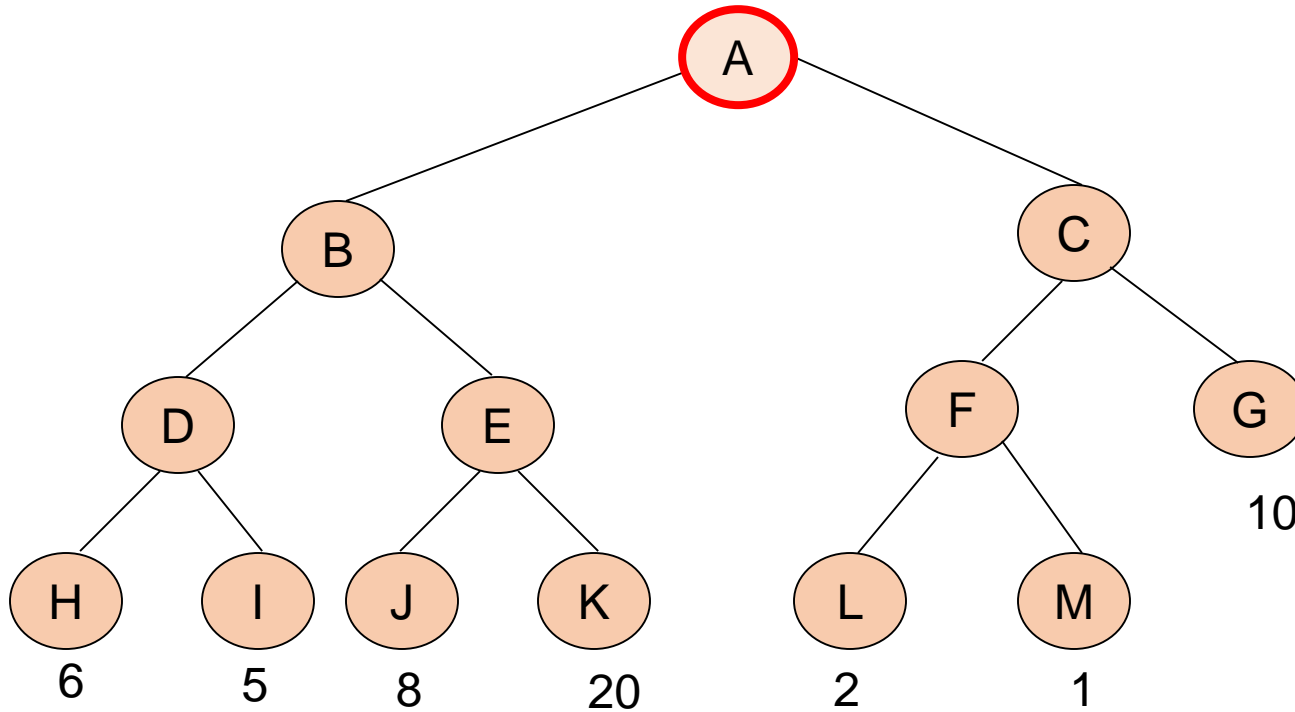


Example of running the Alpha-Beta algorithm

A trace of the program

Node	α	β	Value Function
A:	$-\infty$	$+\infty$	$-\infty$ 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*

```

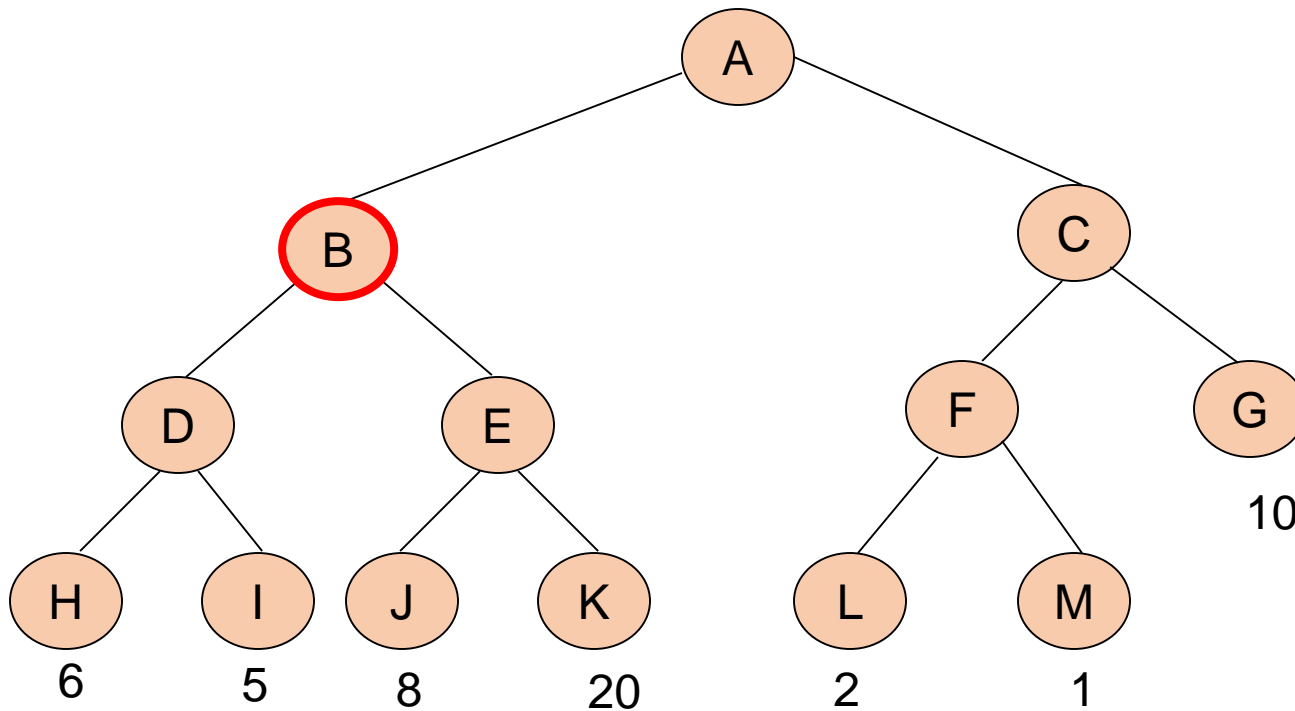
if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
  v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
  if v  $\geq \beta$  then return v
   $\alpha \leftarrow$  MAX ( $\alpha$ , v)
return v
  
```

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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
  if v  $\leq \alpha$  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



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*

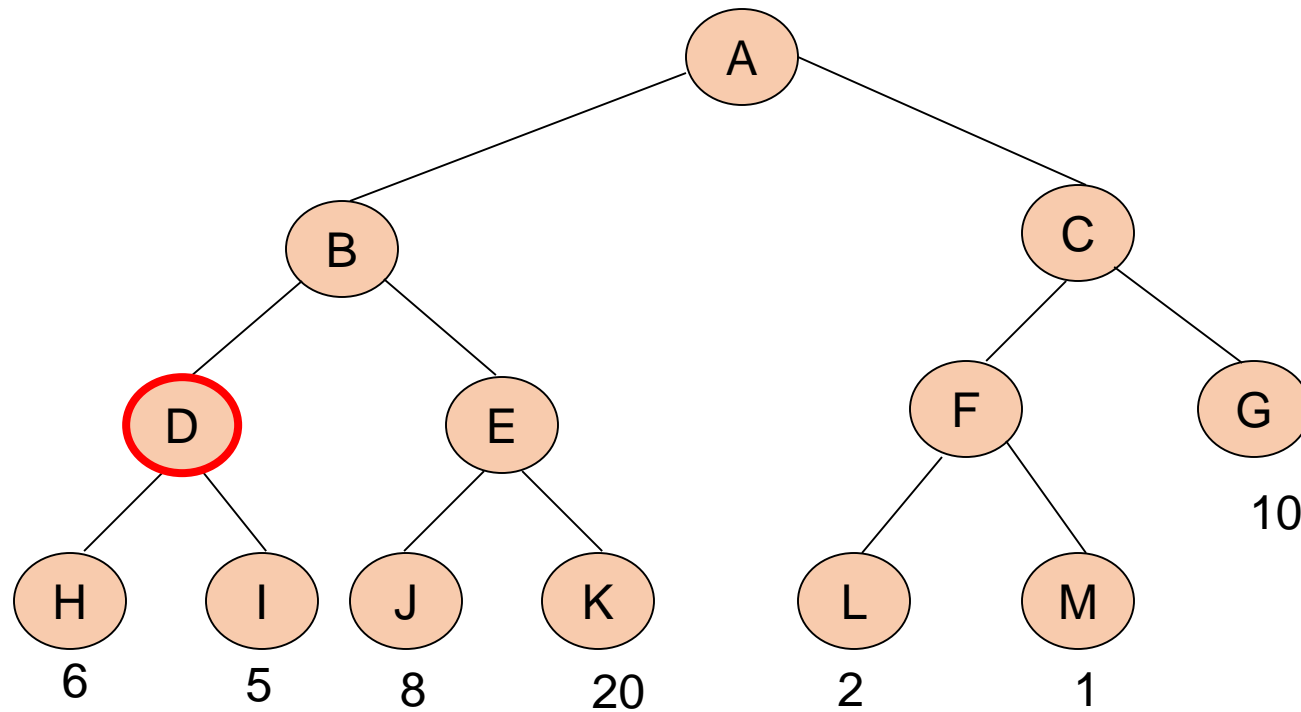
```

if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
    v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  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

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*

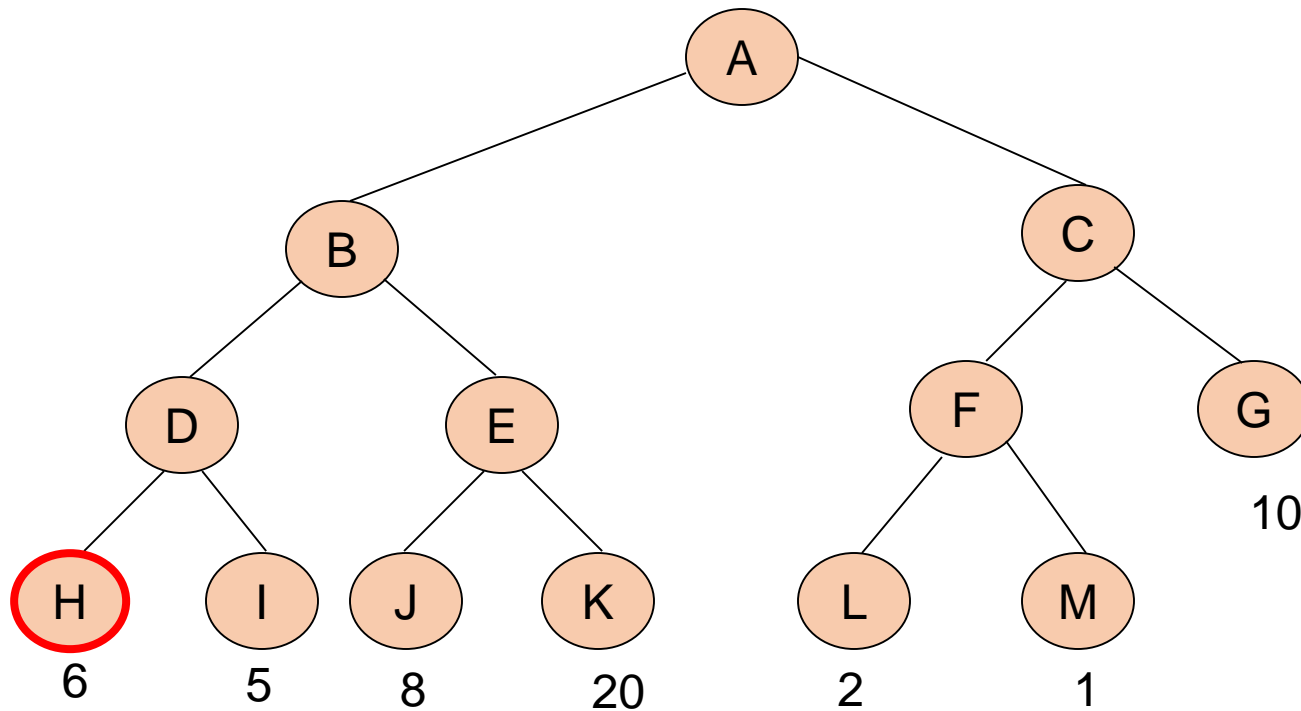
```

if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
  v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
  if v  $\geq \beta$  then return v
   $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
  if v  $\leq \alpha$  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

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*

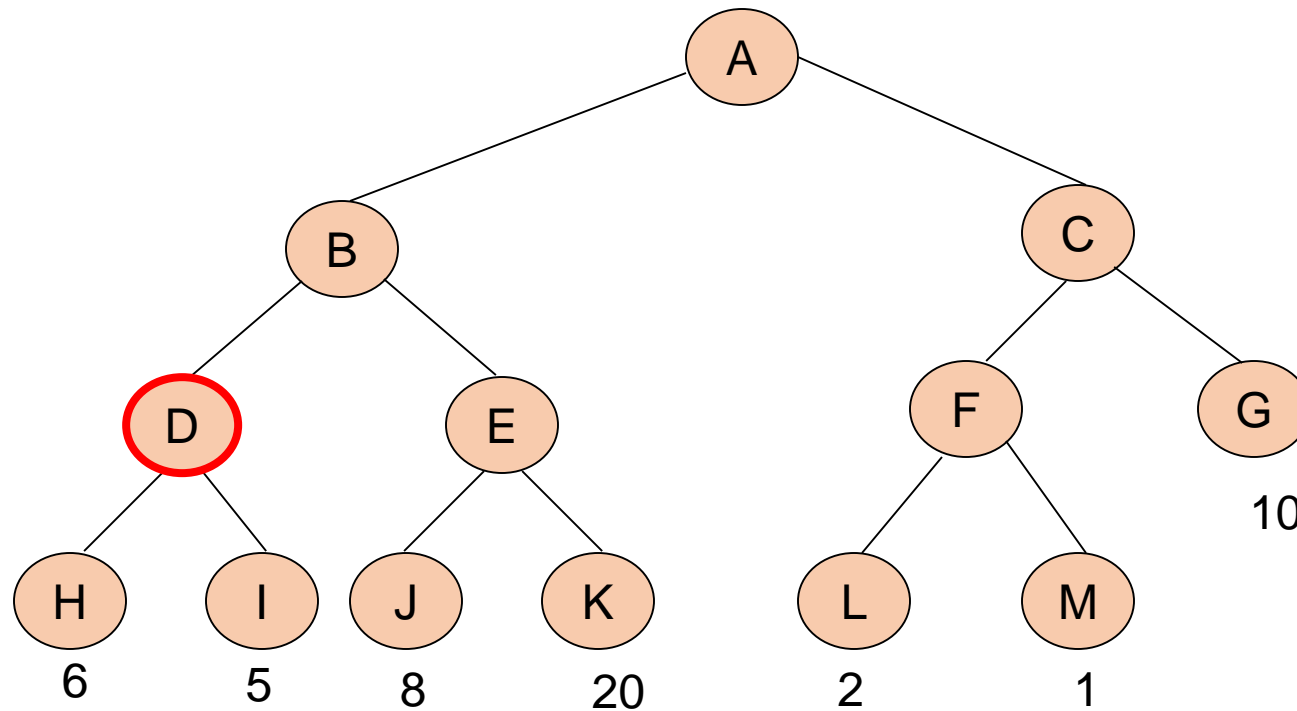
```

if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
  v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
  if v  $\geq \beta$  then return v
   $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
  if v  $\leq \alpha$  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

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*

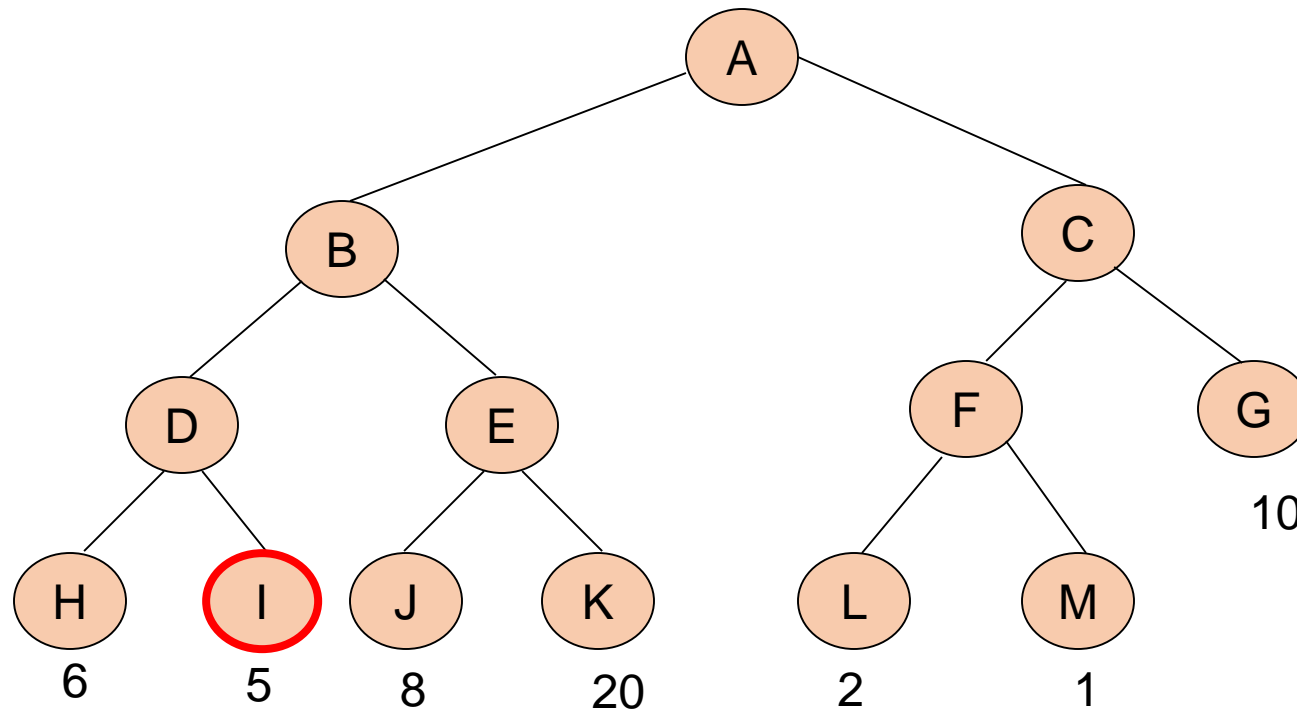
```

if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
    v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  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
I:	5	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*

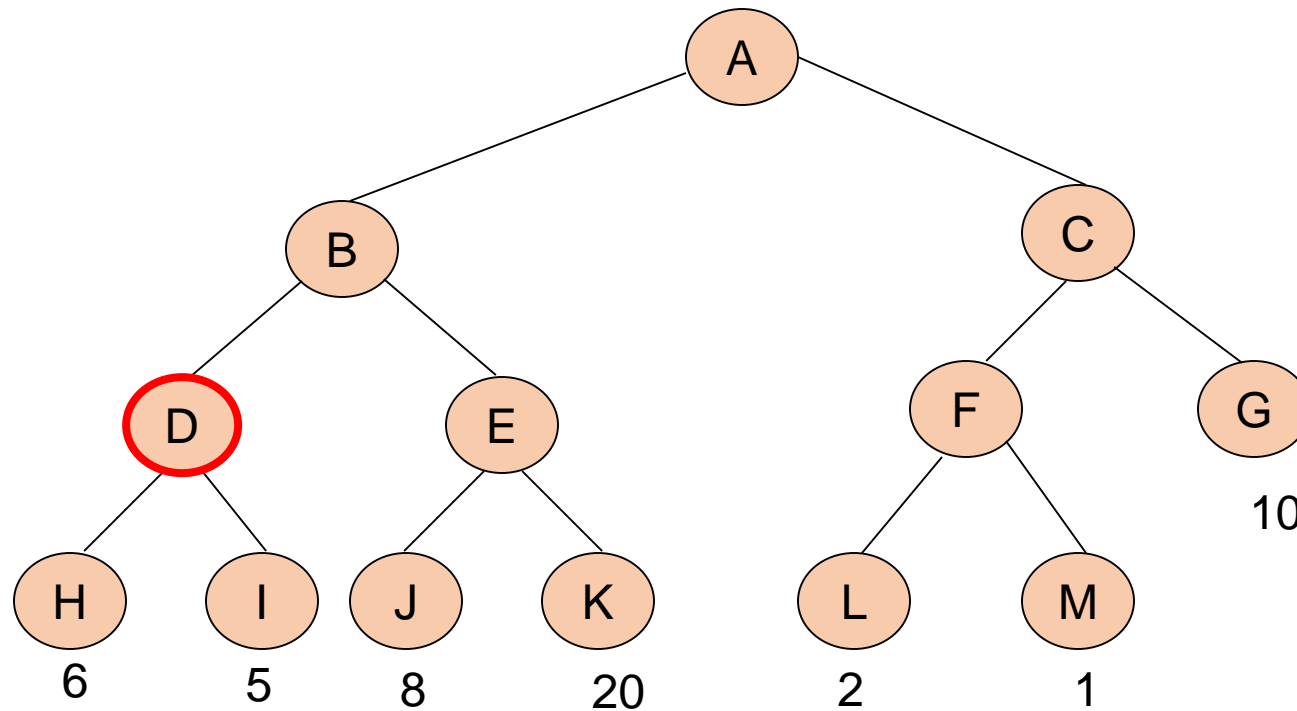
```

if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
    v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  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
I:	5	5	5	
D:	6	$+\infty$	6	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*

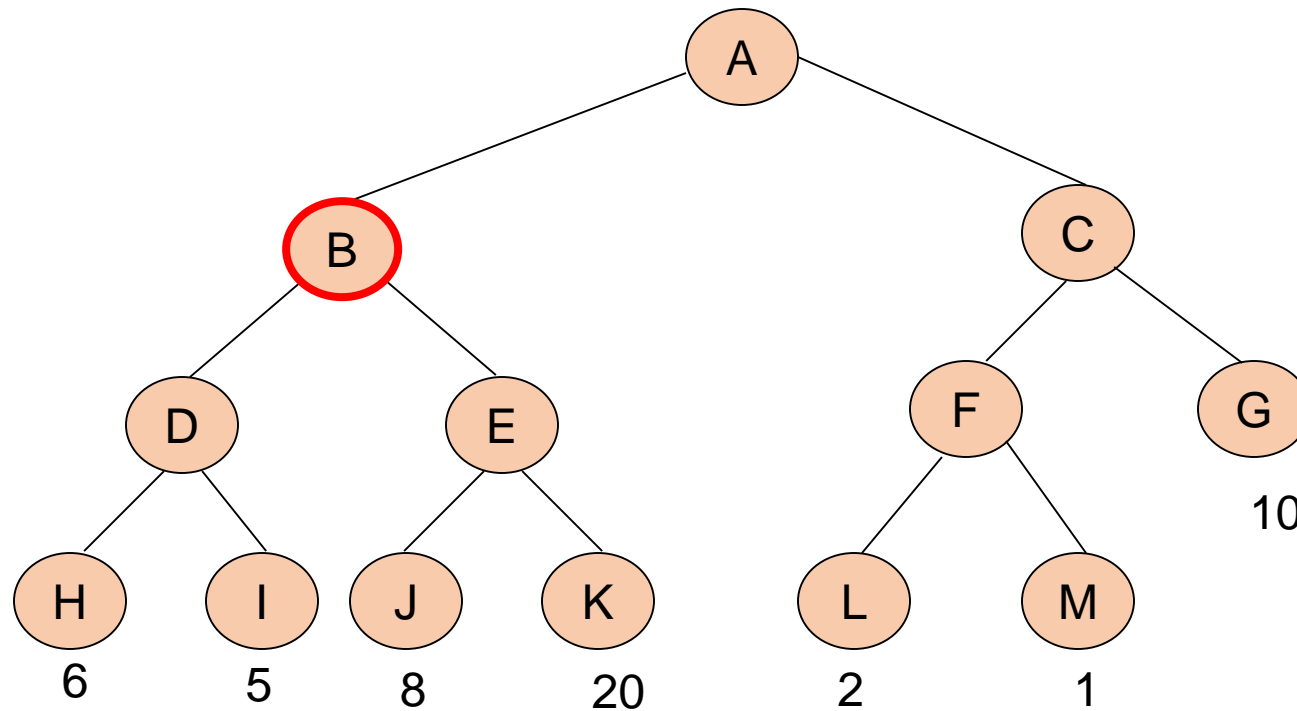
```

if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
    v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  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
I:	5:	5:	5
D:	6:	$+\infty$	6 MAXvalue
B:	$-\infty$:	6:	6 MINvalue

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*

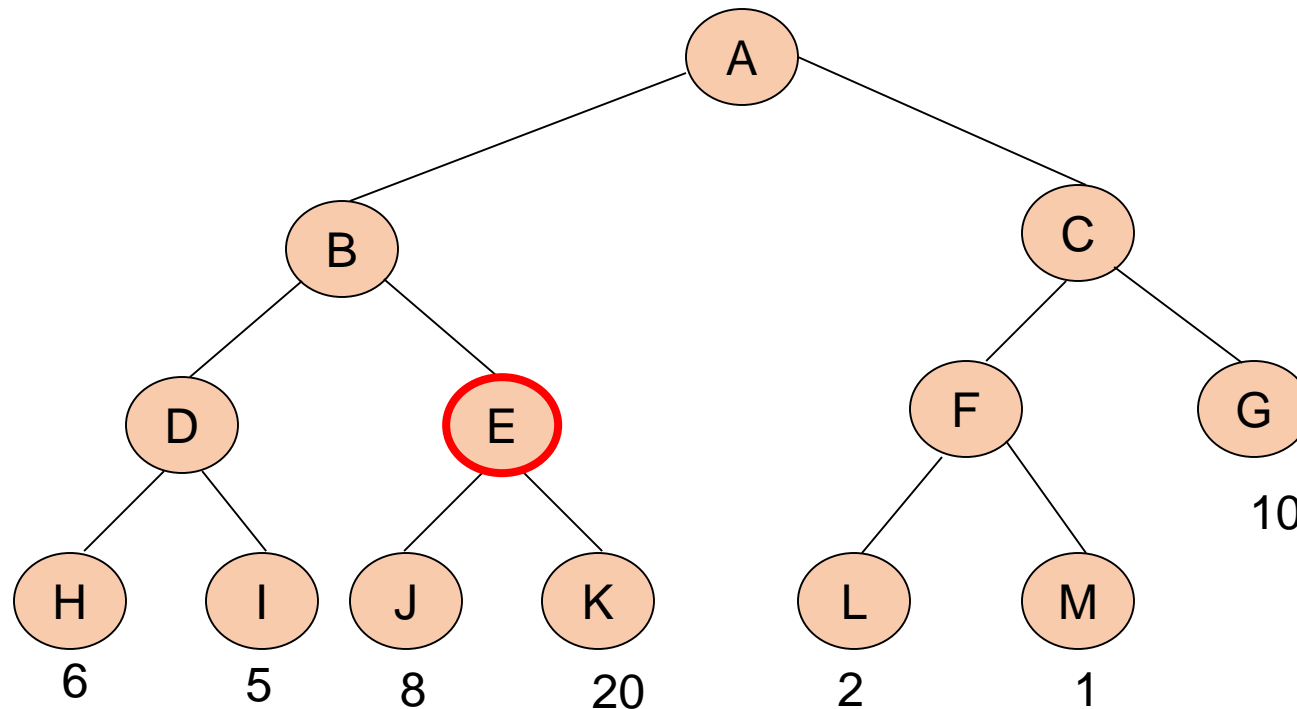
```

if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
  v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
  if v  $\geq \beta$  then return v
   $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
  if v  $\leq \alpha$  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
I:	5	5	5	
D:	6	$+\infty$	6	MAXvalue
B:	$-\infty$	6	6	MINvalue
E:	$-\infty$	6	$-\infty$	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*

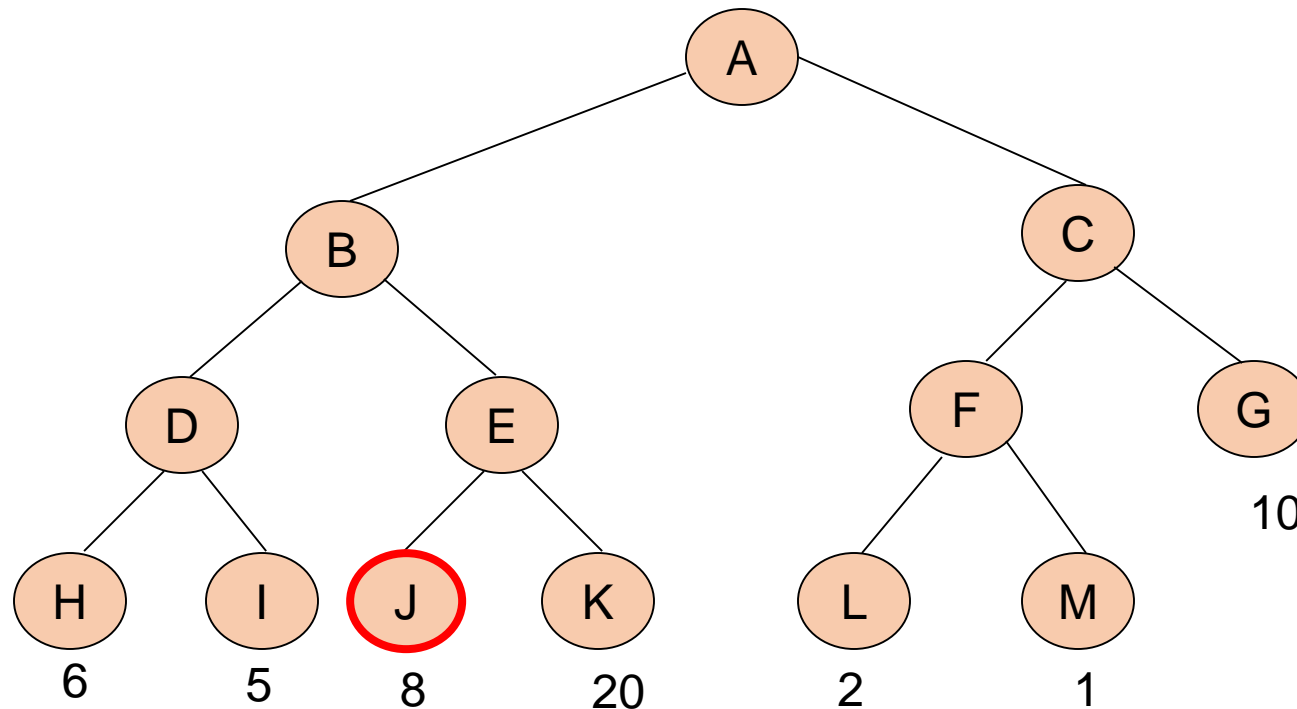
```

if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
    v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  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
I:	5:	5:	5
D:	6:	$+\infty$	6 MAXvalue
B:	$-\infty$:	6:	6 MINvalue
E:	$-\infty$:	6:	$-\infty$ MAXvalue
J:	8:	8:	8

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*

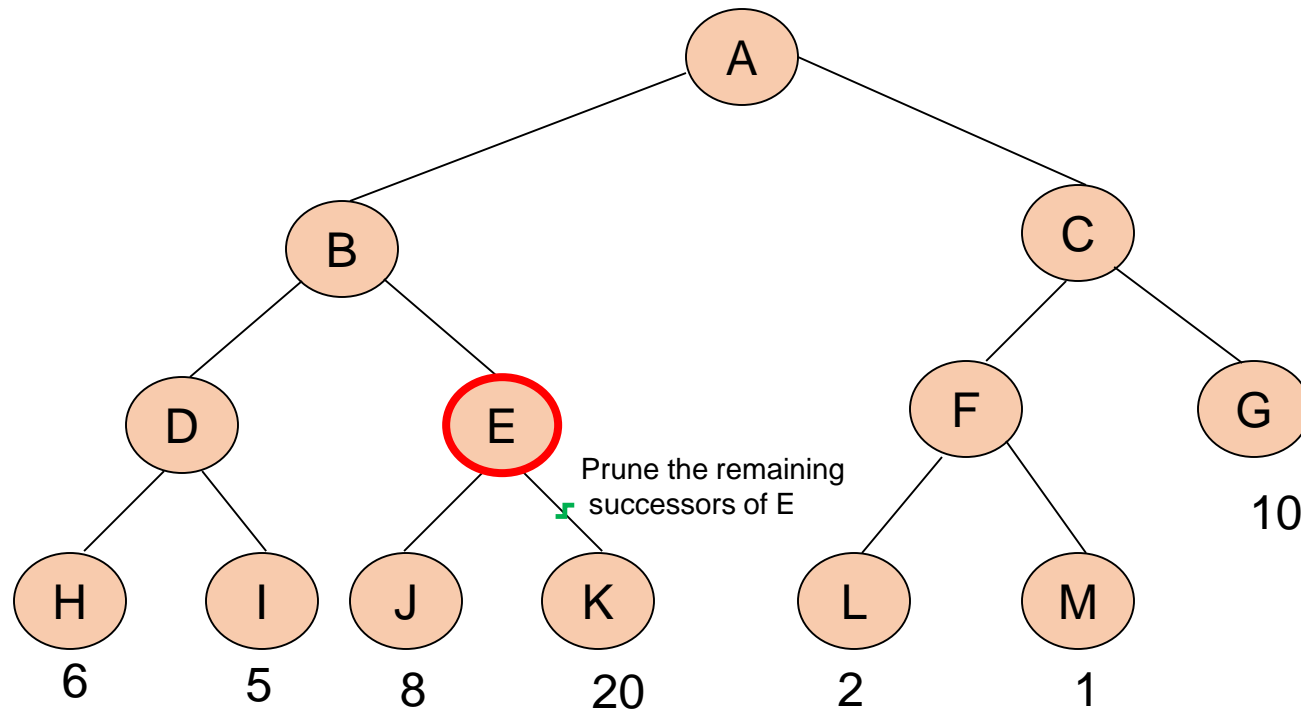
```

if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
  v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
  if v  $\geq \beta$  then return v
   $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
  if v  $\leq \alpha$  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
I:	5	5	5
D:	6	$+\infty$	6 MAXvalue
B:	$-\infty$	6	6 MINvalue
E:	$-\infty$	6	$-\infty$ MAXvalue
J:	8	8	8

PRUNE1(β):E: $-\infty$:6:8 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  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  MAX( $\alpha$ , v)
return v
  
```

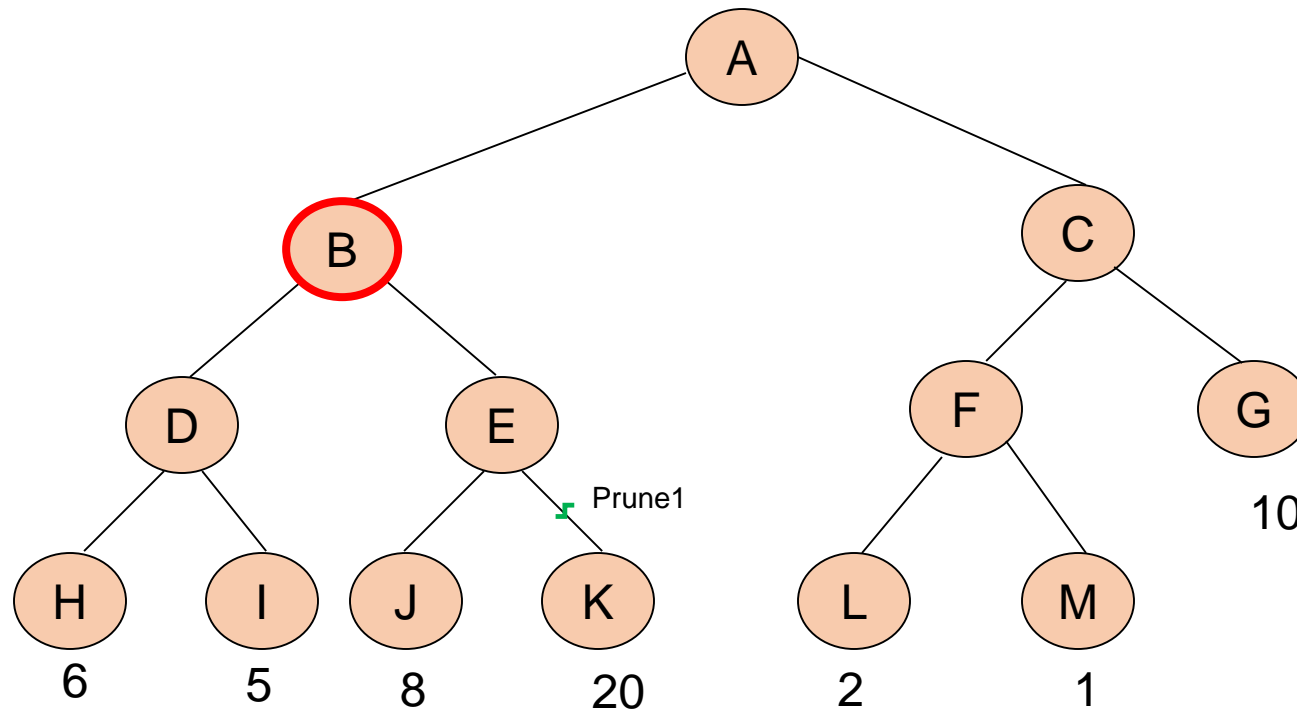
Beta-pruning

Prune all
remaining successors

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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  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
I:	5	5	5
D:	6	$+\infty$	6 MAXvalue
B:	$-\infty$	6	6 MINvalue
E:	$-\infty$	6	$-\infty$ MAXvalue
J:	8	8	8
PRUNE1:E:	$-\infty$	6	8 MAXvalue
B:	$-\infty$	6	6 MINvalue

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  $v \geq \beta$  then return  $v$ 
     $\alpha \leftarrow \text{MAX}(\alpha, v)$ 
return  $v$ 

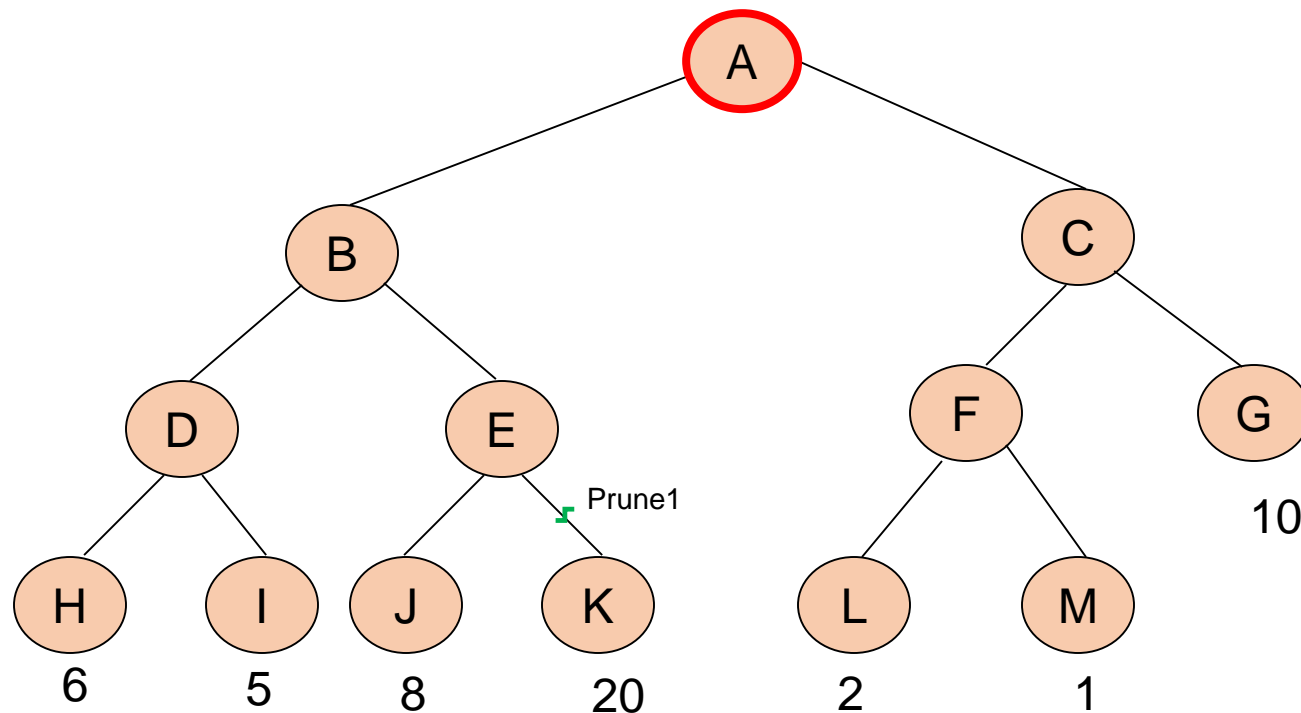
```

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  $v \leq \alpha$  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:	$+\infty$	6	MAXvalue
I:	5:	5:	5	
D:	6:	$+\infty$	6	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
E:	$-\infty$:	6:	$-\infty$	MAXvalue
J:	8:	8:	8	
PRUNE1:E:	$-\infty$:	6:	8	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
A:	6:	$+\infty$:	6	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  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  MAX ( $\alpha$ , v)
return v

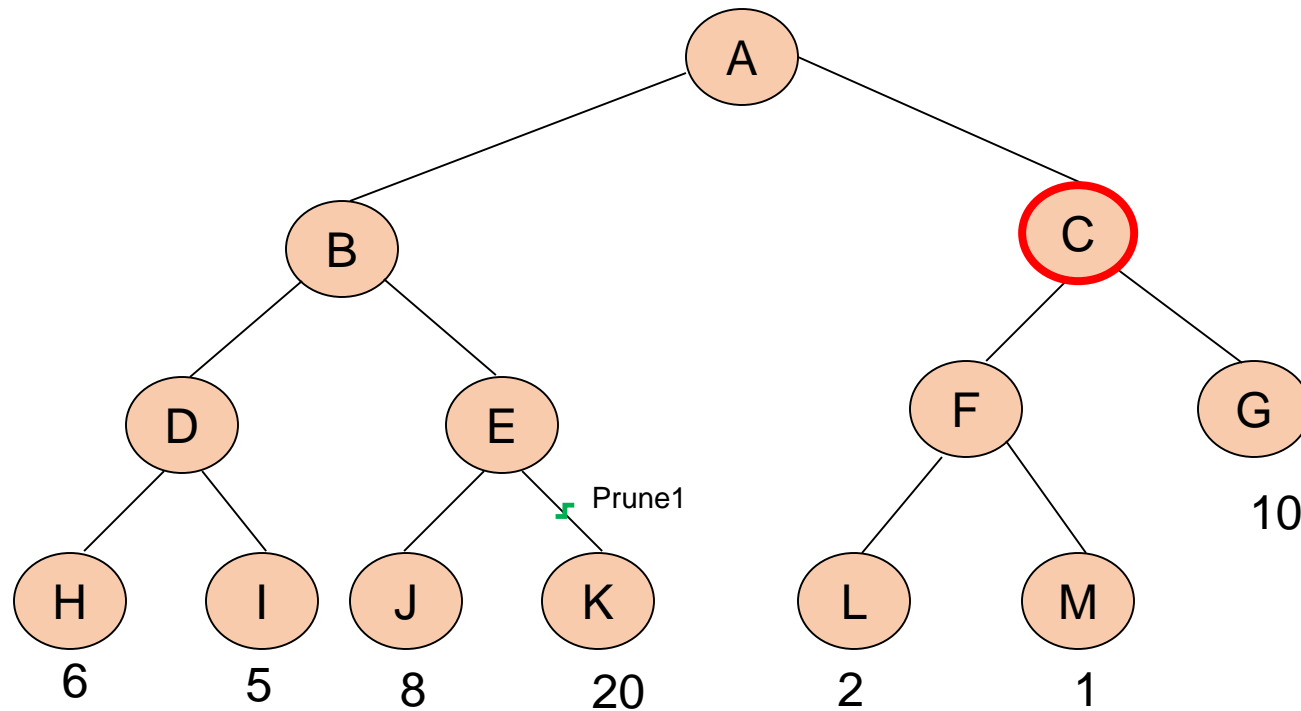
```

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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  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
I:	5:	5:	5
D:	6:	$+\infty$	6 MAXvalue
B:	$-\infty$:	6:	6 MINvalue
E:	$-\infty$:	6:	$-\infty$ MAXvalue
J:	8:	8:	8
PRUNE1:E:	$-\infty$:	6:	8 MAXvalue
B:	$-\infty$:	6:	6 MINvalue
A:	6:	$+\infty$:	6 MAXvalue
C:	6:	$+\infty$:	$+\infty$ MINvalue

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  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  MAX ( $\alpha$ , v)
return v

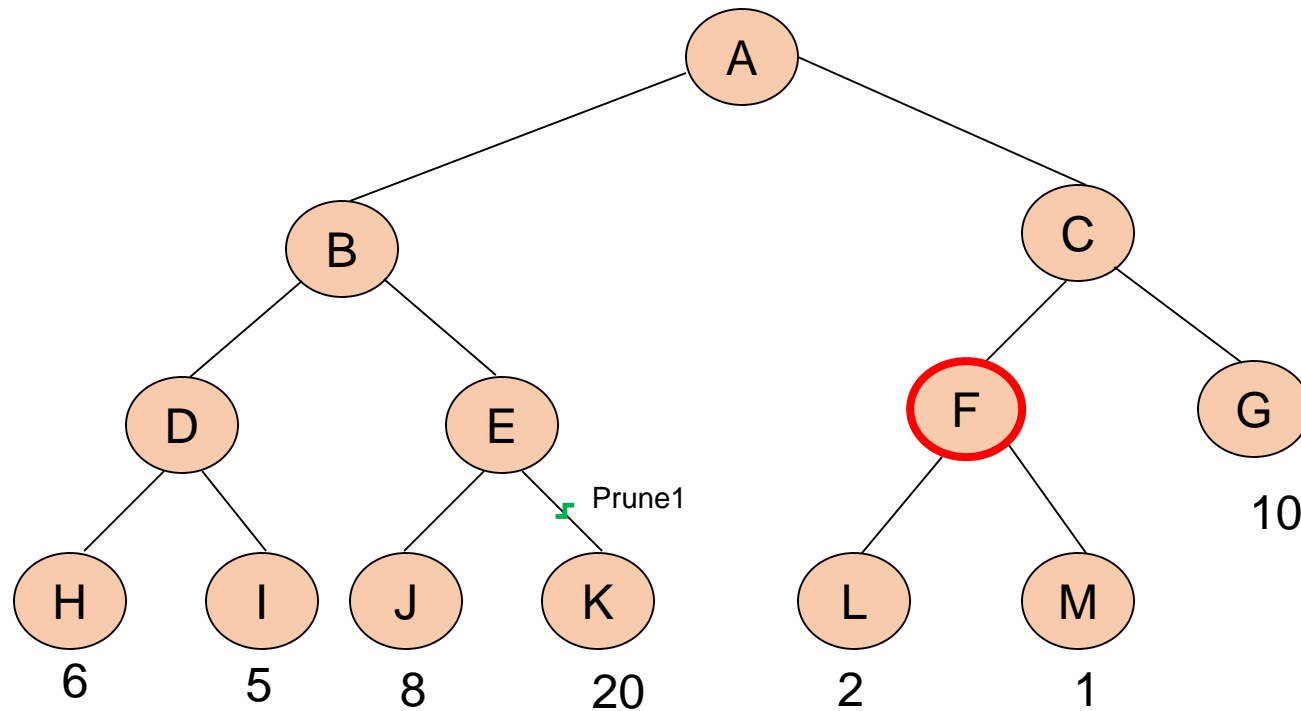
```

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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  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
I:	5:	5:	5	
D:	6:	$+\infty$:	6	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
E:	$-\infty$:	6:	$-\infty$	MAXvalue
J:	8:	8:	8	
PRUNE1:E:	$-\infty$:	6:	8	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
A:	6:	$+\infty$:	6	MAXvalue
C:	6:	$+\infty$:	$+\infty$	MINvalue
F:	6:	$+\infty$:	$-\infty$	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*

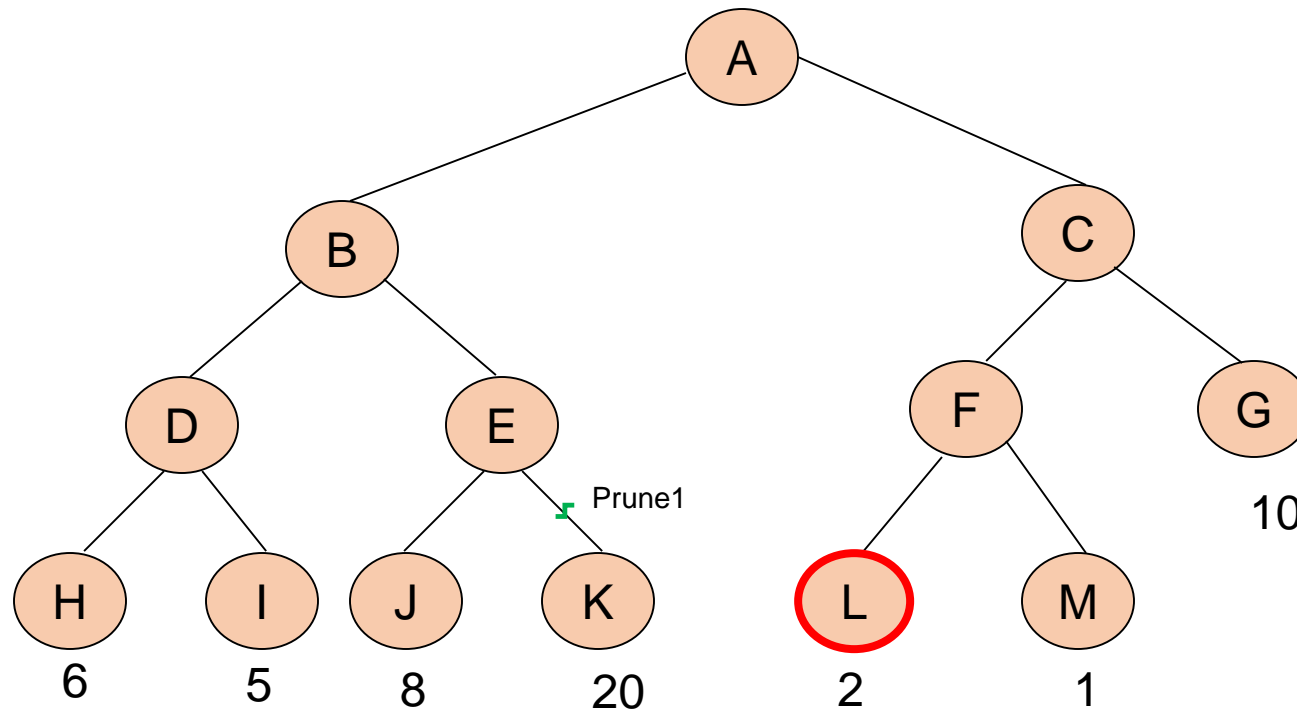
```

if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
    v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  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
I:	5:	5:	5	
D:	6:	$+\infty$	6	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
E:	$-\infty$:	6:	$-\infty$	MAXvalue
J:	8:	8:	8	
PRUNE1:E:	$-\infty$:	6:	8	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
A:	6:	$+\infty$:	6	MAXvalue
C:	6:	$+\infty$:	$+\infty$	MINvalue
F:	6:	$+\infty$:	$-\infty$	MAXvalue
L:	2:	2:	2	

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*

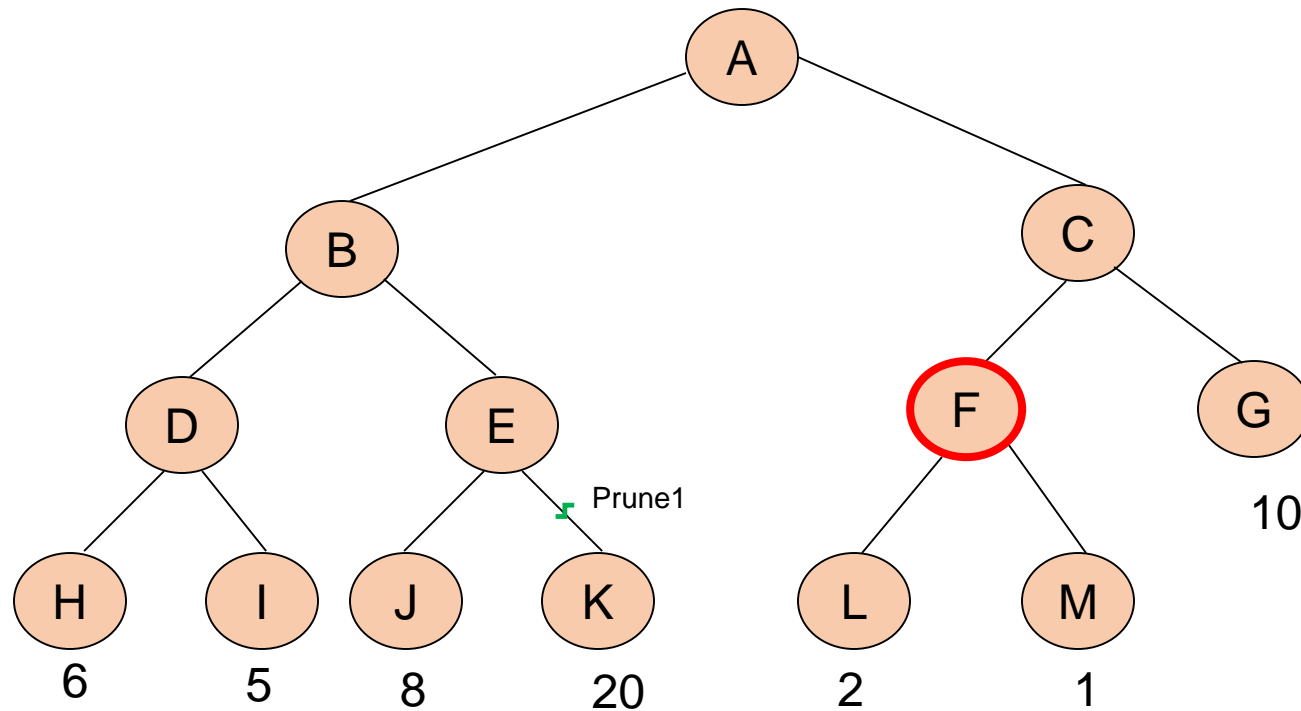
```

if TERMINAL-TEST(state) then return UTILITY(state)
v  $\leftarrow -\infty$ 
for a, s in SUCCESSORS[state] do
    v  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  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
I:	5:	5:	5	
D:	6:	$+\infty$	6	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
E:	$-\infty$:	6:	$-\infty$	MAXvalue
J:	8:	8:	8	
PRUNE1:E:	$-\infty$:	6:	8	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
A:	6:	$+\infty$:	6	MAXvalue
C:	6:	$+\infty$:	$+\infty$	MINvalue
F:	6:	$+\infty$:	$-\infty$	MAXvalue
L:	2:	2:	2	
F:	6:	$+\infty$:	2	MAXvalue

```

function MAX-VALUE (state,  $\alpha$ ,  $\beta$ ) returns a utility value
Inputs: state, current state in game
         $\alpha$ , the best value for MAX along the path to state
         $\beta$ , 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  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  MAX ( $\alpha$ , v)

return v
  
```

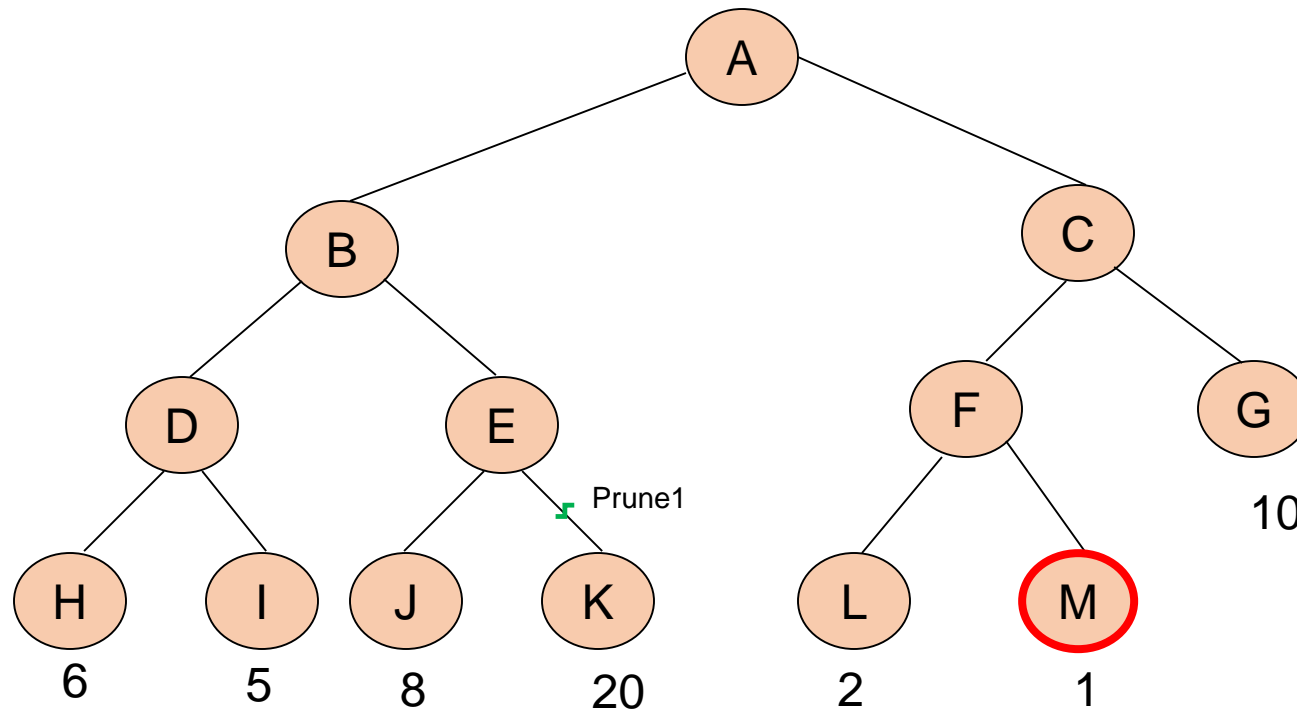
```

function MIN-VALUE (state,  $\alpha$ ,  $\beta$ ) returns a utility value
Inputs: state, current state in game
         $\alpha$ , the best value for MAX along the path to state
         $\beta$ , 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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  then return v
     $\beta \leftarrow$  MIN( $\beta$ , v)

return v
  
```

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



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
I:	5:	5:	5	
D:	6:	$+\infty$	6	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
E:	$-\infty$:	6:	$-\infty$	MAXvalue
J:	8:	8:	8	
PRUNE1:E:	$-\infty$:	6:	8	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
A:	6:	$+\infty$:	6	MAXvalue
C:	6:	$+\infty$:	$+\infty$	MINvalue
F:	6:	$+\infty$:	$-\infty$	MAXvalue
L:	2:	2:	2	
F:	6:	$+\infty$:	2	MAXvalue
M:	1:	1:	1	

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  $v \geq \beta$  then return  $v$ 
     $\alpha \leftarrow \text{MAX}(\alpha, v)$ 

return  $v$ 

```

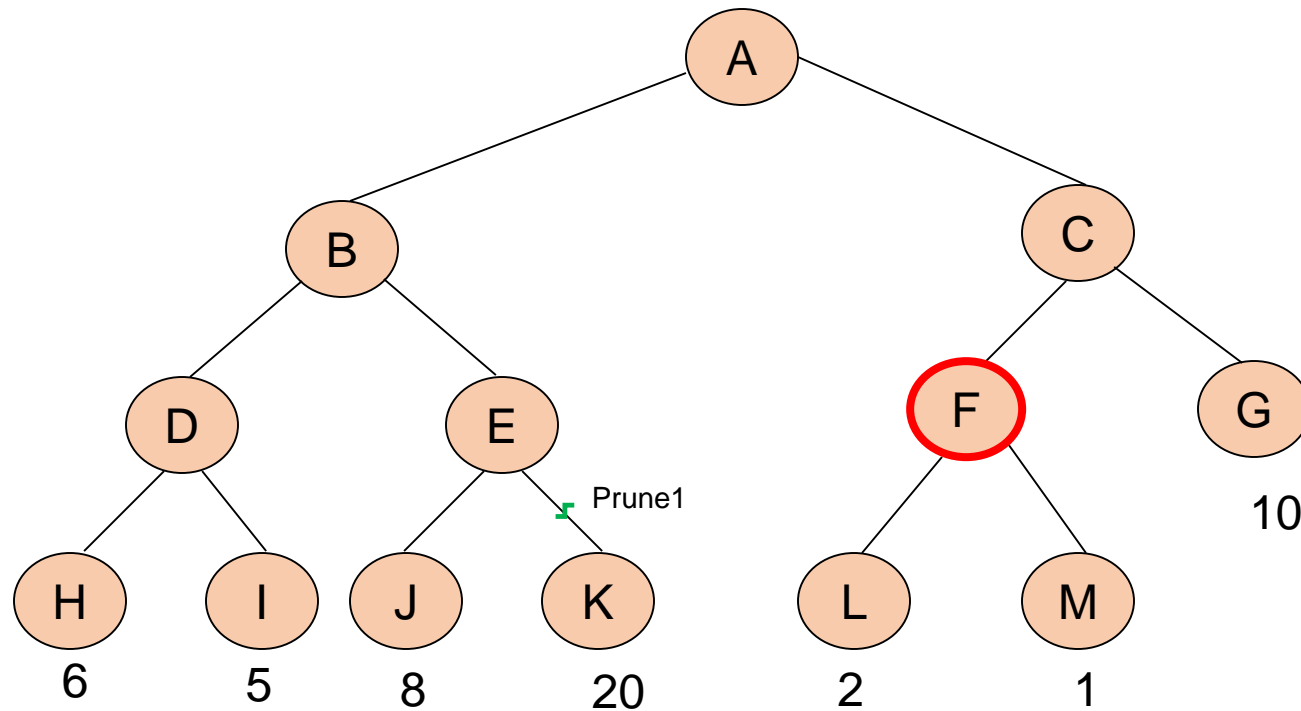
function MIN-VALUE (*state*, α , β) **returns** a utility value
Inputs: *state*, current state in game
 α , the best value for MAX along the path to *state*
 β , the best value for MIN along the path to *state*

```

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  $v \leq \alpha$  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:	$+\infty$	6	MAXvalue
I:	5:	5:	5	
D:	6:	$+\infty$	6	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
E:	$-\infty$:	6:	$-\infty$	MAXvalue
J:	8:	8:	8	
PRUNE1:E:	$-\infty$:	6:	8	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
A:	6:	$+\infty$:	6	MAXvalue
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	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  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  MAX ( $\alpha$ , v)
return v

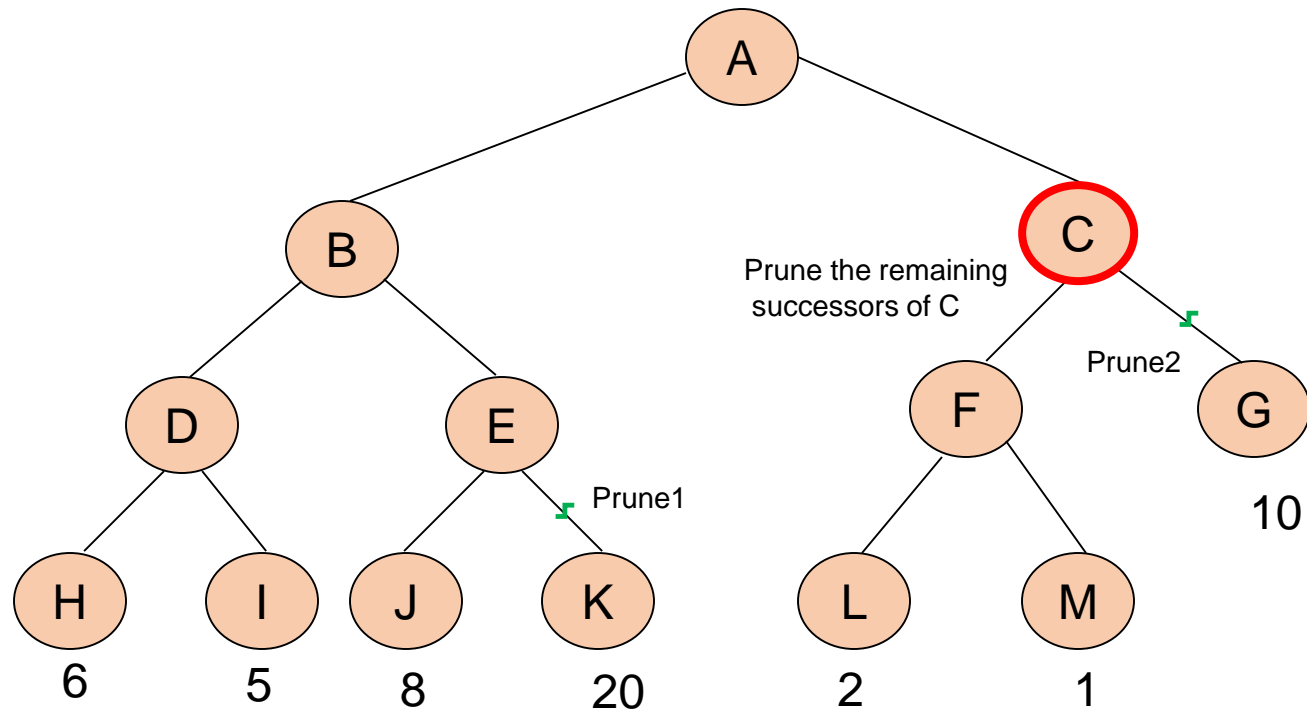
```

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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  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
I:	5:	5:	5	
D:	6:	$+\infty$	6	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
E:	$-\infty$:	6:	$-\infty$	MAXvalue
J:	8:	8:	8	
PRUNE1:E: $-\infty$:6:8 MAXvalue				
B:	$-\infty$:	6:	6	MINvalue
A:	6:	$+\infty$:	6	MAXvalue
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	MAXvalue
PRUNE2:C:6: $+\infty$:2 MINvalue				

```

function MAX-VALUE (state,  $\alpha$ ,  $\beta$ ) returns a utility value
Inputs: state, current state in game
          $\alpha$ , the best value for MAX along the path to
state
          $\beta$ , 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  $v \geq \beta$  then return v
     $\alpha \leftarrow \text{MAX}(\alpha, v)$ 

return v
  
```

```

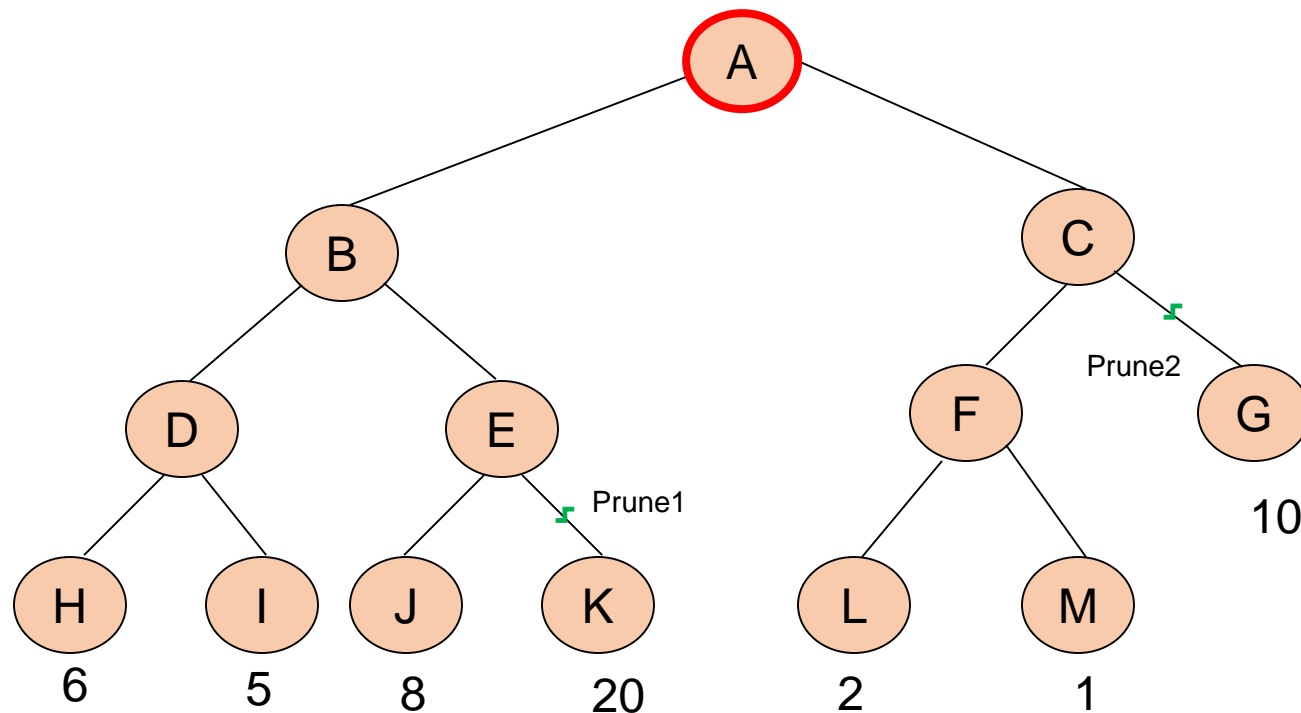
function MIN-VALUE (state,  $\alpha$ ,  $\beta$ ) returns a utility value
Inputs: state, current state in game
          $\alpha$ , the best value for MAX along the path to
state
          $\beta$ , the best value for MIN along the path to state

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

return v
  
```

Prune all remaining successors

Alpha-pruning



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
I:	5:	5:	5	
D:	6:	$+\infty$	6	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
E:	$-\infty$:	6:	$-\infty$	MAXvalue
J:	8:	8:	8	
PRUNE1:E:	$-\infty$:	6:	8	MAXvalue
B:	$-\infty$:	6:	6	MINvalue
A:	6:	$+\infty$:	6	MAXvalue
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	MAXvalue
PRUNE2:C:	6:	$+\infty$:	2	MINvalue
A:	6:	$+\infty$:	6	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  $\leftarrow$  MAX(v, MIN-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\geq \beta$  then return v
     $\alpha \leftarrow$  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  $\leftarrow$  MIN(v, MAX-VALUE(s,  $\alpha$ ,  $\beta$ ))
    if v  $\leq \alpha$  then return v
     $\beta \leftarrow$  MIN( $\beta$ , v)
return v

```