Game Playing
Part 2 Alpha-Beta Pruning

Yingyu Liang

yliang@cs.wisc.edu

Computer Sciences Department

University of Wisconsin, Madison

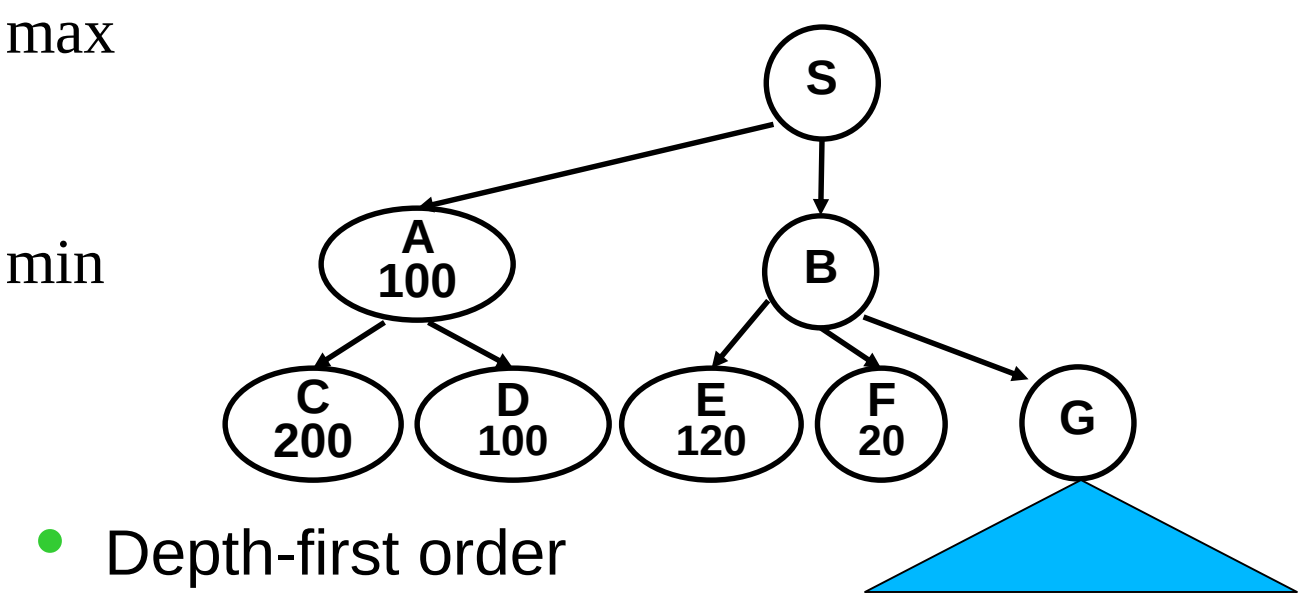
[based on slides from A. Moore http://www.cs.cmu.edu/~awm/tutorials , C. Dyer, J. Skrentny, Jerry Zhu]

alpha-beta pruning

Gives the same game theoretic values as minimax, but prunes part of the game tree.

"If you have an idea that is surely bad, don't take the time to see how truly awful it is." -- Pat Winston

Alpha-Beta Motivation



- Depth-first order
- After returning from A, Max can get at least 100 at S
- After returning from F, Max can get at most 20 at B
- At this point, Max losts interest in B
- There is no need to explore G. The subtree at G is pruned. Saves time.

Alpha-beta pruning

```
function Max-Value (s,α,β)
inputs:
    s: current state in game, Max about to play
    α: best score (highest) for Max along path to s
    β: best score (lowest) for Min along path to s
output: min(β , best-score (for Max) available from s)

    if ( s is a terminal state )
    then return ( terminal value of s )
    else for each s' in Succ(s)
      α := max( α , Min-value(s',α,β))
      if ( α ≥ β ) then return β   /* alpha pruning */
    return α
```

Starting from the root:
Max-Value(root, -∞, +∞)

Alpha-beta pruning

```
function Max-Value (s,α,β)
inputs:
    s: current state in game, Max about to play
    α: best score (highest) for Max along path to s
    β: best score (lowest) for Min along path to s
output: min(β , best-score (for Max) available from s)

    if ( s is a terminal state )
    then return ( terminal value of s )
    else for each s' in Succ(s)
      α := max( α , Min-value(s',α,β))
      if ( α ≥ β ) then return β   /* alpha pruning */
    return α
```

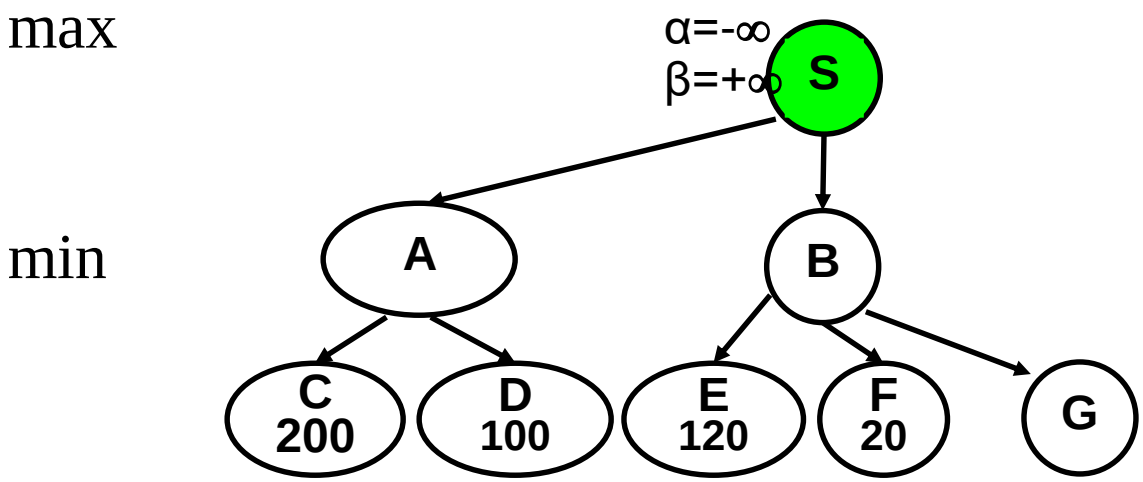
```
function Min-Value(s,α,β)
output: max(α , best-score (for Min) available from s )

    if ( s is a terminal state )
    then return ( terminal value of s)
    else for each s' in Succs(s)
      β := min( β , Max-value(s',α,β))

          if (α ≥ β ) then return α   /* beta pruning */
    return β
```

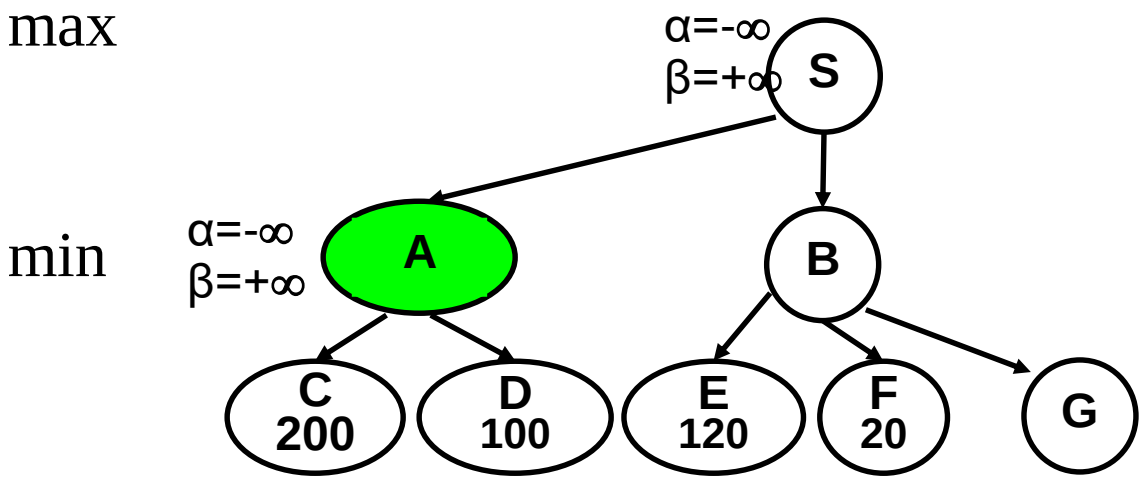
Starting from the root:
Max-Value(root, -∞, +∞)

Alpha-beta pruning example 1

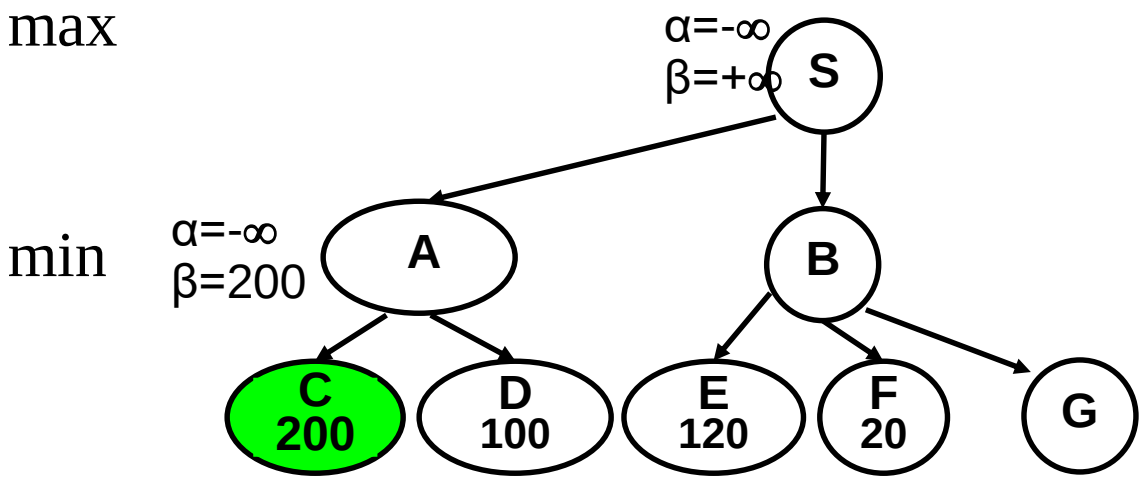


- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

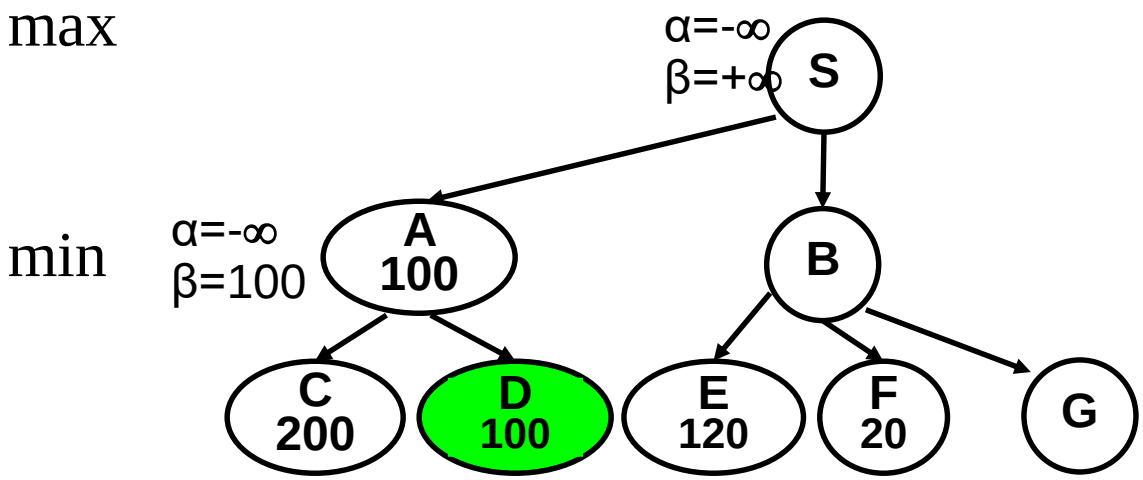
Alpha-beta pruning example 1



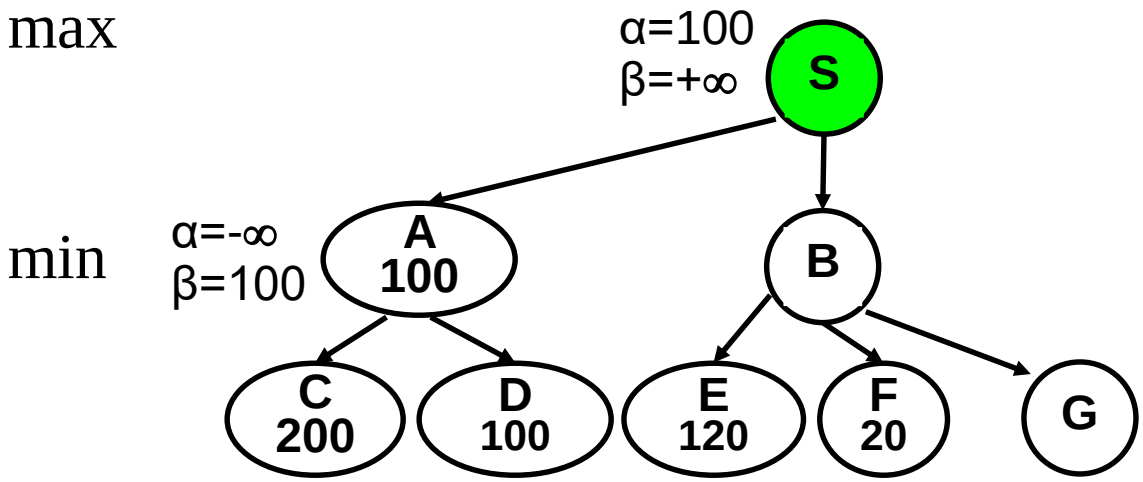
Alpha-beta pruning example 1



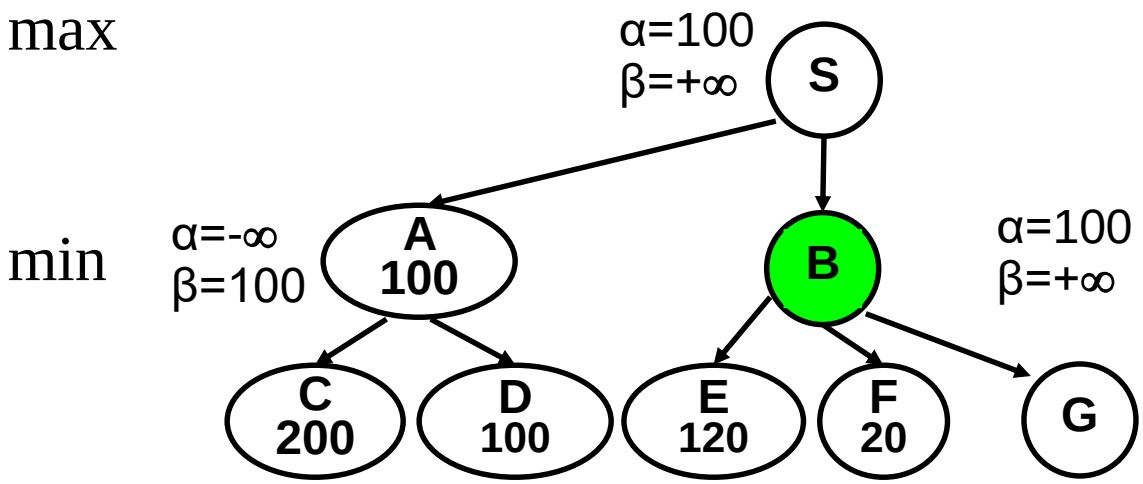
Alpha-beta pruning example 1



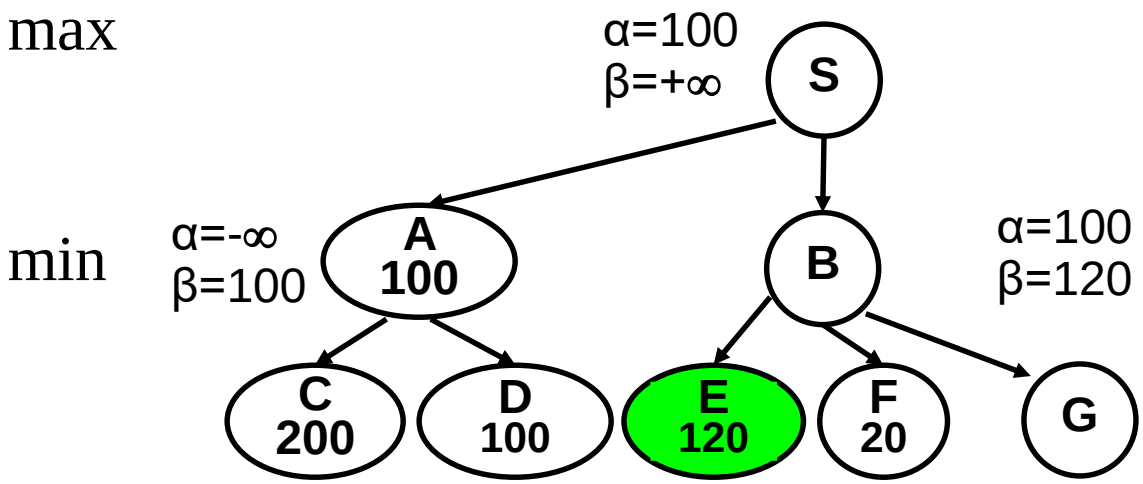
Alpha-beta pruning example 1



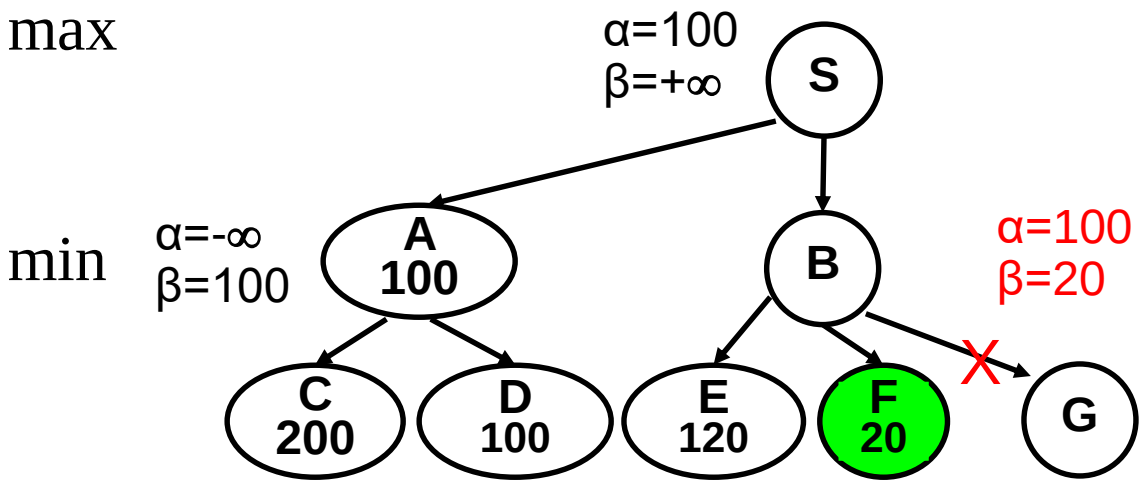
Alpha-beta pruning example 1



Alpha-beta pruning example 1



Alpha-beta pruning example 1



Alpha-beta pruning

```
function Max-Value (s,α,β)
inputs:
    s: current state in game, Max about to play
    α: best score (highest) for Max along path to s
    β: best score (lowest) for Min along path to s
output: min(β , best-score (for Max) available from s)

    if ( s is a terminal state )
    then return ( terminal value of s )
    else for each s' in Succ(s)
      α := max( α , Min-value(s',α,β))
      if ( α ≥ β ) then return β   /* alpha pruning */
    return α
```

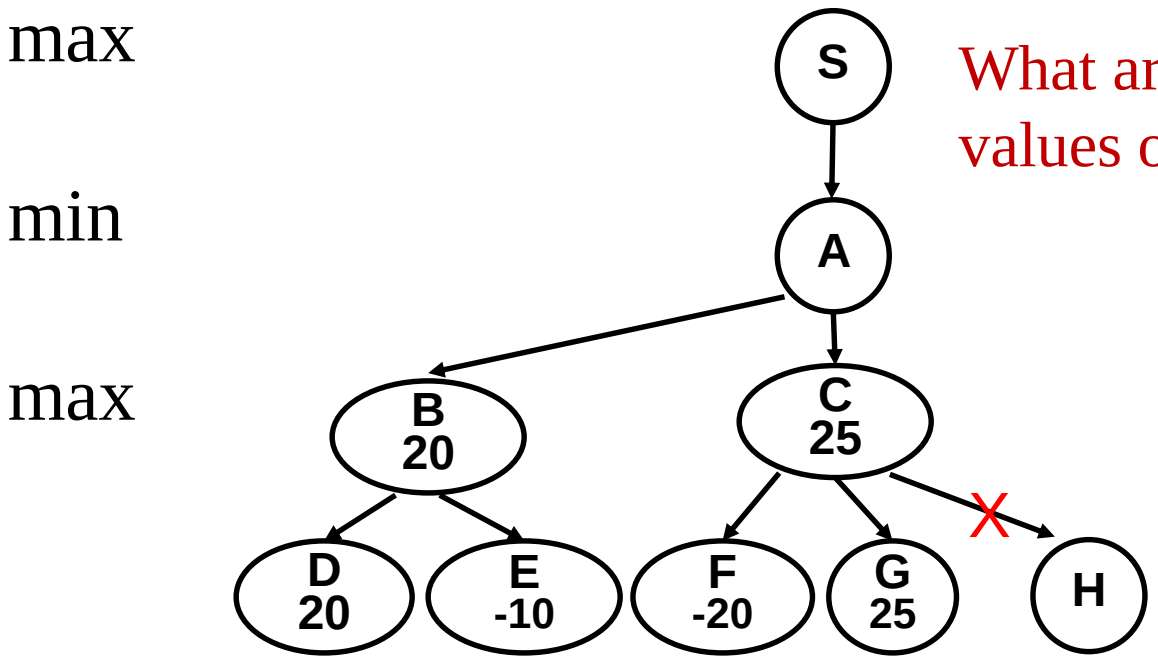
```
function Min-Value(s,α,β)
output: max(α , best-score (for Min) available from s )

    if ( s is a terminal state )
    then return ( terminal value of s)
    else for each s' in Succs(s)
      β := min( β , Max-value(s',α,β))

          if (α ≥ β ) then return α   /* beta pruning */
    return β
```

Starting from the root:
Max-Value(root, -∞, +∞)

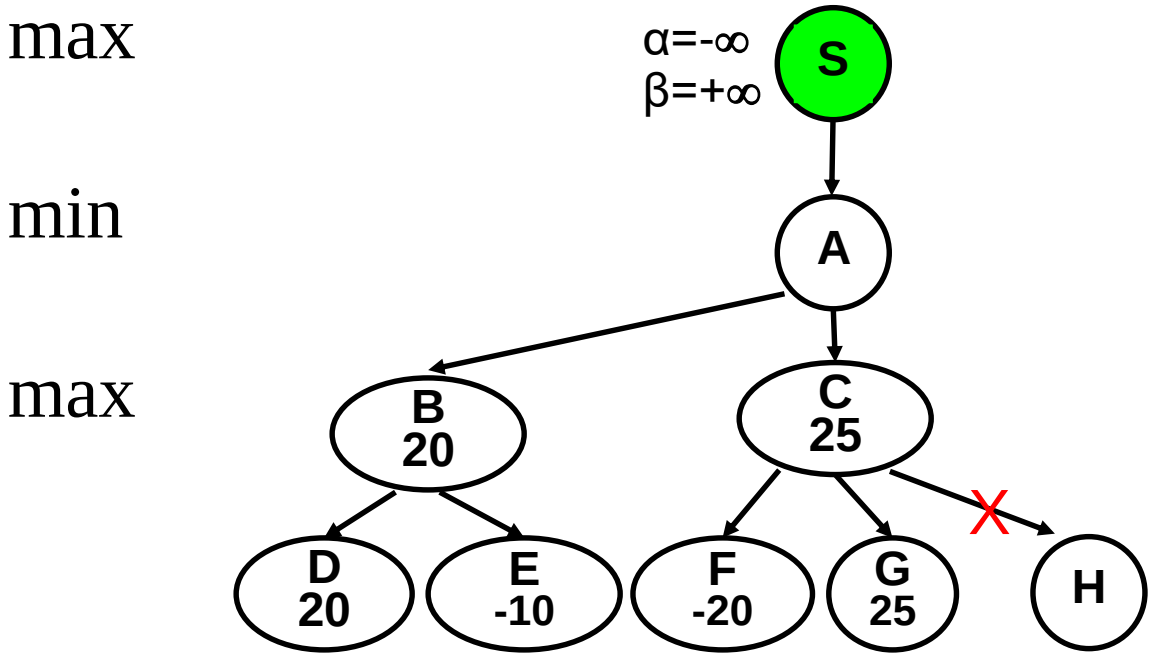
Alpha-beta pruning example 2



What are the alpha and beta values on S?

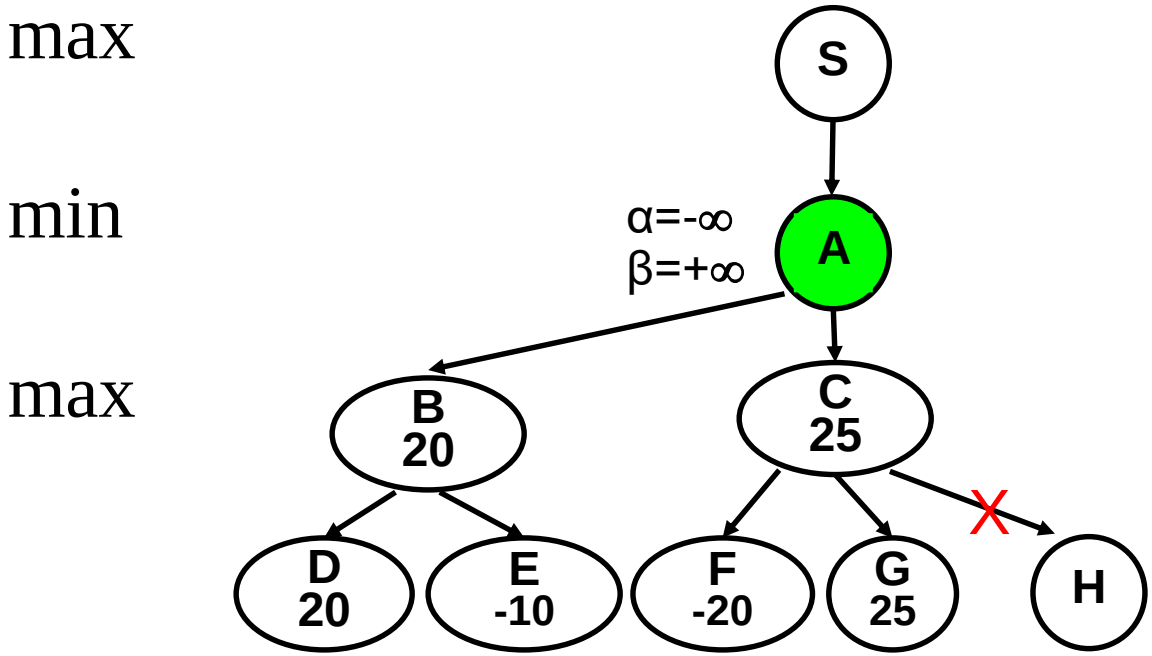
- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

Alpha-beta pruning example 2



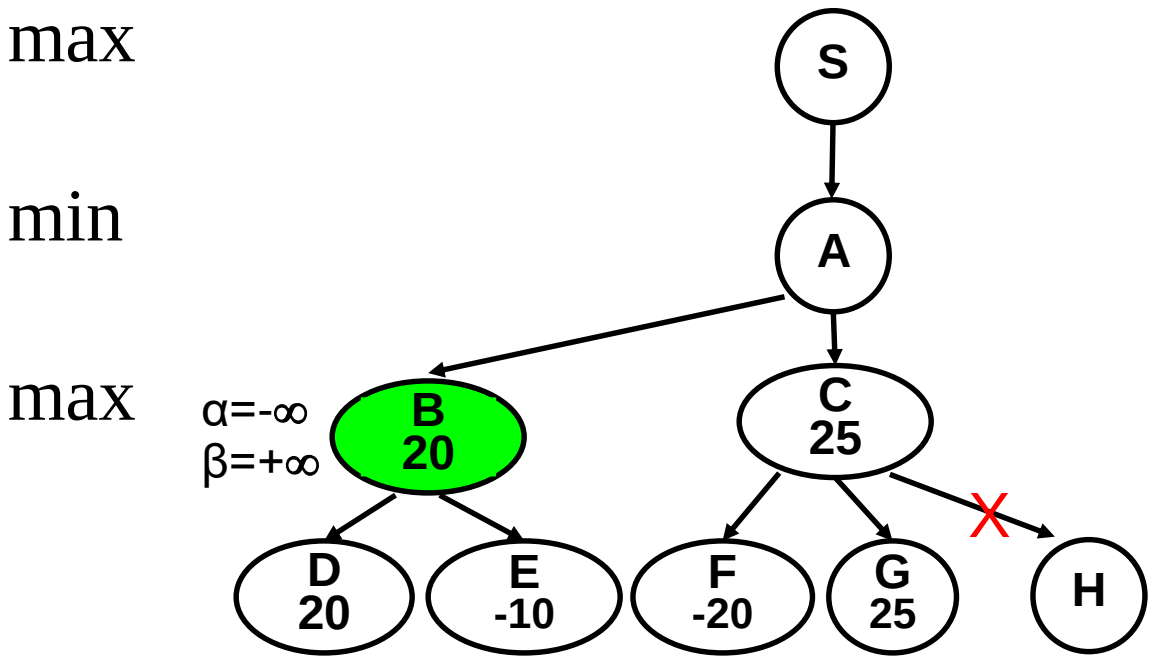
- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

Alpha-beta pruning example 2



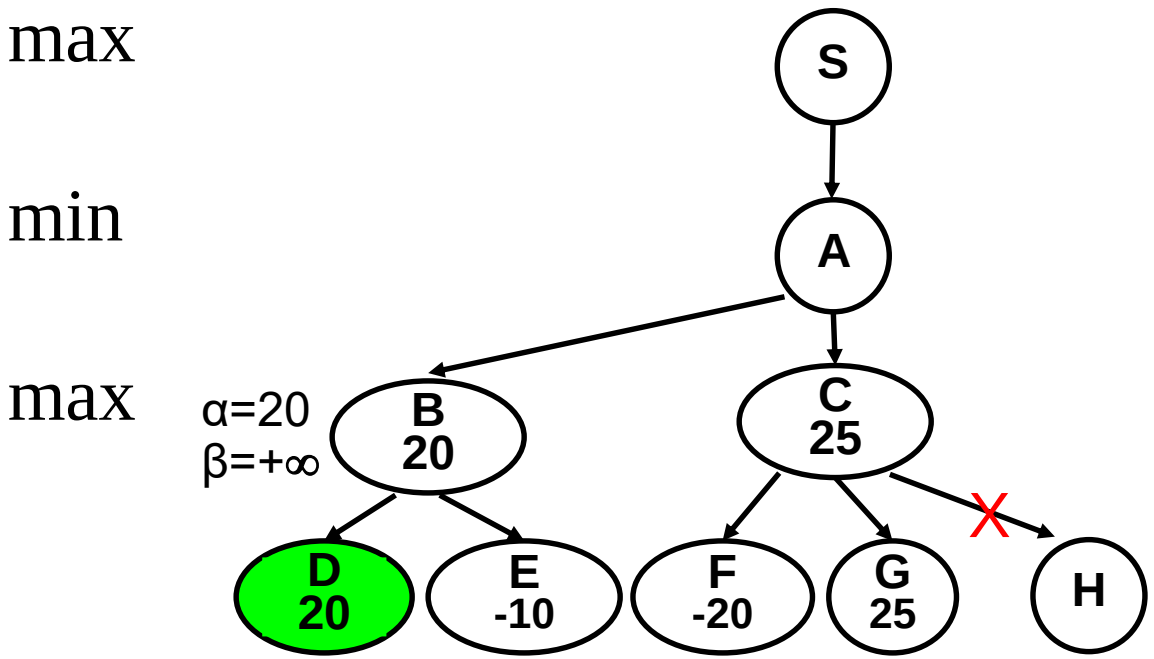
- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

Alpha-beta pruning example 2



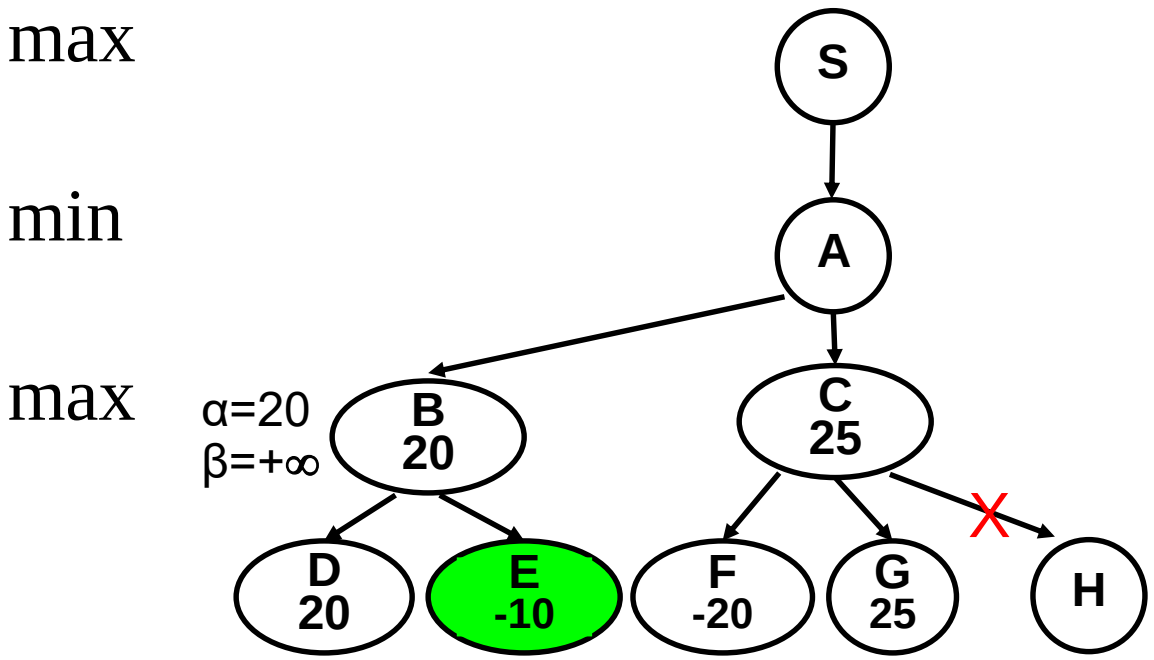
- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

Alpha-beta pruning example 2



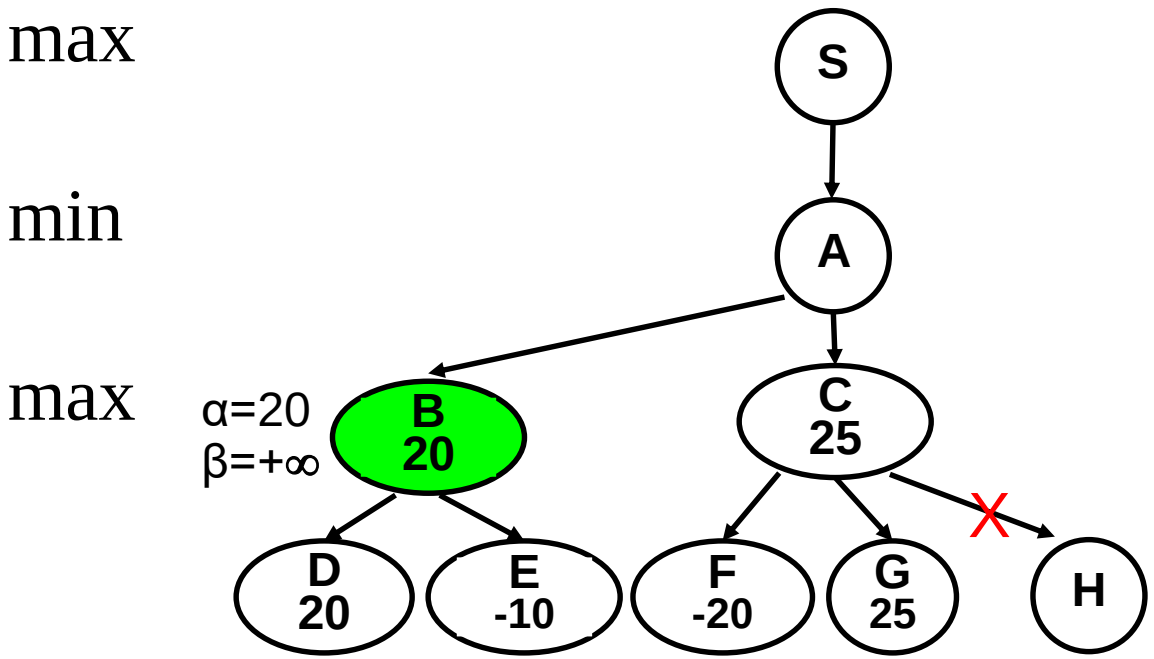
- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

Alpha-beta pruning example 2



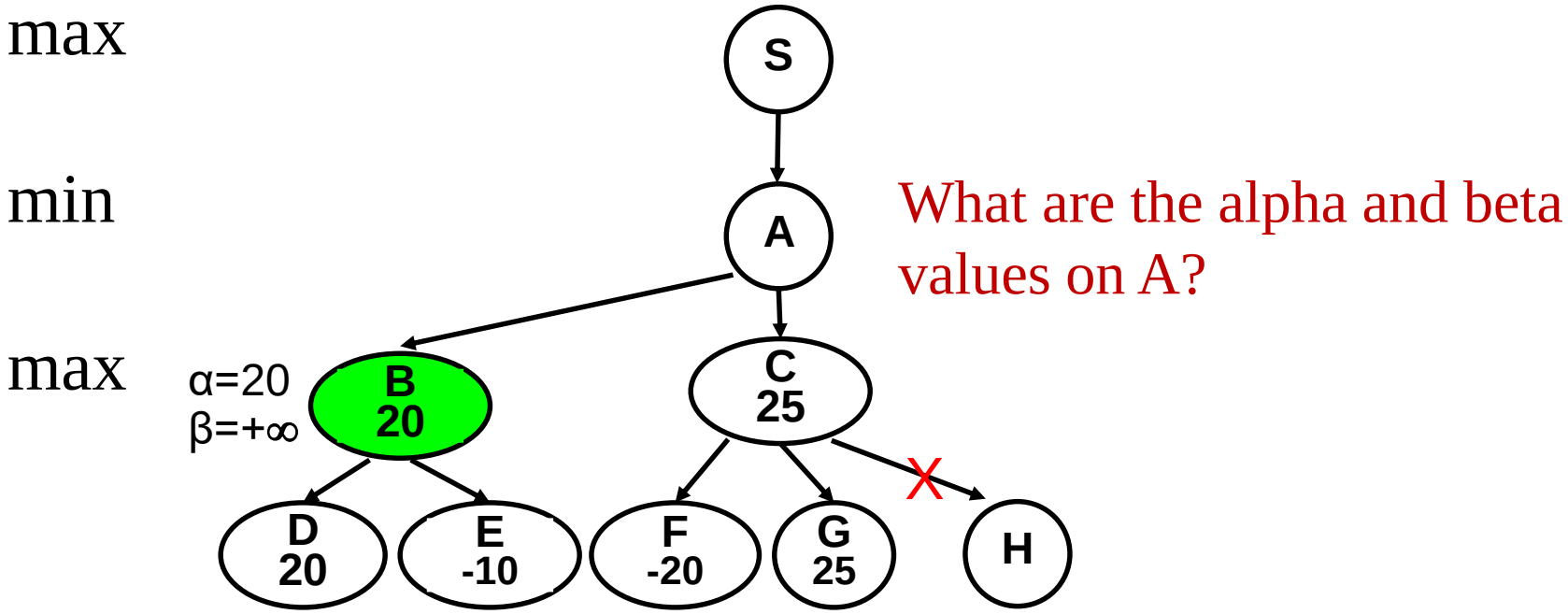
- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

Alpha-beta pruning example 2



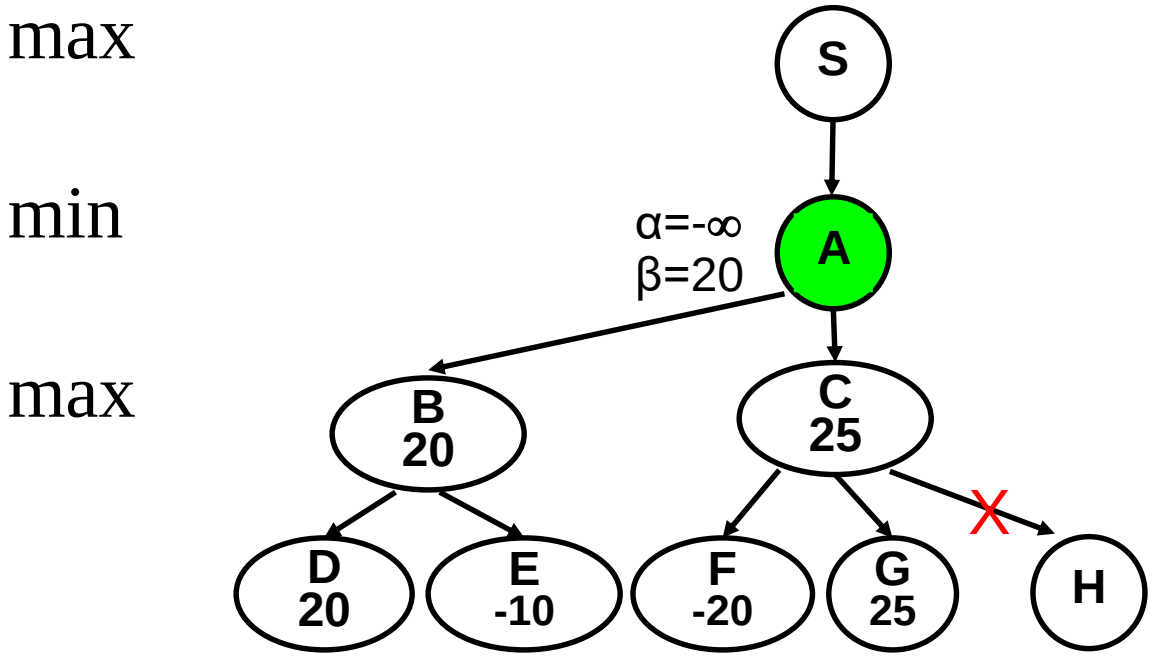
- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

Alpha-beta pruning example 2



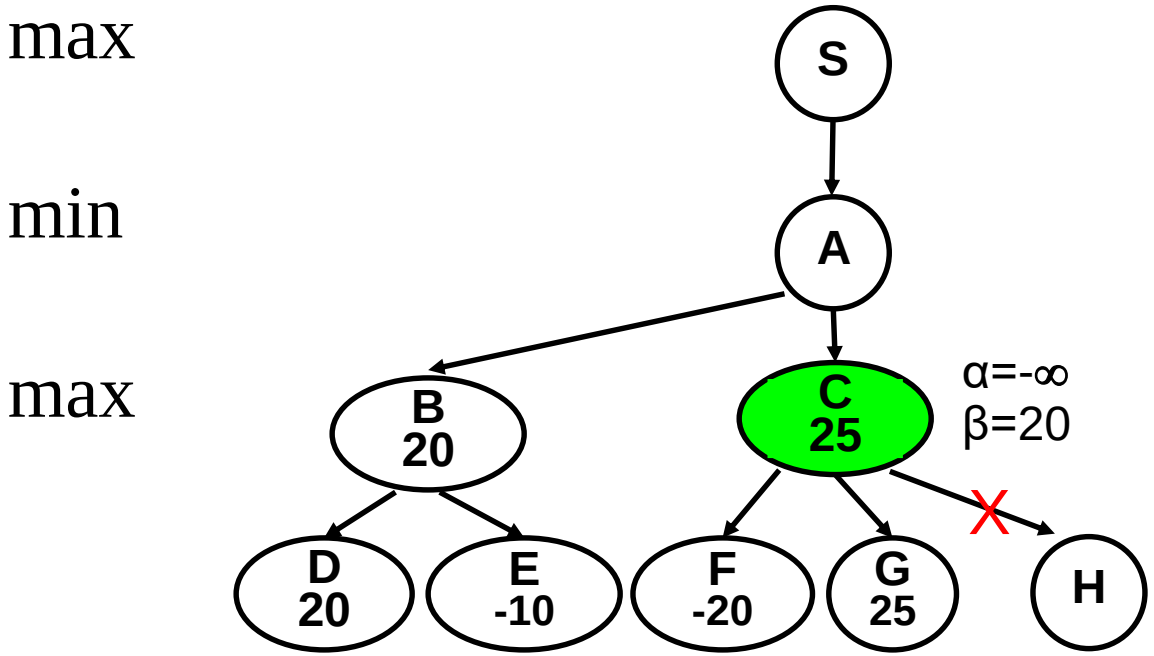
- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

Alpha-beta pruning example 2



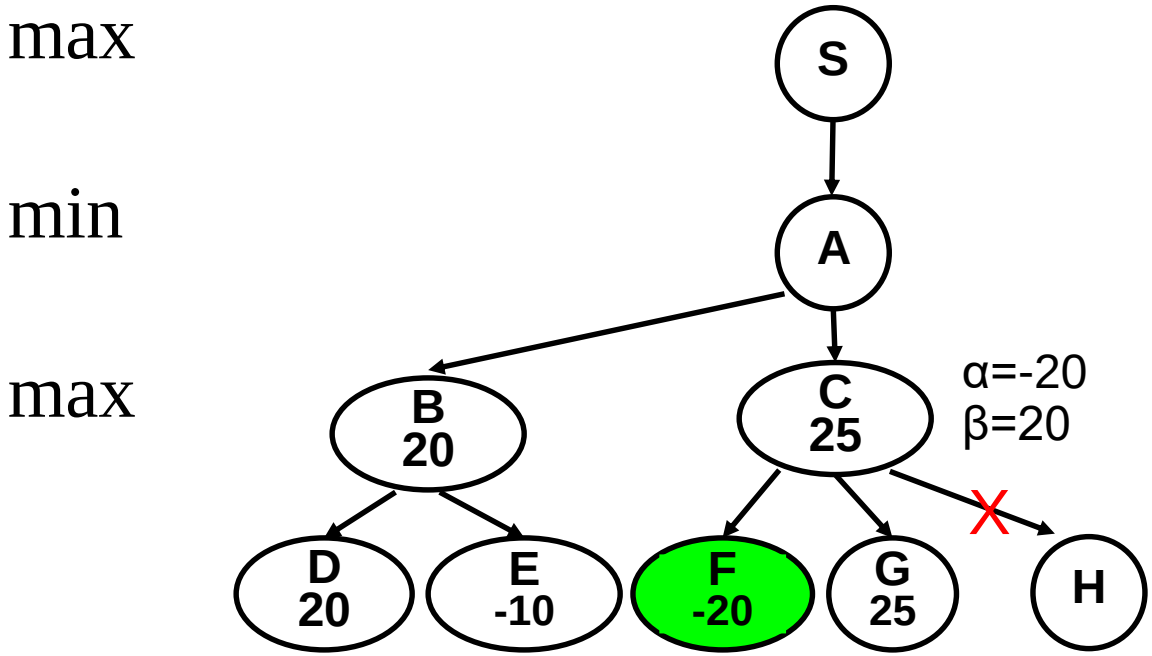
- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

Alpha-beta pruning example 2



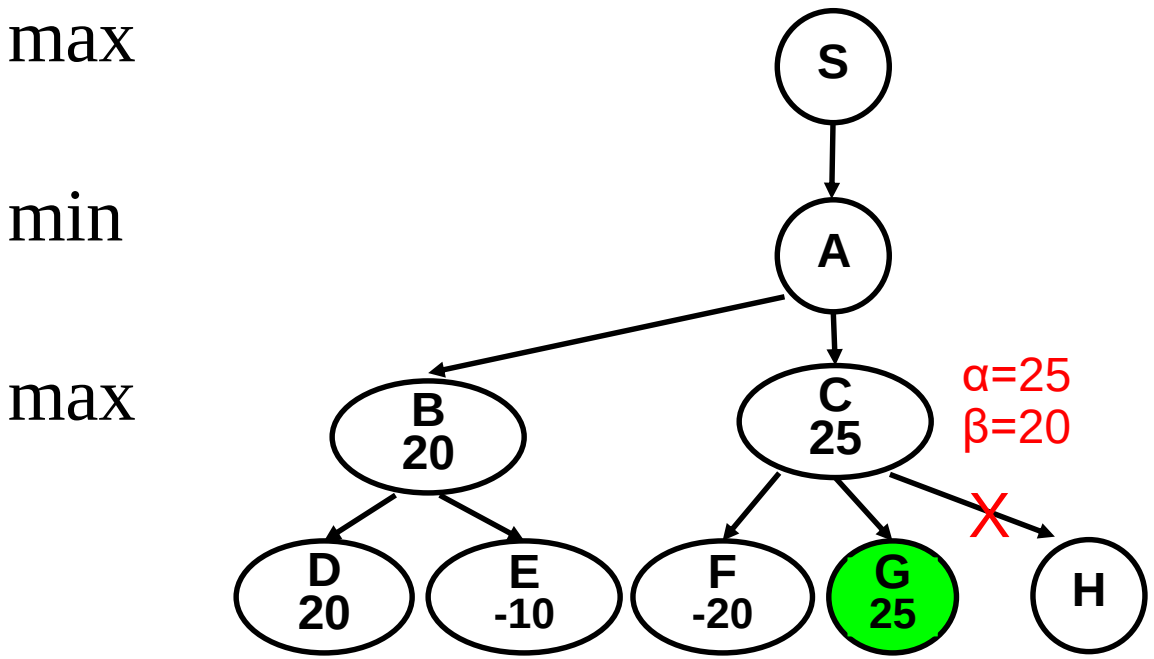
- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

Alpha-beta pruning example 2



- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

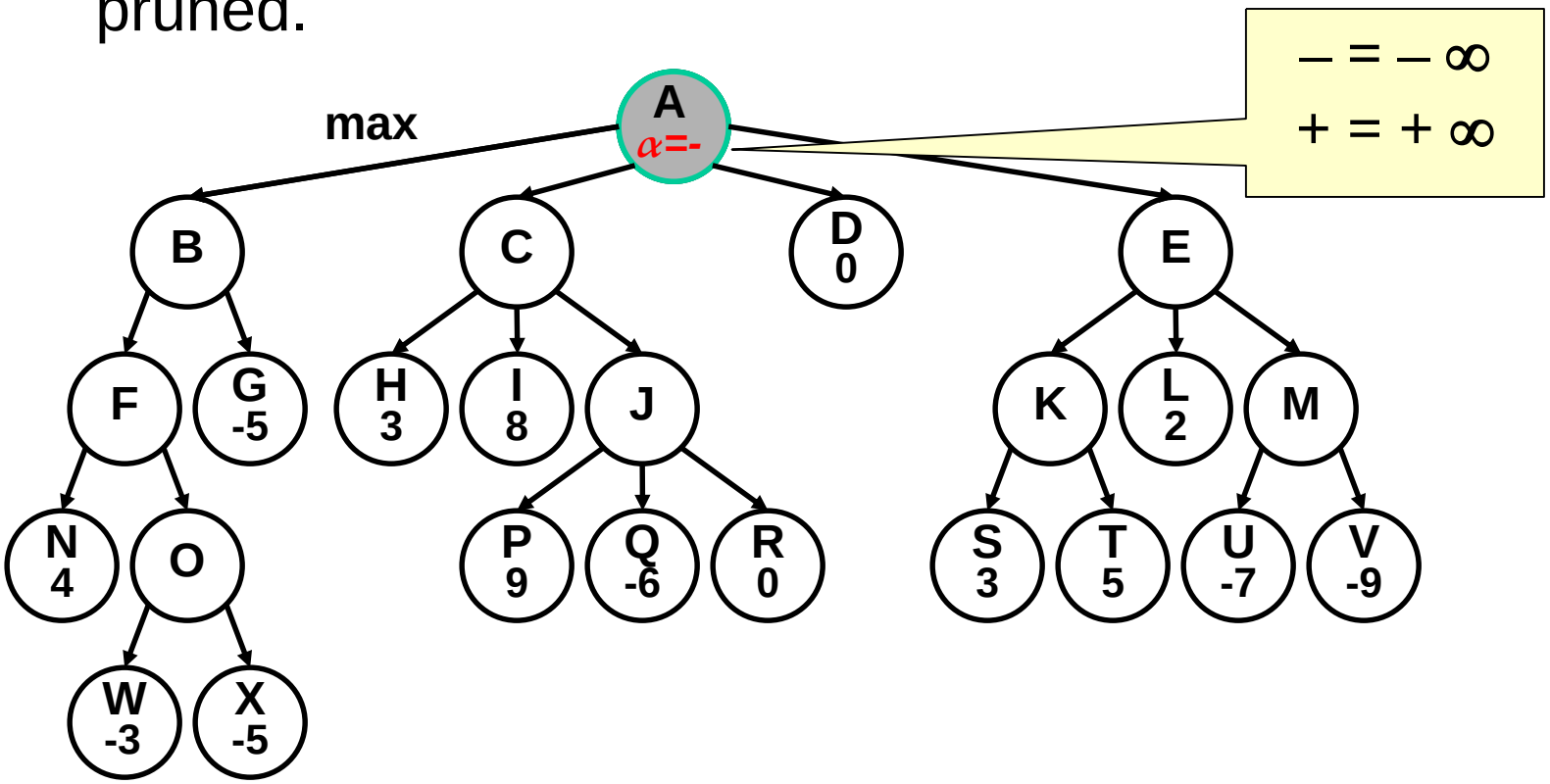
Alpha-beta pruning example 2



- Keep two bounds along the path
  - α: the best Max can do
  - β: the best (smallest) Min can do
- If at anytime α exceeds β, the remaining children are pruned.

Yet another alpha-beta pruning example

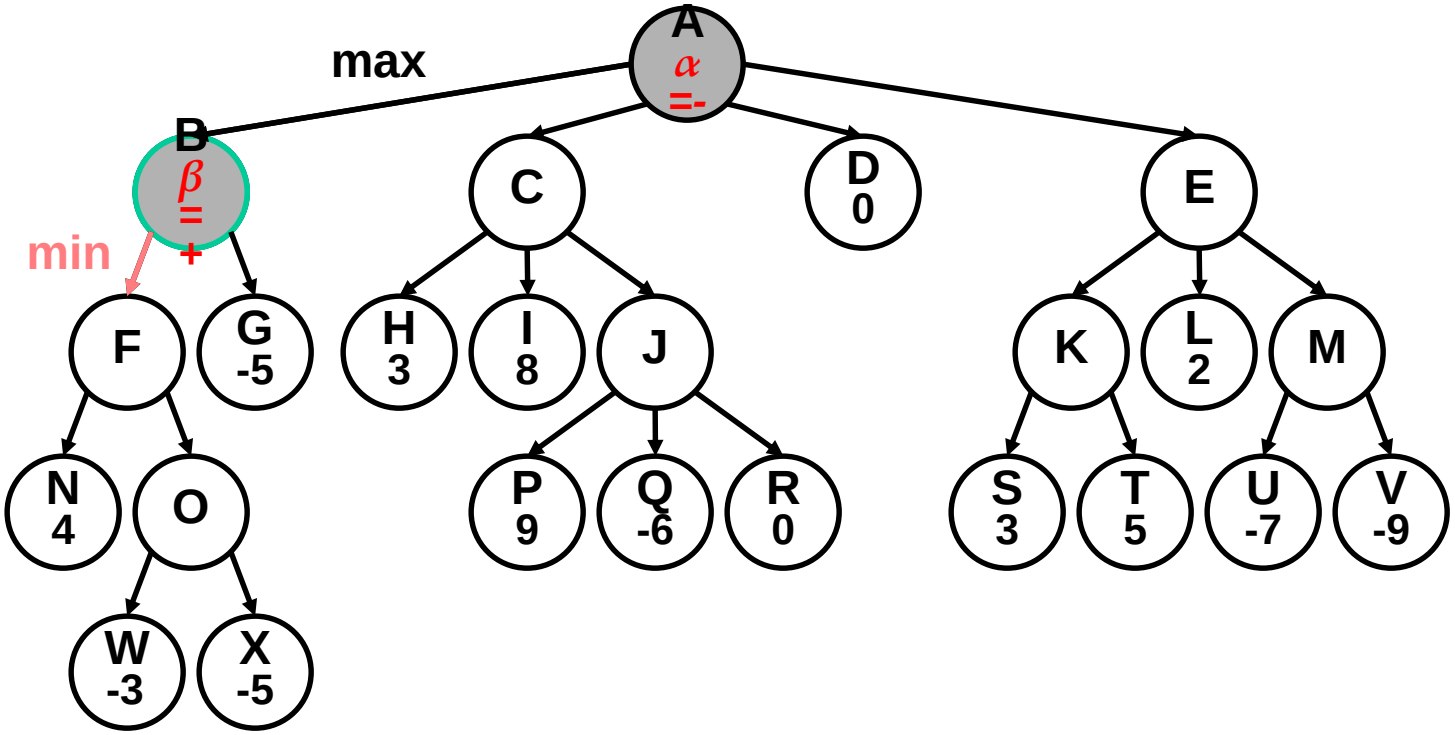
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If at anytime α exceeds β, the remaining children are pruned.



[Example from James Skrentny]

Alpha-beta pruning example

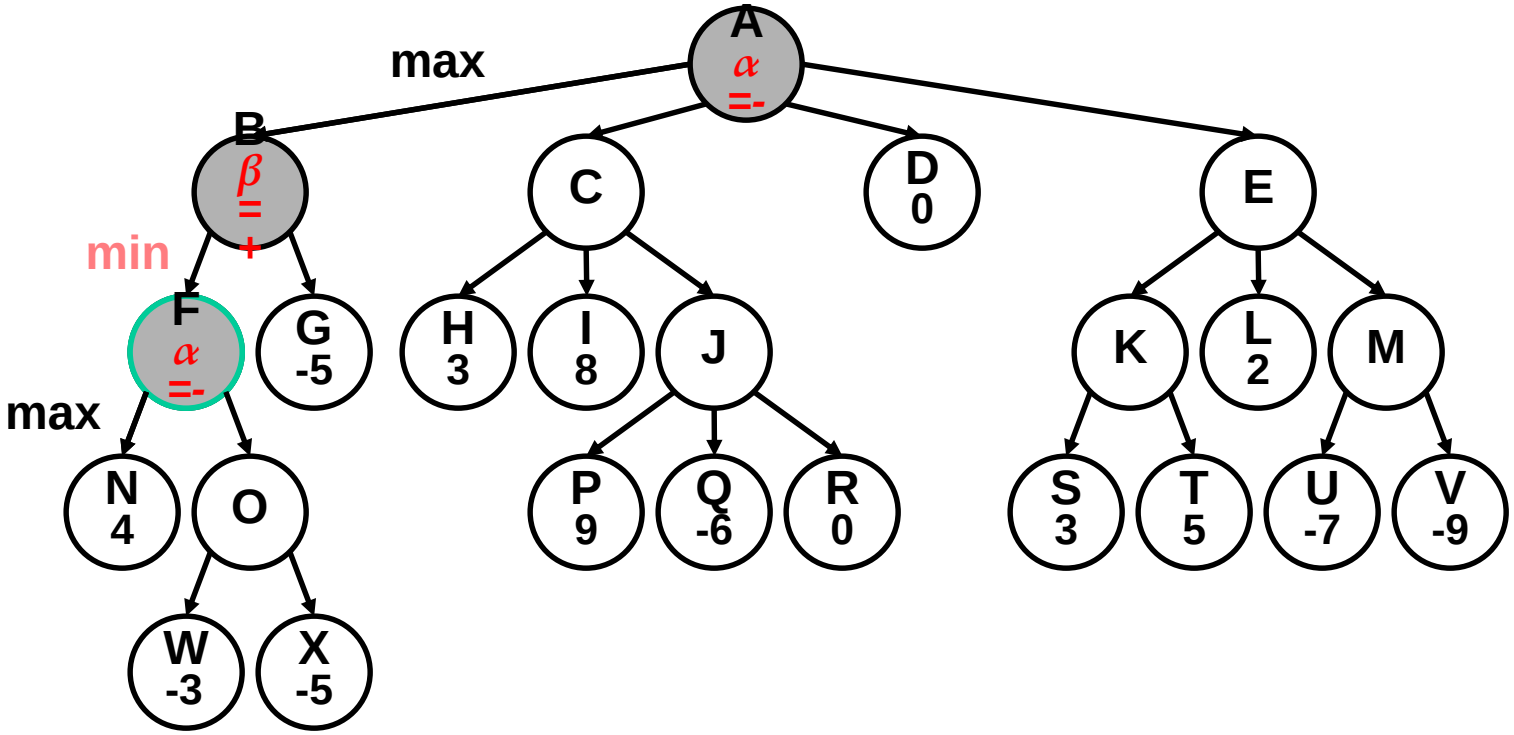
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

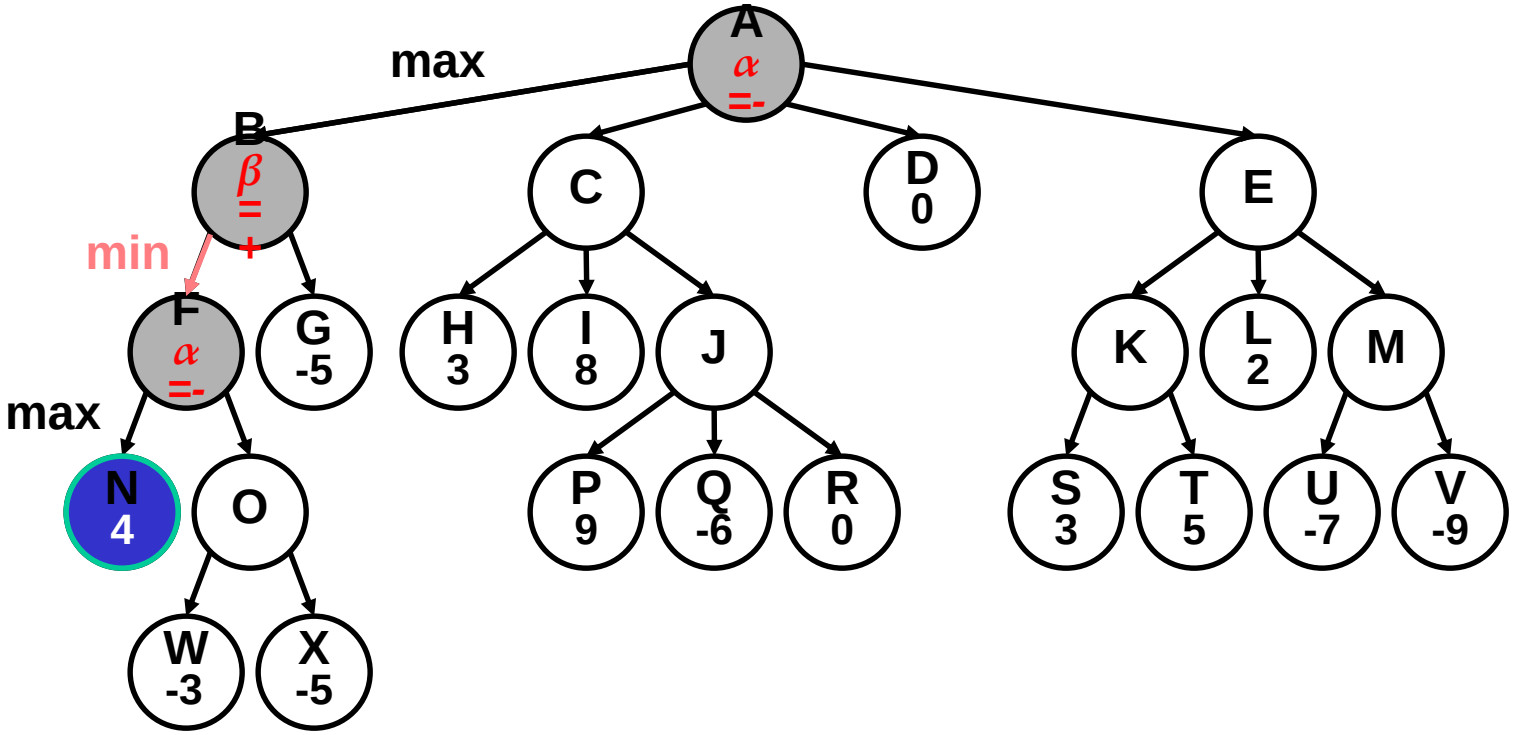
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

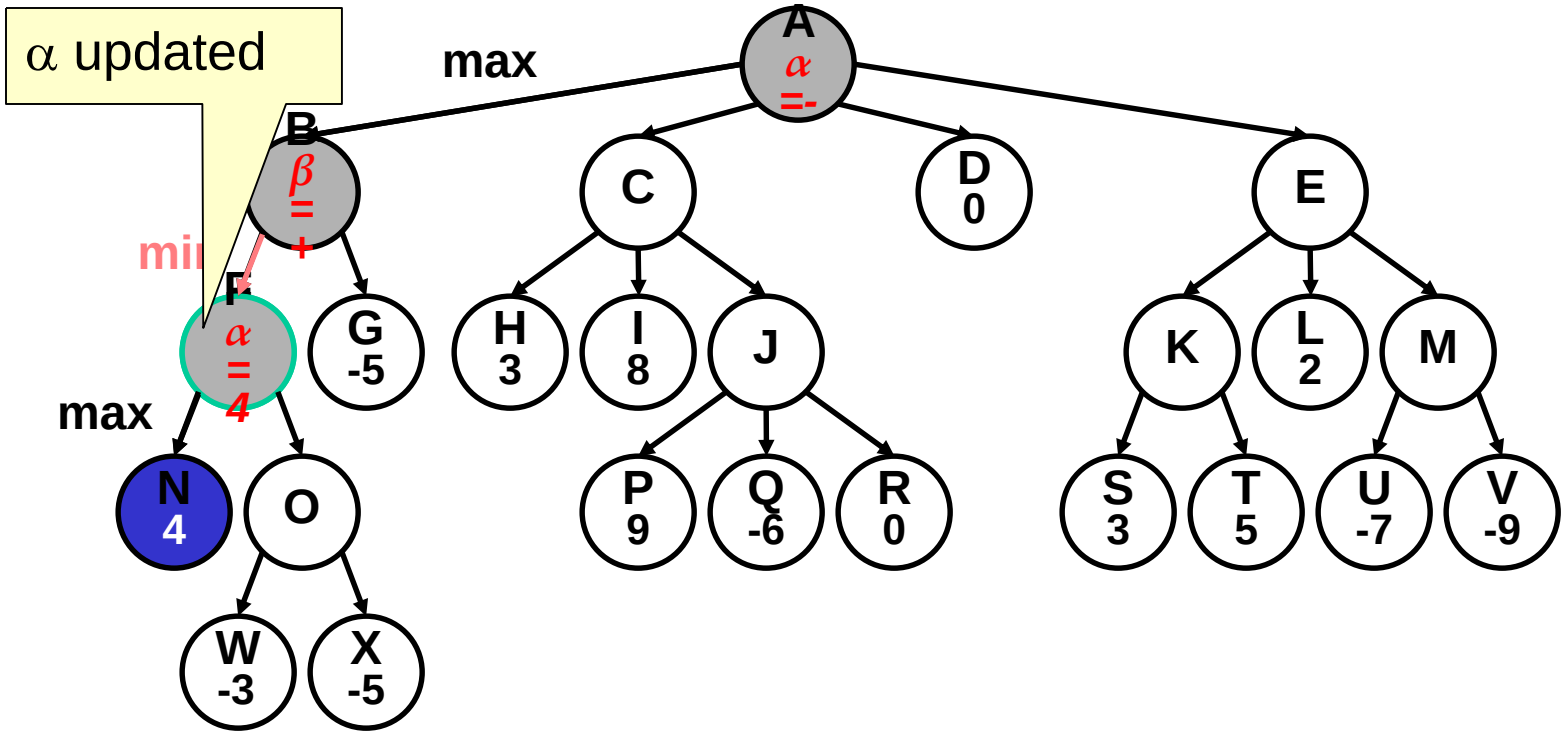
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

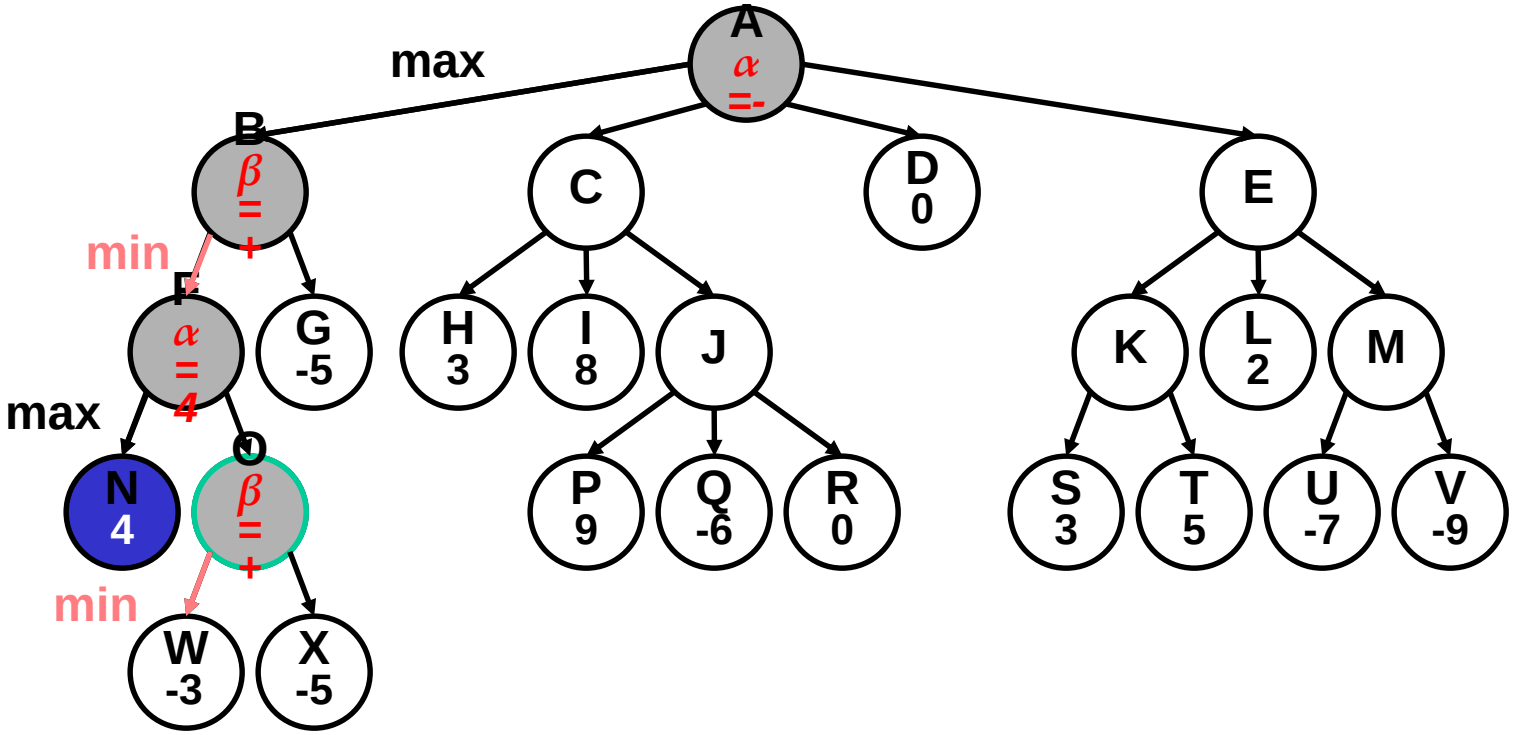
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

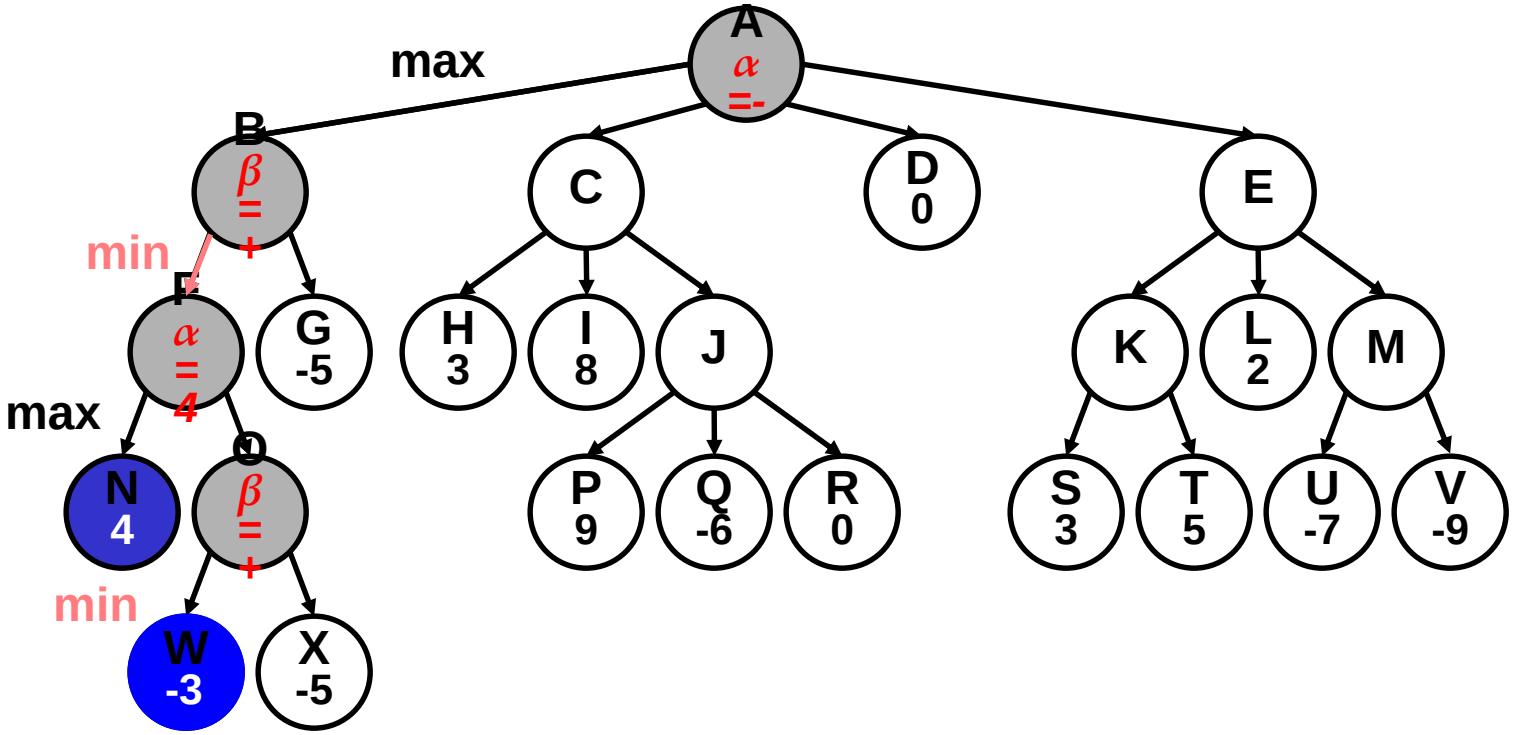
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

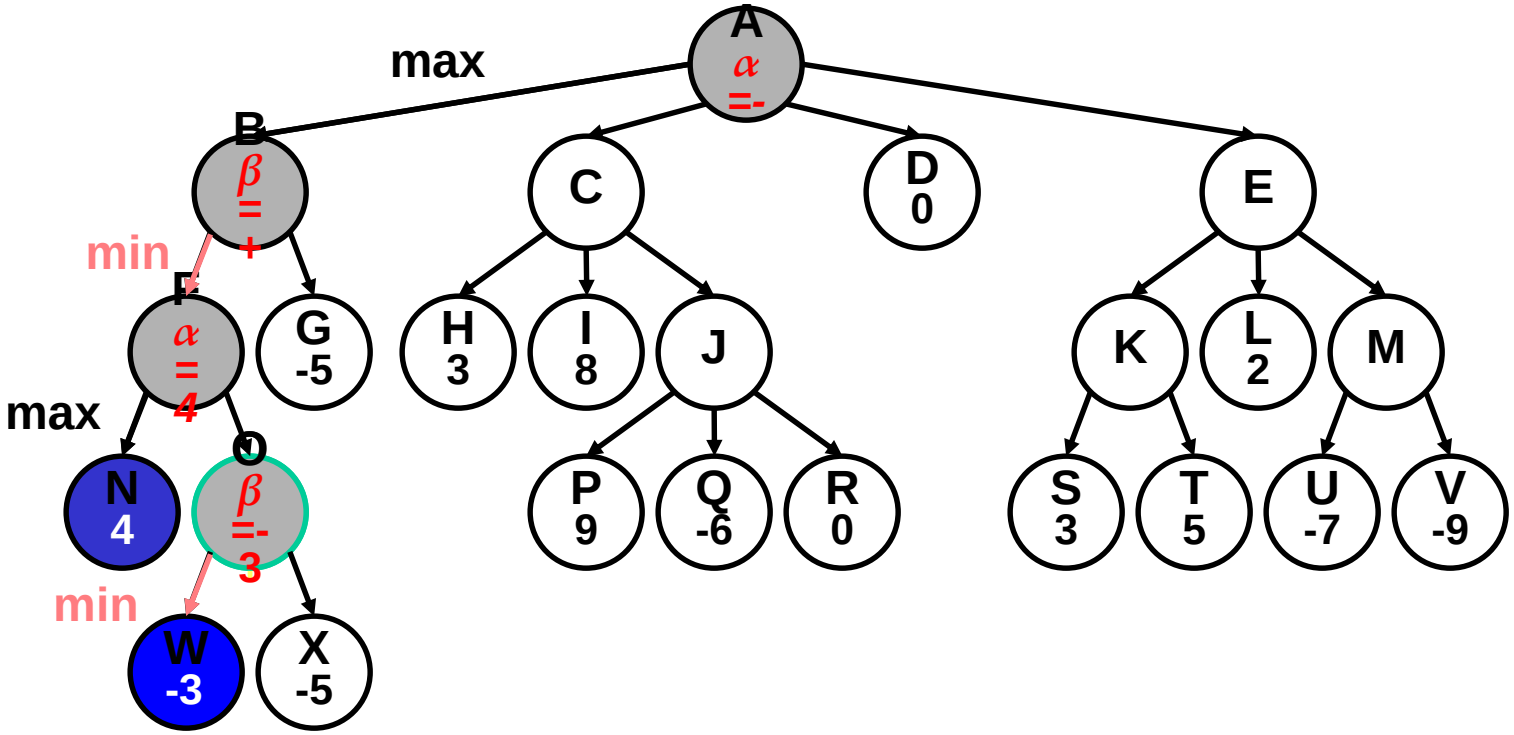
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

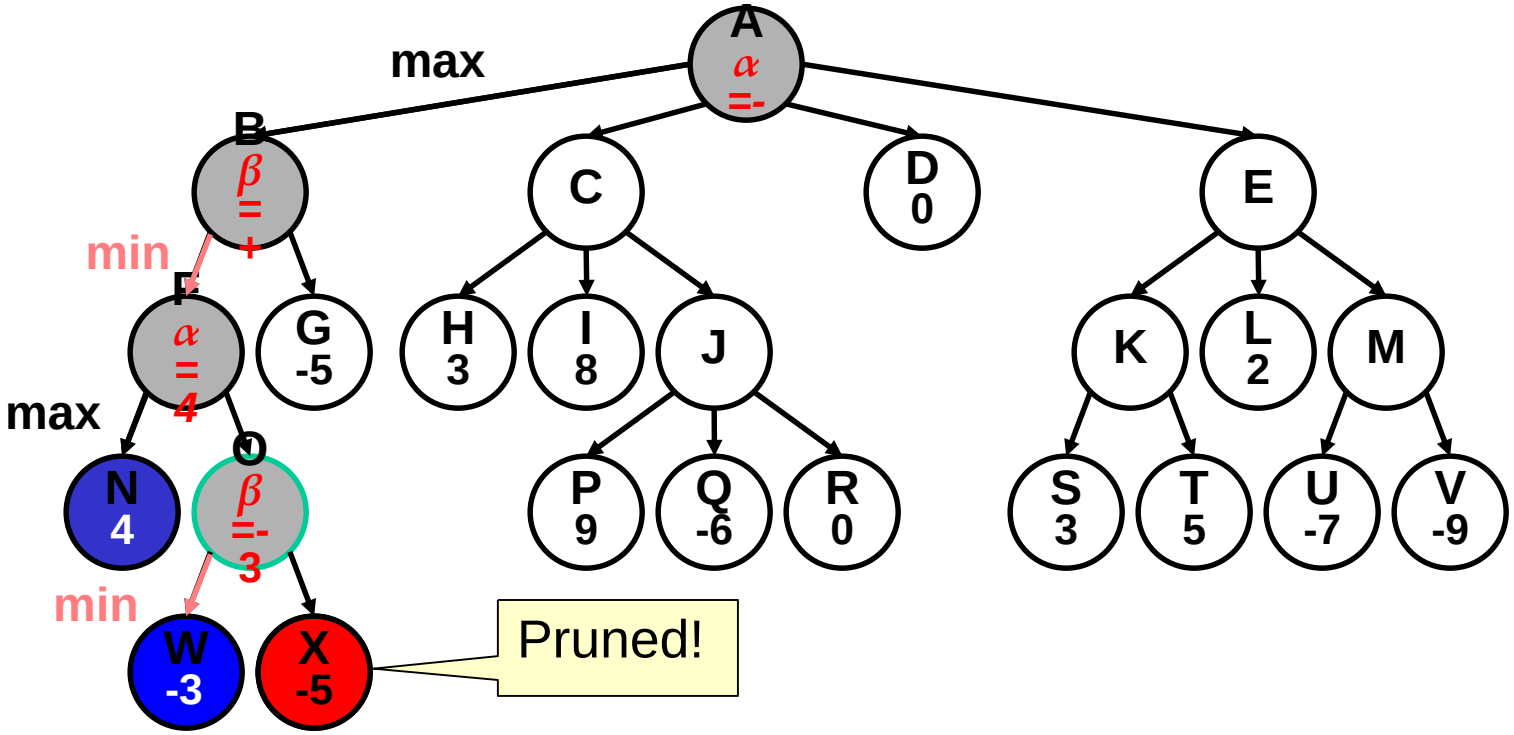
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

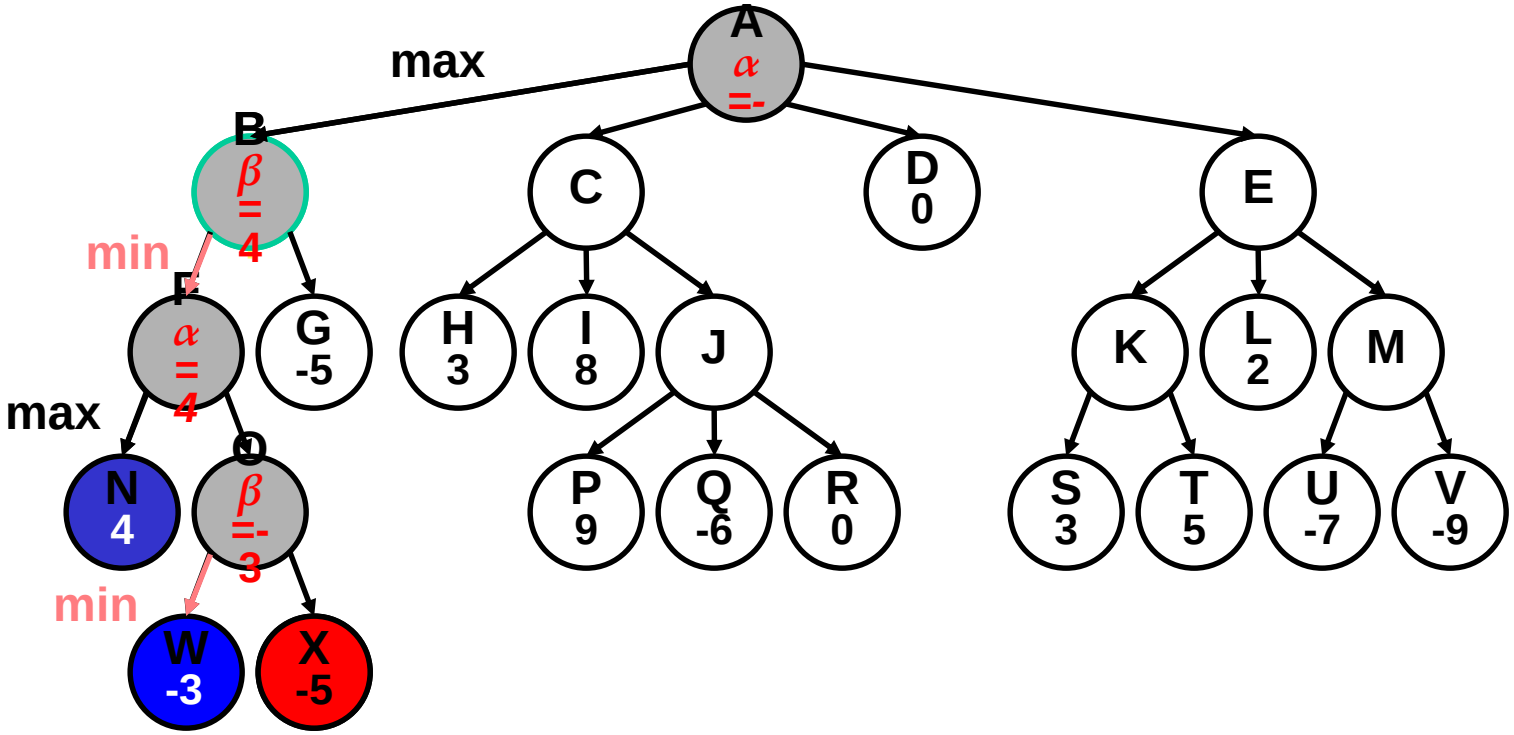
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

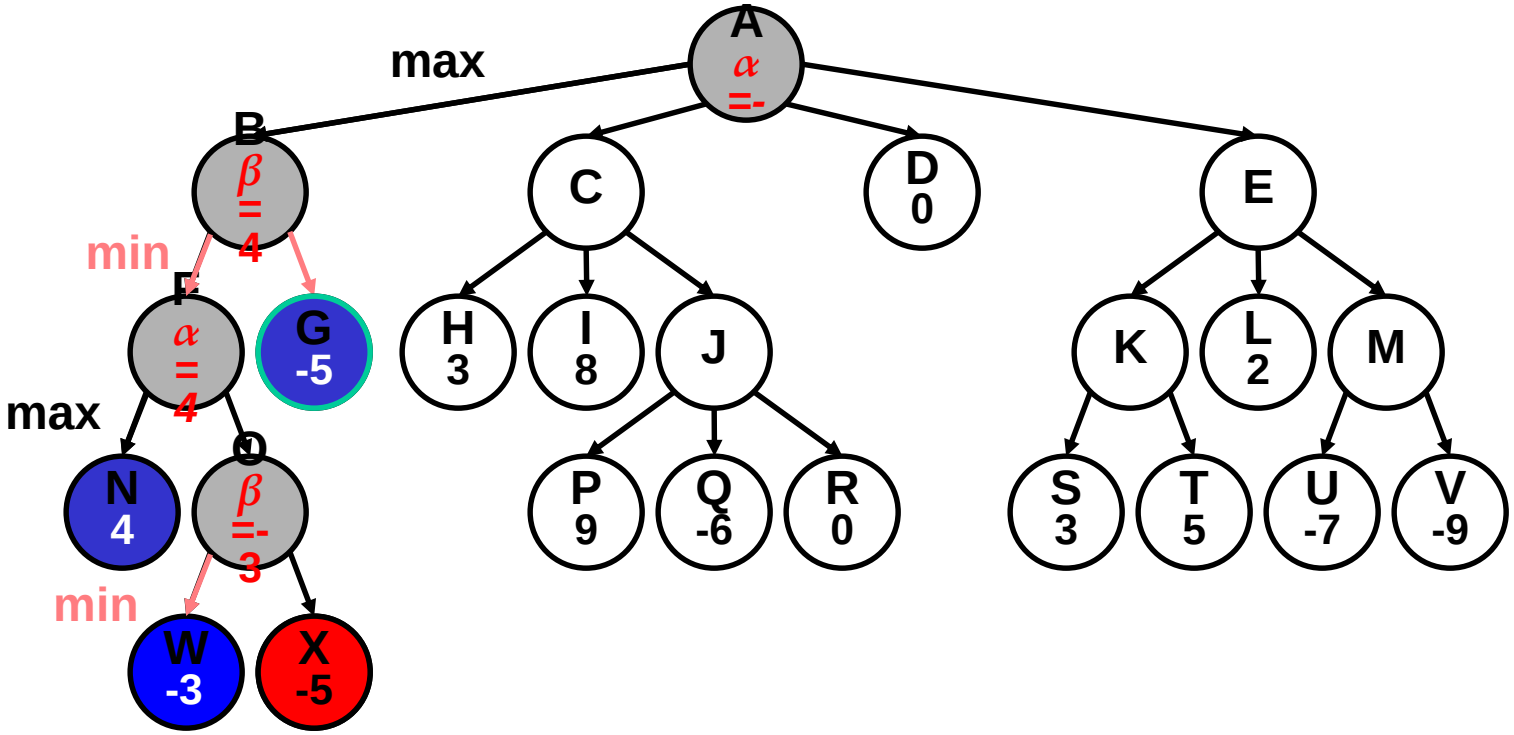
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

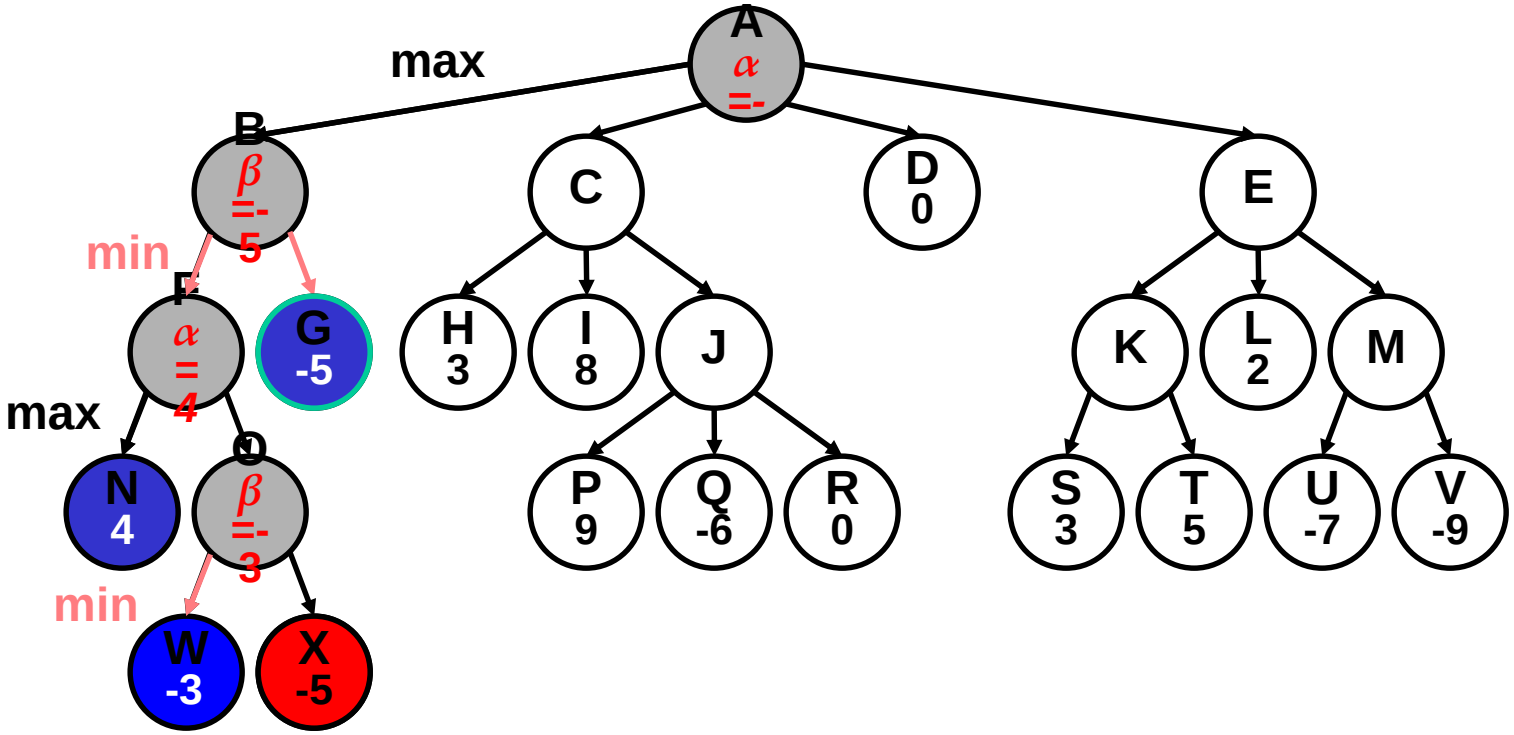
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

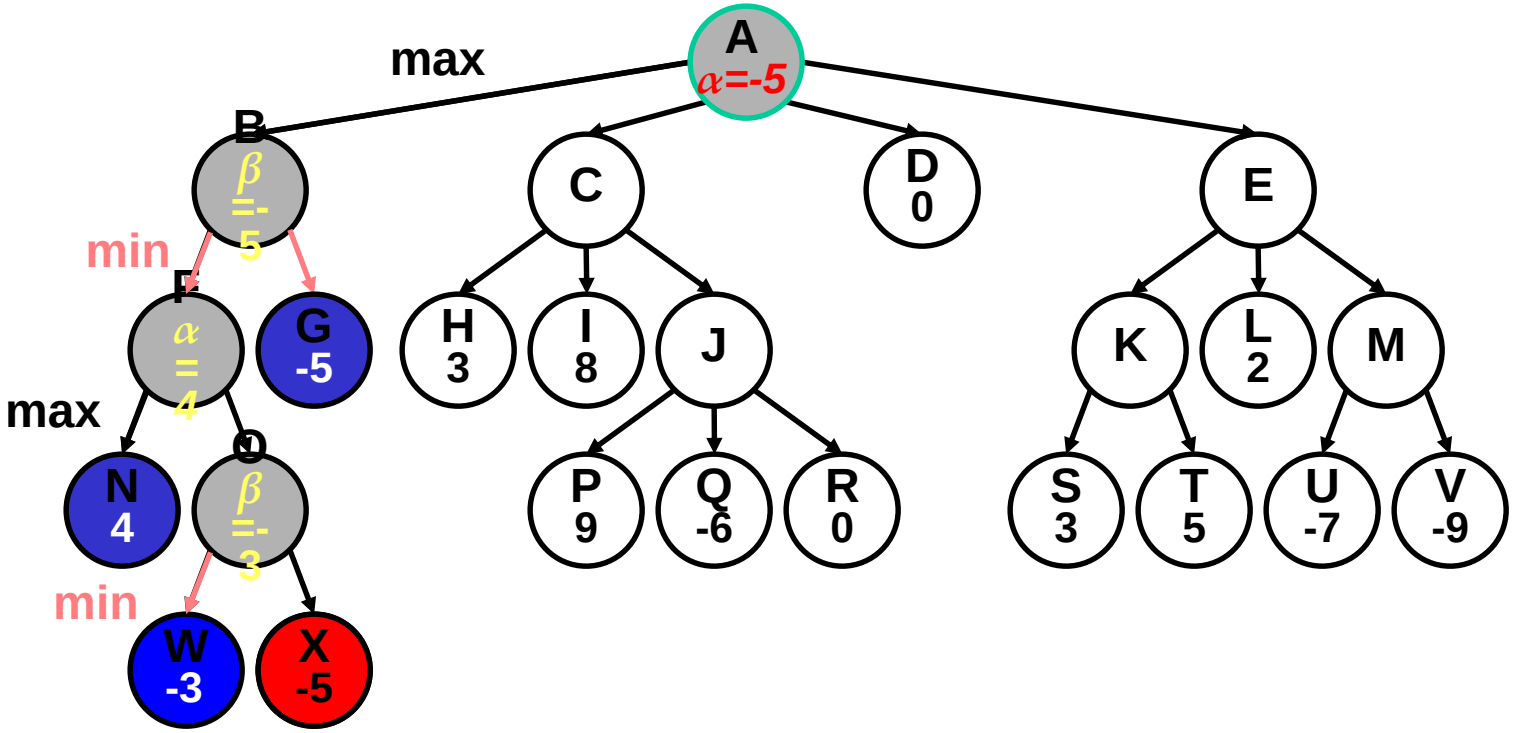
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

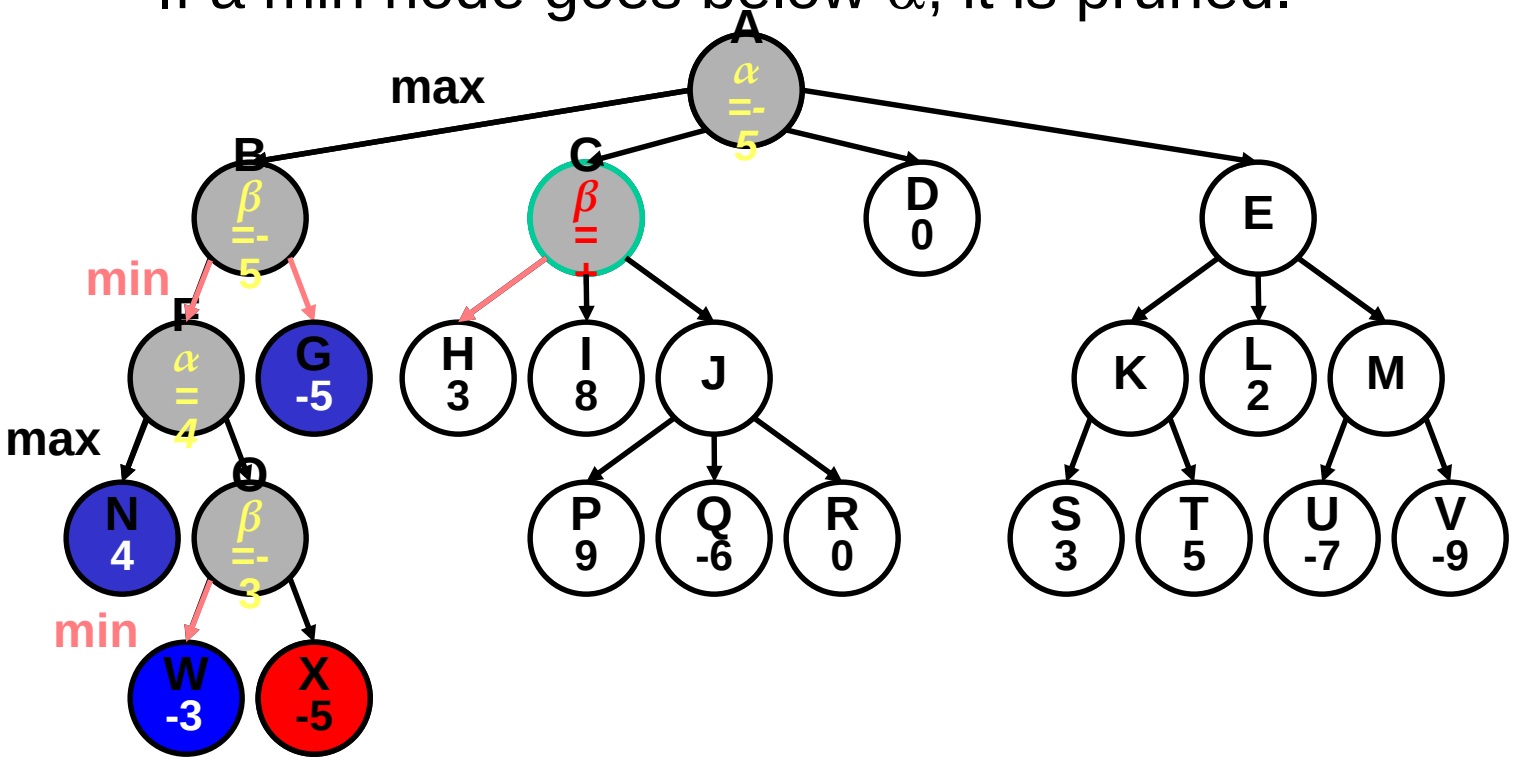
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

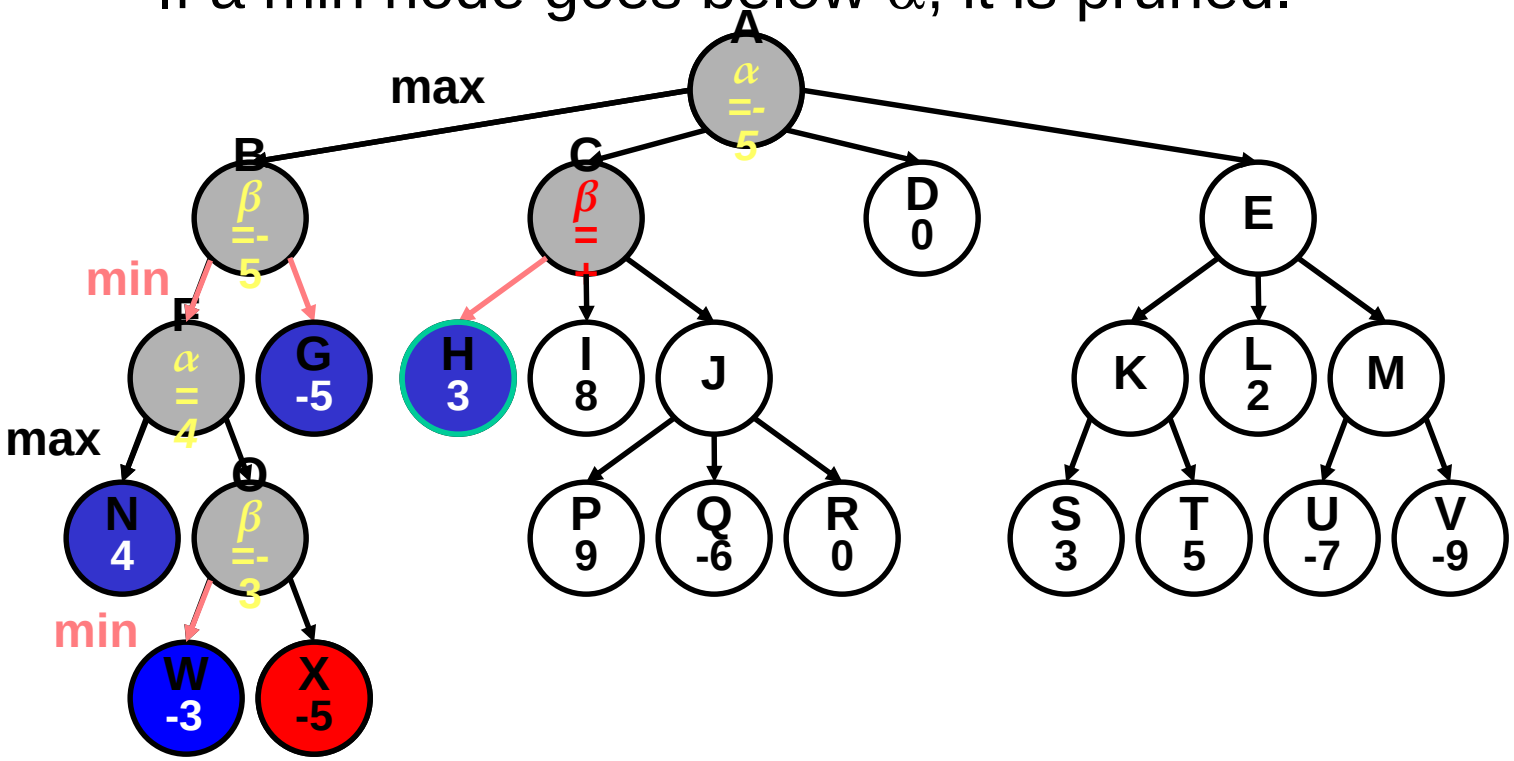
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

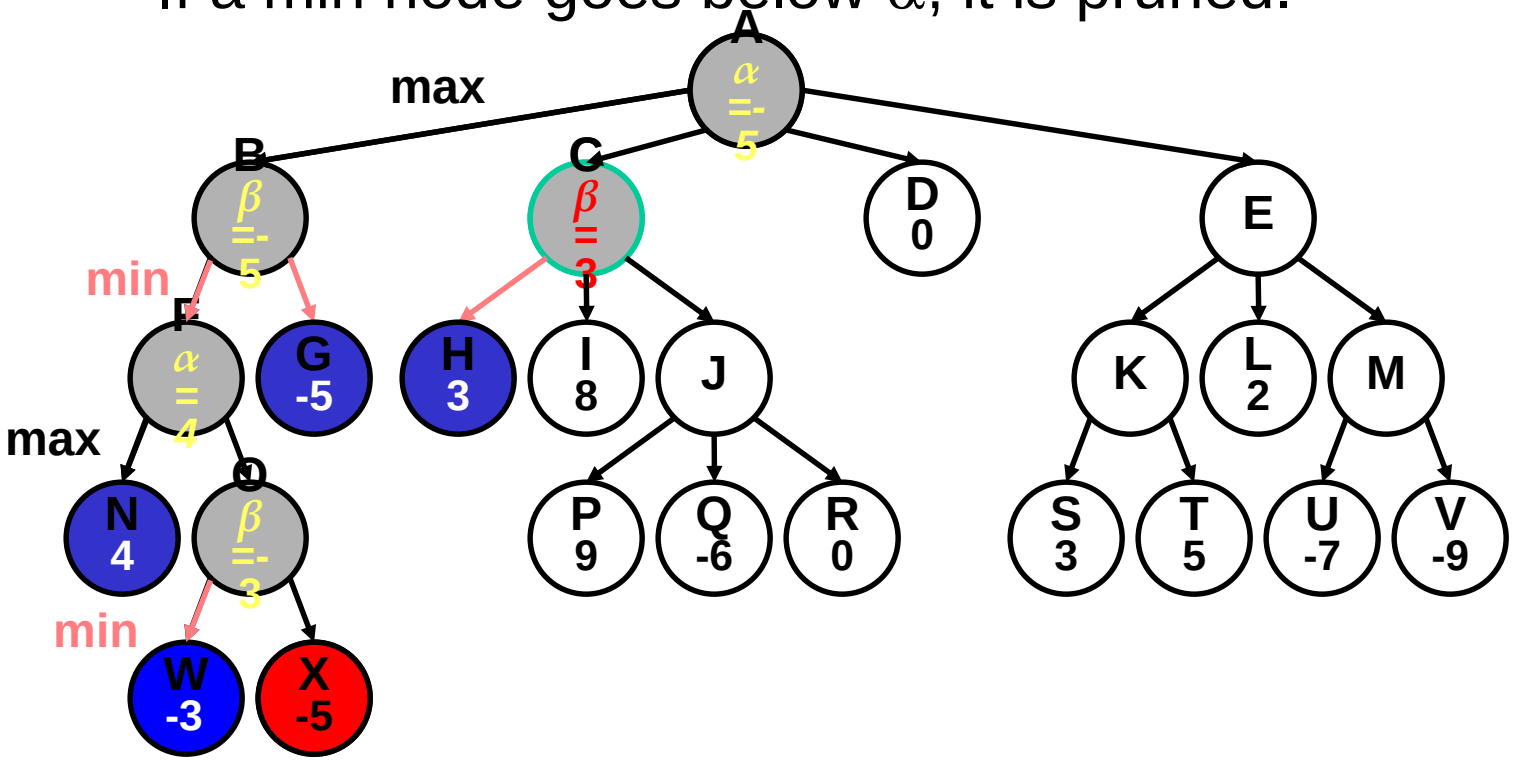
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

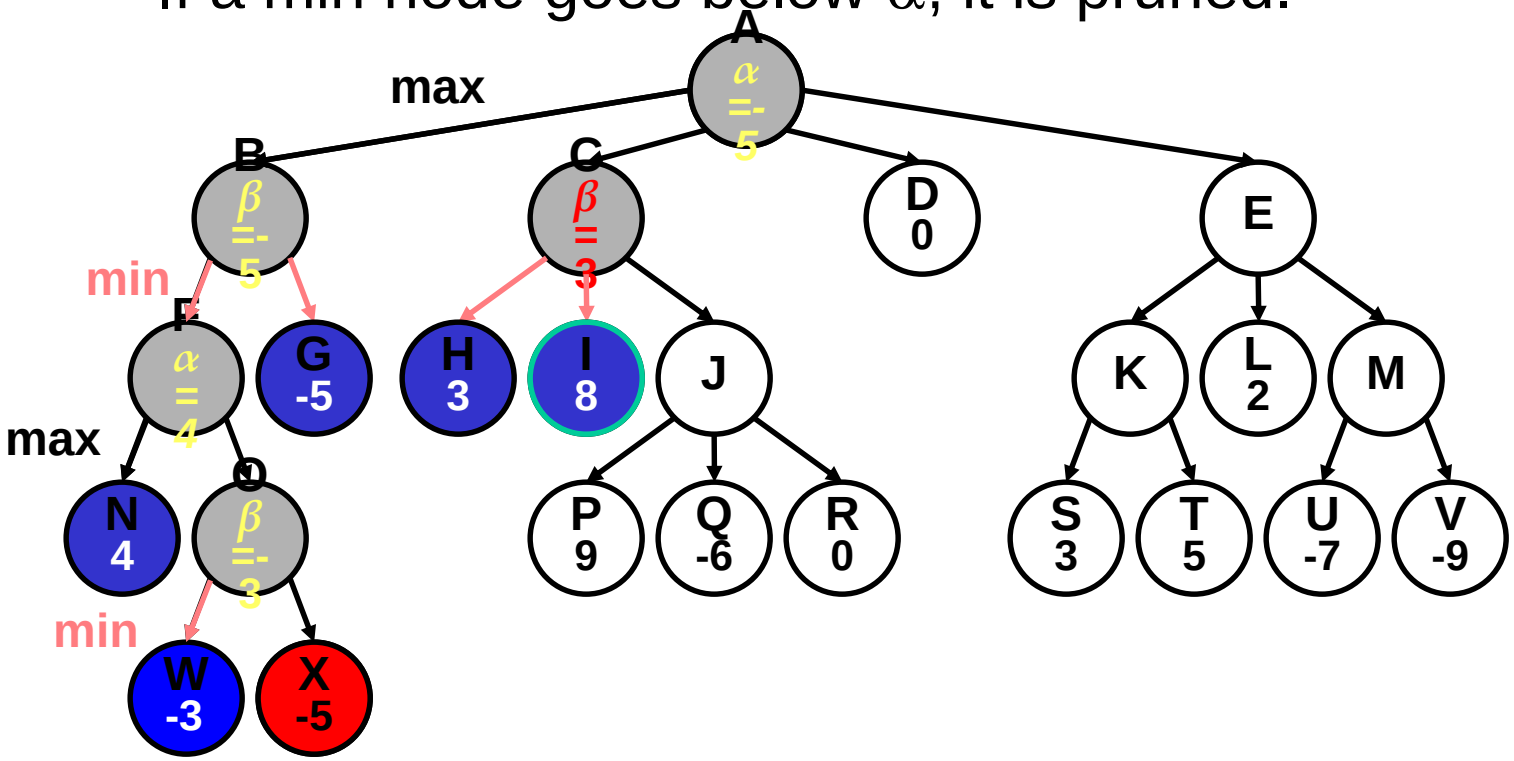
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

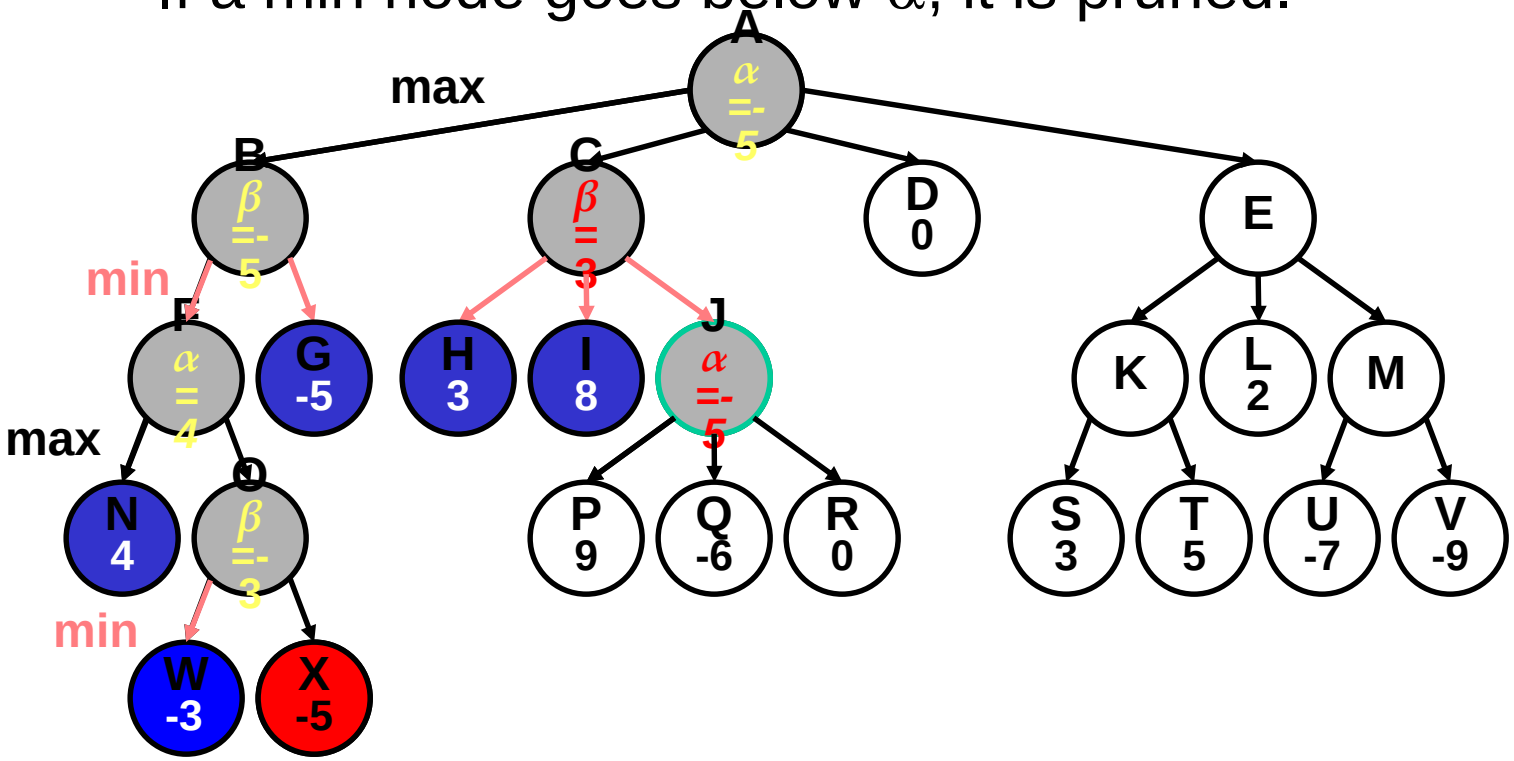
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

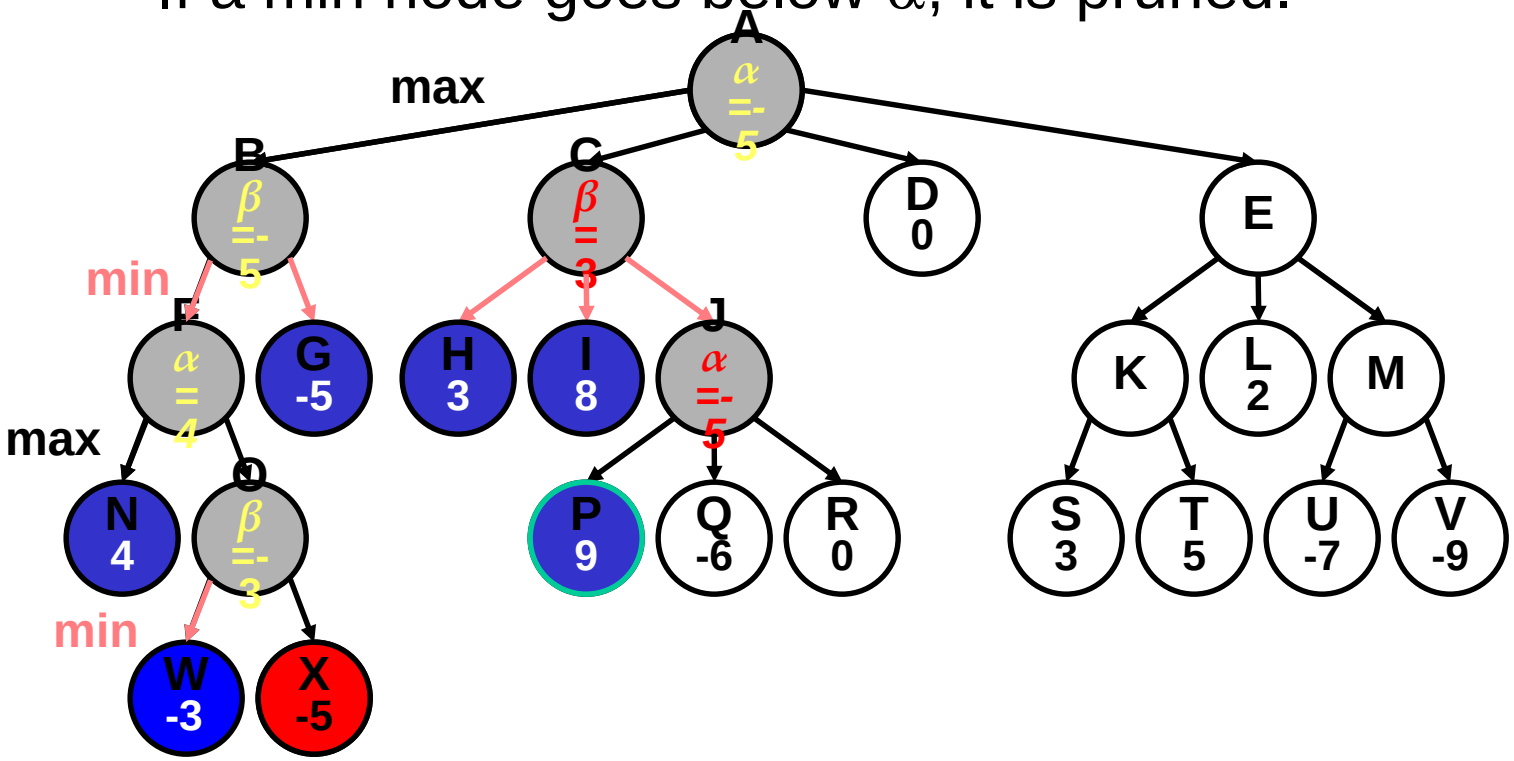
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

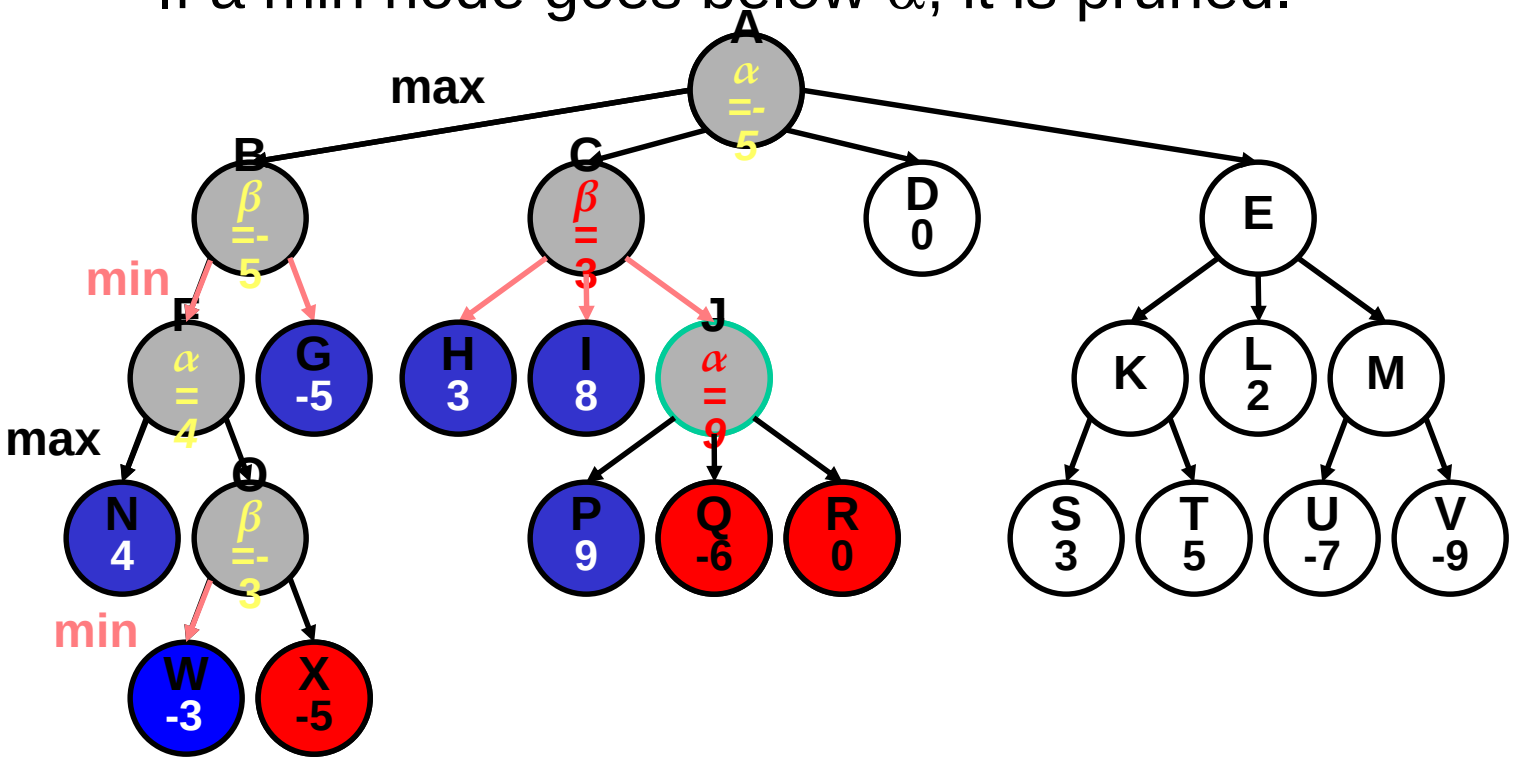
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

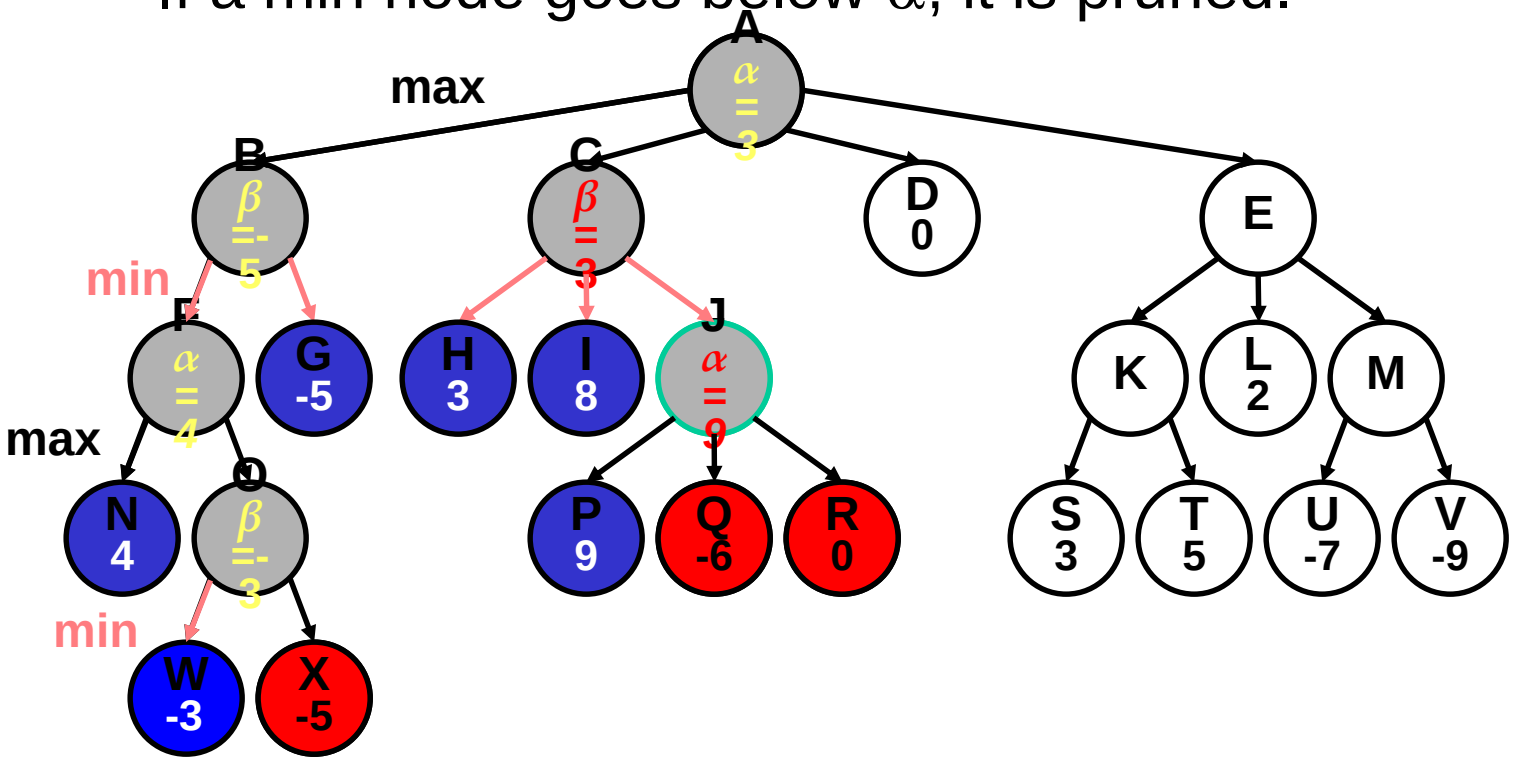
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

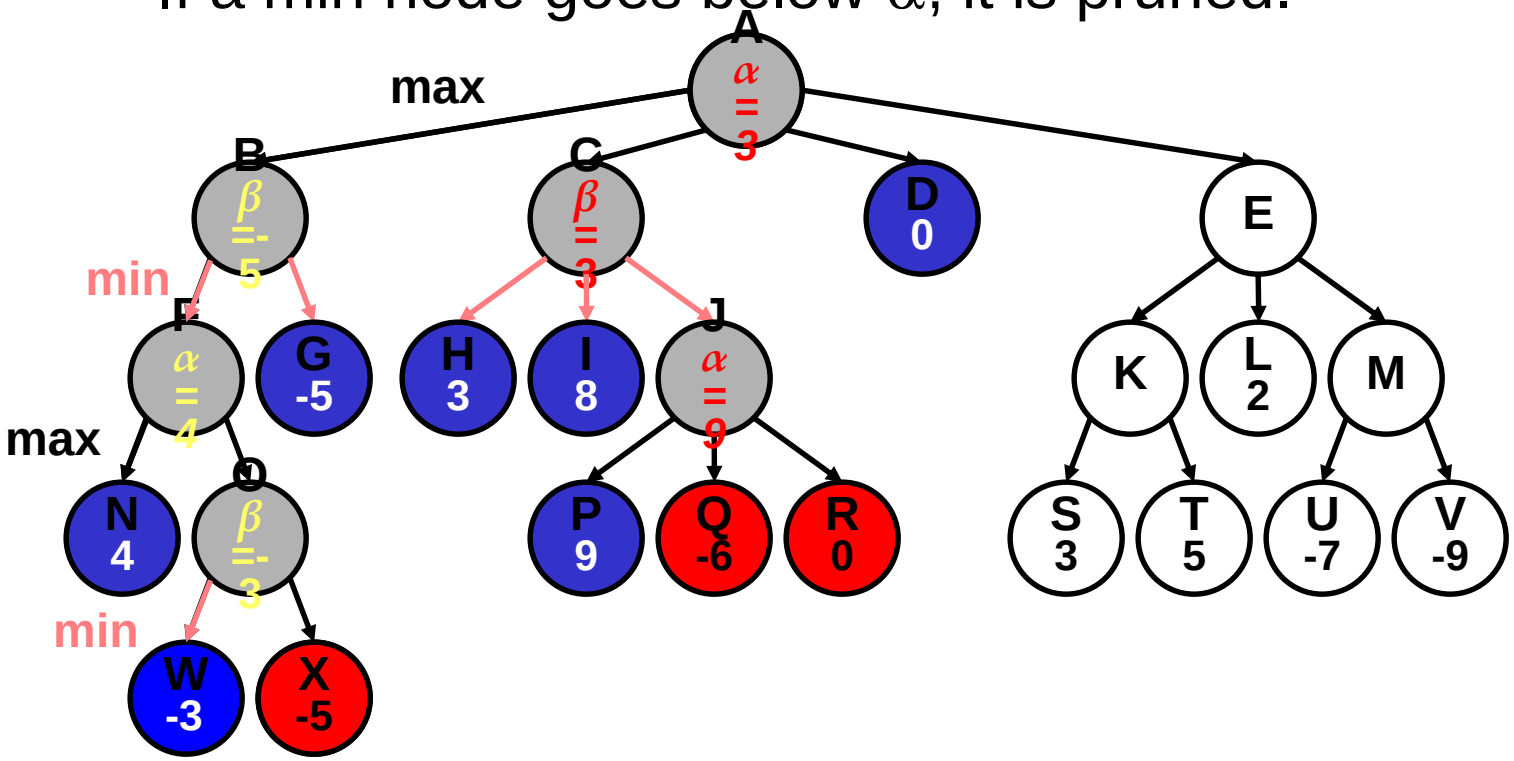
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

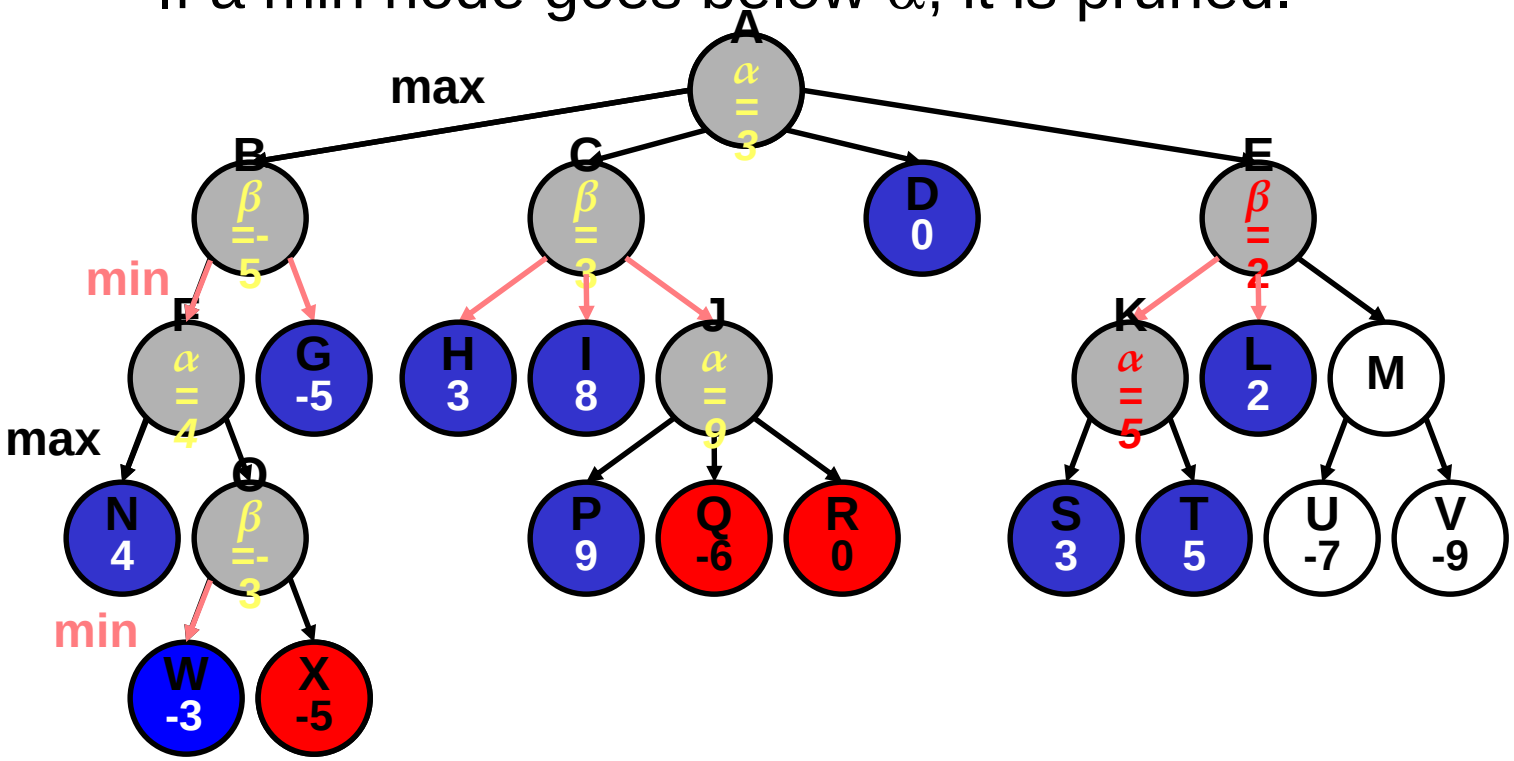
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

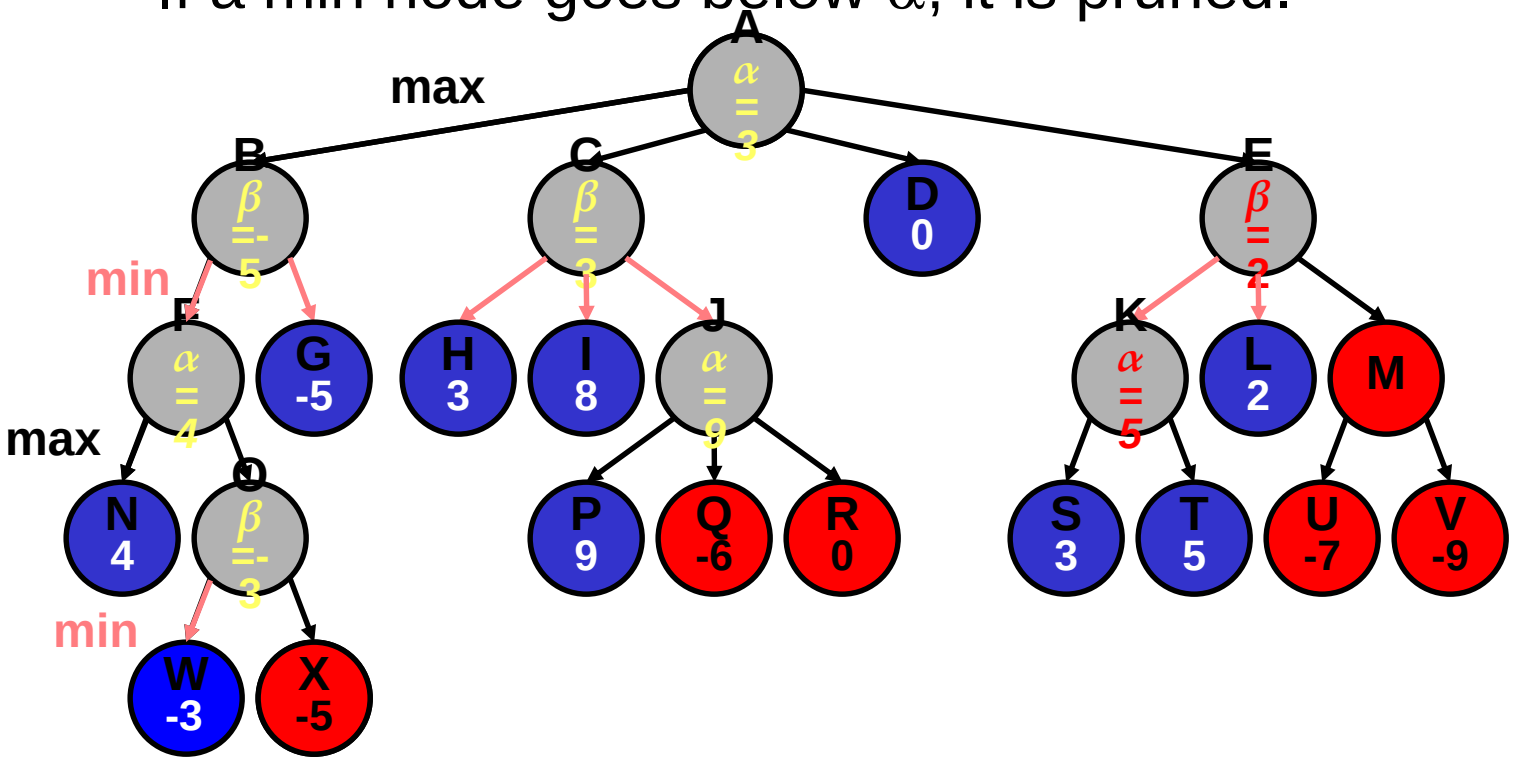
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

Alpha-beta pruning example

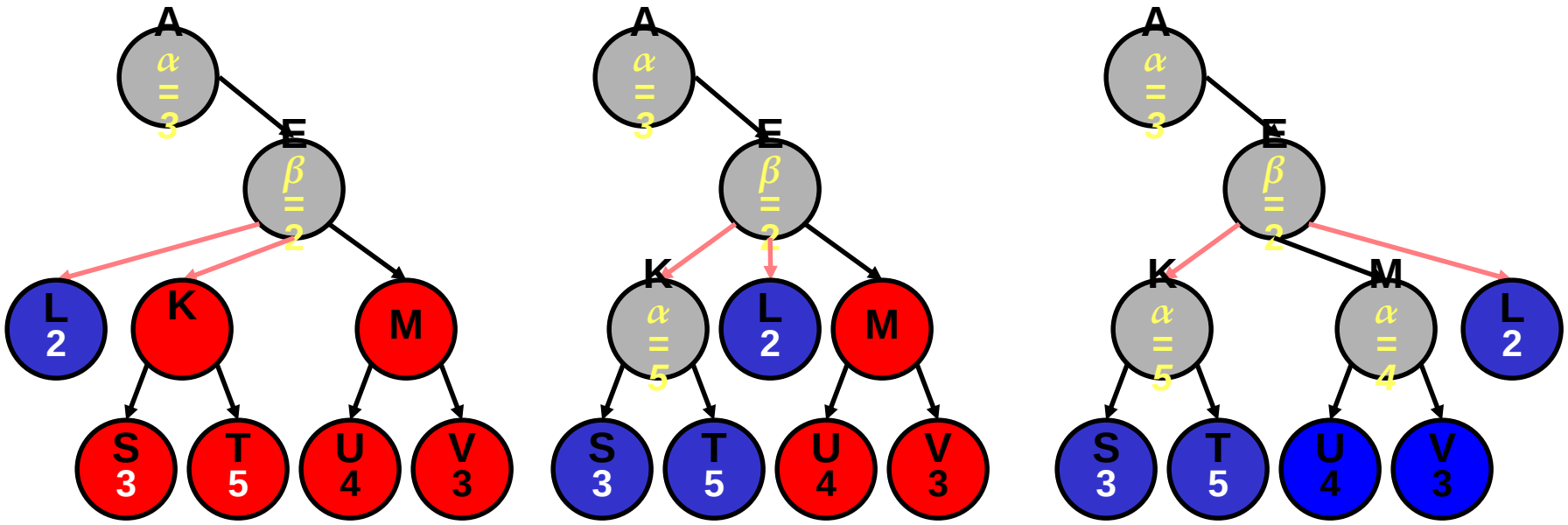
- Keep two bounds along the path
  - α: the best Max can do on the path
  - β: the best (smallest) Min can do on the path
- If a max node exceeds β, it is pruned.
- If a min node goes below α, it is pruned.



[Example from James Skrentny]

How effective is alpha-beta pruning?

- Depends on the order of successors!



- In the best case, the number of nodes to search is O($b^{m/2}$), the square root of minimax's cost.
- This occurs when each player's best move is the leftmost child.
- In DeepBlue (IBM Chess), the average branching factor was about 6 with alpha-beta instead of 35-40 without.
- The worst case is no pruning at all.