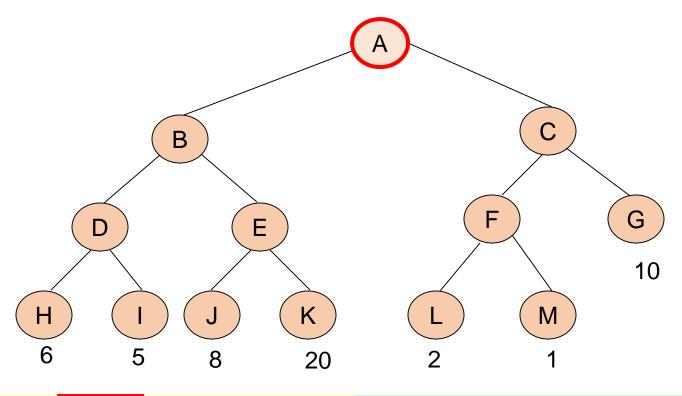
Example of running the Alpha-Beta algorithm

A trace of the program

Node α β Value Function

A: -∞: +∞: -∞ *MAXvalue*



function MAX-VALUE (state, α , β) **returns** a utility value **Inputs:** state, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (*state*, α , β) **returns** *a utility value* **Inputs:** *state*, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

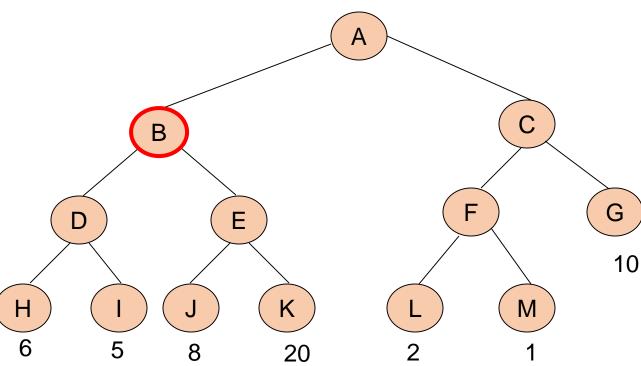
 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow \text{MIN}(\beta, v)$



function MAX-VALUE (state, α , β) **returns** a utility value

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

if $v \ge \beta$ then return v $\alpha \leftarrow MAX(\alpha, v)$

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

 α , the best value for MAX along the path to *state* β , the best value for MIN along the path to *state*

Inputs: state, current state in game

for a, s in SUCCESSORS[state] do

 $v \leftarrow -\infty$

return v

function MIN-VALUE (state, α , β) returns a utility value Inputs: state, current state in game α , the best value for MAX along the path to state β , the best value for MIN along the path to state if TERMINAL-TEST(state) then return UTILITY(state) $v \leftarrow +\infty$ for a, s in SUCCESSORS[state] do $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$ if $v \leq \alpha$ then return $v \in \beta \leftarrow \text{MIN}(\beta, v)$ return $v \in \beta \leftarrow \text{MIN}(\beta, v)$

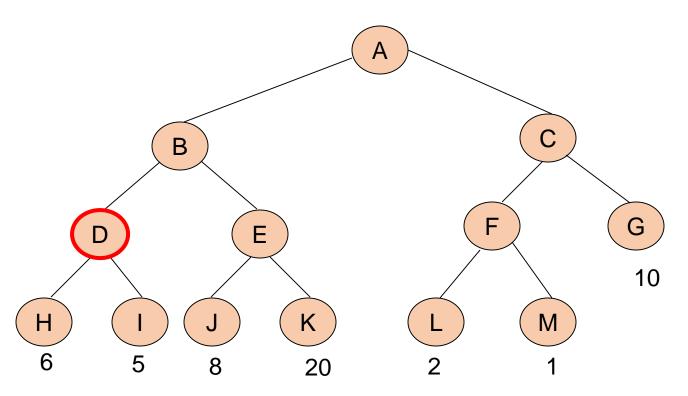
Node

A: -∞: +∞:

Value Function

-∞: +∞: +∞ *MINvalue*

-∞ MAXvalue



 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (*state*, α , β) **returns** *a utility value* **Inputs:** *state*, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$

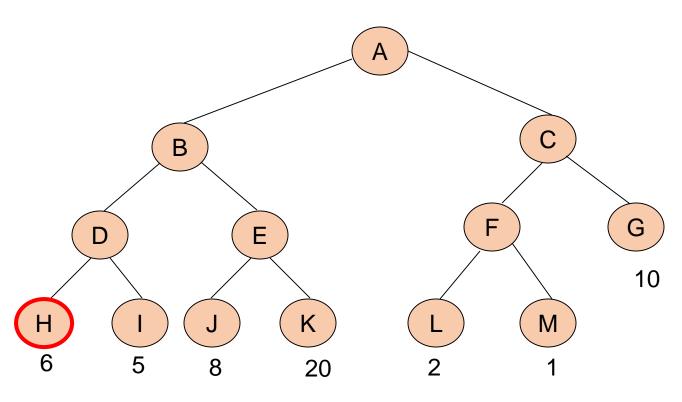
return v

Node α β Value Function

A: -∞: +∞: -∞ *MAXvalu*e

B: $-\infty$: $+\infty$: $+\infty$ *MINvalue*

D: -∞: +∞: -∞ *MAXvalue*



A: $-\infty$: $+\infty$: $-\infty$ *MAXvalue* B: $-\infty$: $+\infty$: $+\infty$ *MINvalue* D: $-\infty$: $+\infty$: $-\infty$ MAX value

Node

6:

6: 6

Value Function

function MAX-VALUE (state, α , β) **returns** a utility value **Inputs:** *state*, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (state, α , β) **returns** a utility value Inputs: state, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) then return UTILITY(state)

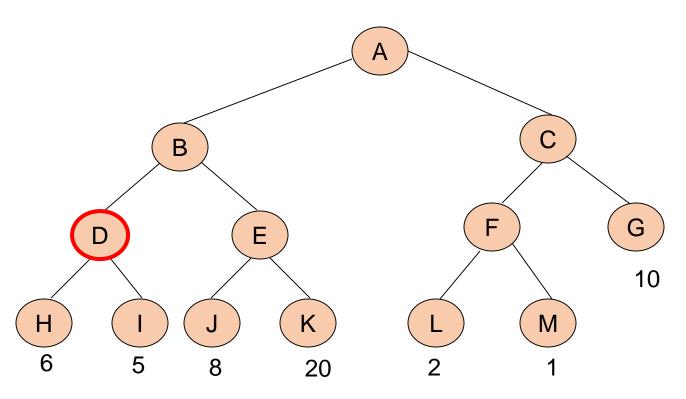
 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$



 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (*state*, α , β) **returns** *a utility value* **Inputs:** *state*, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow \text{MIN}(\beta, v)$

return v

Node α β Value Function

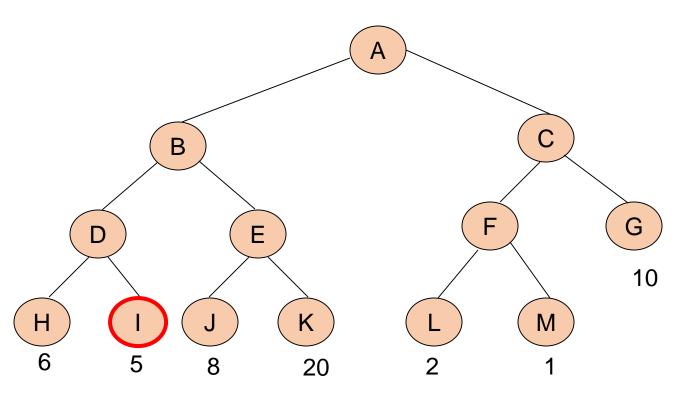
A: $-\infty$: $+\infty$: $-\infty$ *MAXvalue*

B: $-\infty$: $+\infty$: $+\infty$ *MINvalue*

D: $-\infty$: $+\infty$: $-\infty$ *MAXvalue*

H: 6: 6: 6

D: 6: +∞: 6 *MAXvalue*



A: $-\infty$: $+\infty$: $-\infty$ *MAXvalue* B: $-\infty$: $+\infty$: $+\infty$ *MINvalue* D: $-\infty$: $+\infty$: $-\infty$ MAX value 6: 6: +∞: 6 *MAXvalue* 5:

Node

β Value Function

6: 6

5: 5

function MAX-VALUE (state, α , β) **returns** a utility value **Inputs:** *state*, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (state, α , β) **returns** a utility value Inputs: state, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

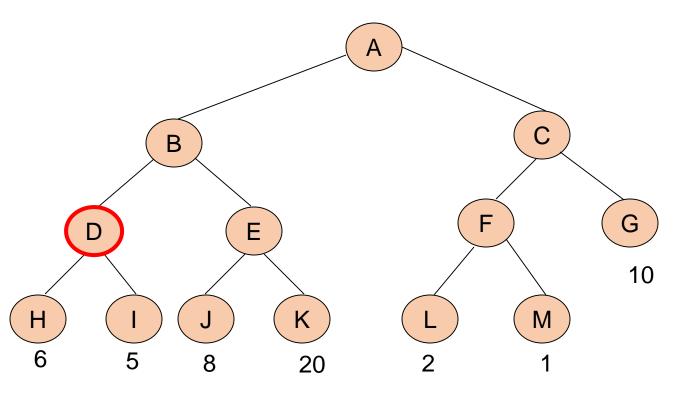
 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$



function MIN-VALUE (*state*, α , β) **returns** *a utility value* **Inputs:** *state*, current state in game α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$

return v

function MAX-VALUE (state, α , β) **returns** a utility value **Inputs:** state, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) then return UTILITY(state) $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

Node α β Value Function

A: $-\infty$: $+\infty$: $-\infty$ *MAXvalue*

B: $-\infty$: $+\infty$: $+\infty$ *MINvalue*

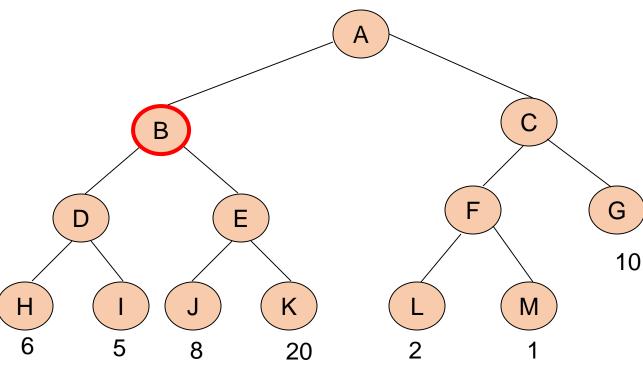
D: $-\infty$: $+\infty$: $-\infty$ *MAXvalue*

H: 6: 6: 6

D: $6: +\infty$: 6 *MAXvalue*

l: 5: 5: 5

D: 6: +∞: 6 *MAXvalue*



function MAX-VALUE (state, α , β) **returns** a utility value

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

Inputs: *state*, current state in game

 $v \leftarrow -\infty$

function MIN-VALUE (state, α , β) **returns** a utility value Inputs: state, current state in game α , the best value for MAX along the path to *state* β , the best value for MIN along the path to *state* **if** TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

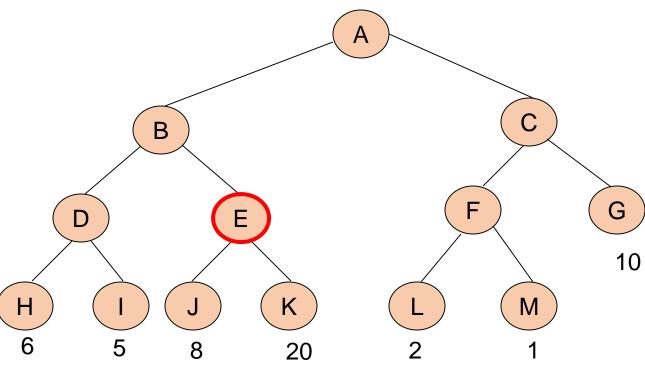
if TERMINAL-TEST(*state*) **then return** UTILITY(*state*) $v \leftarrow +\infty$ for a, s in SUCCESSORS[state] do for a, s in SUCCESSORS[state] do $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$ $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$ if $v \ge \beta$ then return vif $v \le \alpha$ then return v $\alpha \leftarrow \text{MAX}(\alpha, v)$ $\beta \leftarrow MIN(\beta, v)$ return v return v

Node **β** Value Function A: $-\infty$: $+\infty$: $-\infty$ *MAXvalue* B: $-\infty$: $+\infty$: $+\infty$ MINvalue D: $-\infty$: $+\infty$: $-\infty$ MAX value H: 6: 6: 6 6: +∞: 6 MAXvalue

5: 5: 5

D: 6: $+\infty$: 6 MAX value

B: **-**∞: 6: 6 MINvalue



H:

function MAX-VALUE (state, α , β) **returns** a utility value **Inputs:** state, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (*state*, α , β) **returns** *a utility value* **Inputs:** *state*, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$

return v

Node α β Value Function

A: $-\infty$: $+\infty$: $-\infty$ *MAXvalue*

B: $-\infty$: $+\infty$: $+\infty$ *MINvalue*

D: $-\infty$: $+\infty$: $-\infty$ *MAXvalue*

H: 6: 6: 6

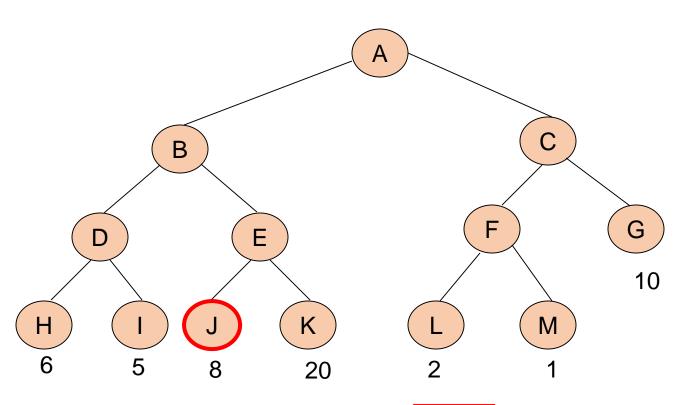
D: $6: +\infty$: 6 *MAXvalue*

l: 5: 5: 5

D: 6: $+\infty$: 6 MAX value

B: **-**∞: 6: 6 *MINvalue*

E: $-\infty$: 6: $-\infty$ MAX value



A: $-\infty$: $+\infty$: $-\infty$ *MAXvalue* B: $-\infty$: $+\infty$: $+\infty$ *MINvalue* D: $-\infty$: $+\infty$: $-\infty$ MAX value H: 6: 6: 6 6: +∞: 6 MAXvalue 5: 5: 5 6: +∞: 6 *MAXvalue* B: **-**∞: 6: 6 MINvalue 6: -∞ MAXvalue 8: 8

β Value Function

Node

function MAX-VALUE (*state*, α , β) **returns** *a utility value* **Inputs:** *state*, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (state, α , β) **returns** a utility value **Inputs**: state, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) then return UTILITY(state)

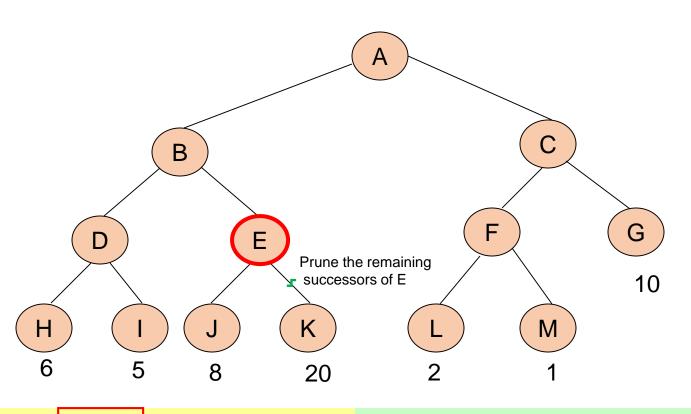
 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

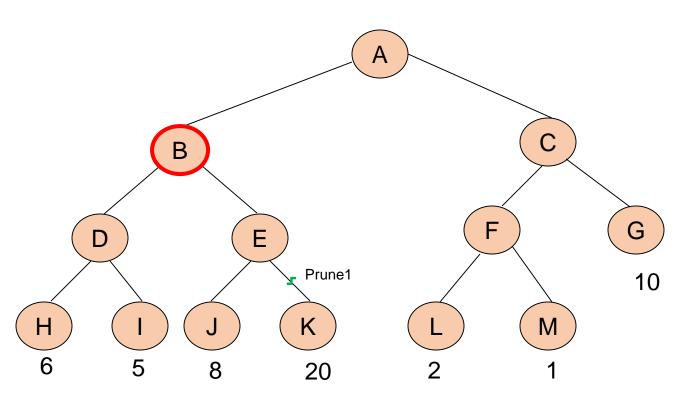
if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$



```
Node
            β Value Function
A: -∞: +∞:
             -∞ MAXvalue
B: -\infty: +\infty: +\infty MINvalue
D: -\infty: +\infty: -\infty MAX value
     6:
          6:
             6
     6: +∞:
              6 MAXvalue
    5:
          5:
              5
              6 MAXvalue
B: -∞:
             6 MINvalue
         6: -∞ MAXvalue
         8: 8
PRUNE1(β):E:-∞:6:8 MAXvalue
```

```
function MAX-VALUE (state, \alpha, \beta) returns a utility value
                                                                             function MIN-VALUE (state, \alpha, \beta) returns a utility value
Inputs: state, current state in game
                                                                             Inputs: state, current state in game
                     \alpha, the best value for MAX along the path to state
                                                                                                   \alpha, the best value for MAX along the path to state
                     \beta, the best value for MIN along the path to state
                                                                                                   \beta, the best value for MIN along the path to state
   if TERMINAL-TEST(state) then return UTILITY(state)
                                                                                if TERMINAL-TEST(state) then return UTILITY(state)
   v \leftarrow -\infty
                                                                                v \leftarrow +\infty
   for a, s in SUCCESSORS[state] do
                                                                                 for a, s in SUCCESSORS[state] do
                     v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))
                                                                                                  v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))
                    if v \ge \beta then return v
                                                                                                   if v \le \alpha then return v
                     \alpha \leftarrow \text{MAX}(\alpha, v)
                                                                                                   \beta \leftarrow \text{MIN}(\beta, v)
   return v
                                                                              return v
                                                Beta-prunning
                              Prune all
              remaining successors
```



```
Node
            β Value Function
A: -\infty: +\infty: -\infty MAX value
B: -\infty: +\infty: +\infty MINvalue
D: -\infty: +\infty: -\infty MAX value
H:
     6:
          6: 6
     6: +\infty: 6 MAX value
    5:
          5:
               5
    6: +∞: 6 MAXvalue
B: -∞:
          6: 6 MINvalue
E: -\infty: 6: -\infty MAX value
J: 8:
          8: 8
PRUNE1:E:-∞:6:8 MAXvalue
B:-∞:
         6: 6 MINvalue
```

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (state, α , β) **returns** a utility value **Inputs**: state, current state in game

 α , the best value for MAX along the path to state

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

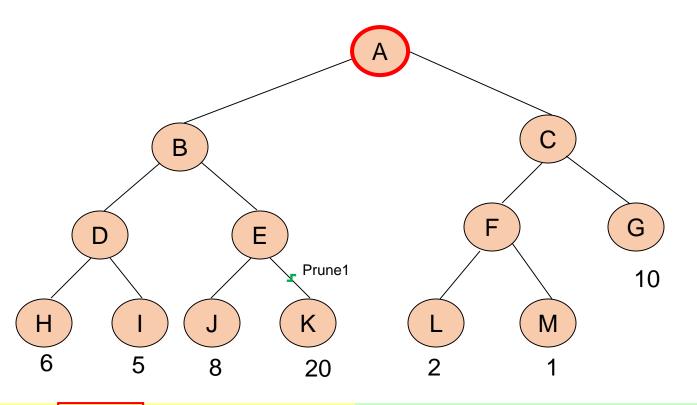
 $v \leftarrow +\infty$

for *a, s* **in** SUCCESSORS[*state*] **do**

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow \text{MIN}(\beta, v)$



```
Node
               Value Function
               -∞ MAXvalue
 B: -\infty: +\infty: +\infty MINvalue
 \bigcap \cdot -\infty \cdot +\infty : -\infty MAXvalue
 H:
      6:
            6: 6
      6: +\infty: 6 MAX value
      5:
           5:
                5
     6: +∞: 6 MAXvalue
 B: -∞:
           6: 6 MINvalue
           6: -∞ MAXvalue
 J: 8:
           8: 8
 PRUNE1:E:-∞:6:8 MAXvalue
 B:-∞:
           6: 6 MINvalue
 A: 6: +∞:
               6 MAXvalue
```

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (*state*, α , β) **returns** *a utility value* **Inputs:** *state*, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

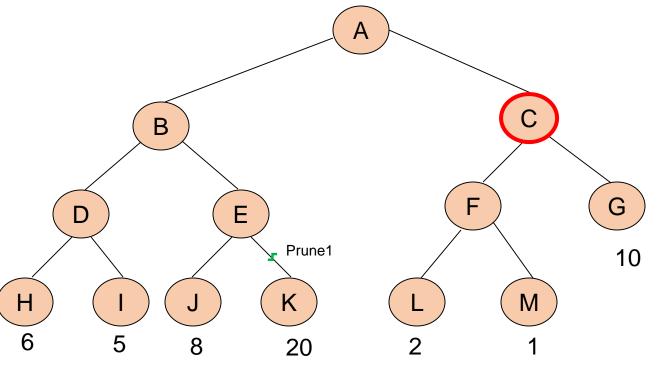
 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$



6: +∞: 6 *MAXvalue* B: **-**∞: 6: 6 MINvalue 6: -∞ MAXvalue 8: 8: 8 PRUNE1:E:-∞:6:8 MAXvalue B:**-**∞: 6: 6 MINvalue A: 6: +∞: 6 MAXvalue C: 6: +∞: +∞ *MINvalue* **function** MIN-VALUE (state, α , β) **returns** a utility value α , the best value for MAX along the path to *state* β , the best value for MIN along the path to *state* **if** TERMINAL-TEST(*state*) **then return** UTILITY(*state*) $v \leftarrow +\infty$

Node

6:

5:

β Value Function

A: $-\infty$: $+\infty$: $-\infty$ *MAX value* B: $-\infty$; $+\infty$; $+\infty$ MINvalue

D: $-\infty$: $+\infty$: $-\infty$ MAX value

6: 6

5:

6: +∞: 6 *MAXvalue*

5

function MAX-VALUE (state, α , β) **returns** a utility value **Inputs:** *state*, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

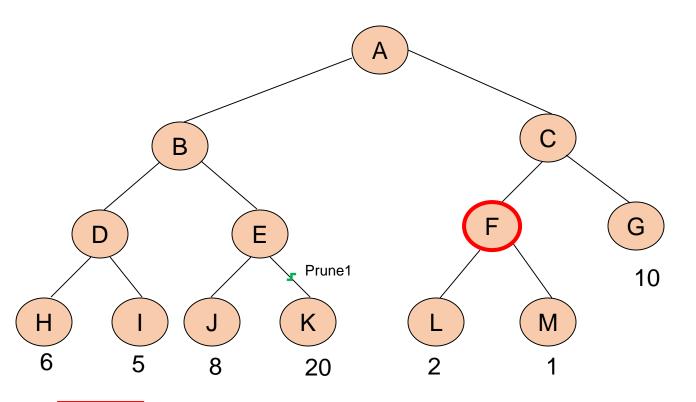
Inputs: state, current state in game

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$



 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (state, α , β) **returns** a utility value **Inputs:** *state*, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$

return v

Node **β** Value Function

A: $-\infty$: $+\infty$: $-\infty$ MAX value

B: $-\infty$: $+\infty$: $+\infty$ *MINvalue*

 $\cap -\infty + \infty - \infty$ MAX value

6: 6: 6

6: $+\infty$: 6 MAX value

5: 5:

6: $+\infty$: 6 MAX value

B: **-**∞: 6: 6 MINvalue

F. **-**∞. 6: -∞ MAXvalue

J: 8: 8: 8

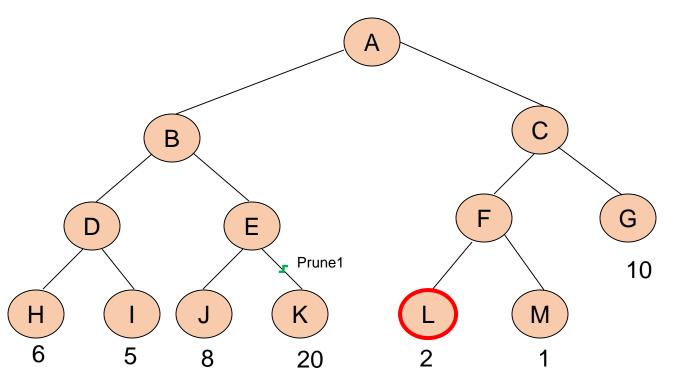
PRUNE1:E:-∞:6:8 MAXvalue

B:**-**∞: 6: 6 MINvalue

A: 6: $+\infty$: 6 MAX value

C: 6: $+\infty$: $+\infty$ MINvalue

F: 6: +∞: -∞_*MAXvalue*



 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow -\infty$

for *a, s* **in** SUCCESSORS[*state*] **do**

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (state, α , β) **returns** a utility value **Inputs**: state, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow \text{MIN}(\beta, v)$

return v

Node α β Value Function

A: -∞: +∞: -∞ MAXvalue

B: $-\infty$: $+\infty$: $+\infty$ *MINvalue*

D: $-\infty$: $+\infty$: $-\infty$ *MAXvalue*

H: 6: 6: 6

D: $6: +\infty$: 6 *MAXvalue*

l: 5: 5: 5

D: 6: +∞: 6 *MAXvalue*

B: **-**∞: 6: 6 *MINvalue*

E: $-\infty$: 6: $-\infty$ MAX value

J: 8: 8: 8

PRUNE1:E:-∞:6:8 MAXvalue

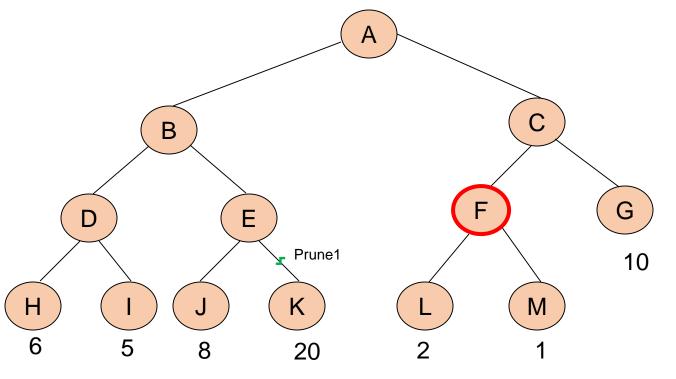
B:-∞: 6: 6 *MINvalue*

A: $6: +\infty$: 6 *MAXvalue*

C: 6: $+\infty$: $+\infty$ MINvalue

F: 6: $+\infty$: $-\infty$ *MAXvalue*

L: 2: 2: 2



 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (state, α , β) **returns** a utility value **Inputs:** state, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to state

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$

return v

Node α β Value Function

A: -∞: +∞: -∞ MAXvalue

B: $-\infty$: $+\infty$: $+\infty$ *MINvalue*

D: $-\infty$: $+\infty$: $-\infty$ MAX value

H: 6: 6: 6

D: $6: +\infty$: 6 *MAXvalue*

l: 5: 5: 5

D: 6: +∞: 6 *MAXvalue*

B: **-**∞: 6: 6 *MINvalue*

E: $-\infty$: 6: $-\infty$ MAX value

J: 8: 8: 8

PRUNE1:E:-∞:6:8 MAXvalue

B:**-**∞: 6: 6 *MINvalue*

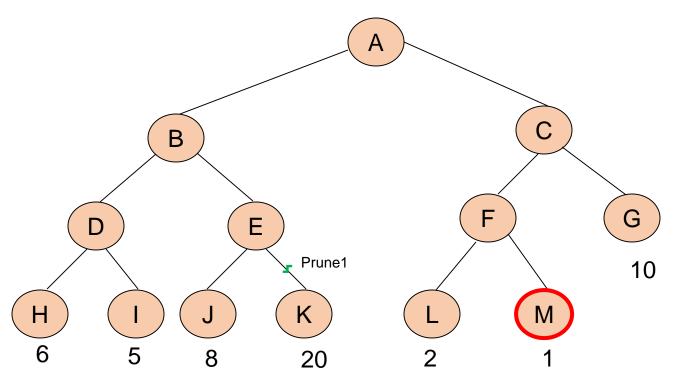
A: $6: +\infty$: 6 *MAXvalue*

C: 6: $+\infty$: $+\infty$ MINvalue

F: 6: $+\infty$: $-\infty$ *MAXvalue*

L: 2: 2: 2

F: 6: +∞: 2 *MAXvalue*



 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (state, α , β) **returns** a utility value **Inputs**: state, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) then return UTILITY(state)

 $v \leftarrow +\infty$

for *a, s* **in** SUCCESSORS[*state*] **do**

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$

return v

Node α β Value Function

A: -∞: +∞: -∞ MAXvalue

B: $-\infty$: $+\infty$: $+\infty$ *MINvalue*

D: $-\infty$: $+\infty$: $-\infty$ *MAXvalue*

H: 6: 6: 6

D: 6: **+**∞: 6 *MAXvalue*

l: 5: 5: 5

D: 6: $+\infty$: 6 MAX value

B: **-**∞: 6: 6 *MINvalue*

E: $-\infty$: 6: $-\infty$ MAX value

J: 8: 8: 8

PRUNE1:E:-∞:6:8 MAXvalue

B:**-**∞: 6: 6 *MINvalue*

A: $6: +\infty$: 6 *MAXvalue*

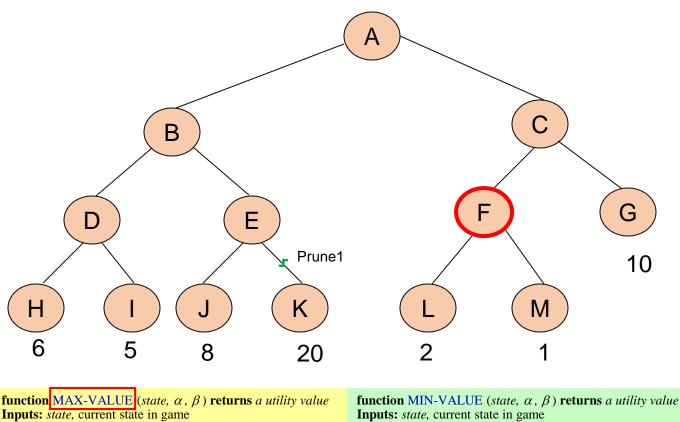
C: 6: $+\infty$: $+\infty$ MINvalue

F: 6: $+\infty$: $-\infty$ *MAXvalue*

L: 2: 2: 2

F: 6: +∞: 2 *MAXvalue*

M: 1: 1: 1



function MAX-VALUE (state, α , β) **returns** a utility value

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow MIN(\beta, v)$

return v

Node β Value Function

A: $-\infty$: $+\infty$: $-\infty$ MAX value

B: $-\infty$: $+\infty$: $+\infty$ *MINvalue*

 $\bigcap : -\infty : +\infty : -\infty$ *MAXvalue*

6: 6: 6

6: $+\infty$: 6 MAX value

1: 5: 5: 5

6: +∞: 6 *MAXvalue*

6: 6 MINvalue

6: -∞ MAXvalue

J: 8: 8: 8

PRUNE1:E:-∞:6:8 MAXvalue

B:**-**∞: 6: 6 MINvalue

A: $6: +\infty$: 6 *MAXvalue*

C: 6: $+\infty$: $+\infty$ MINvalue

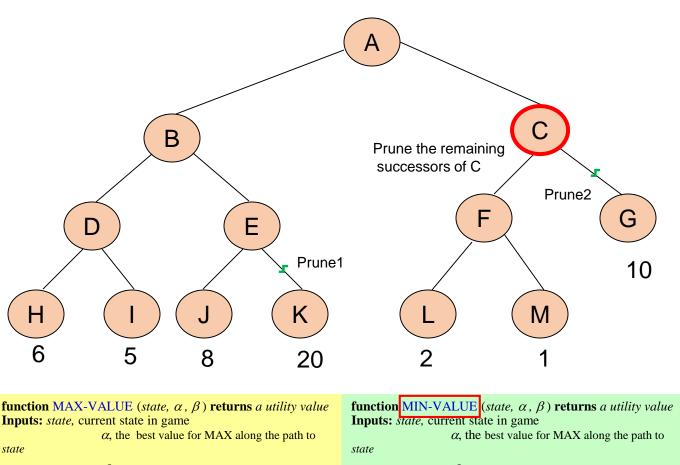
F: 6: $+\infty$: $-\infty$ MAX value

L: 2: 2: 2

F: 6: +∞: 2 MAXvalue

M: 1: 1:

F: 6:**+**∞: 2 MAXvalue



Inputs: *state*, current state in game

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v = 4 $\beta \leftarrow MIN(\beta, v)$

return v

Alpha-prunning

Prune all

remaining successors

```
Node
           β Value Function
```

-∞ MAXvalue

B: $-\infty$: $+\infty$: $+\infty$ *MINvalue*

 $D: -\infty: +\infty: -\infty$ *MAXvalue*

6: 6: 6

6: +∞: 6 MAXvalue

5: 5: 5

6: $+\infty$: 6 MAX value

B: **-**∞: 6: 6 MINvalue

6: -∞ MAXvalue

J: 8: 8: 8

PRUNE1:E:-∞:6:8 MAXvalue

B:**-**∞: 6: 6 MINvalue

A: 6: **+**∞: 6 MAXvalue

C: 6: $+\infty$: $+\infty$ MINvalue

F: 6: $+\infty$: $-\infty$ MAX value

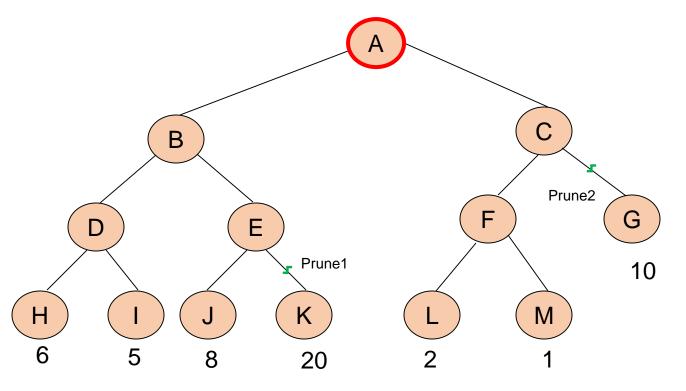
L: 2: 2:

F: 6: +∞: 2 MAXvalue

M: 1: 1:

F: 6:**+**∞: 2 MAXvalue

PRUNE2:C:6:+∞:2 MINvalue



 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow -\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(s, \alpha, \beta))$

if $v \ge \beta$ then return v

 $\alpha \leftarrow \text{MAX}(\alpha, v)$

return v

function MIN-VALUE (*state*, α , β) **returns** *a utility value* **Inputs:** *state*, current state in game

 α , the best value for MAX along the path to *state*

 β , the best value for MIN along the path to *state*

if TERMINAL-TEST(state) **then return** UTILITY(state)

 $v \leftarrow +\infty$

for a, s in SUCCESSORS[state] do

 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(s, \alpha, \beta))$

if $v \le \alpha$ then return v

 $\beta \leftarrow \text{MIN}(\beta, v)$

return v

N	lode	α	β	Value	Function
---	------	---	---	-------	-----------------

A: -∞: +∞: -∞ MAXvalue

B: $-\infty$: $+\infty$: $+\infty$ *MINvalue*

D: $-\infty$: $+\infty$: $-\infty$ MAX value

H: 6: 6: 6

D: $6: +\infty$: 6 *MAXvalue*

l: 5: 5: 5

D: 6: $+\infty$: 6 MAX value

B: **-**∞: 6: 6 *MINvalue*

E: $-\infty$: 6: $-\infty$ *MAXvalue*

J: 8: 8: 8

PRUNE1:E:-∞:6:8 MAXvalue

B:-∞: 6: 6 *MINvalue*

A: 6: $+\infty$: 6 MAX value

C: 6: $+\infty$: $+\infty$ MINvalue

F: 6: $+\infty$: $-\infty$ *MAXvalue*

L: 2: 2: 2

F: 6: $+\infty$: 2 *MAXvalue*

M: 1: 1: 1

F: $6:+\infty$: 2 MAX value

PRUNE2:C:6:+∞:2 MINvalue

A: 6:+∞: 6 *MAXvalue*