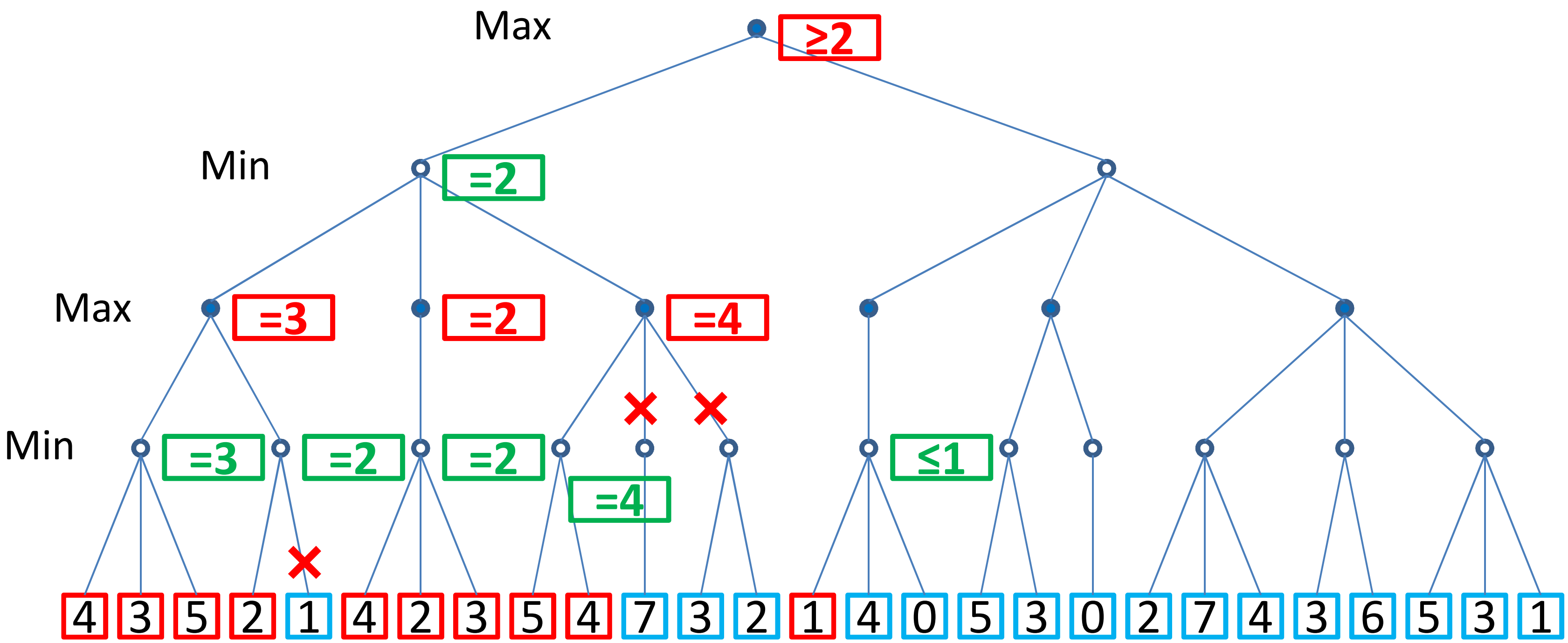
MiniMax with $\alpha\beta$-pruning



MiniMax with $\alpha\beta$-pruning

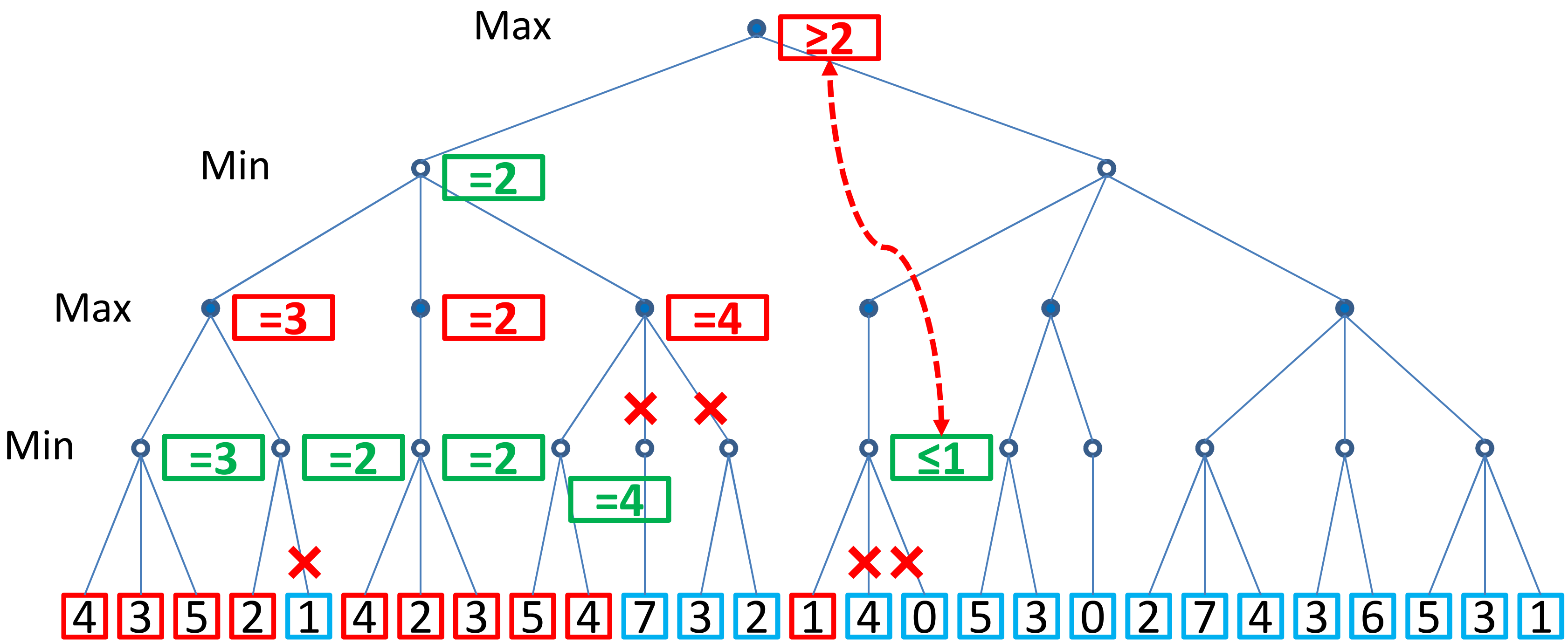


MiniMax with $\alpha\beta$-pruning

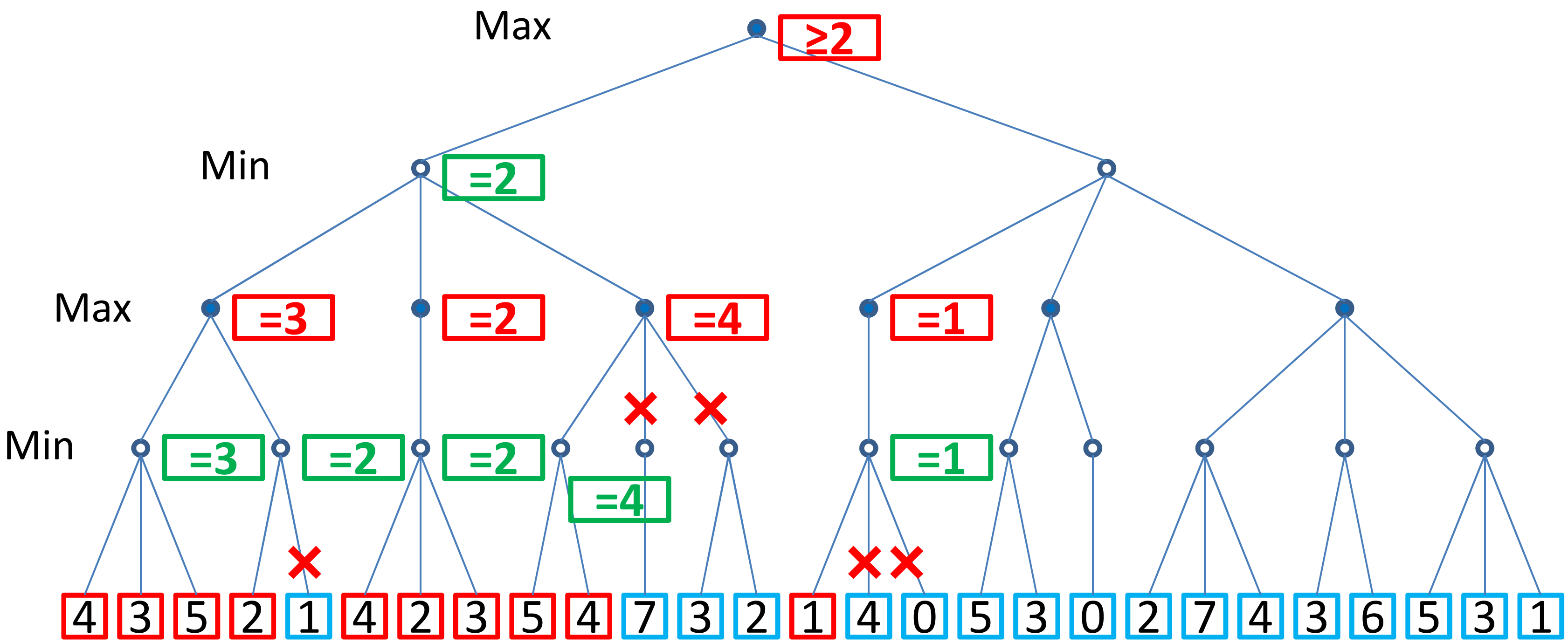


MiniMax with $\alpha\beta$-pruning

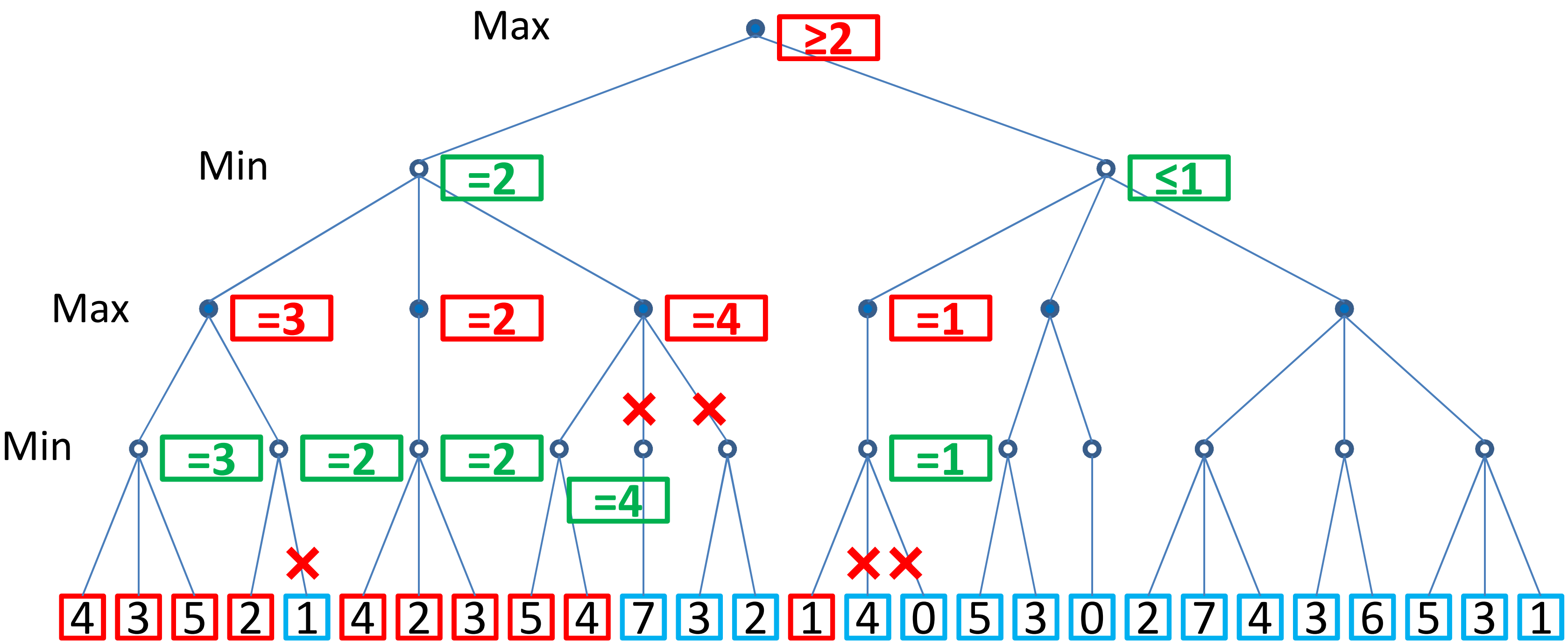
- **"Deep" cut-off: Ancestor β-node ≥ α-node**



MiniMax with $\alpha\beta$-pruning

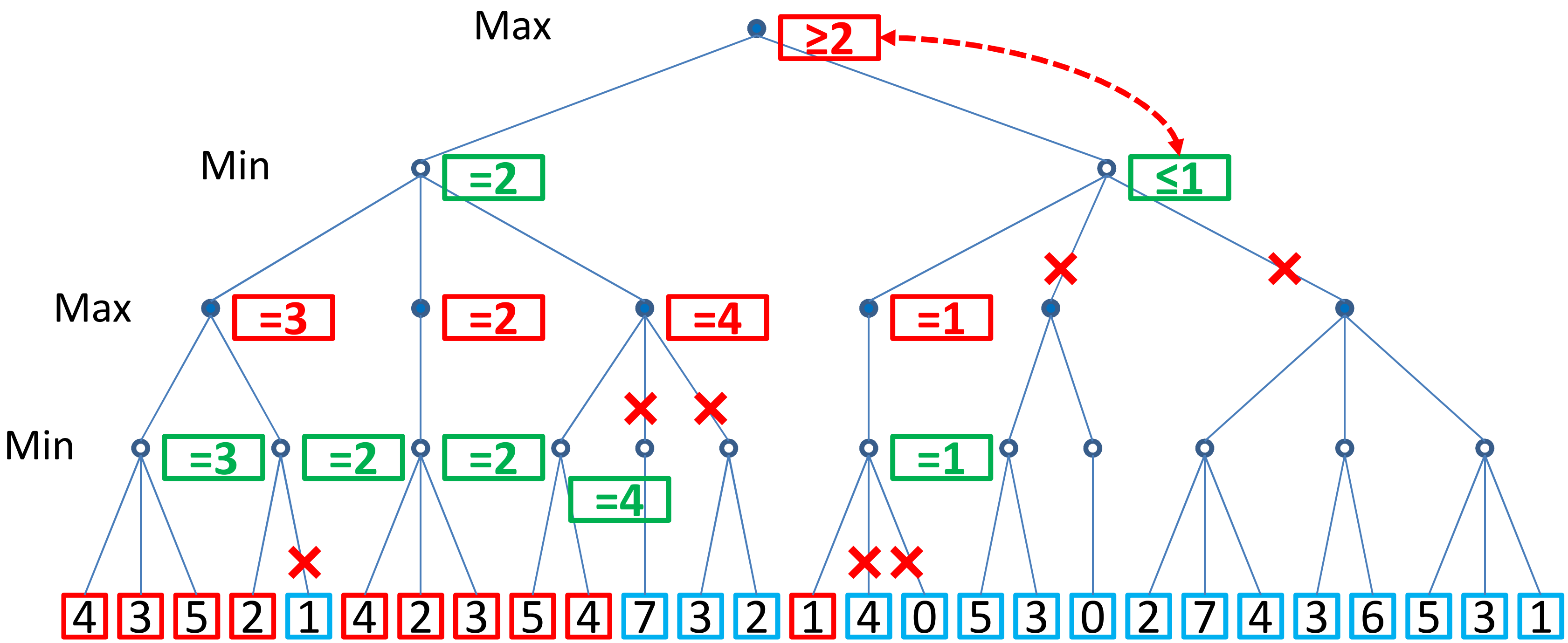


MiniMax with $\alpha\beta$-pruning



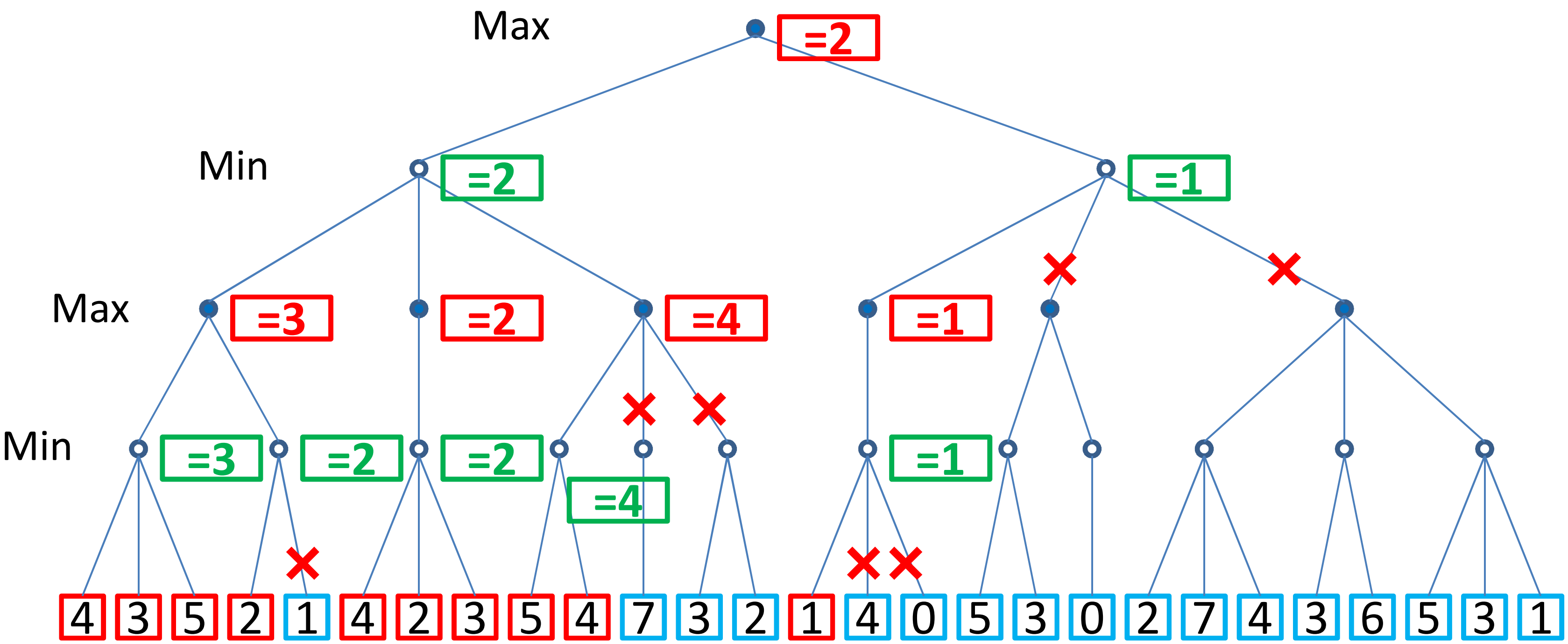
MiniMax with $\alpha\beta$-pruning

- **Prune: Parent β-node ≥ Child α-node**



MiniMax with $\alpha\beta$-pruning

- **17 static evaluations saved**

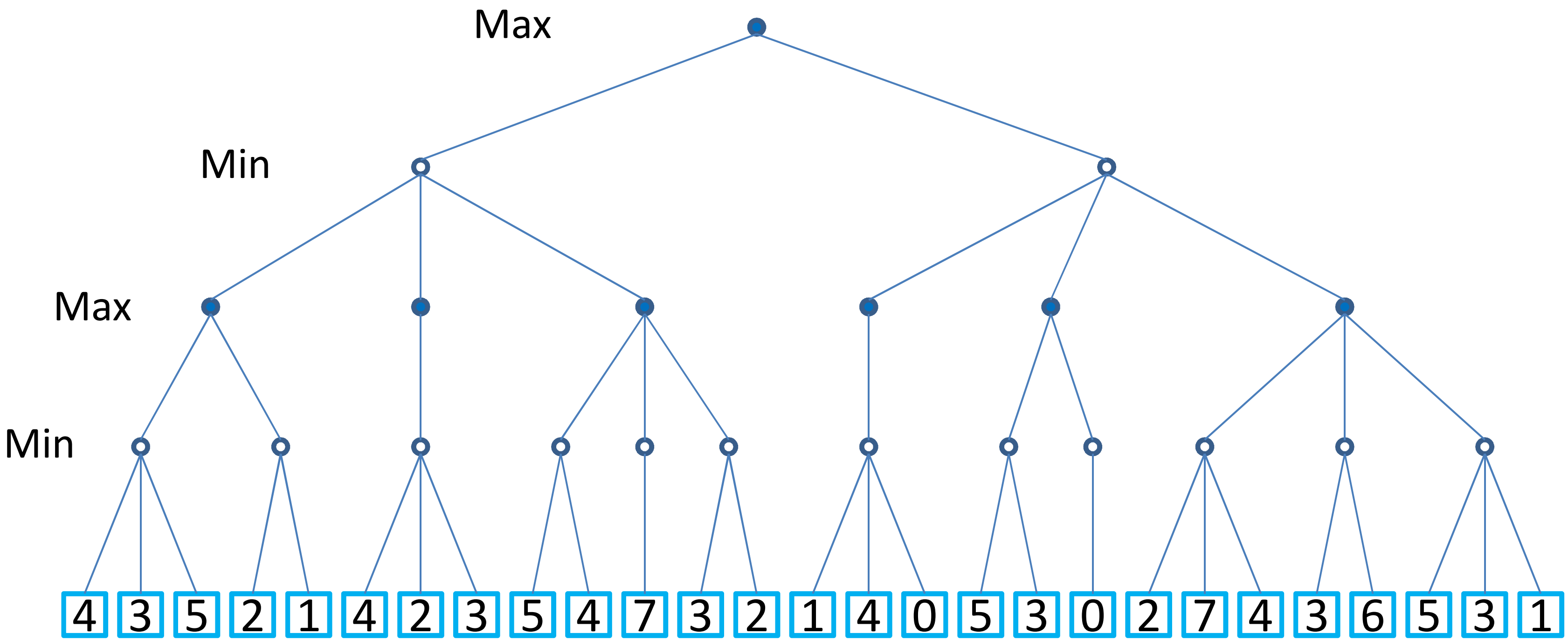


MiniMax & Constraint Processing: MiniMax Algorithm

PROBLEM 2

Problem 2

- Can the nodes be ordered in such a way that $\alpha\beta$-pruning can cut off more branches?

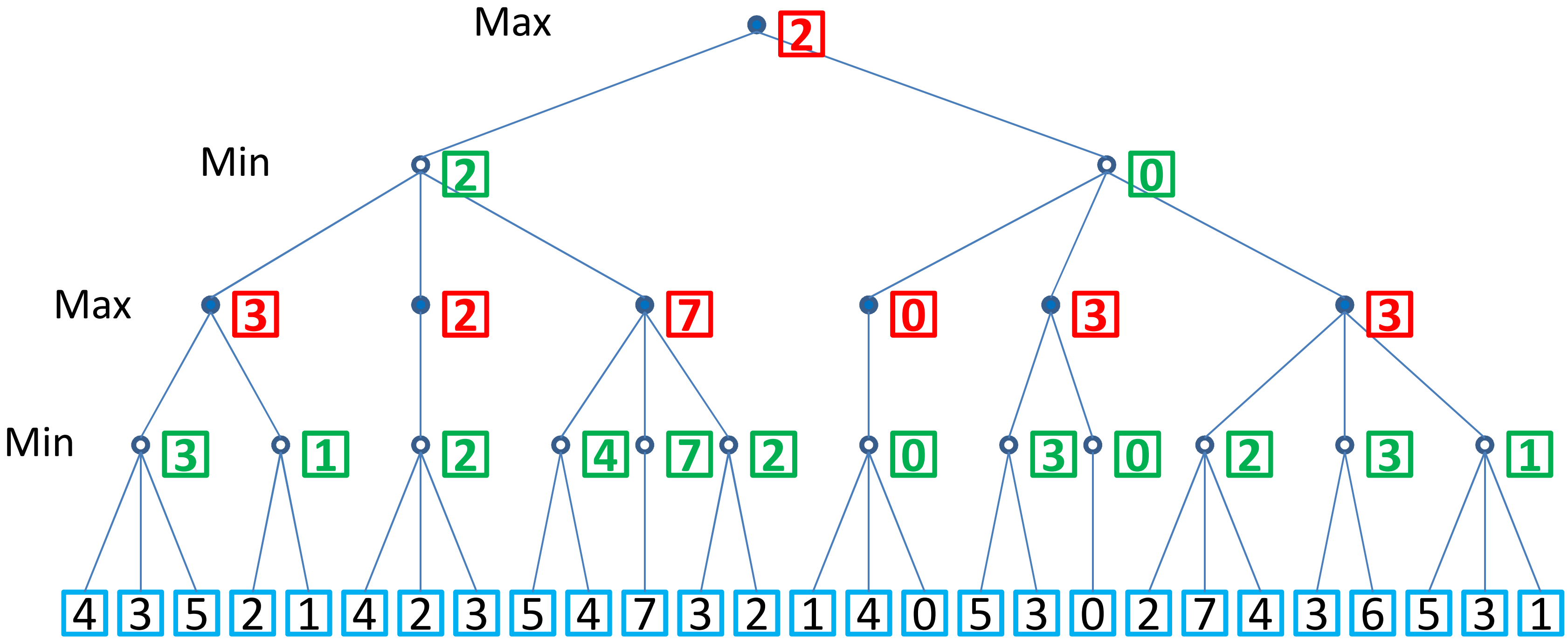


MiniMax & Constraint Processing: MiniMax Algorithm

OPTIMIZING $\alpha\beta$-PRUNING

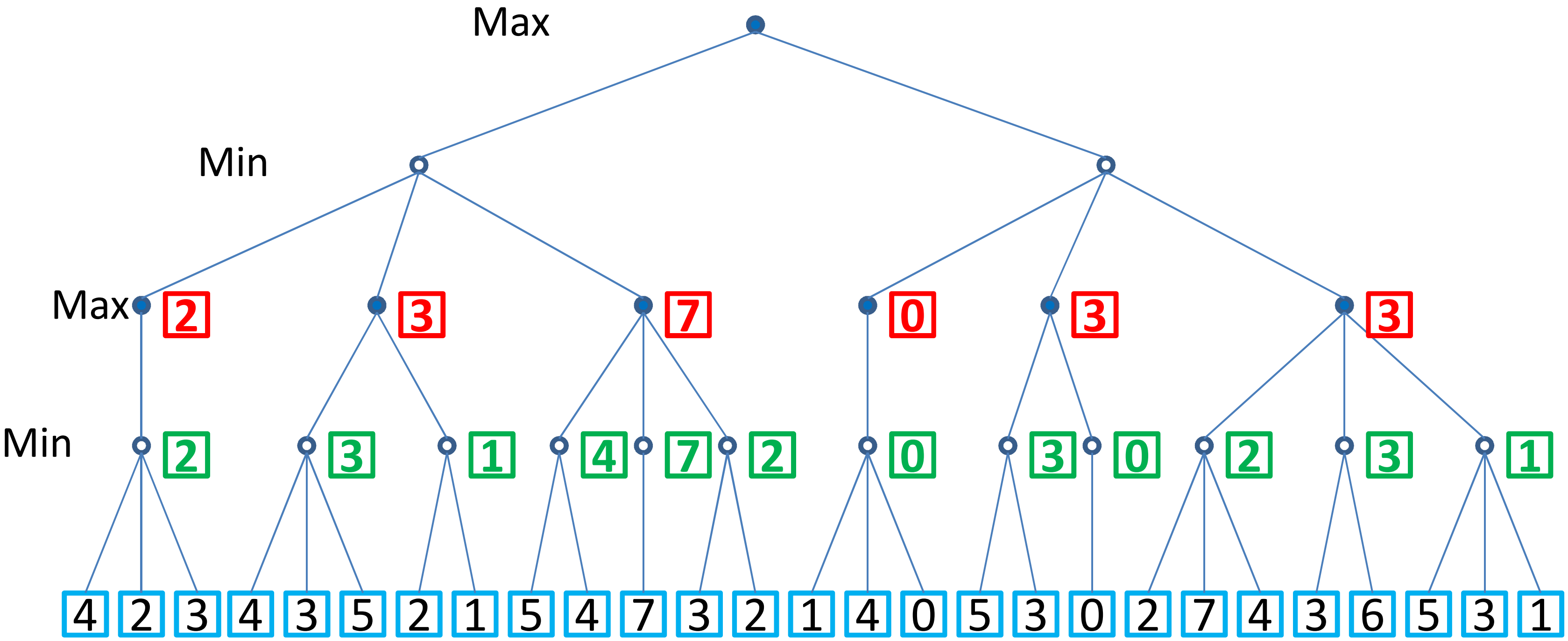
Optimizing $\alpha\beta$-Pruning

- **Best case:** Each layer best node left-to-right



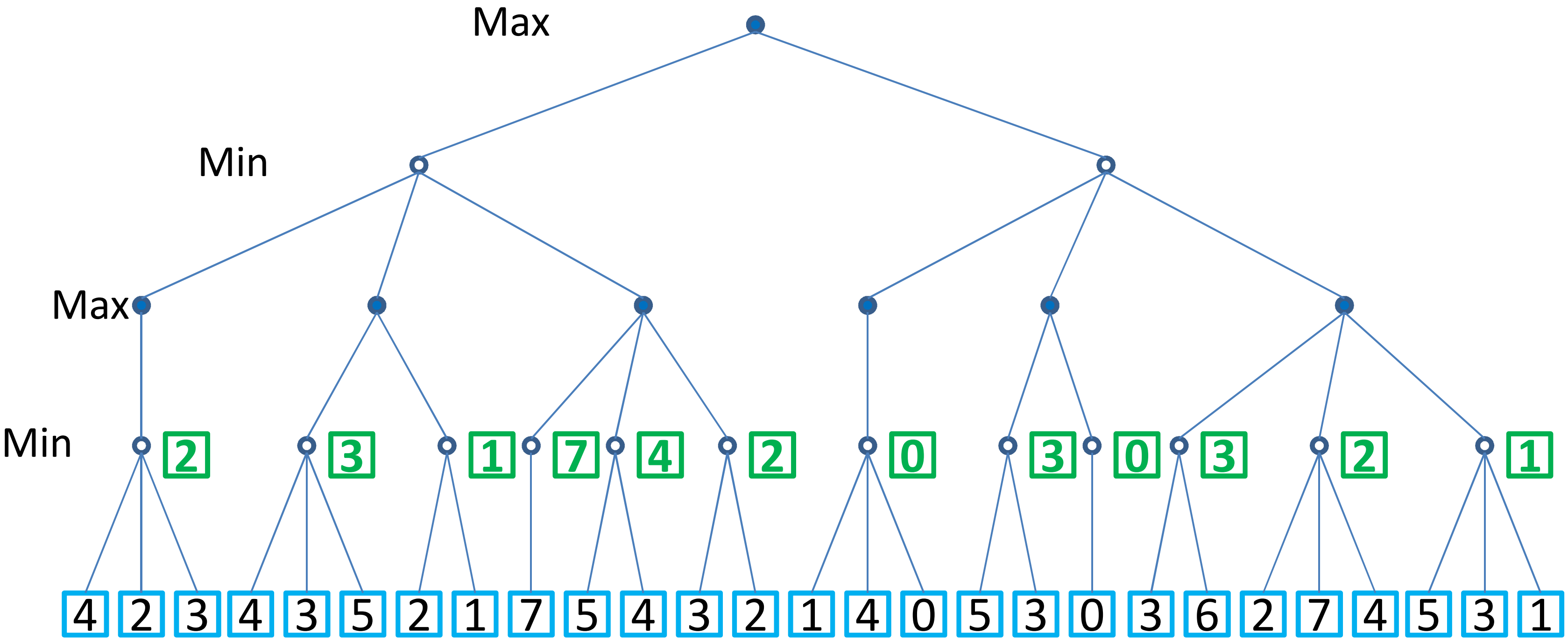
Optimizing $\alpha\beta$-Pruning

- **Best case:** Each layer best node left-to-right



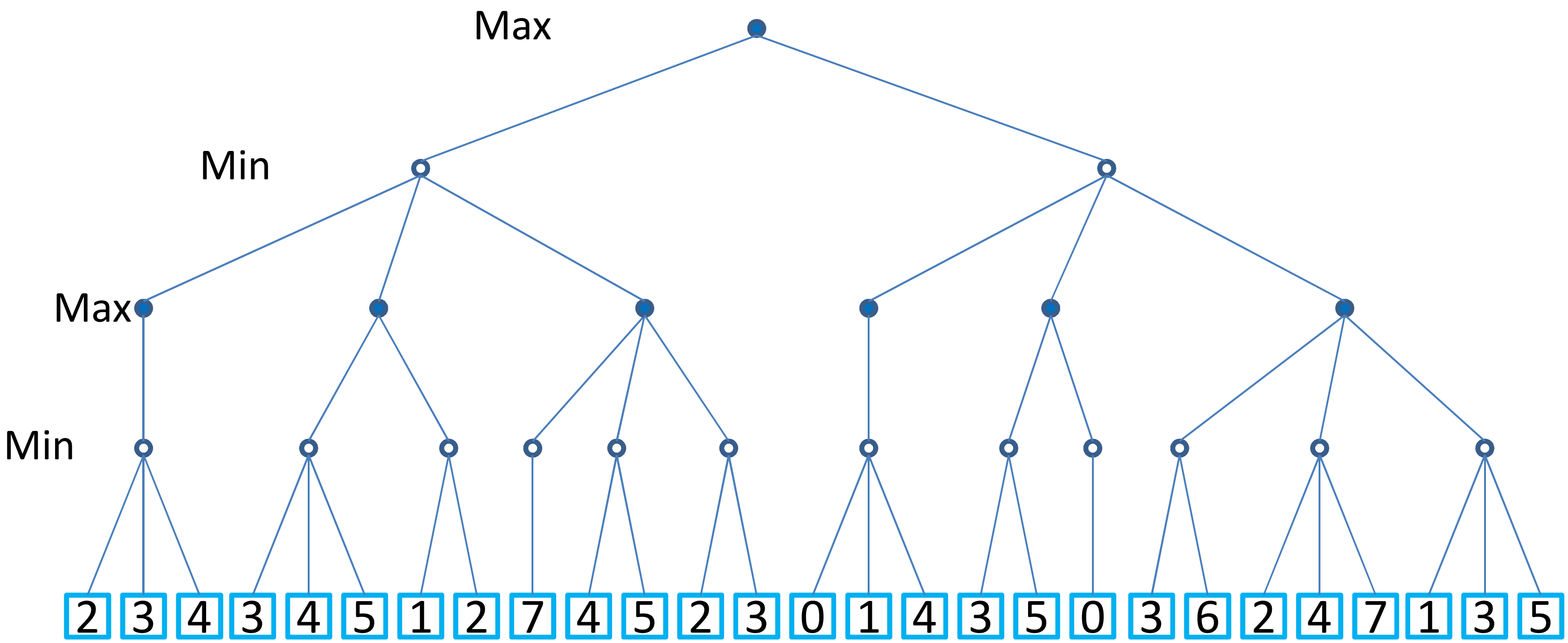
Optimizing $\alpha\beta$-Pruning

- **Best case:** Each layer best node left-to-right



Optimizing $\alpha\beta$-Pruning

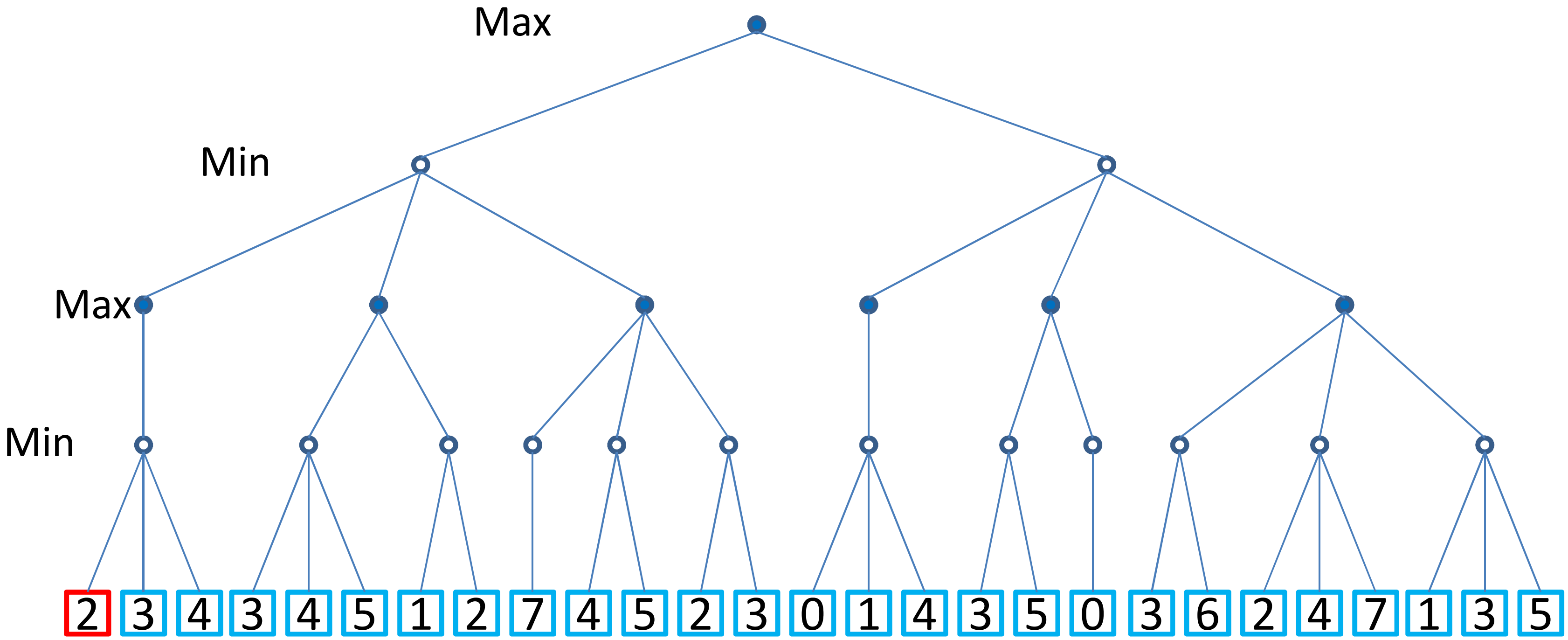
- **Best case:** Each layer best node left-to-right



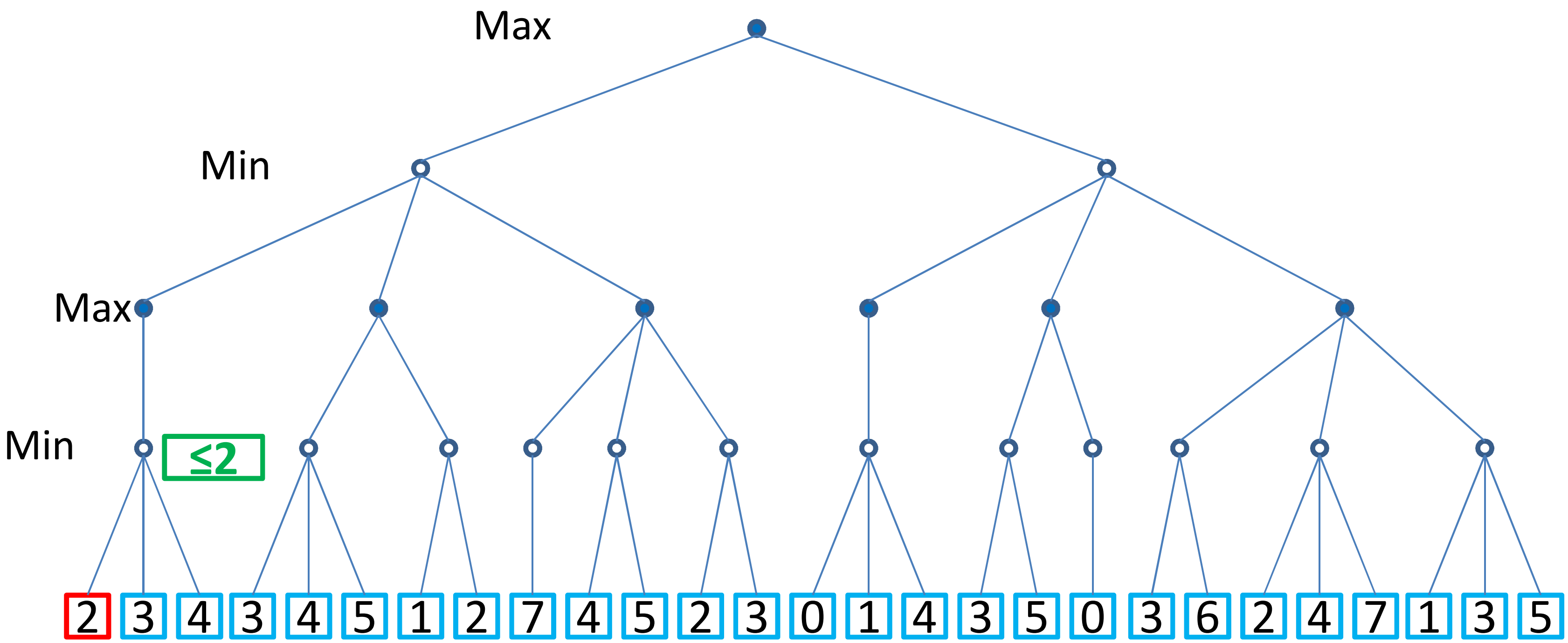
MiniMax & Constraint Processing: MiniMax Algorithm

MINIMAX WITH $\alpha\beta$-PRUNING

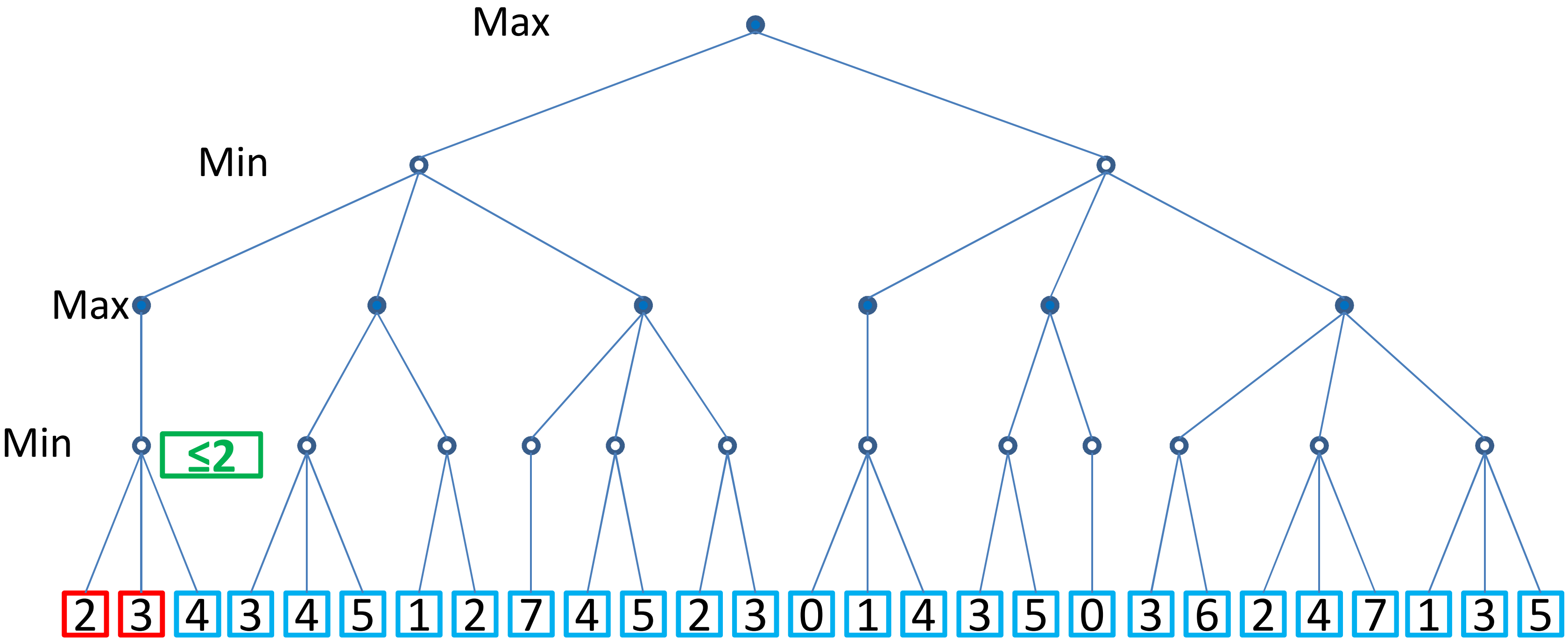
Minimax with $\alpha\beta$-Pruning



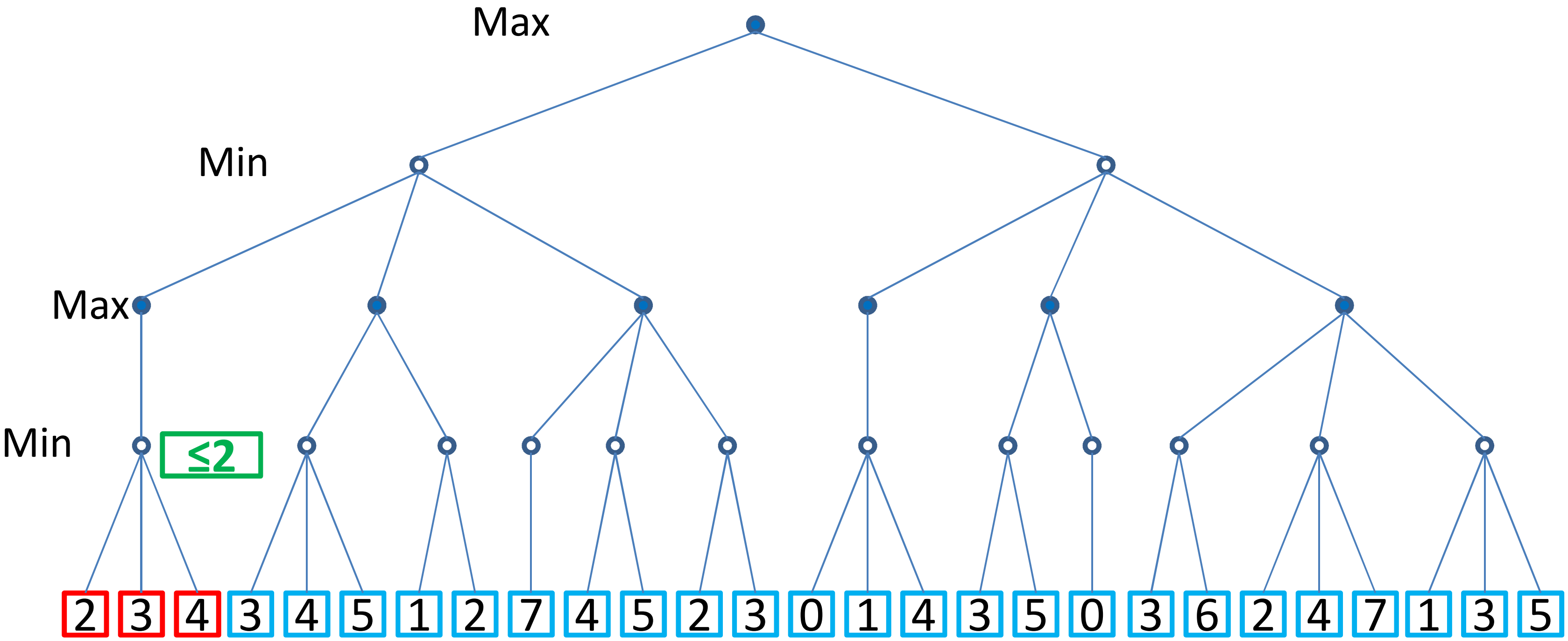
Minimax with $\alpha\beta$-Pruning



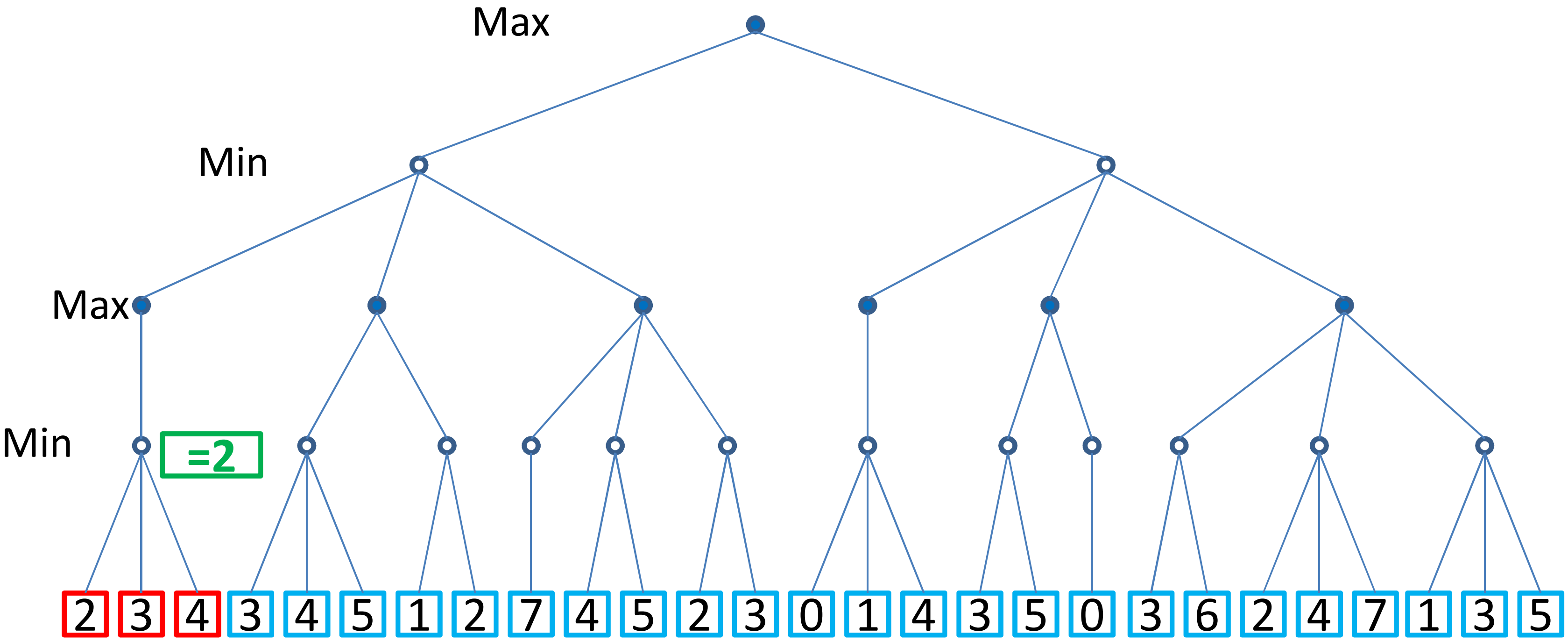
Minimax with $\alpha\beta$-Pruning



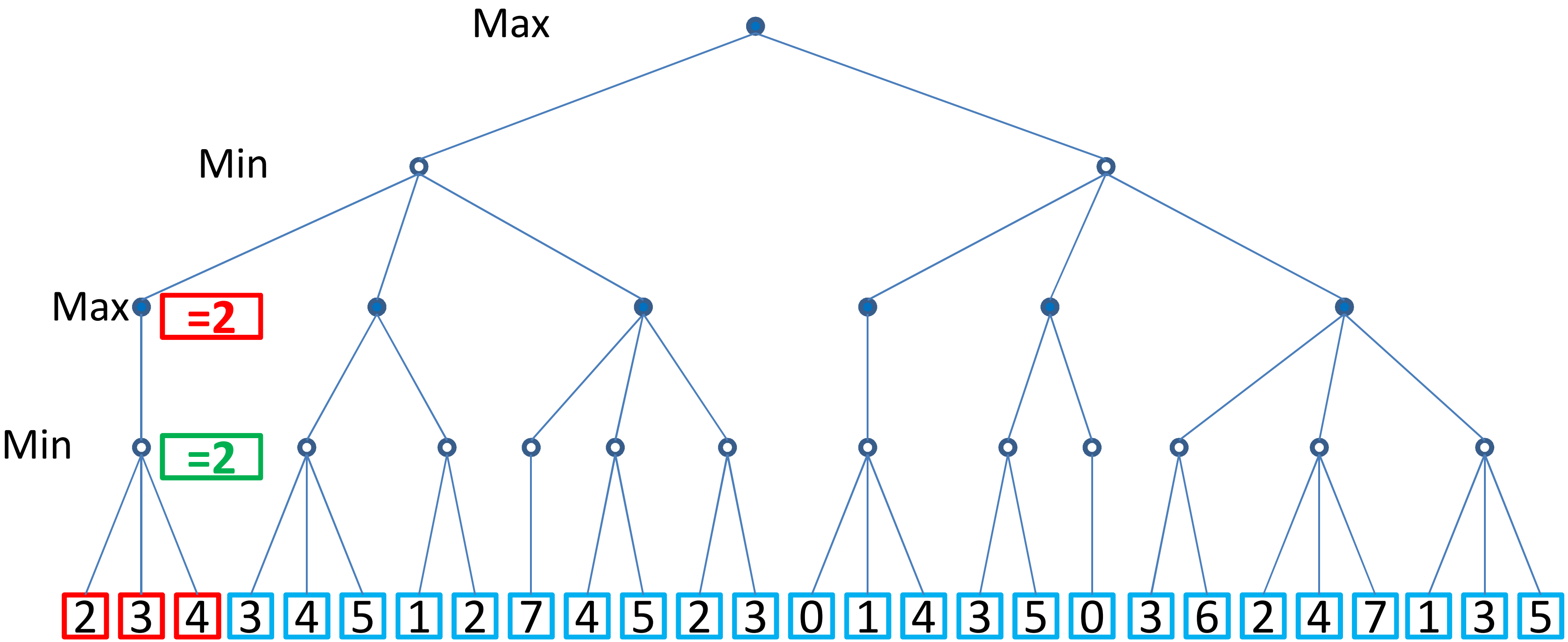
Minimax with $\alpha\beta$-Pruning



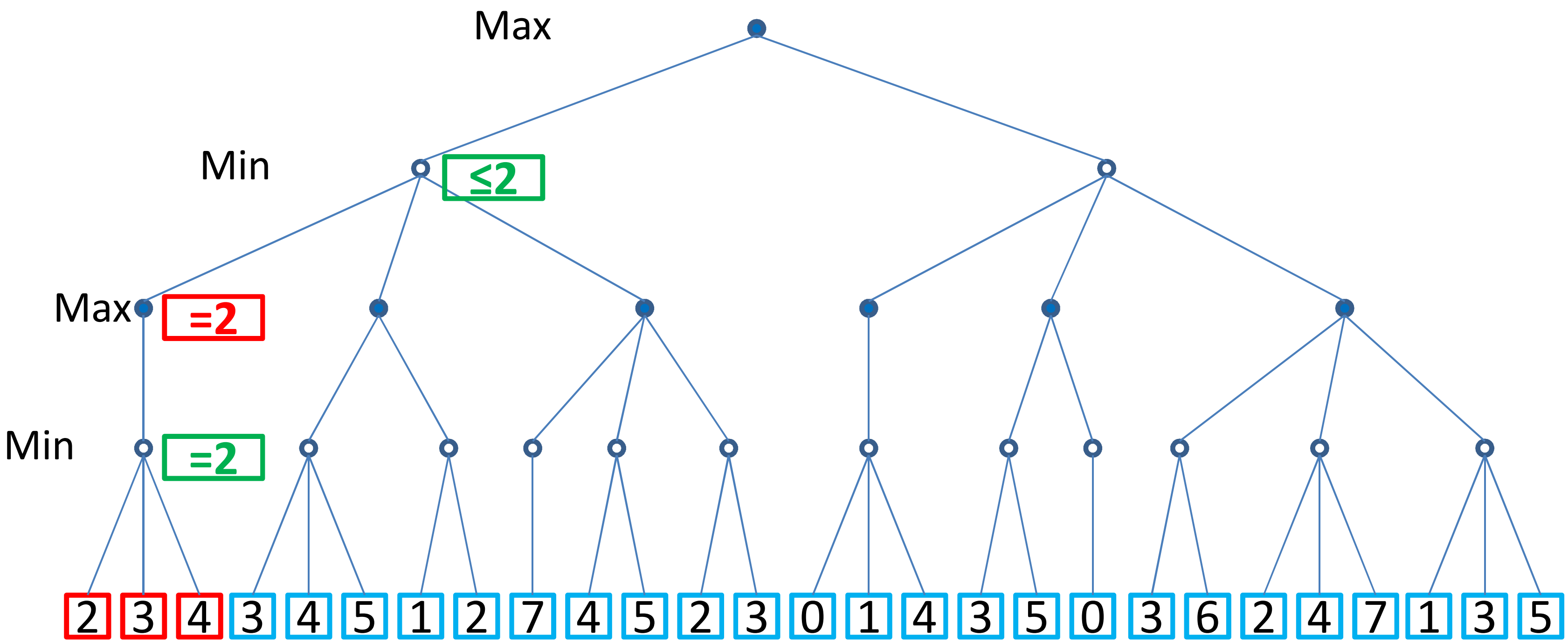
Minimax with $\alpha\beta$-Pruning



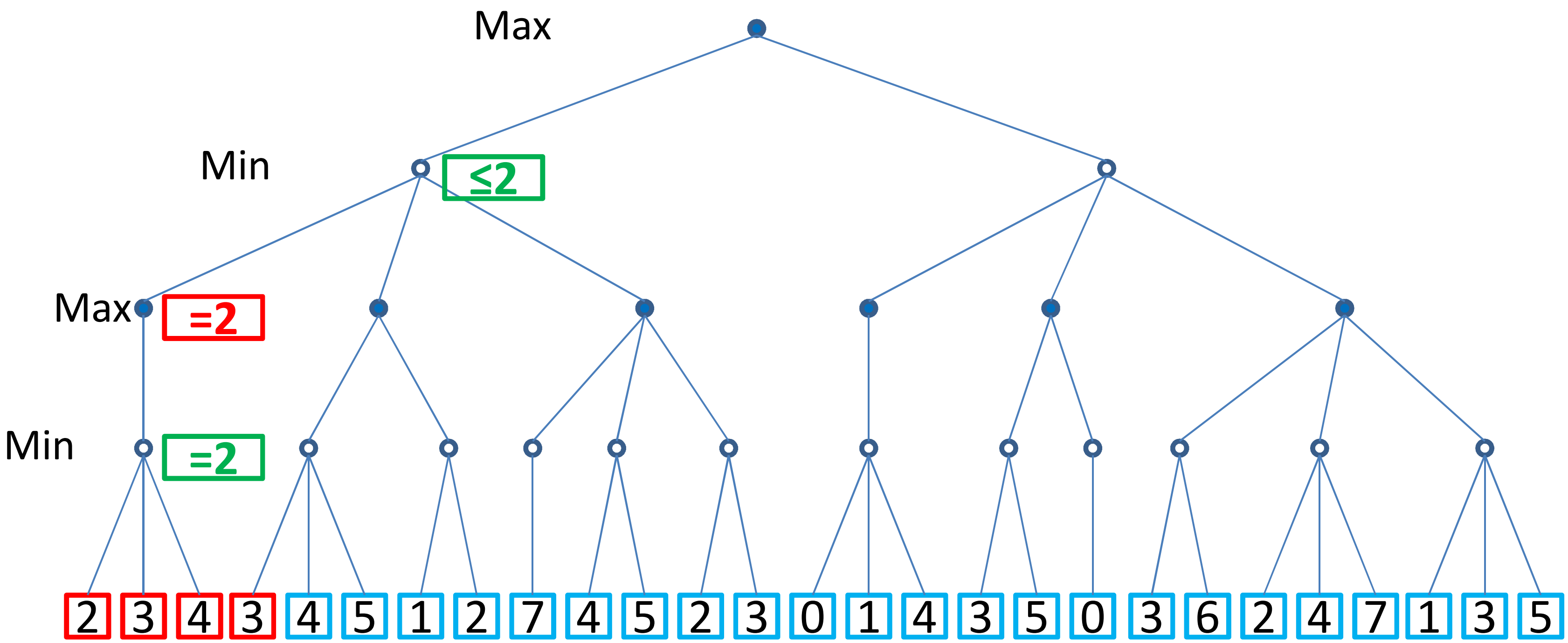
Minimax with $\alpha\beta$-Pruning



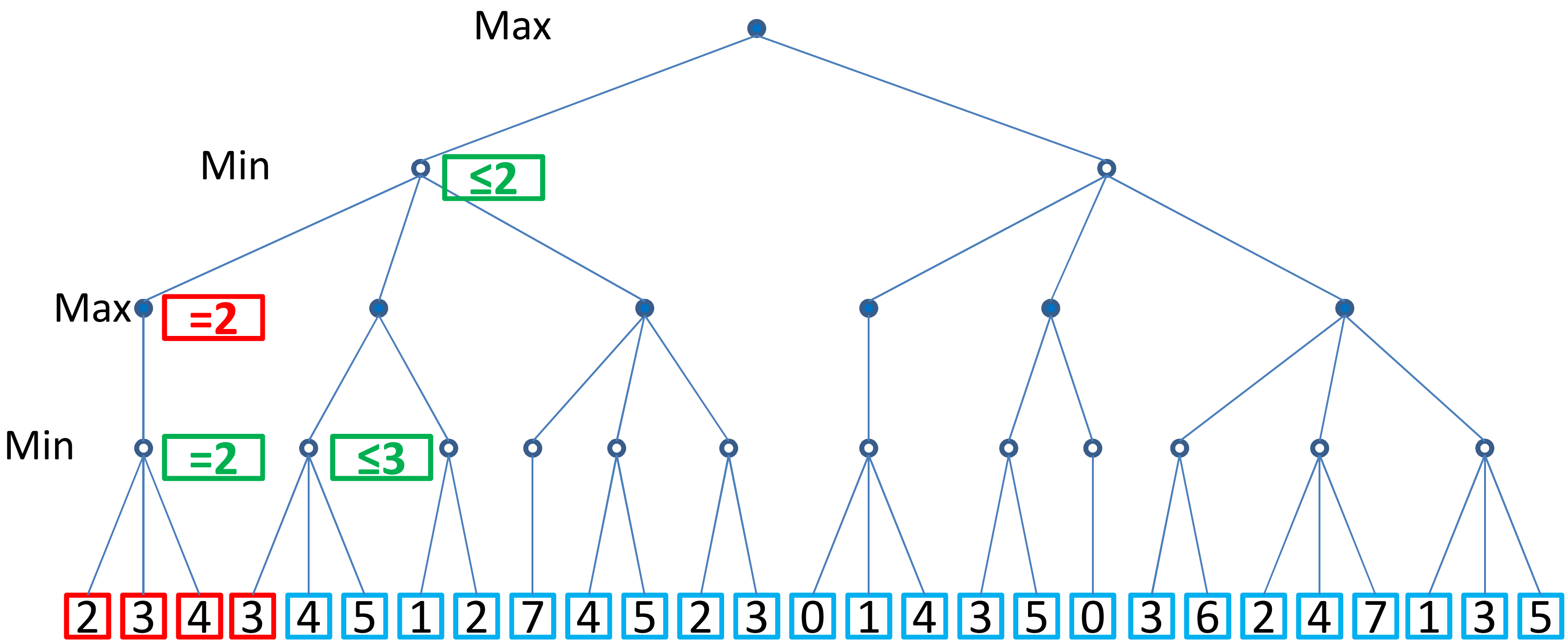
Minimax with $\alpha\beta$-Pruning



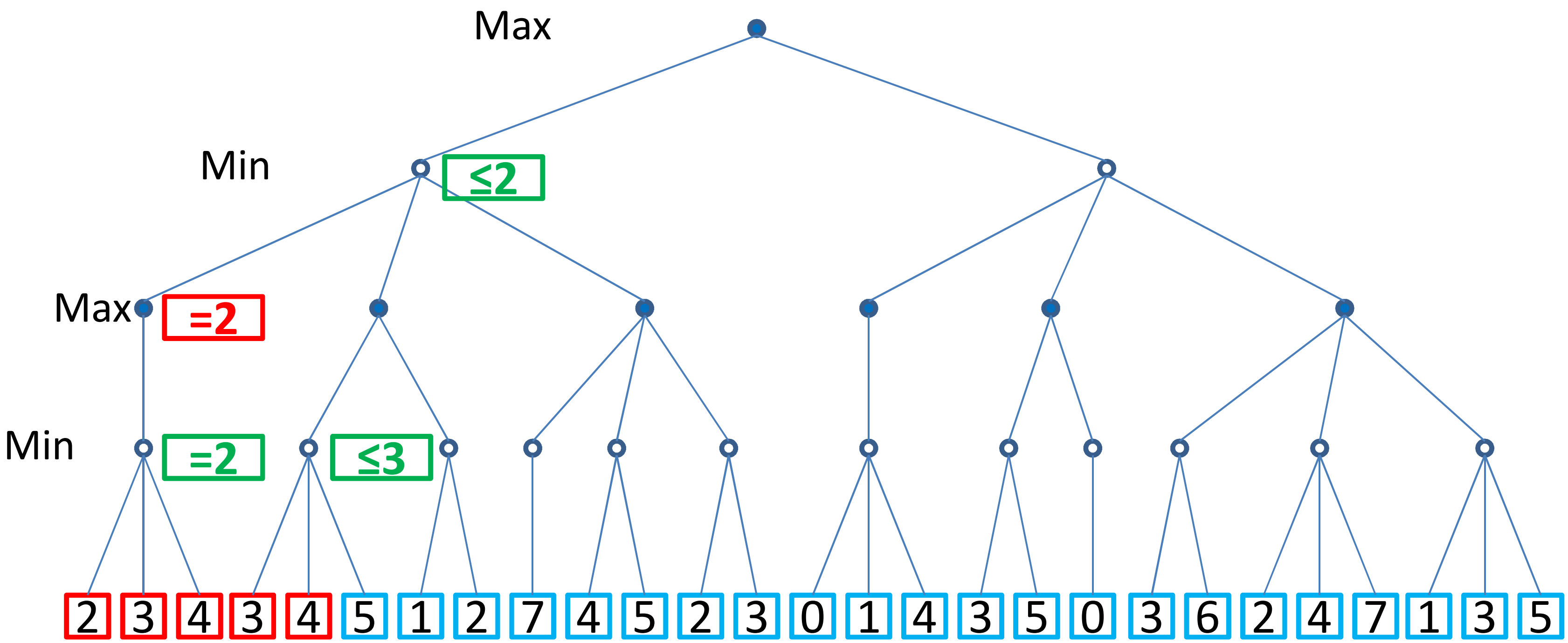
Minimax with $\alpha\beta$-Pruning



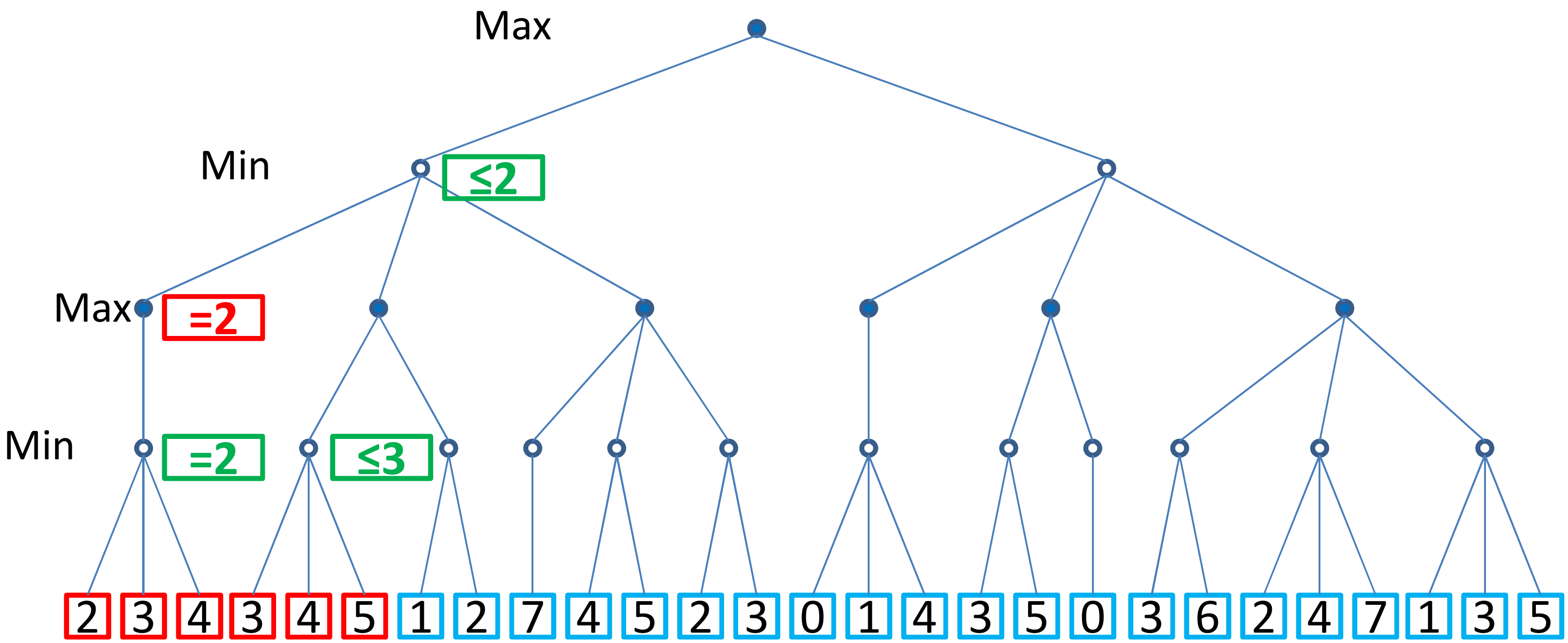
Minimax with $\alpha\beta$-Pruning



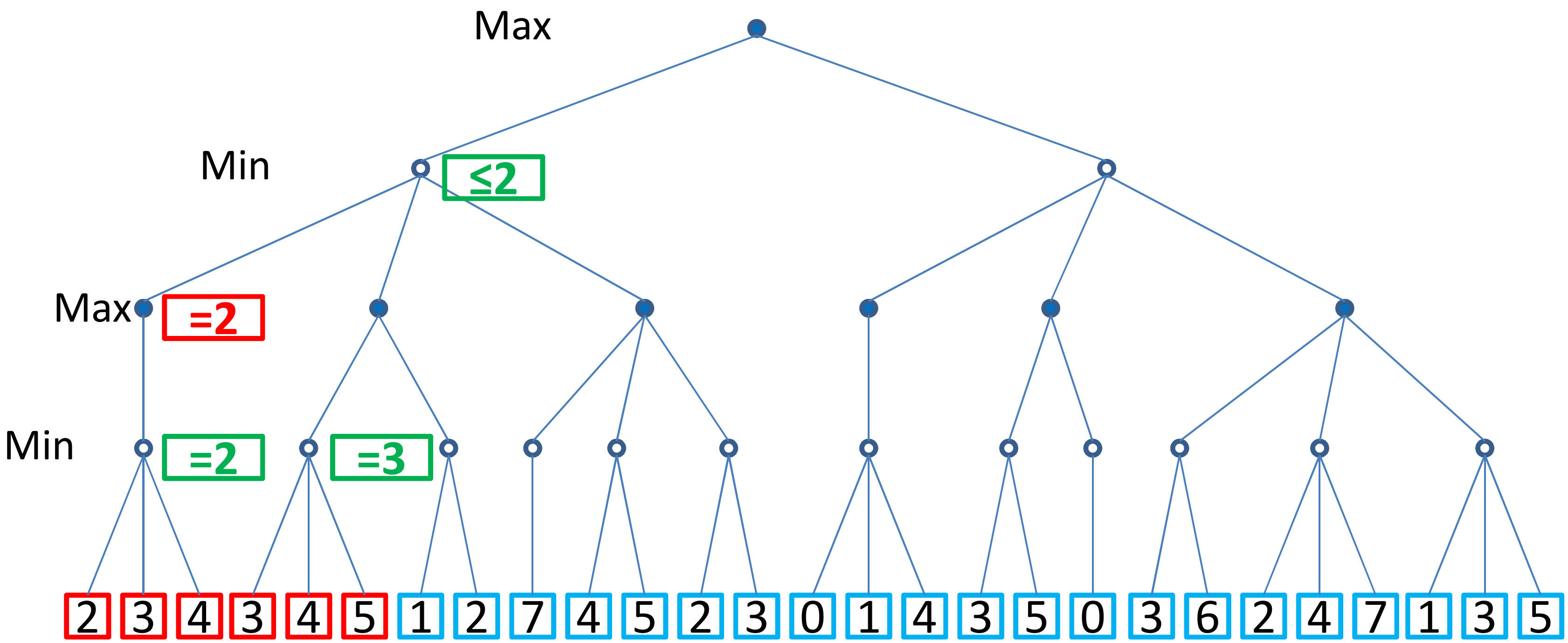
Minimax with $\alpha\beta$-Pruning



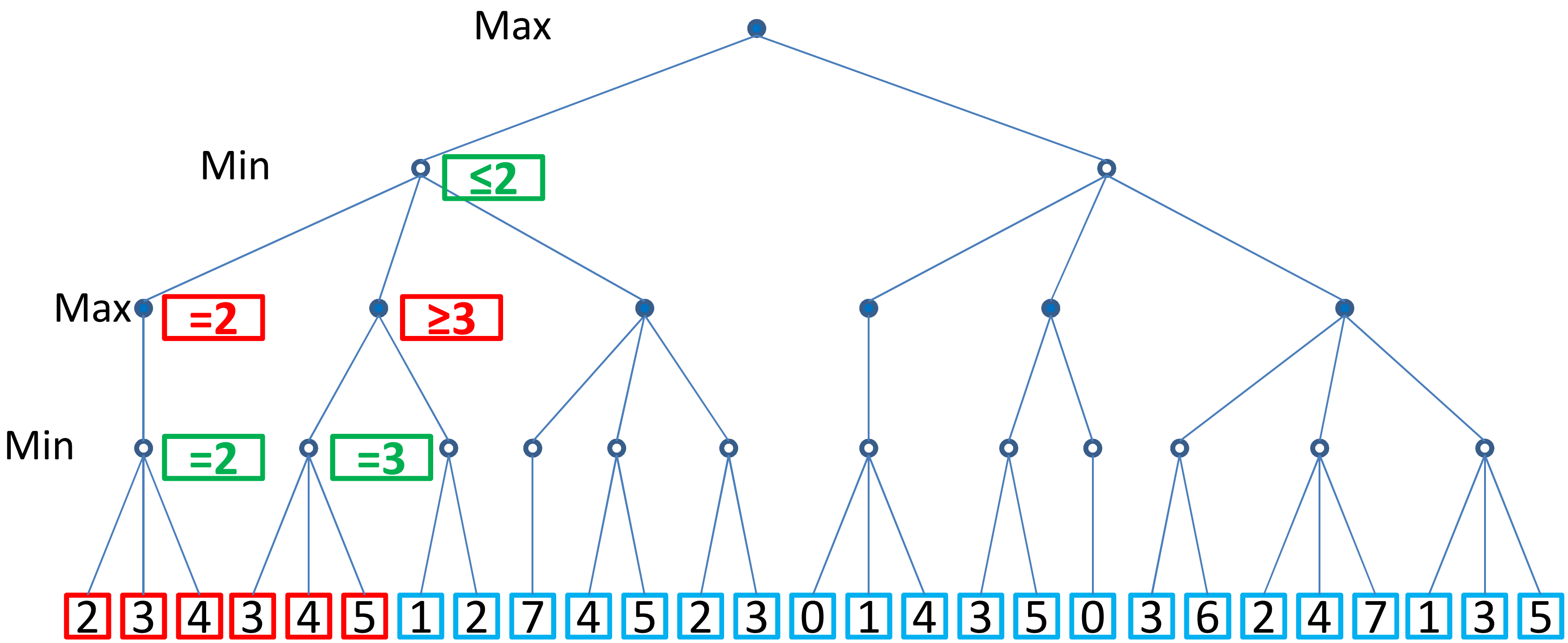
Minimax with $\alpha\beta$-Pruning



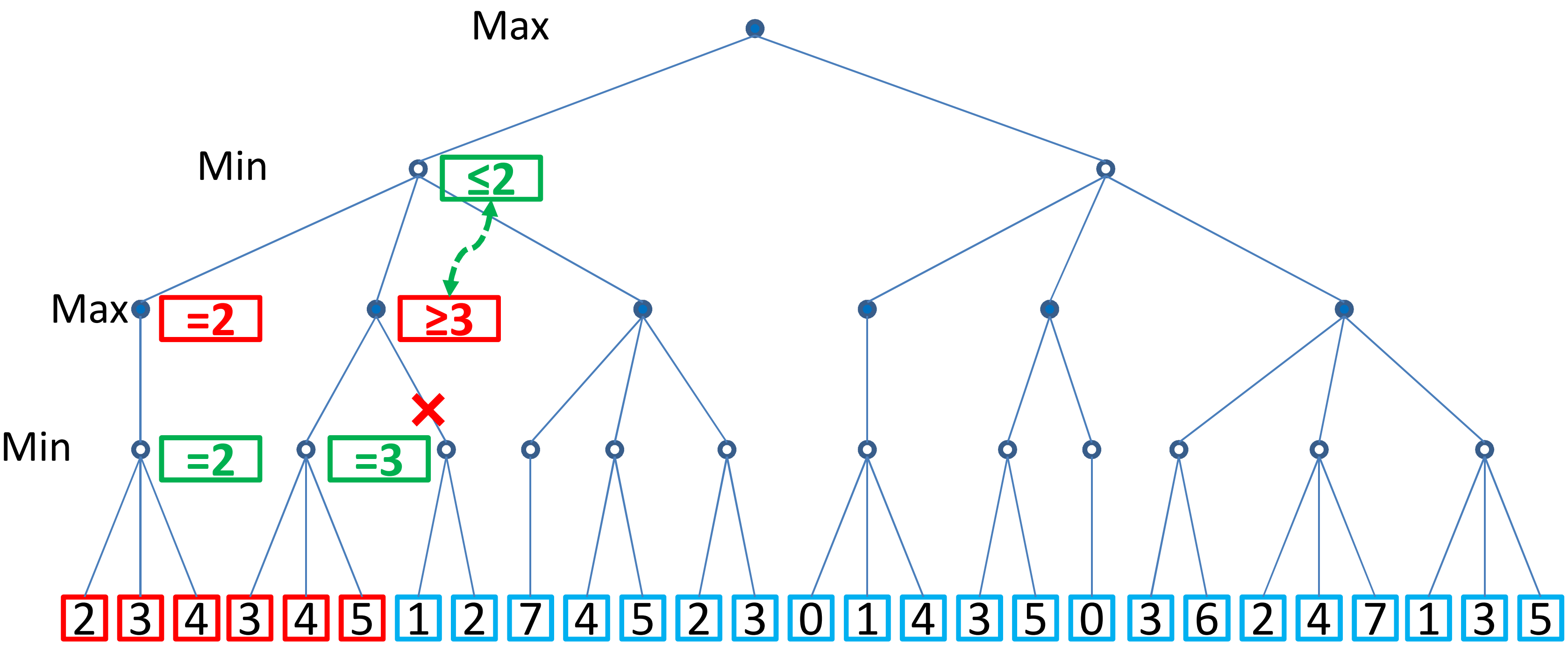
Minimax with $\alpha\beta$-Pruning



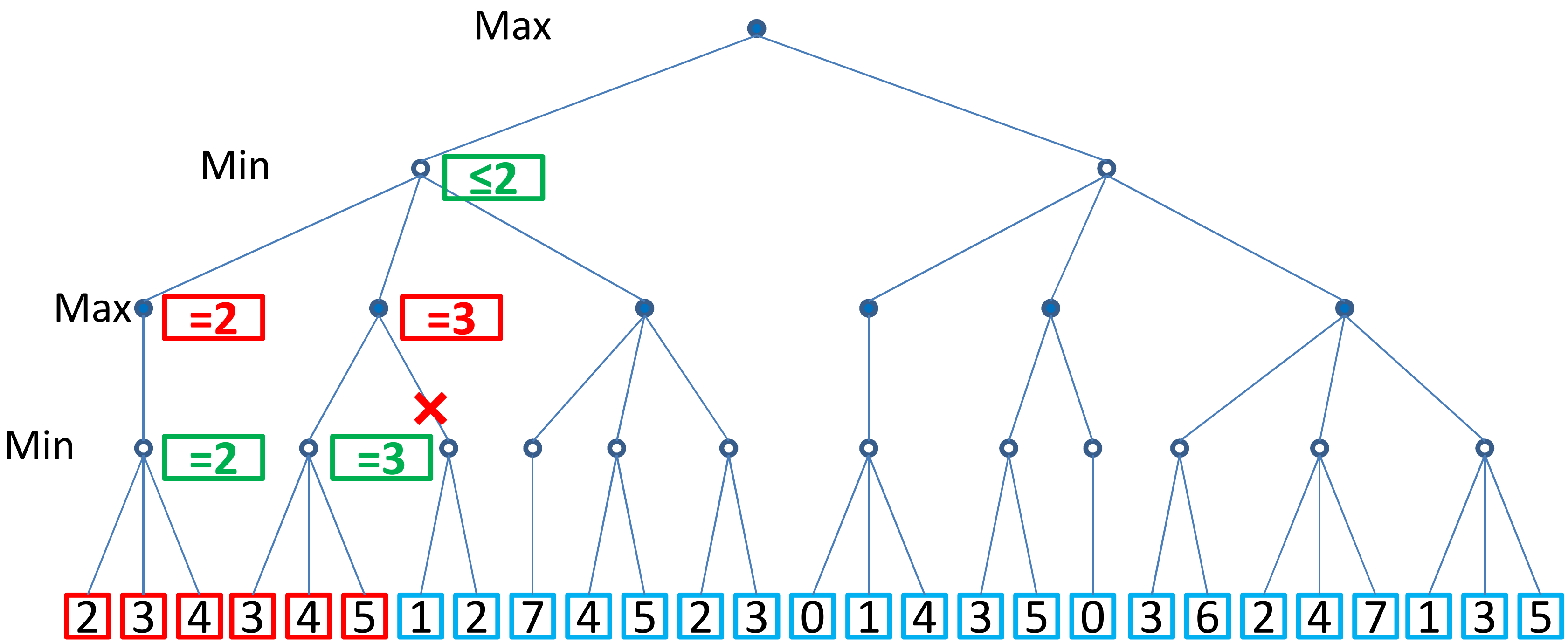
Minimax with $\alpha\beta$-Pruning



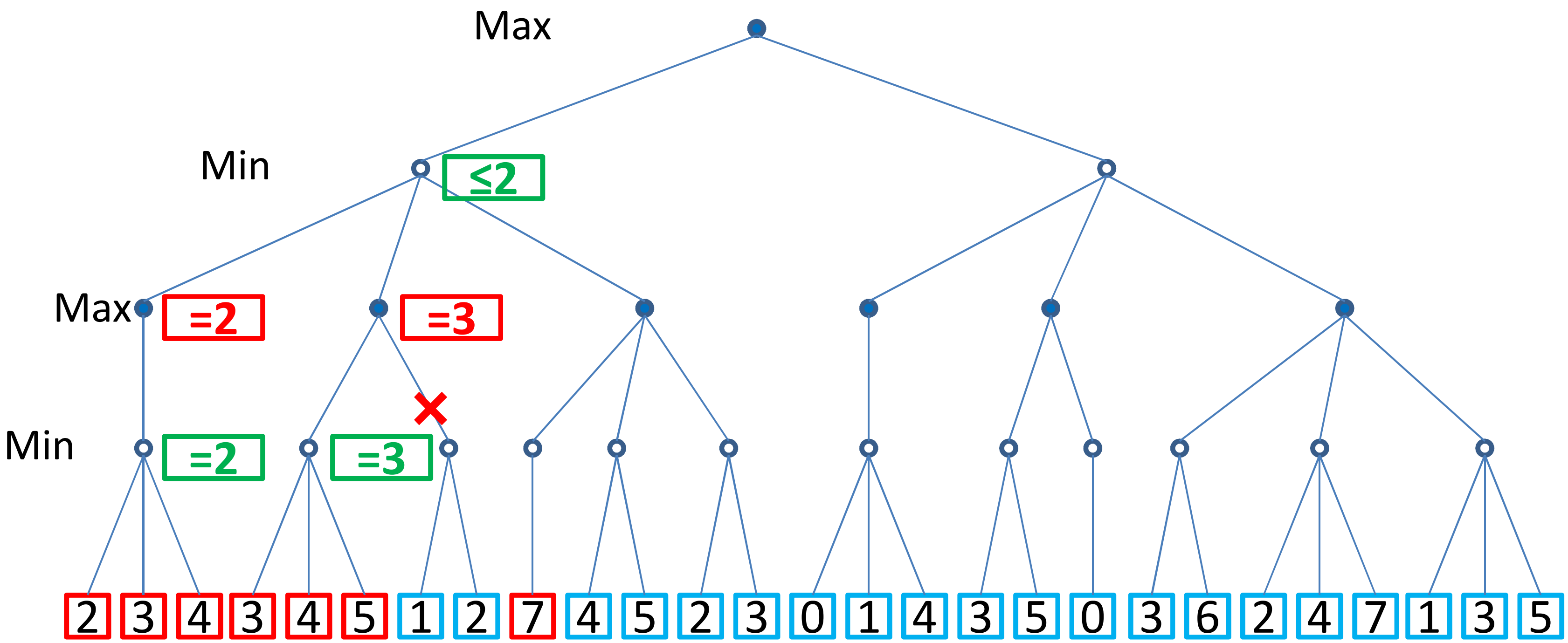
Minimax with $\alpha\beta$-Pruning



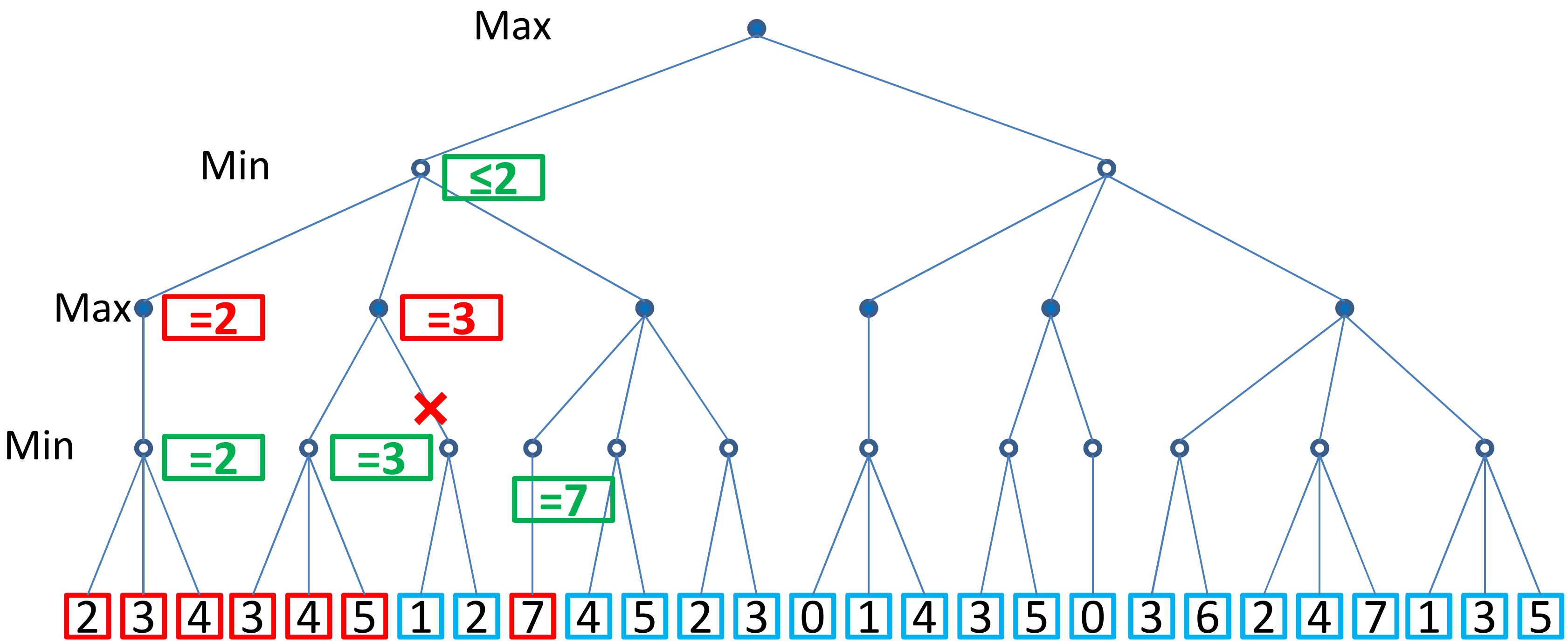
Minimax with $\alpha\beta$-Pruning



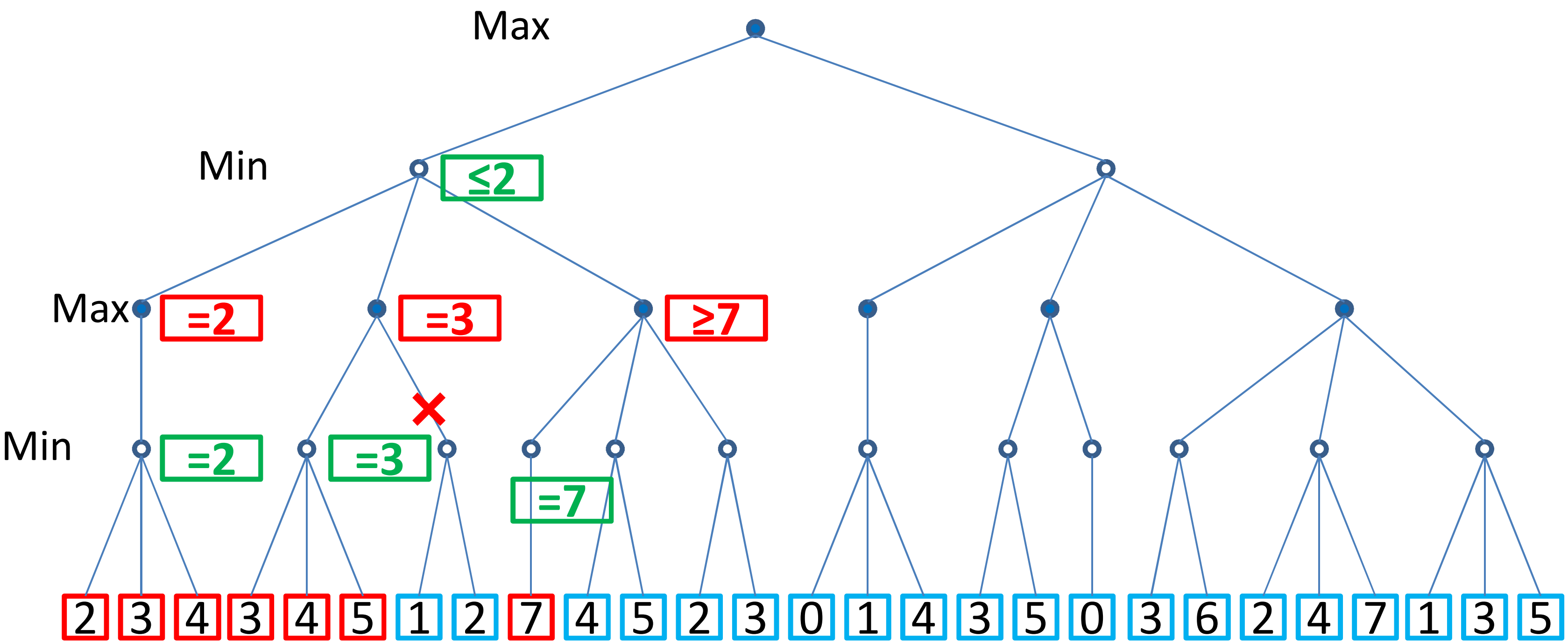
Minimax with $\alpha\beta$-Pruning



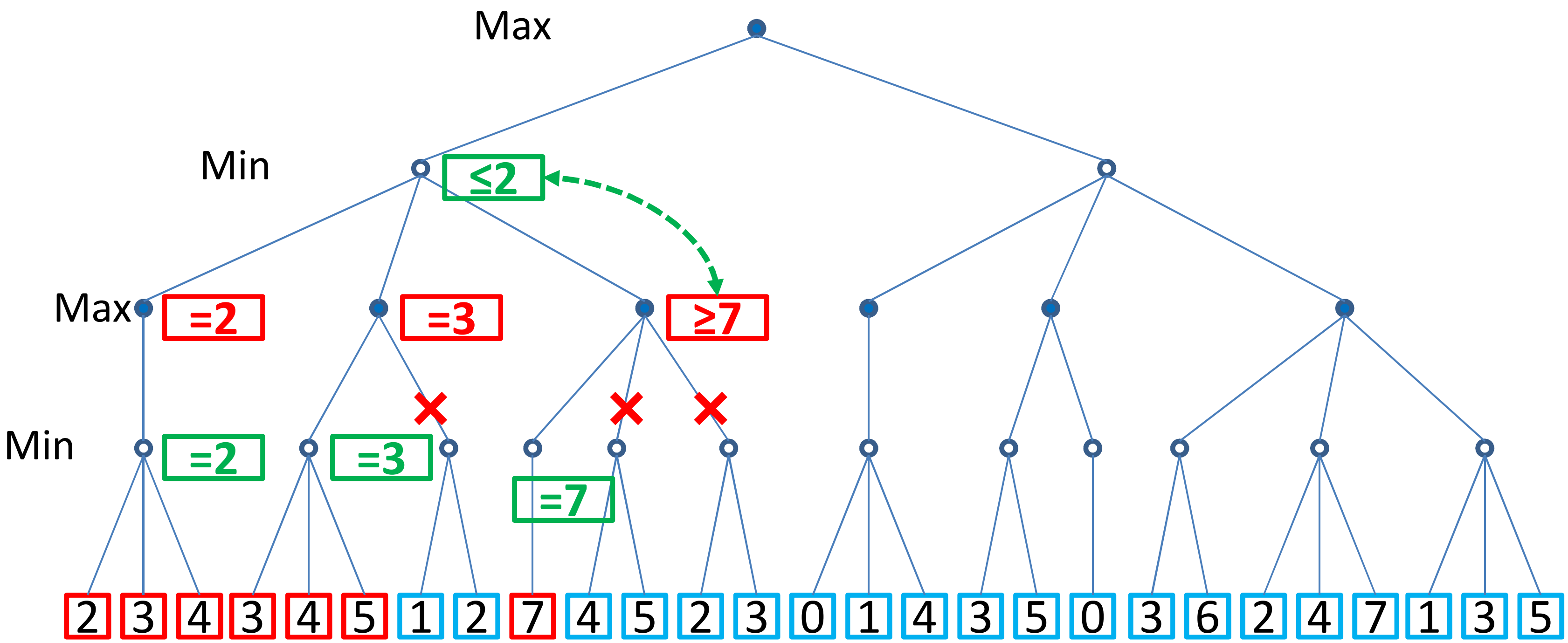
Minimax with $\alpha\beta$-Pruning



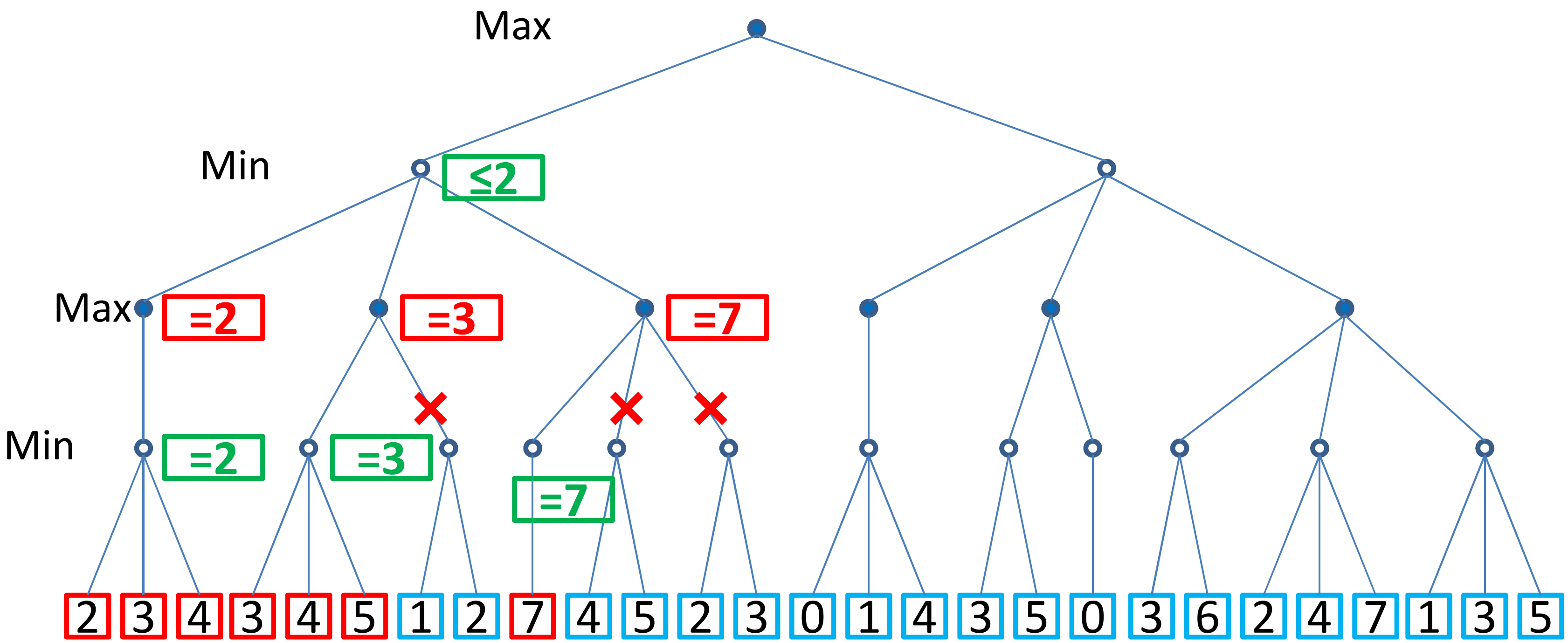
Minimax with $\alpha\beta$-Pruning



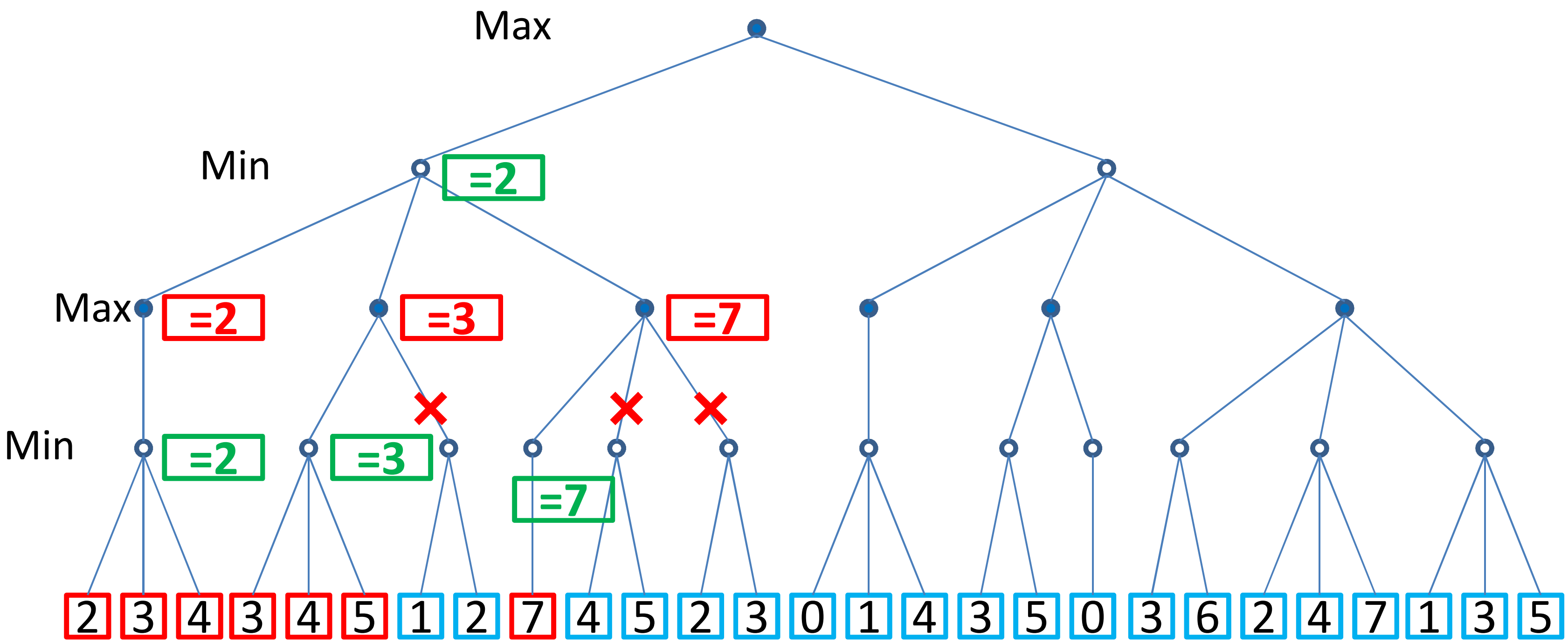
Minimax with $\alpha\beta$-Pruning



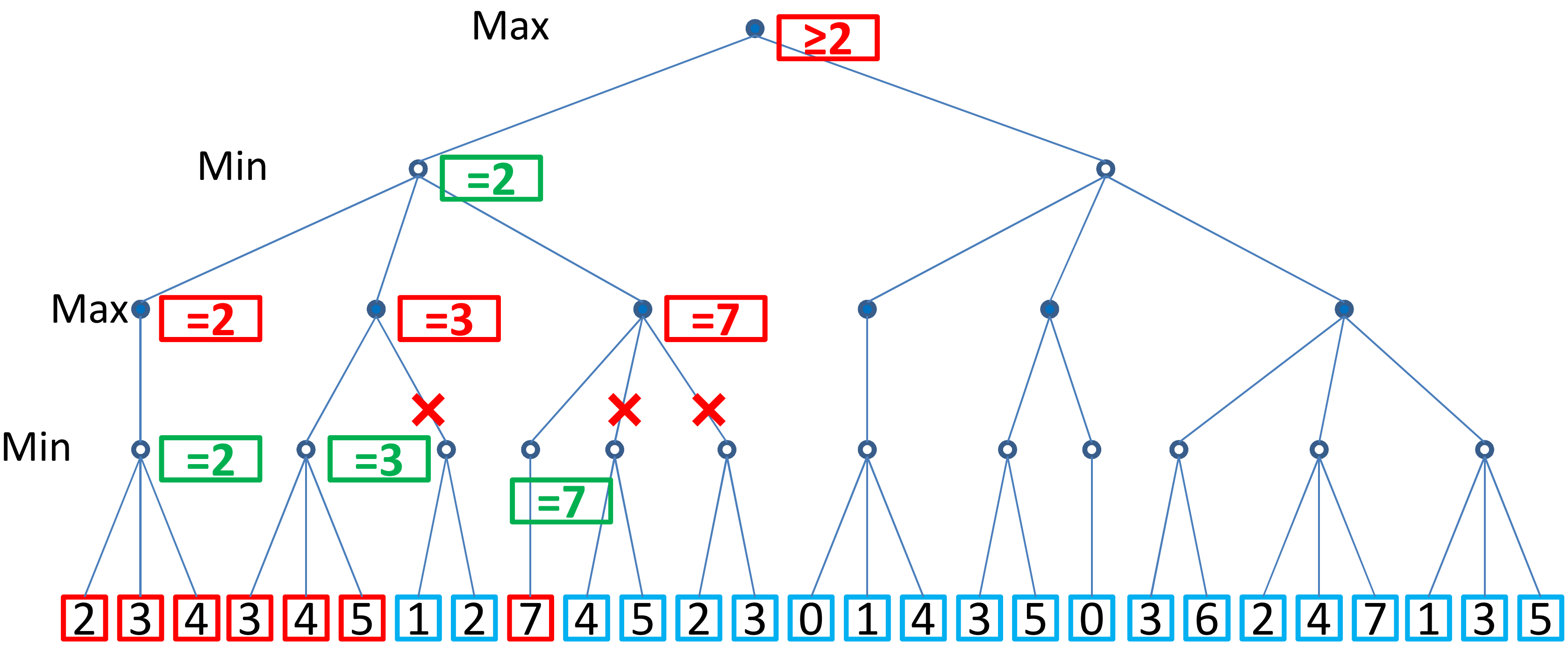
Minimax with $\alpha\beta$-Pruning



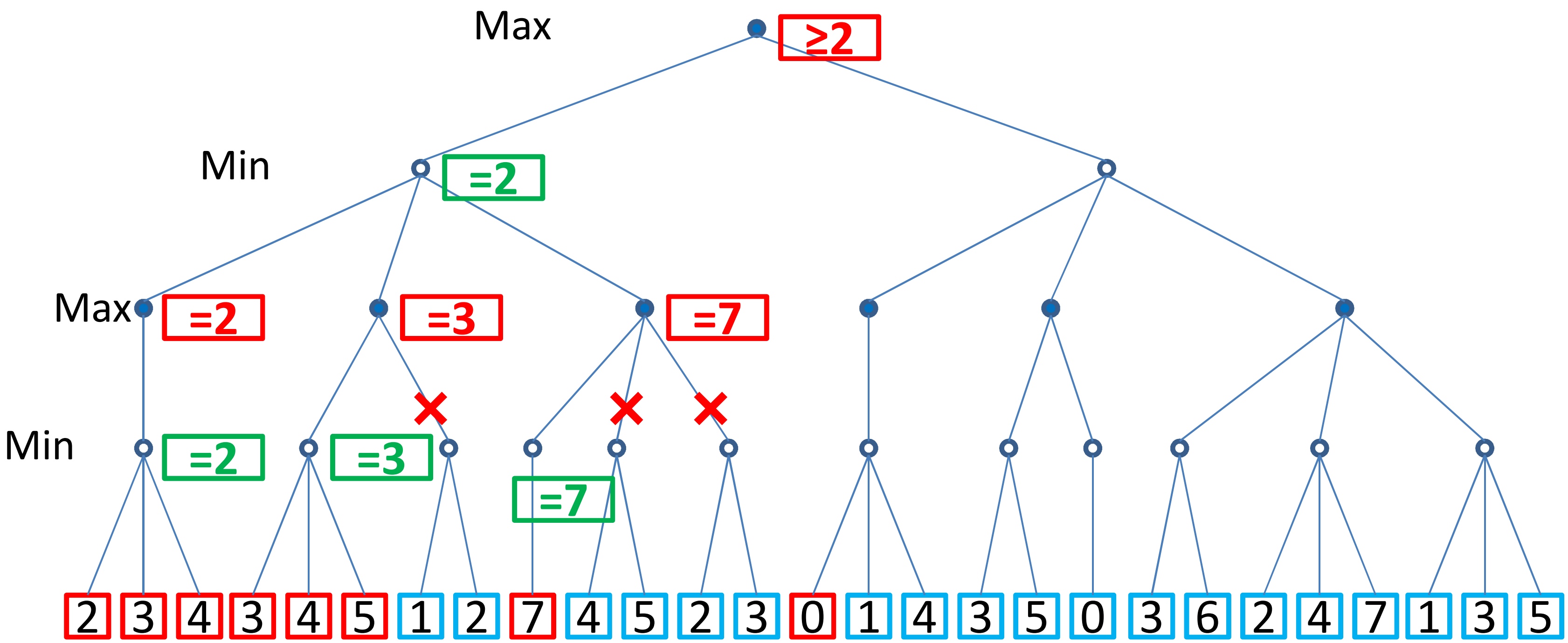
Minimax with $\alpha\beta$-Pruning



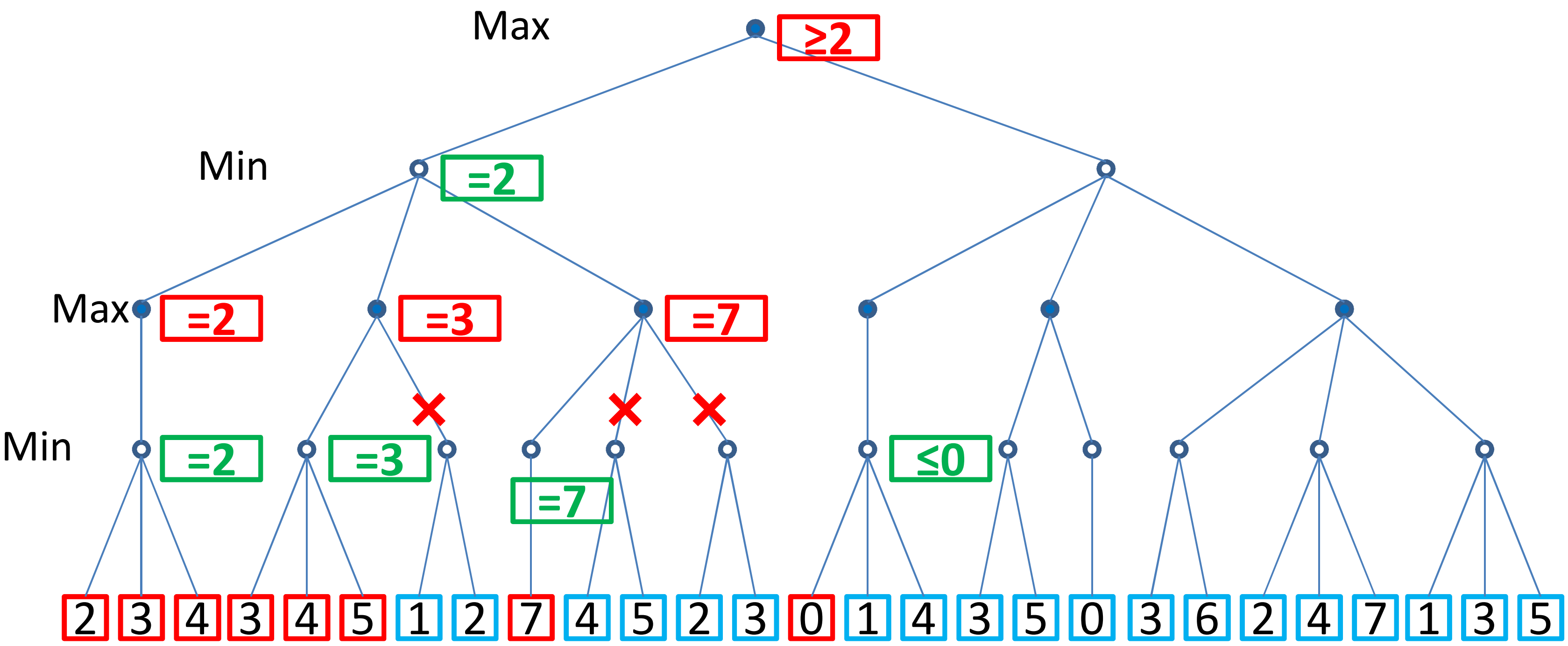
Minimax with $\alpha\beta$-Pruning



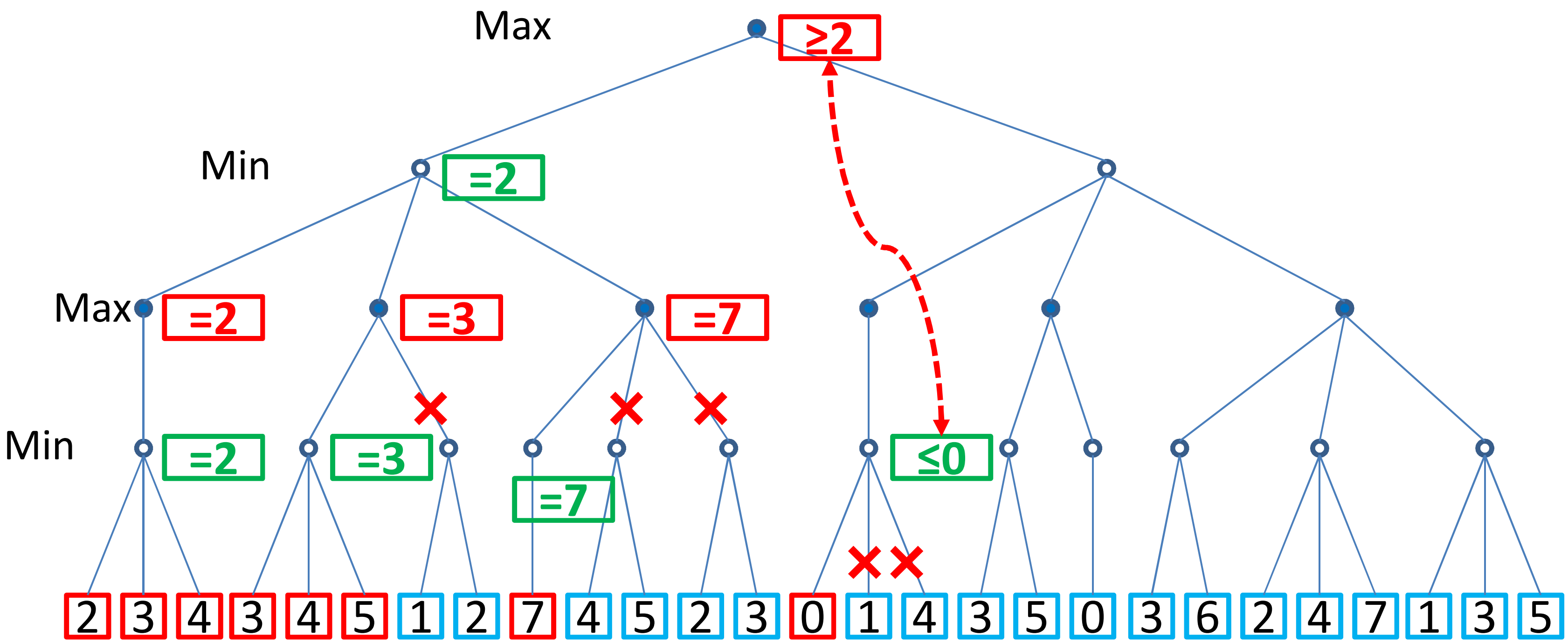
Minimax with $\alpha\beta$-Pruning



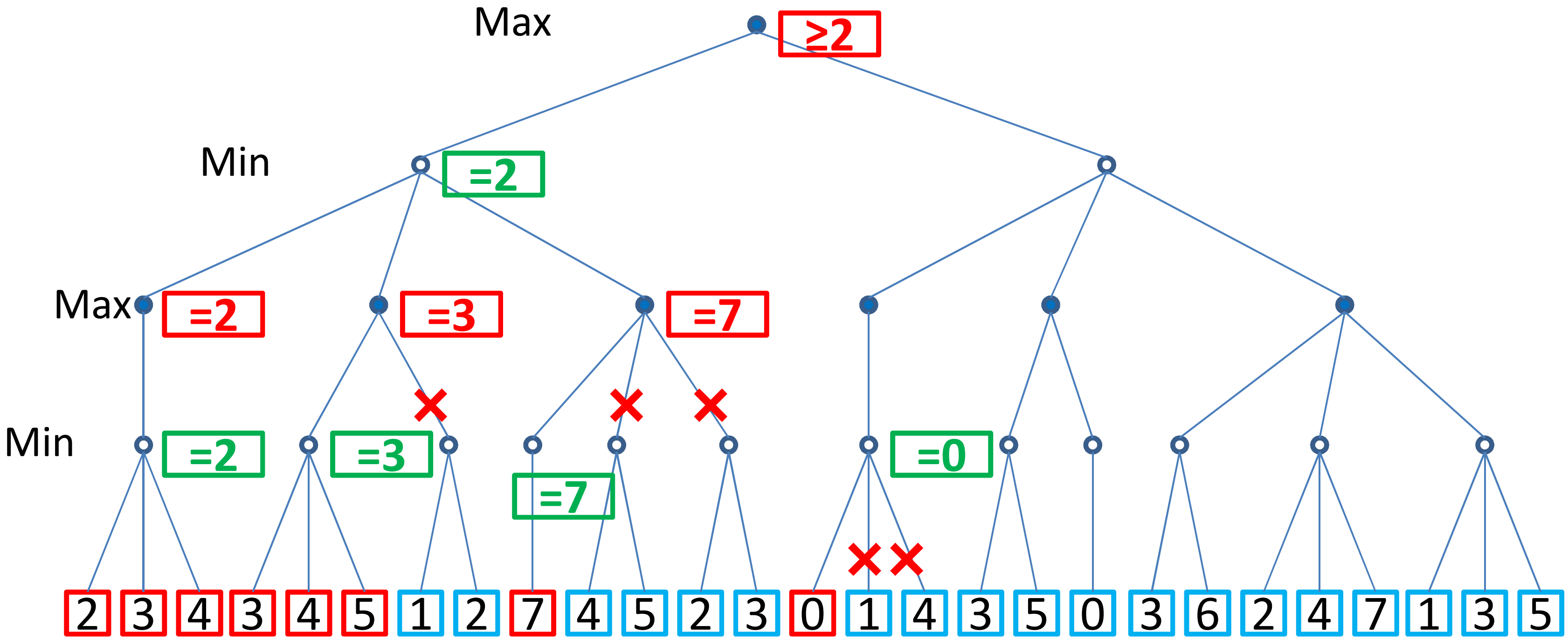
Minimax with $\alpha\beta$-Pruning



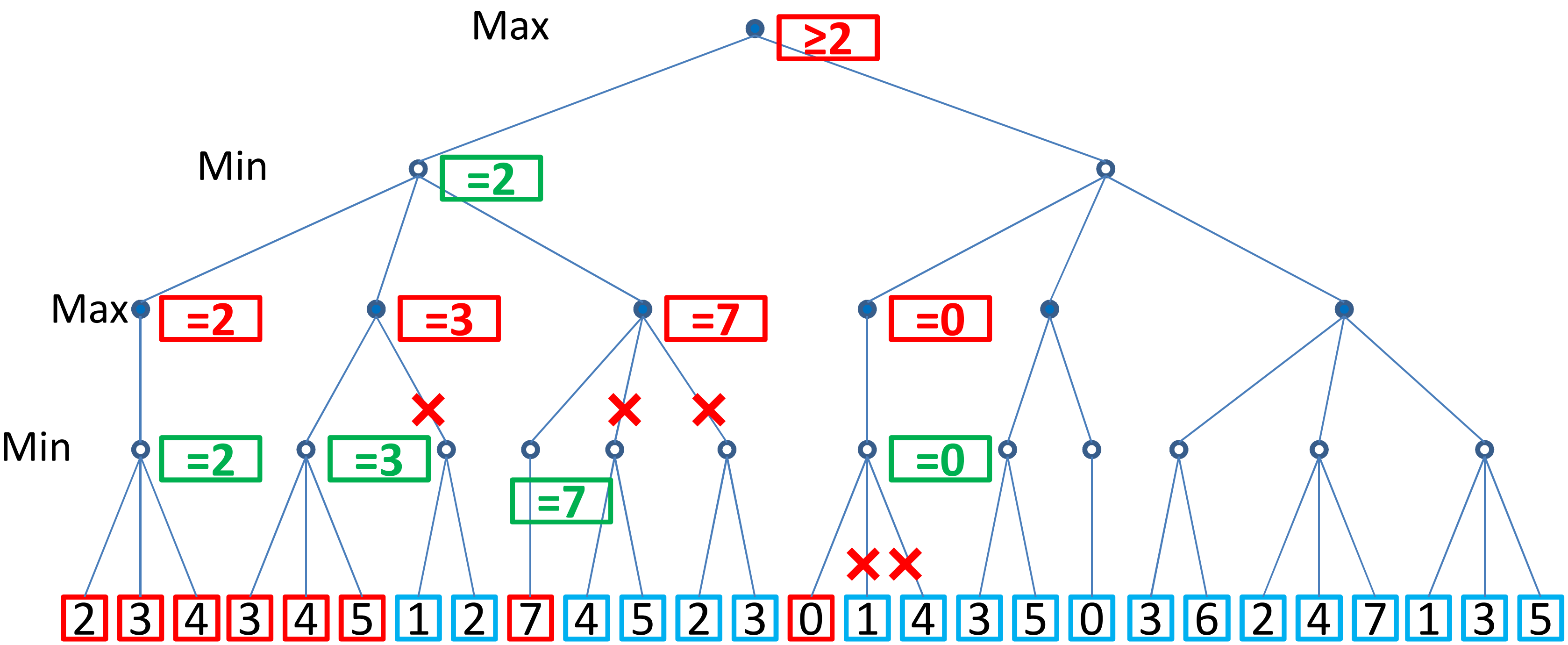
Minimax with $\alpha\beta$-Pruning



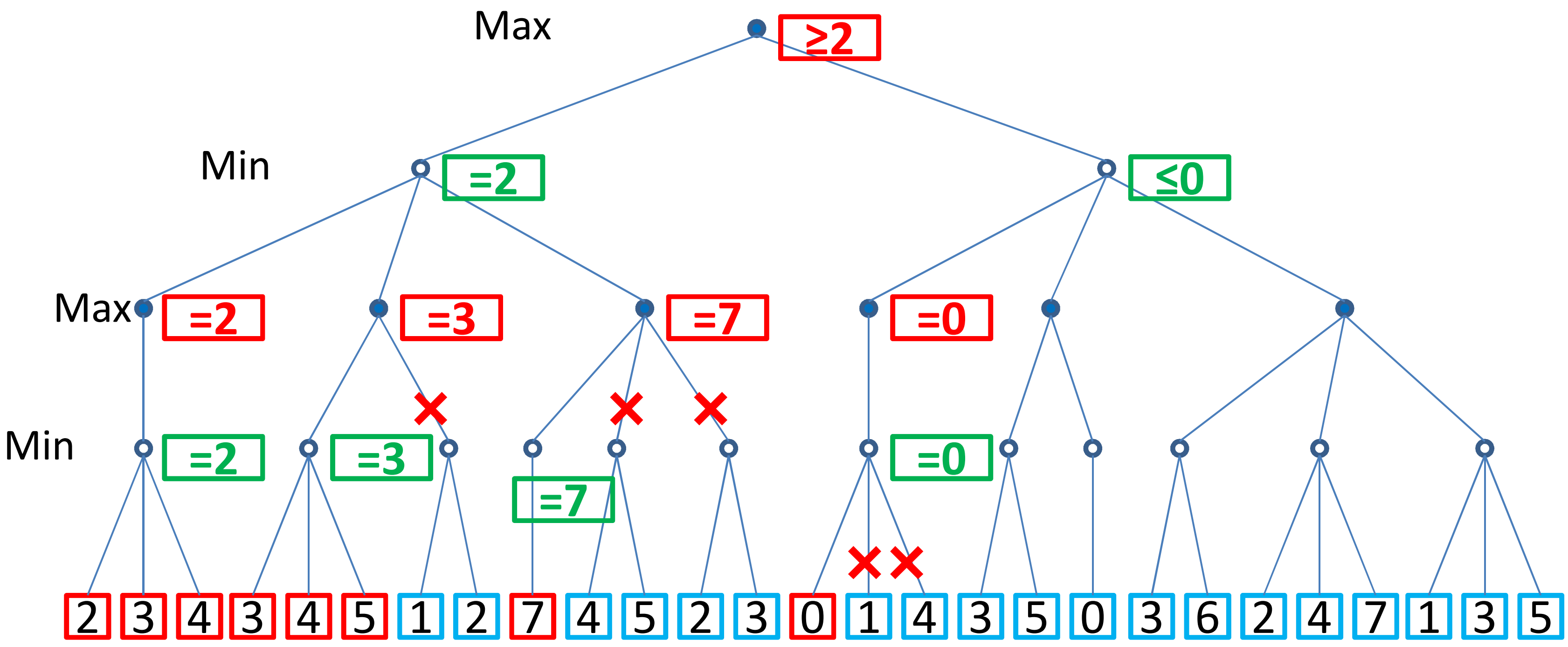
Minimax with $\alpha\beta$-Pruning



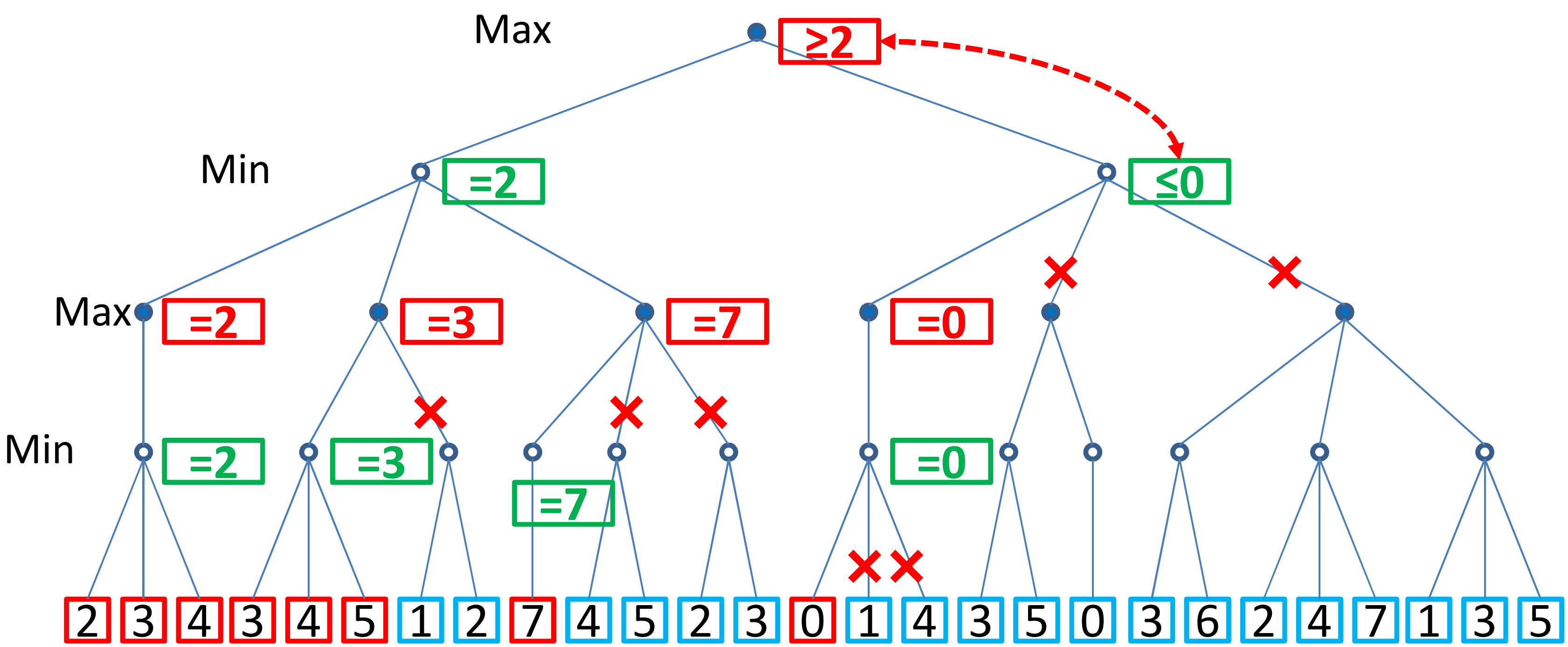
Minimax with $\alpha\beta$-Pruning



Minimax with $\alpha\beta$-Pruning

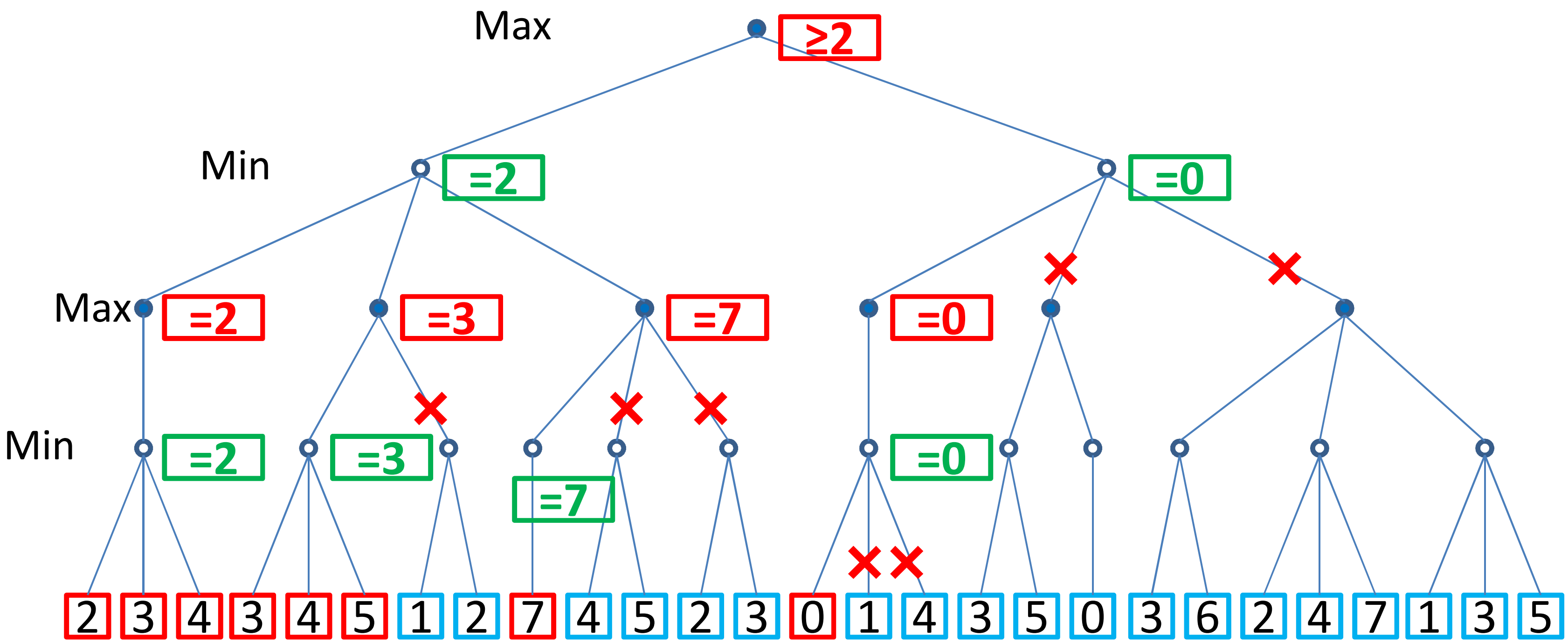


Minimax with $\alpha\beta$-Pruning



Minimax with $\alpha\beta$-Pruning

- **19 static evaluations saved**



Exercises: Artificial Intelligence

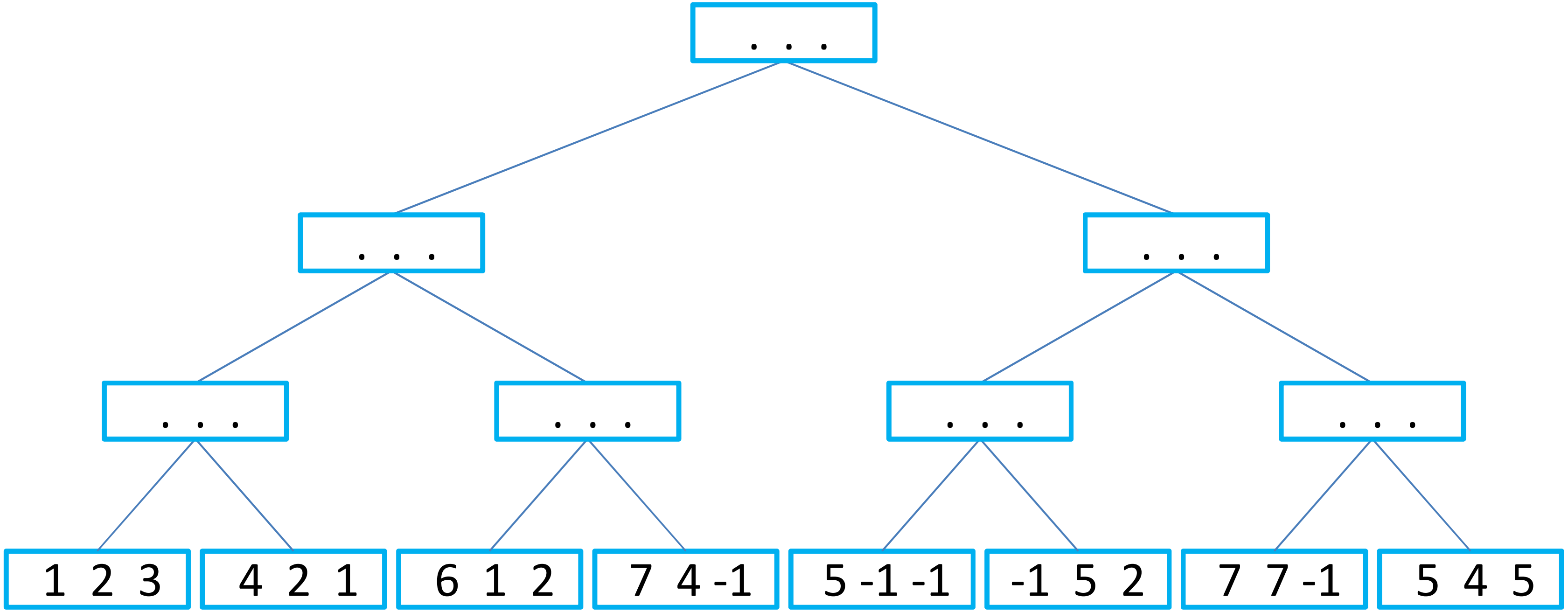
MiniMax & Constraint Processing:
MiniMax Algorithm for 3 Players

MiniMax & Constraint Processing: MiniMax Algorithm for 3 Players

PROBLEM

Problem

- Come up with a MiniMax algorithm for 3 players and apply on the figure below.

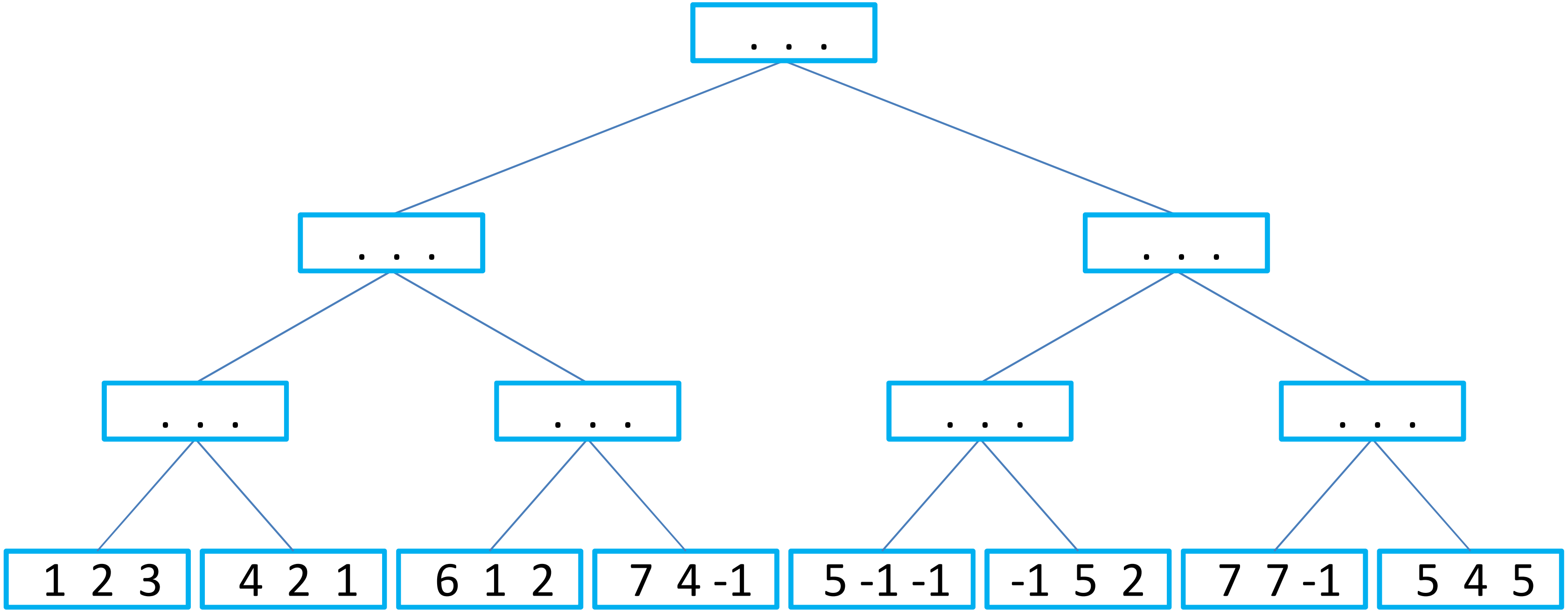


MiniMax & Constraint Processing: MiniMax Algorithm

MINIMAX FOR 3 PLAYERS

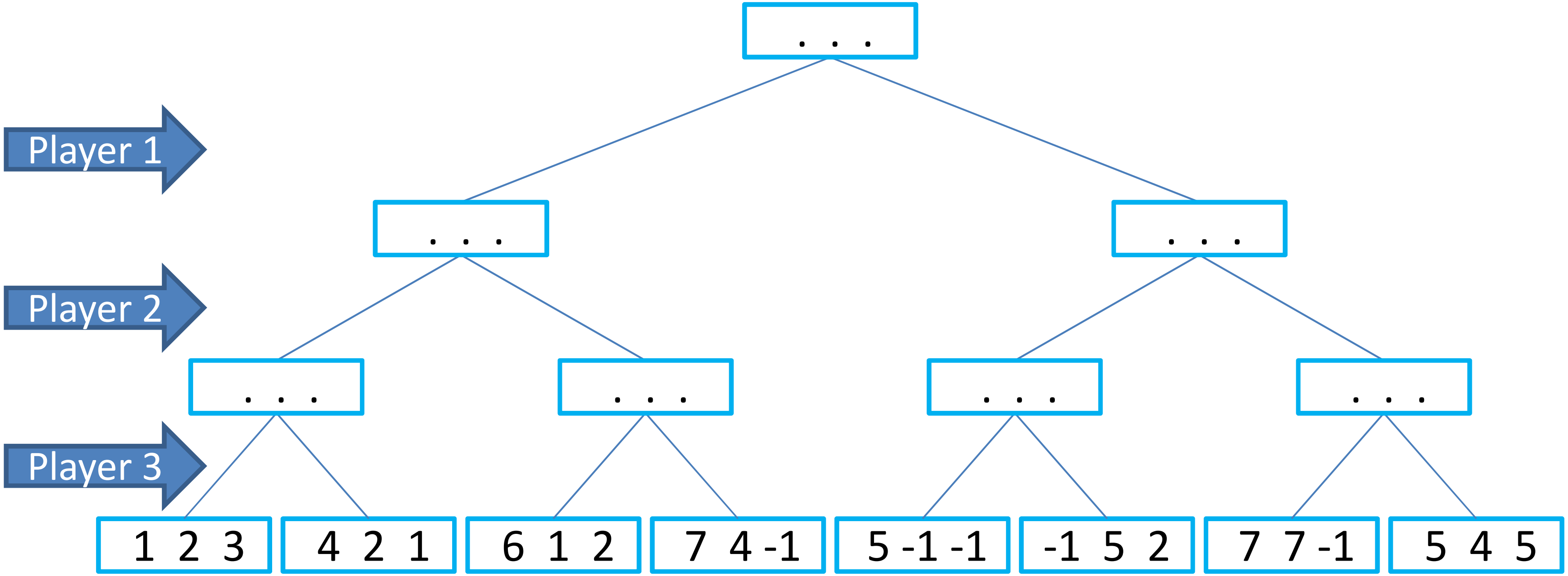
MiniMax For 3 Players

- All players are Max
- Evaluation function given by vector



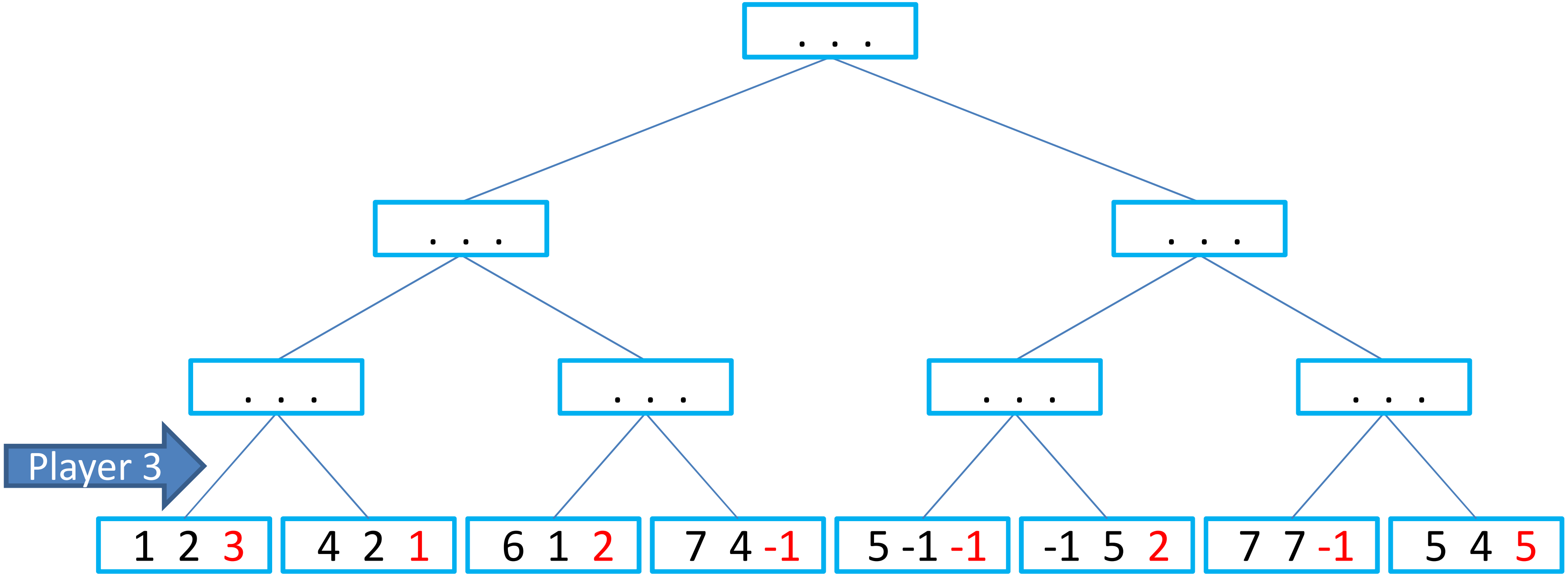
MiniMax For 3 Players

- Each layer assigned to 1 player
- Turn: every 3 layers



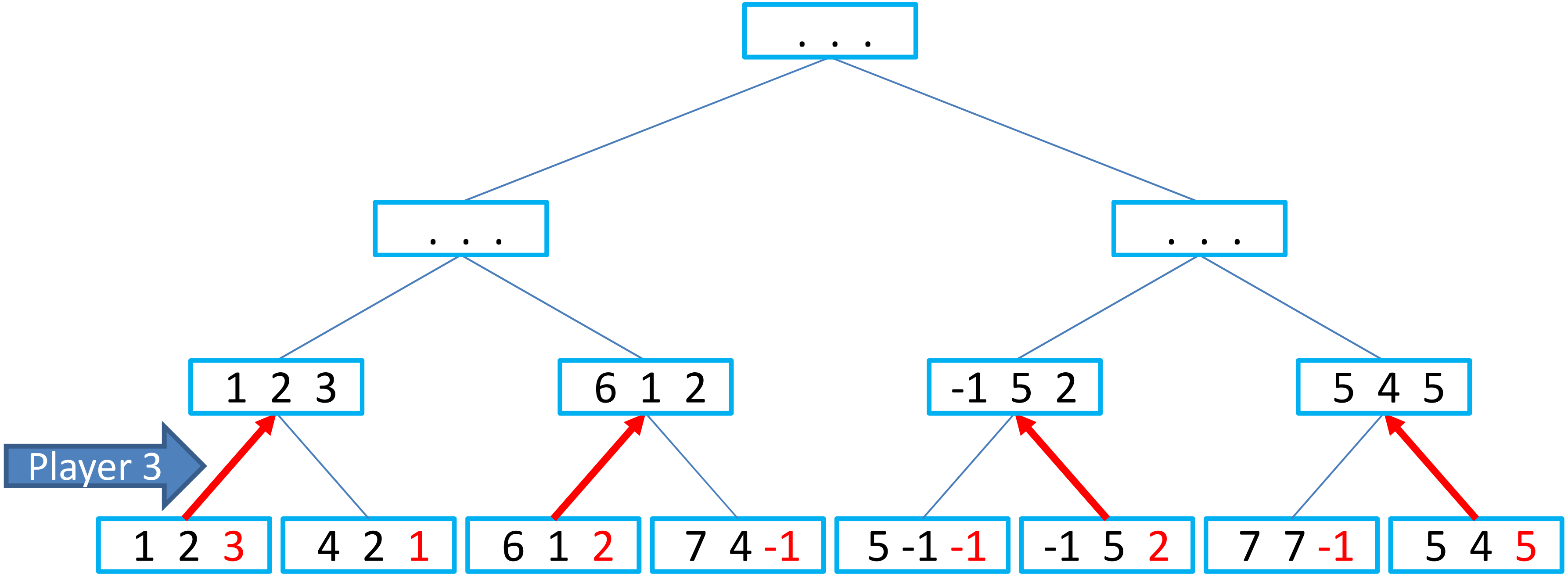
MiniMax For 3 Players

- Max third player: third position of vector



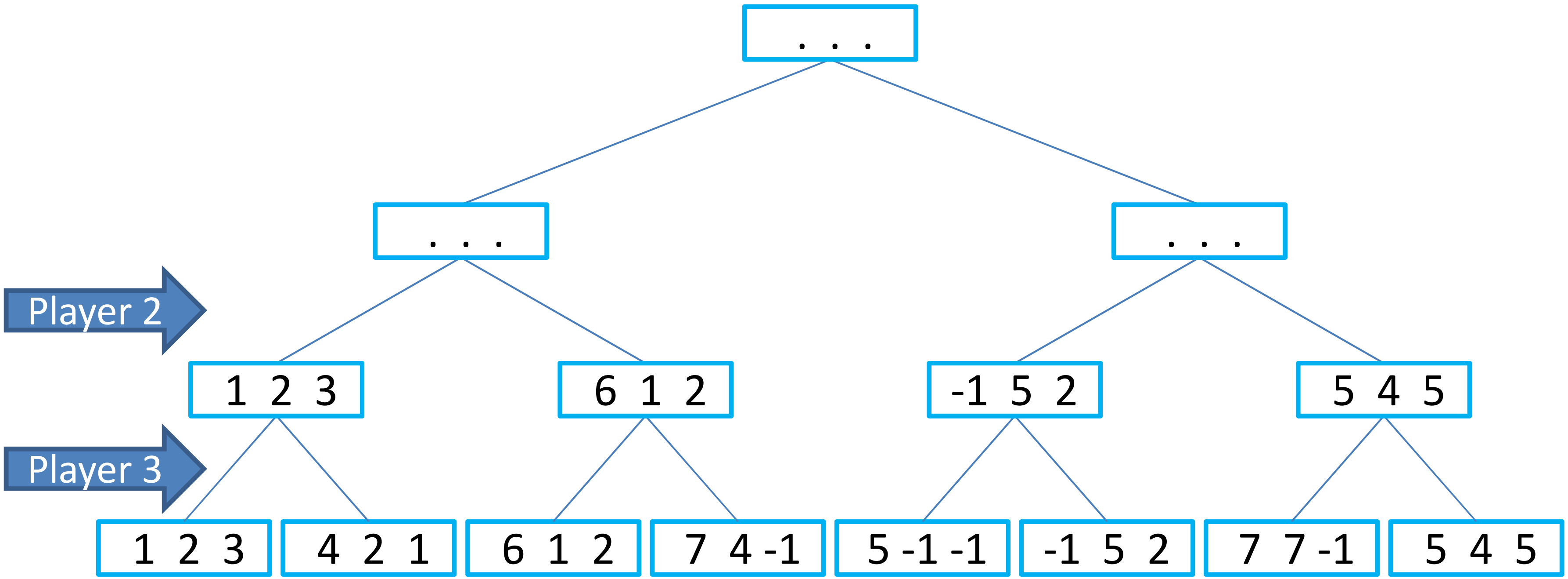
MiniMax For 3 Players

- MaxThirdPlayer([1,2,3],[4,2,1]) = [1,2,3]
- MaxThirdPlayer([6,1,2],[7,4,-1]) = [6,1,2]
- MaxThirdPlayer([5,-1,-1],[-1,5,2]) = [-1,5,2]
- MaxThirdPlayer([7,7,-1],[5,4,5]) = [5,4,5]



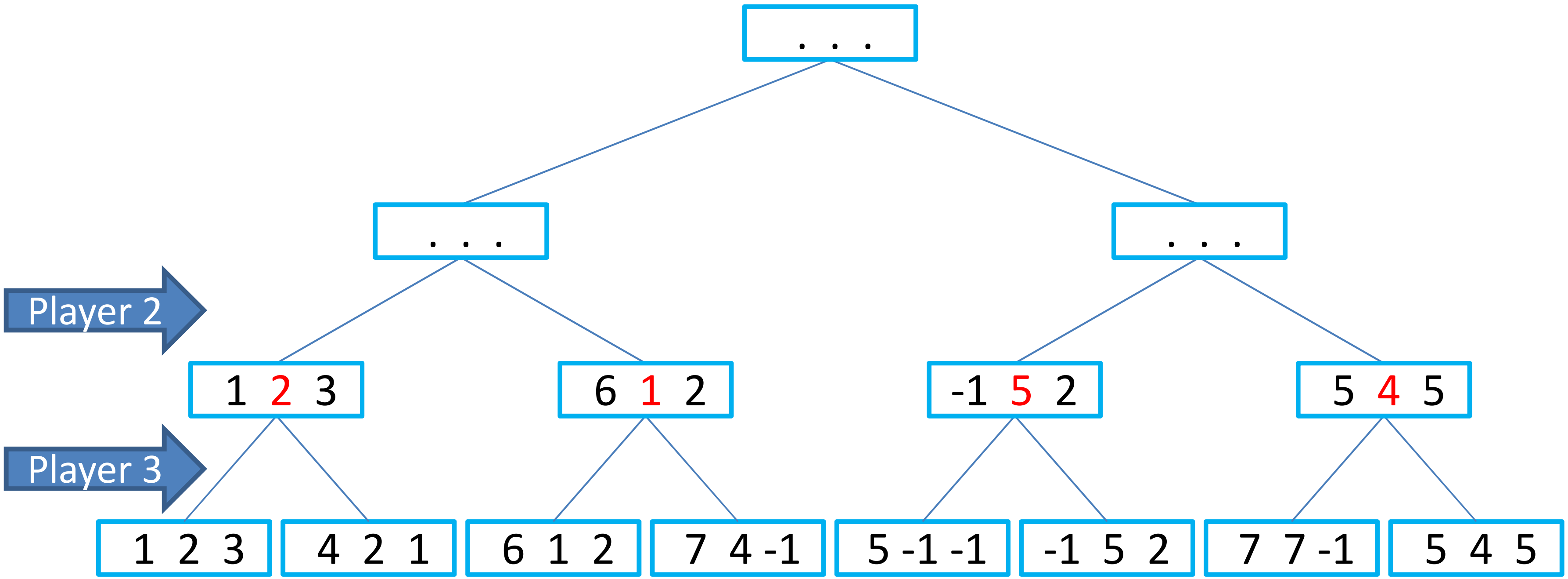
MiniMax For 3 Players

- Second player’s move



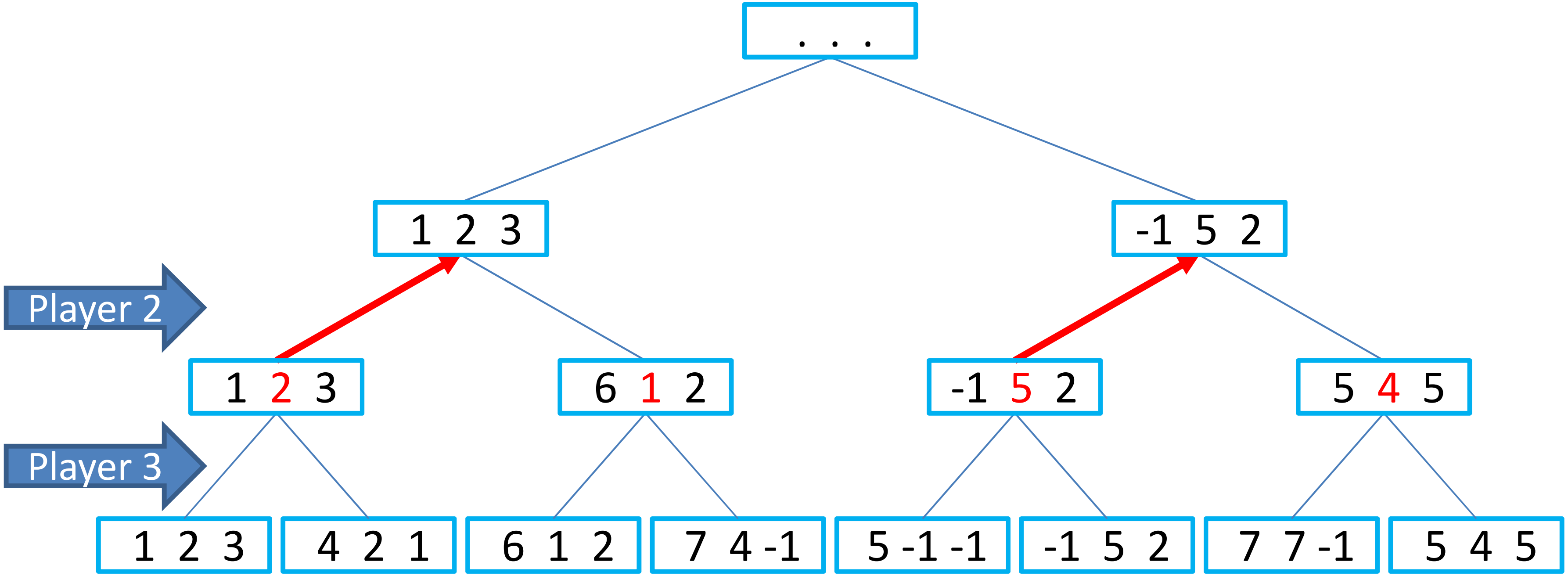
MiniMax For 3 Players

- Max second player: second position of vector



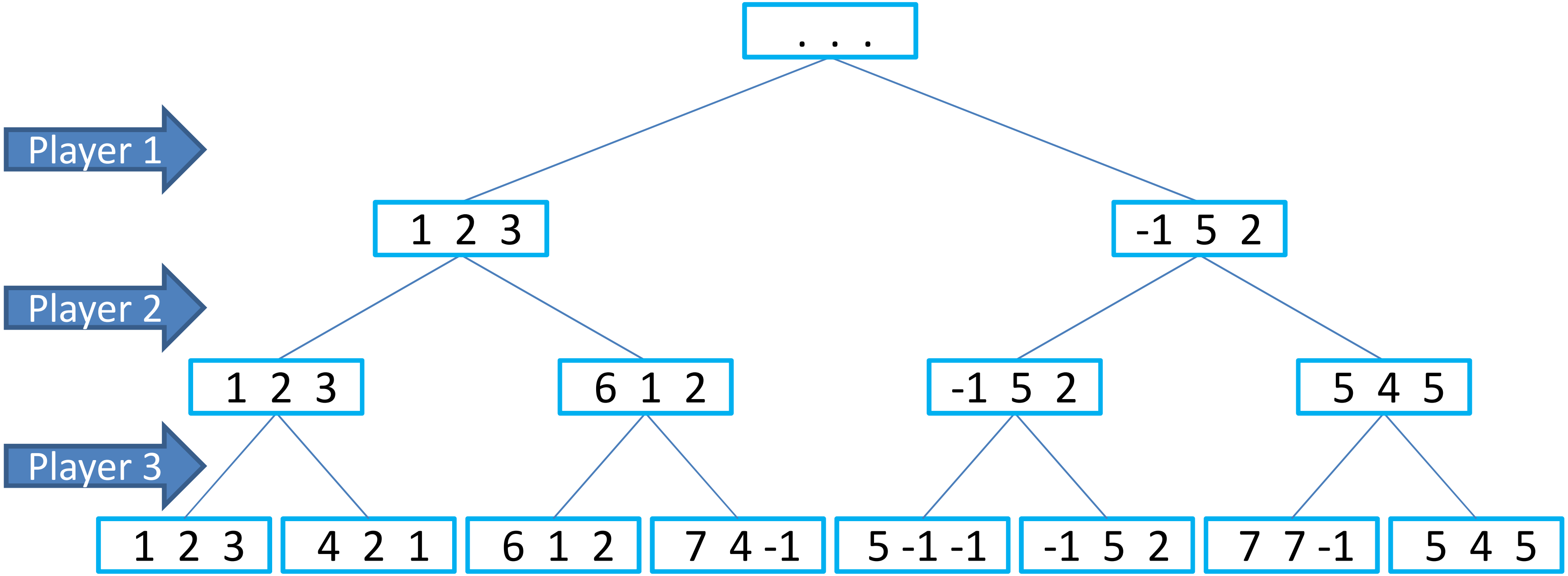
MiniMax For 3 Players

- MaxSecondPlayer([1,2,3],[6,1,2]) = [1,2,3]
- MaxSecondPlayer([-1,5,2],[5,4,5]) = [-1,5,2]



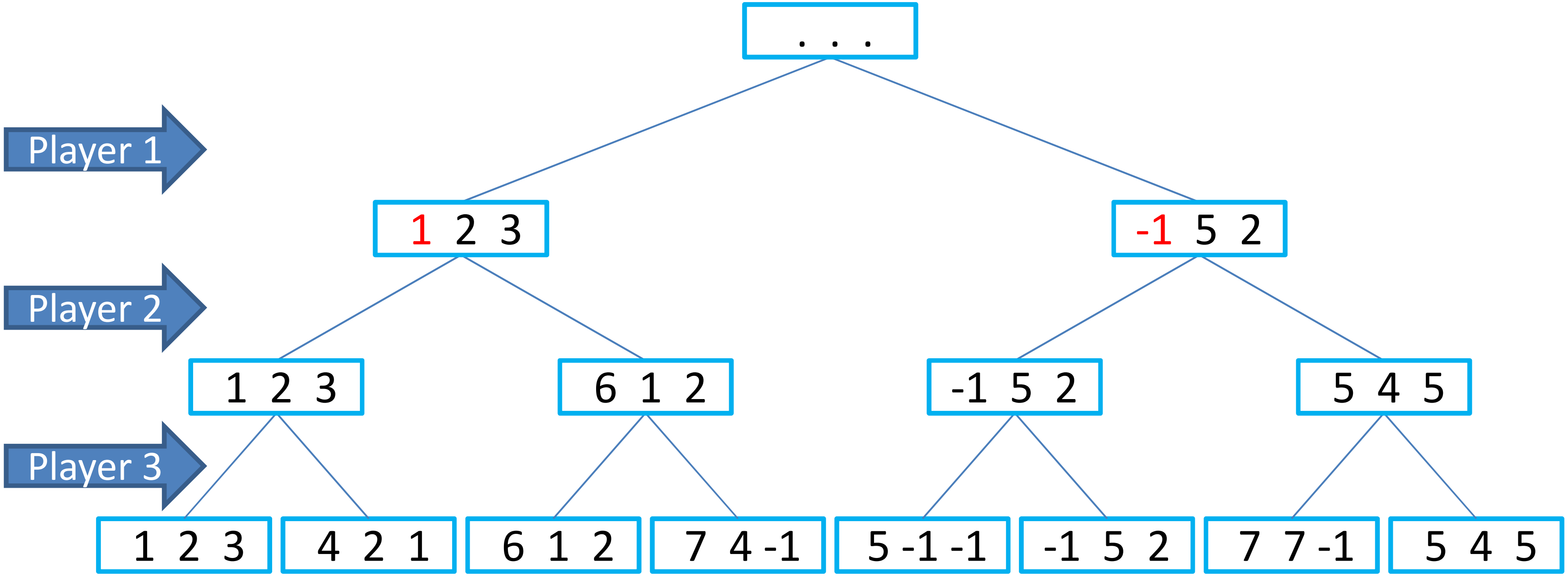
MiniMax For 3 Players

- First player's move



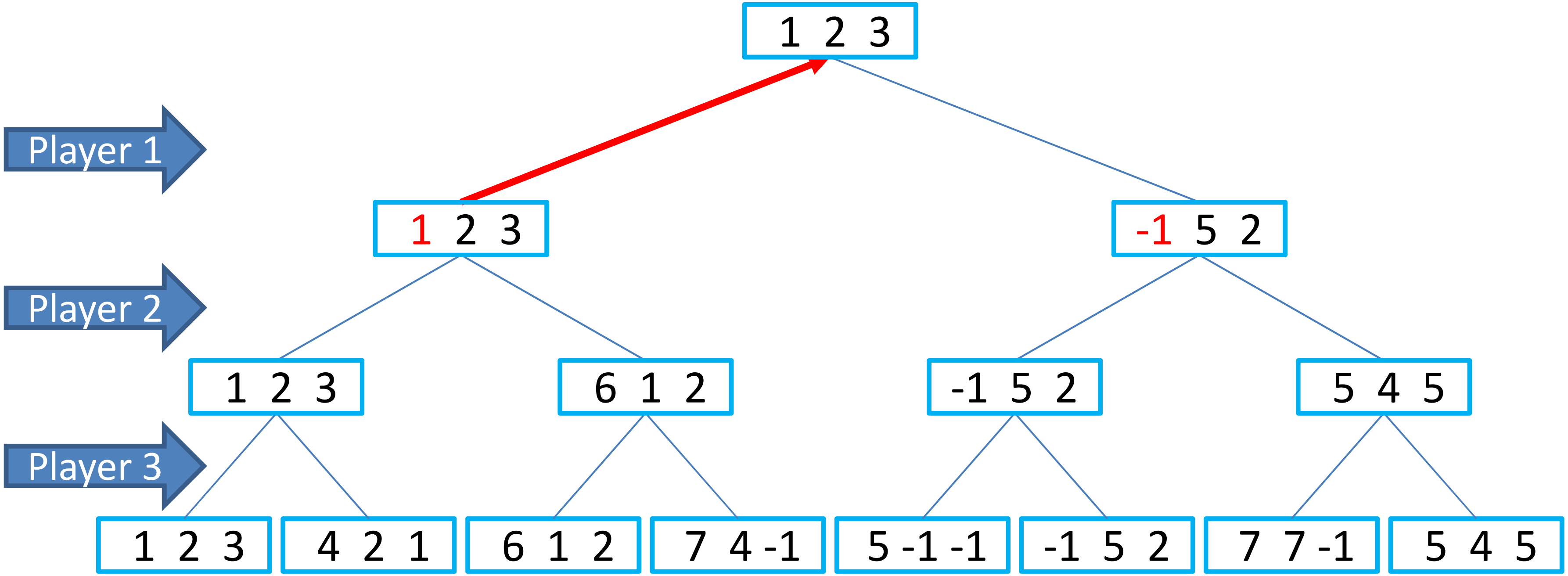
MiniMax For 3 Players

- Max first player: first position of vector



MiniMax For 3 Players

- MaxFirstPlayer([1,2,3],[-1,5,4]) = [1,2,3]


1 2 3
Player 1
1 2 3
-1 5 2
Player 2
1 2 3
6 1 2
-1 5 2
5 4 5
Player 3
1 2 3
4 2 1
6 1 2
7 4 -1
5 -1 -1
-1 5 2
7 7 -1
5 4 5