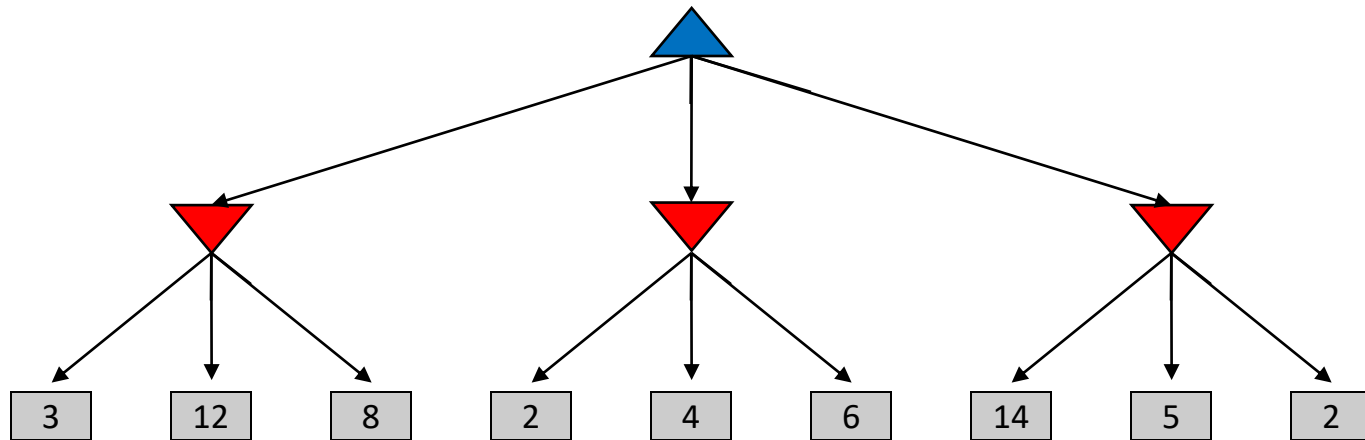


HỆ THỐNG TÌM KIẾM TRONG MÔI TRƯỜNG ĐỐI KHÁNG (PHẦN 2)

Ôn lại thuật toán minimax

MAX

MIN

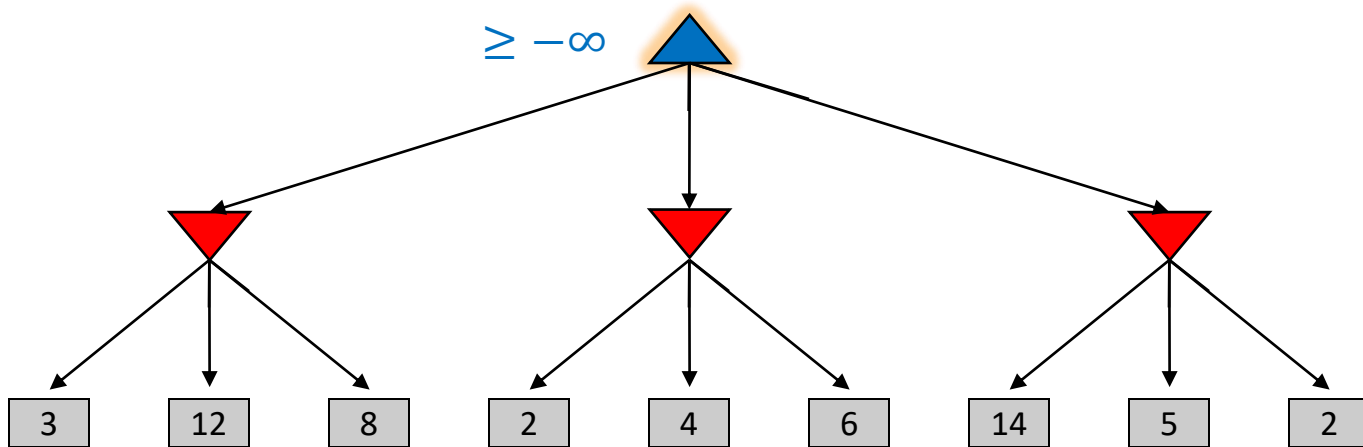


Ôn lại thuật toán minimax

MAX

$\geq -\infty$

MIN



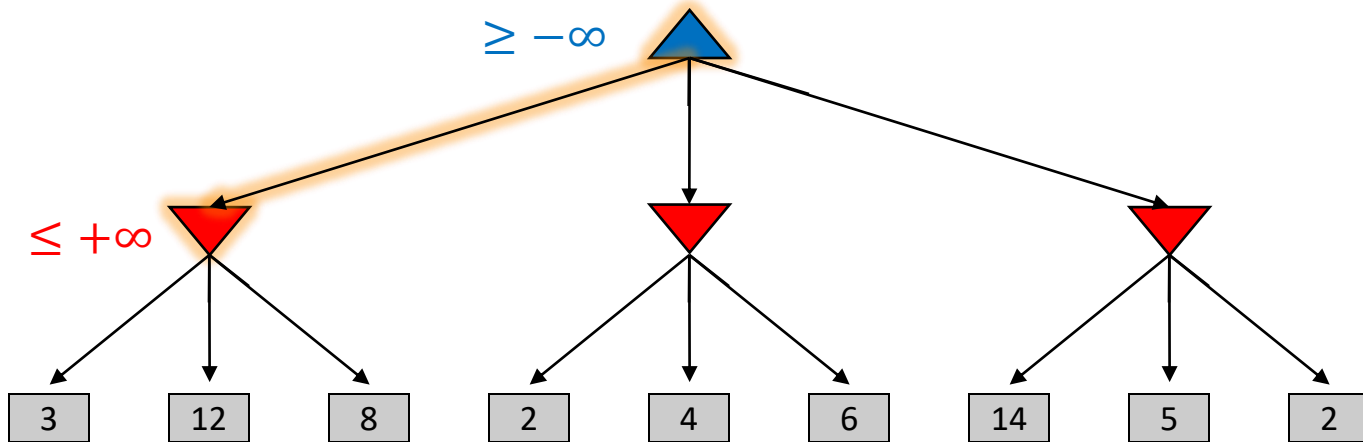
Ôn lại thuật toán minimax

MAX

$\geq -\infty$

MIN

$\leq +\infty$



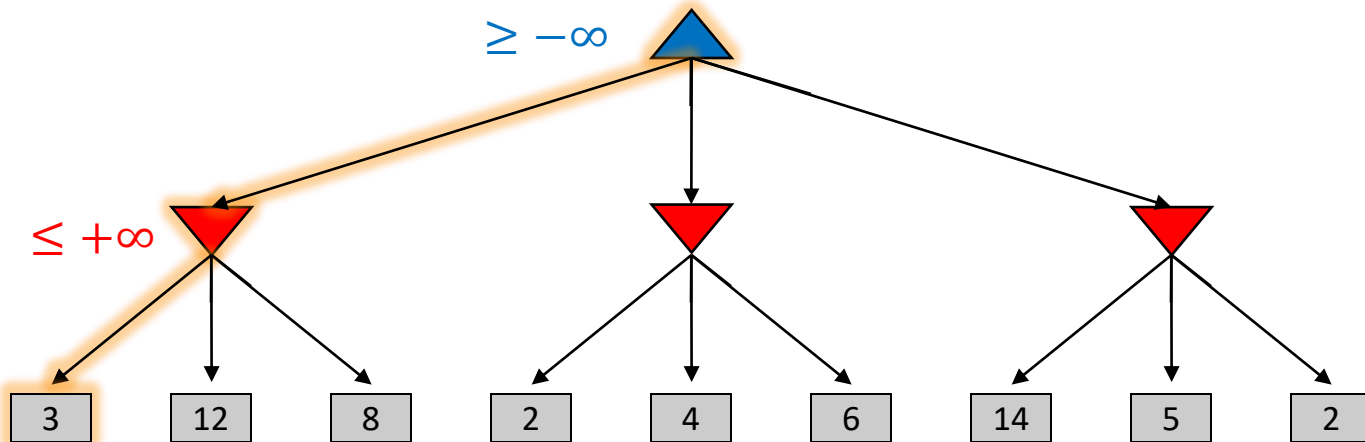
Ôn lại thuật toán minimax

MAX

$\geq -\infty$

MIN

$\leq +\infty$



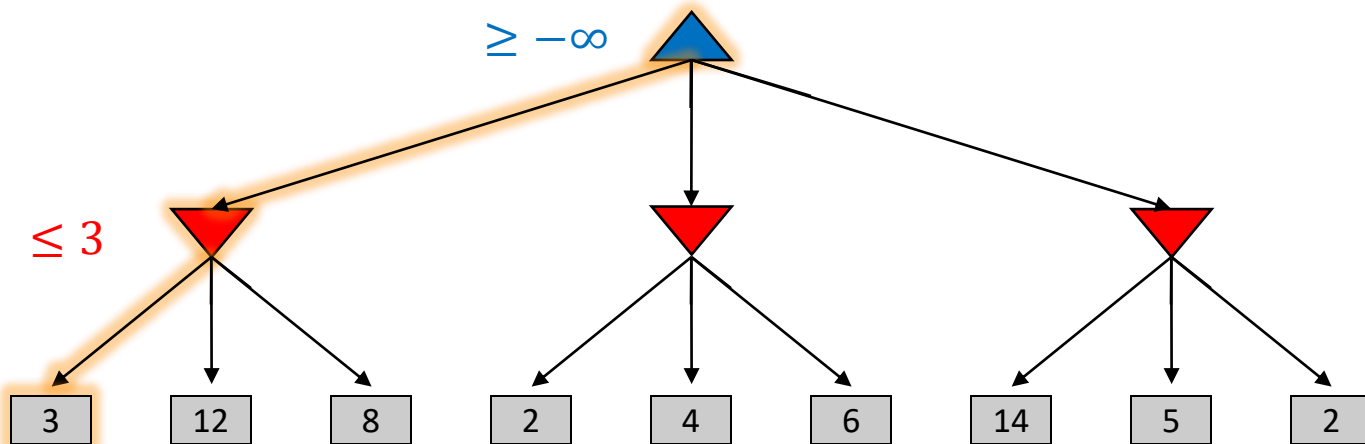
Ôn lại thuật toán minimax

MAX

$\geq -\infty$

MIN

≤ 3



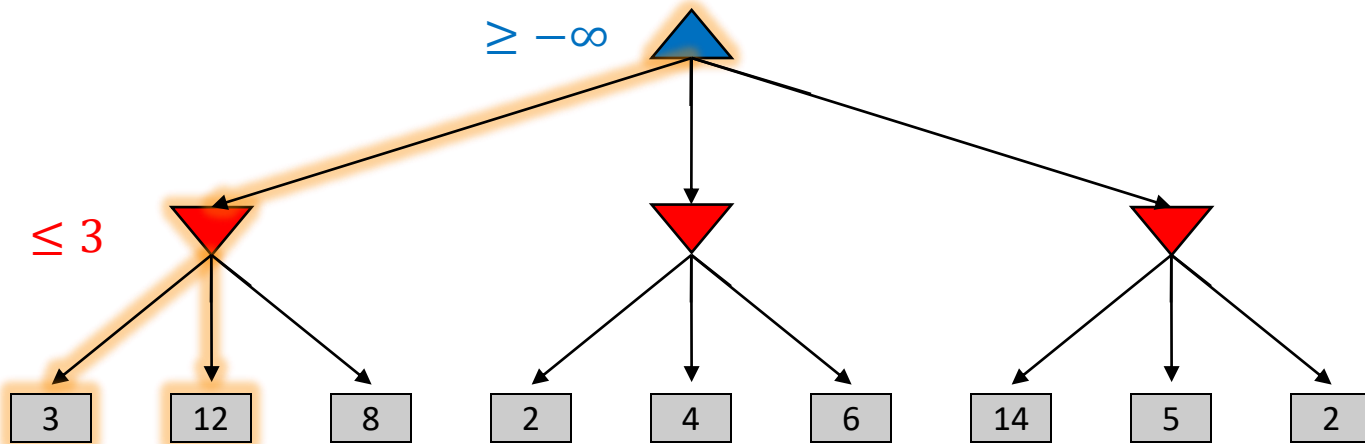
Ôn lại thuật toán minimax

MAX

$\geq -\infty$

MIN

≤ 3



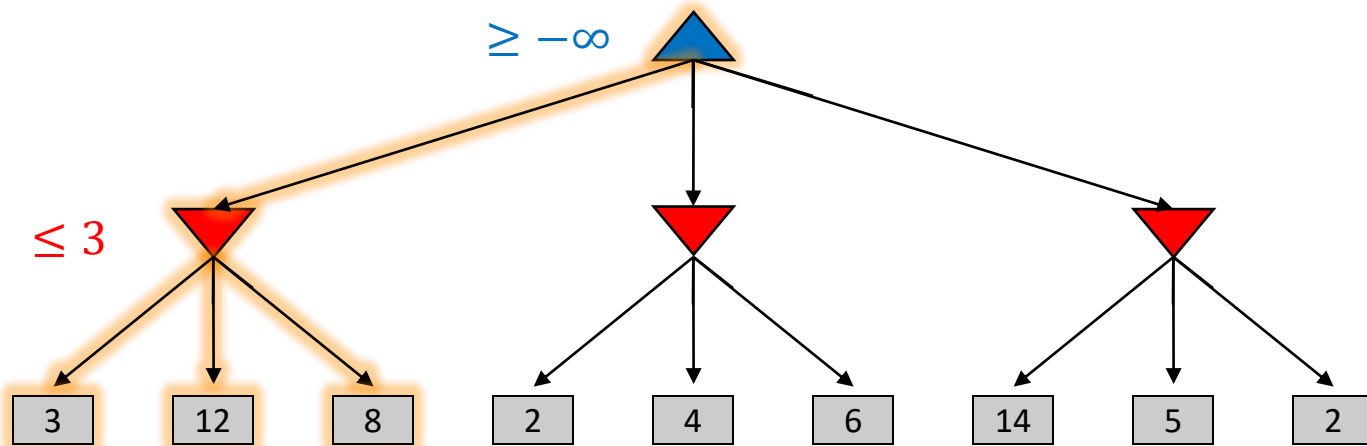
Ôn lại thuật toán minimax

MAX

$\geq -\infty$

MIN

≤ 3



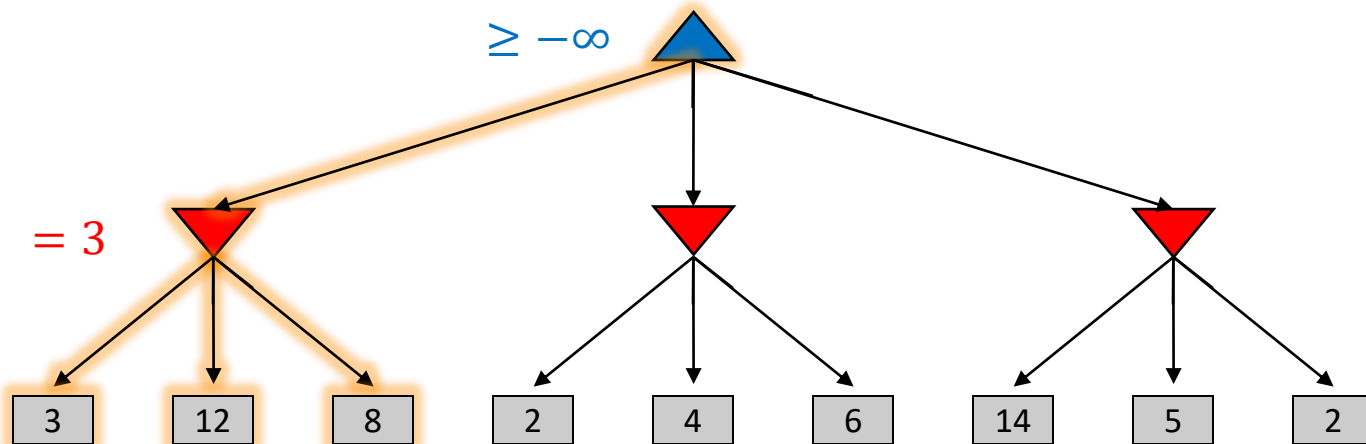
Ôn lại thuật toán minimax

MAX

$\geq -\infty$

MIN

$= 3$



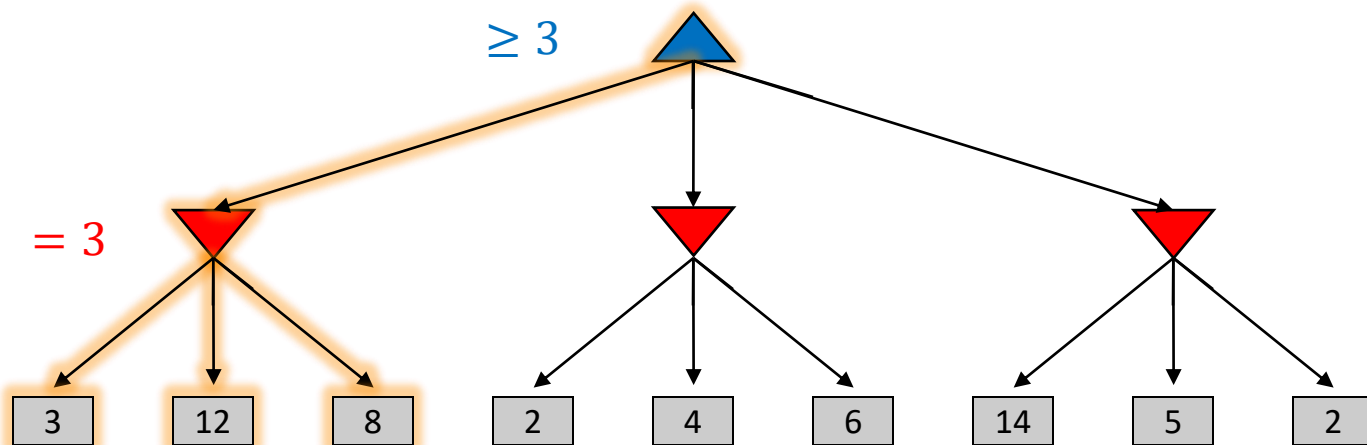
Ôn lại thuật toán minimax

MAX

≥ 3

MIN

$= 3$



Ôn lại thuật toán minimax

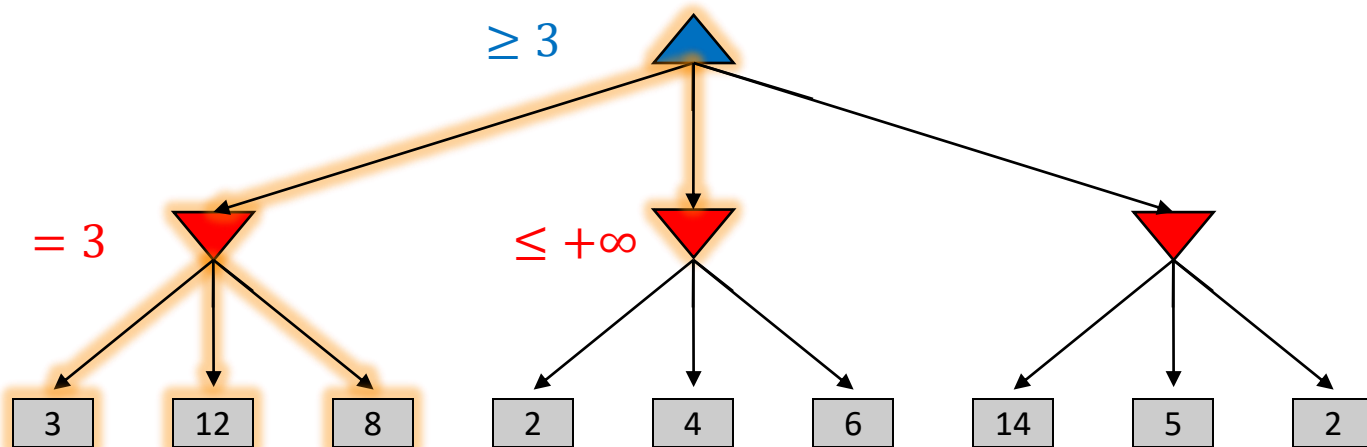
MAX

≥ 3

MIN

$= 3$

$\leq +\infty$



Ôn lại thuật toán minimax

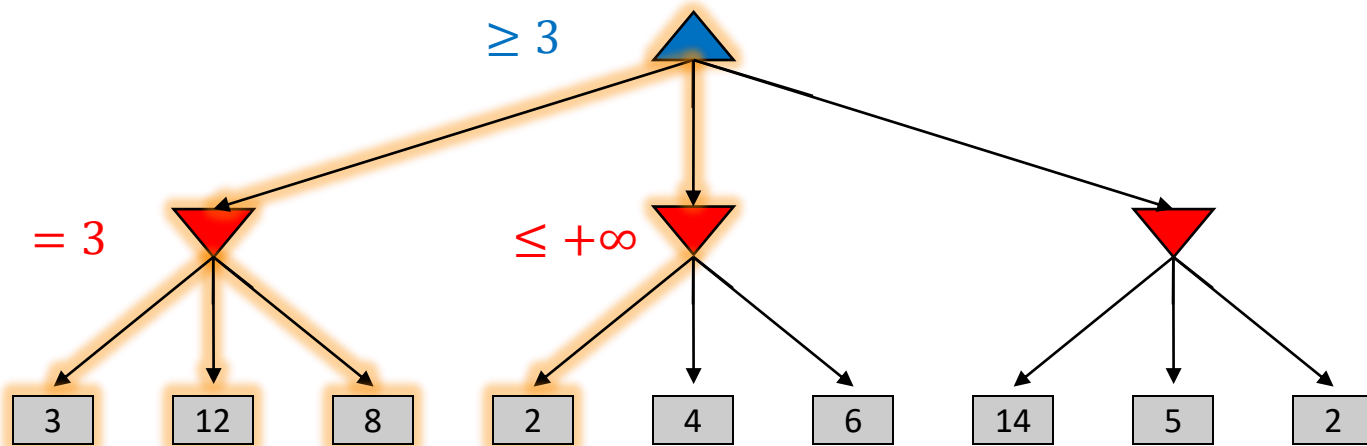
MAX

≥ 3

MIN

$= 3$

$\leq +\infty$



Ôn lại thuật toán minimax

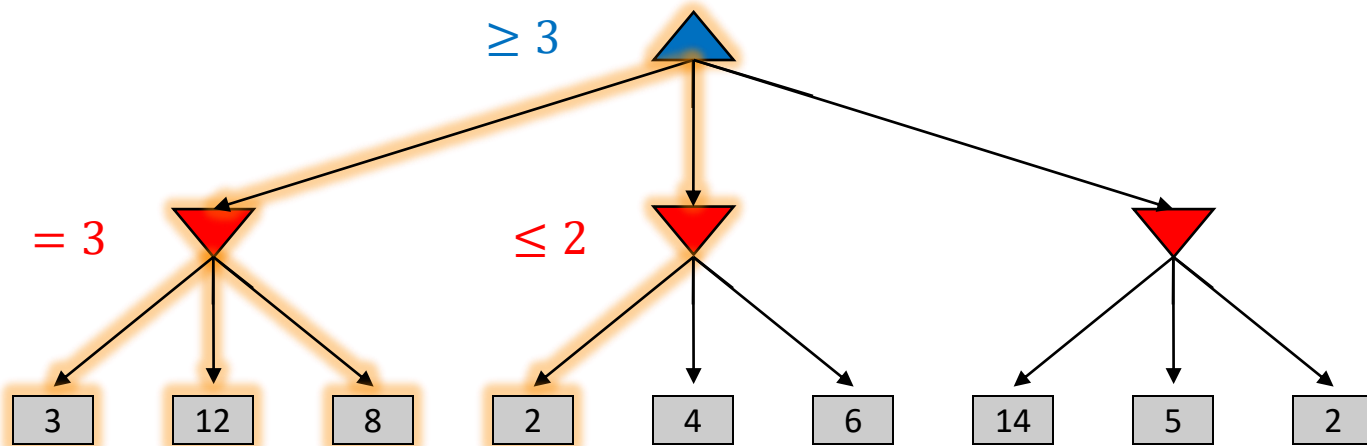
MAX

≥ 3

MIN

$= 3$

≤ 2



Ôn lại thuật toán minimax

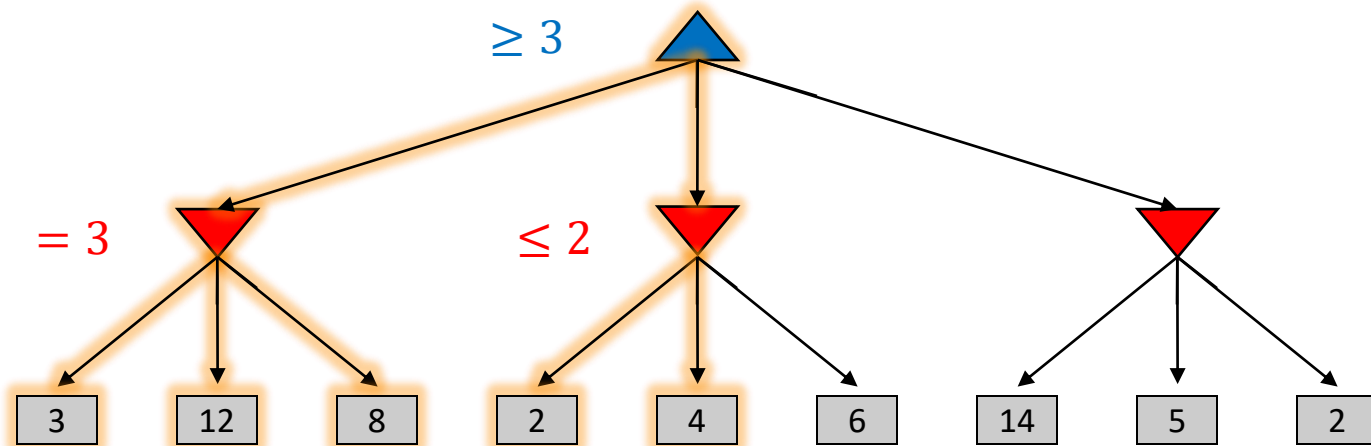
MAX

≥ 3

MIN

$= 3$

≤ 2



Ôn lại thuật toán minimax

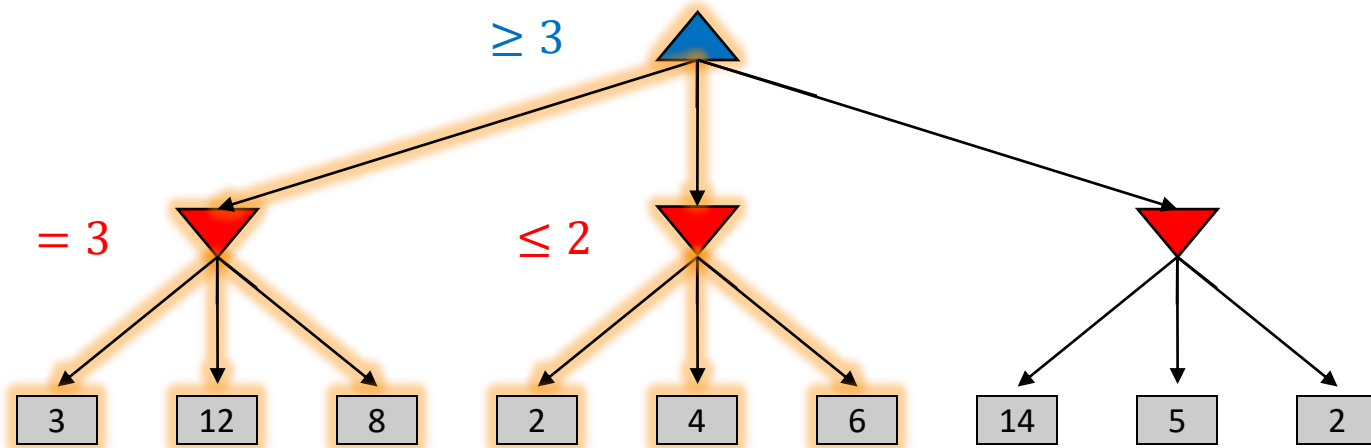
MAX

≥ 3

MIN

$= 3$

≤ 2



Ôn lại thuật toán minimax

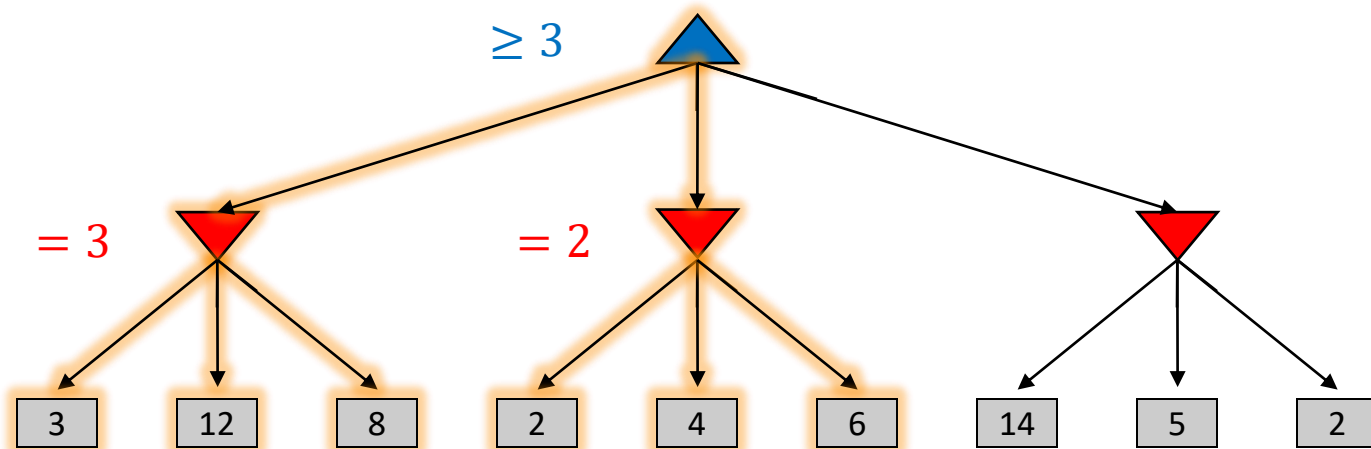
MAX

≥ 3

MIN

$= 3$

$= 2$



Ôn lại thuật toán minimax

MAX

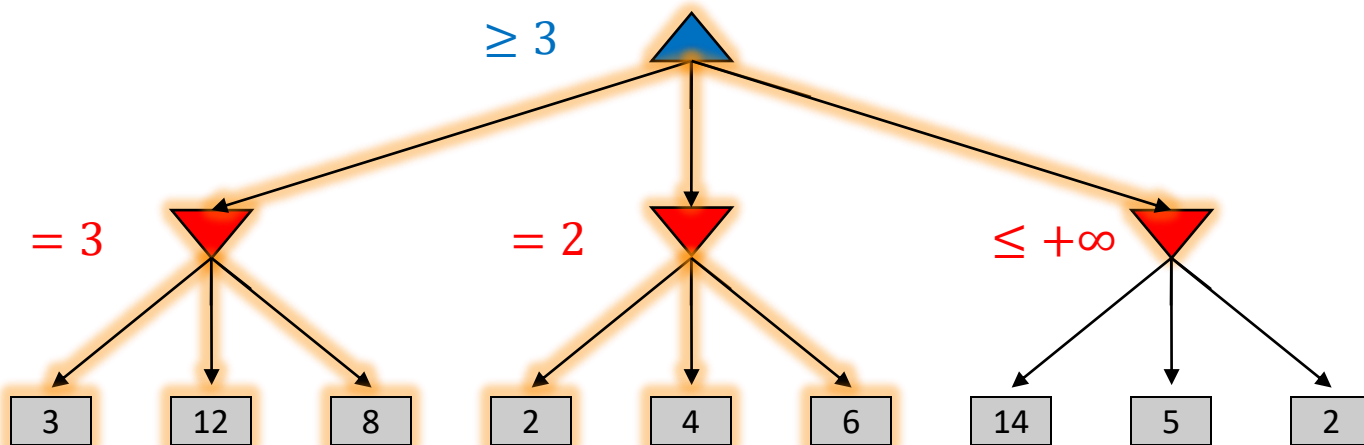
≥ 3

MIN

$= 3$

$= 2$

$\leq +\infty$



Ôn lại thuật toán minimax

MAX

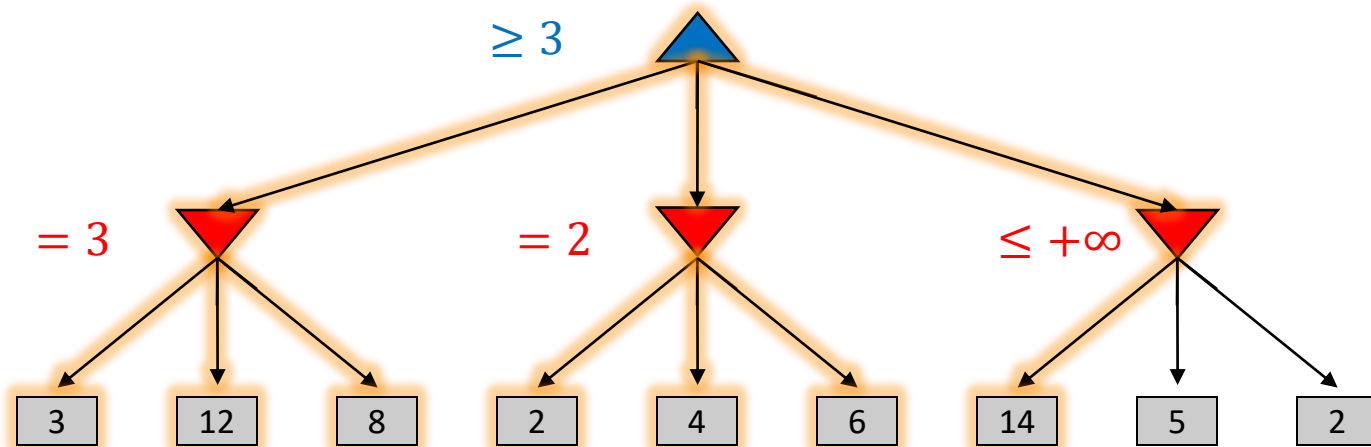
≥ 3

MIN

$= 3$

$= 2$

$\leq +\infty$



Ôn lại thuật toán minimax

MAX

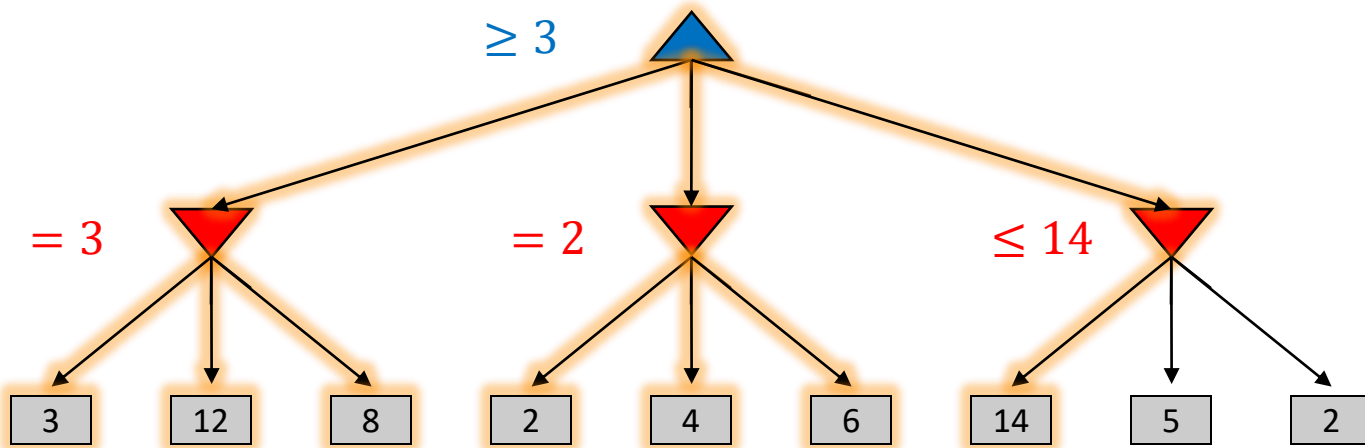
≥ 3

MIN

$= 3$

$= 2$

≤ 14



Ôn lại thuật toán minimax

MAX

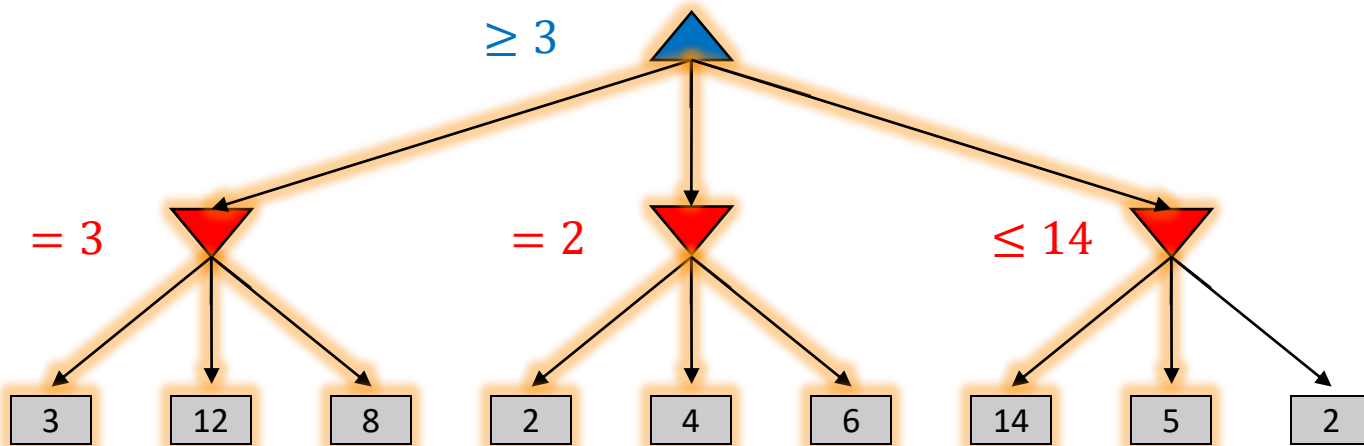
≥ 3

MIN

$= 3$

$= 2$

≤ 14



Ôn lại thuật toán minimax

MAX

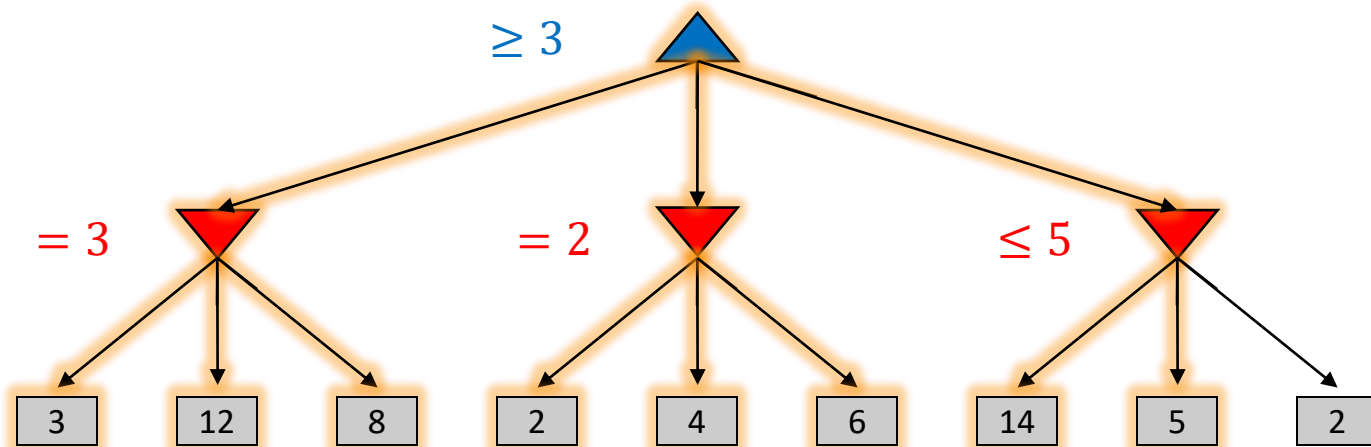
≥ 3

MIN

$= 3$

$= 2$

≤ 5



Ôn lại thuật toán minimax

MAX

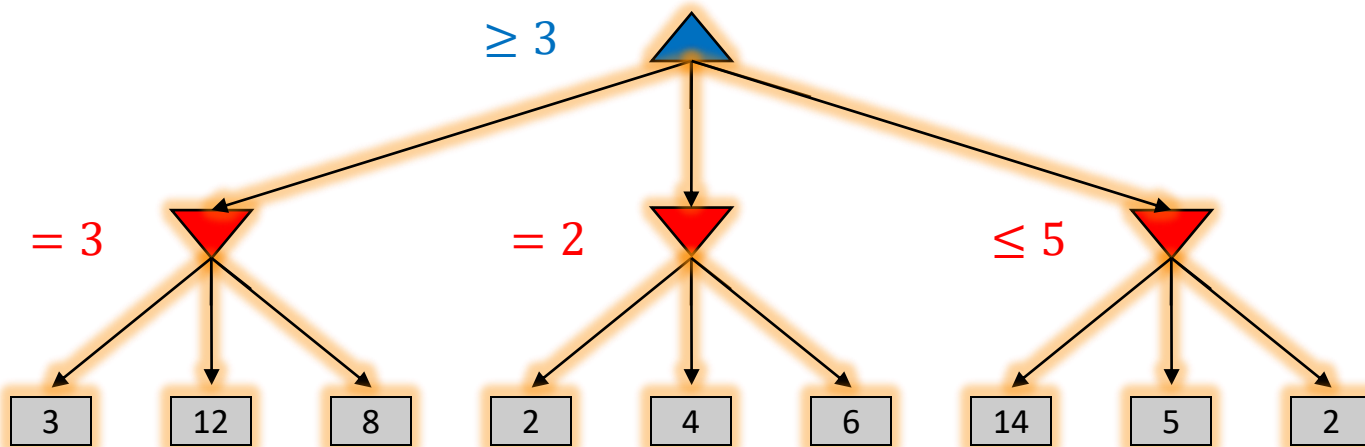
≥ 3

MIN

$= 3$

$= 2$

≤ 5

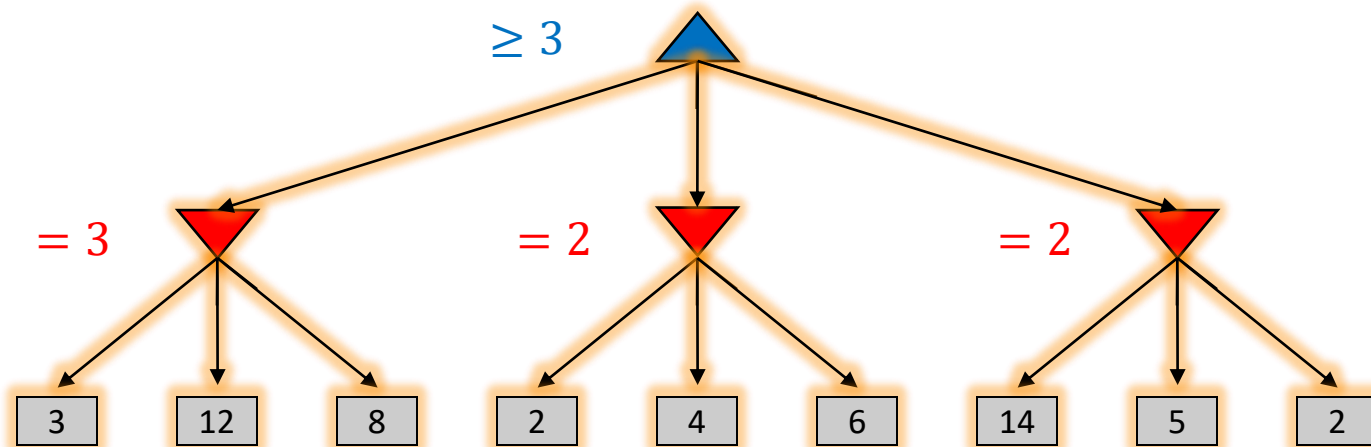


Ôn lại thuật toán minimax

MAX

≥ 3

MIN



Ôn lại thuật toán minimax

MAX

= 3

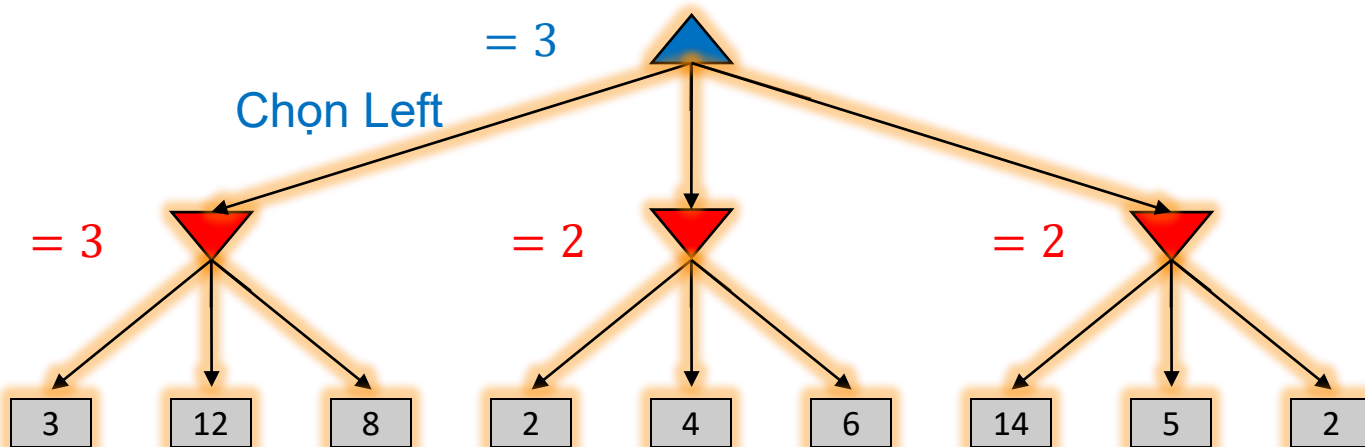
Chọn Left

MIN

= 3

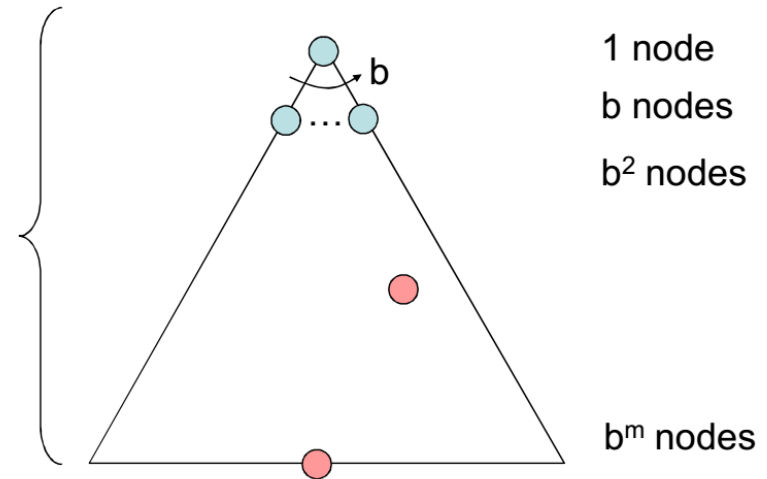
= 2

= 2



Vấn đề của thuật toán minimax

- Số node (trạng thái) phải duyệt?
 - Nguyên cả cây!
 - Giả sử cây có hệ số phân nhánh b và có độ sâu $m \rightarrow$ Số node của cây: $1 + b + b^2 + \dots + b^m = O(b^m)$
 - Cờ vua có $b \approx 35$, $m \approx 100 \rightarrow$ Minimax sẽ chạy mãi mãi ...
- Ta sẽ điều chỉnh lại thuật toán minimax để khả thi hơn

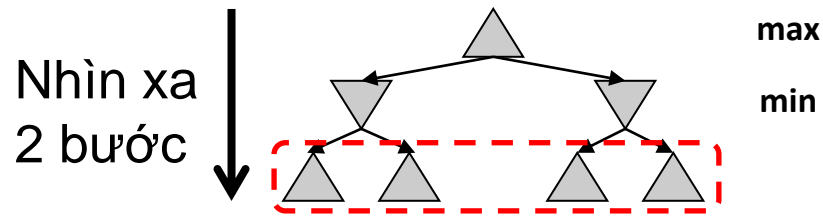


Tổng thể

- Làm cho thuật toán minimax khả thi: giới hạn số bước nhìn về tương lai phía trước
- Làm cho thuật toán minimax tốt hơn nữa: tỉa alpha-beta

Làm cho thuật toán minimax khả thi

- Từ trạng thái hiện tại, ta muốn biết là nên thực hiện hành động nào?
- Thay vì nhìn về tương lai cho tới khi game kết thúc (là điều không khả thi), ta chỉ nhìn về tương lai một số bước



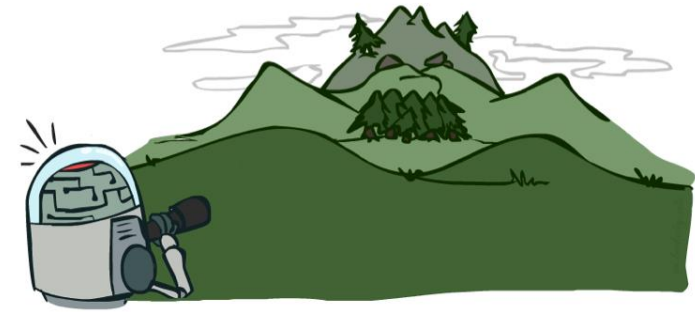
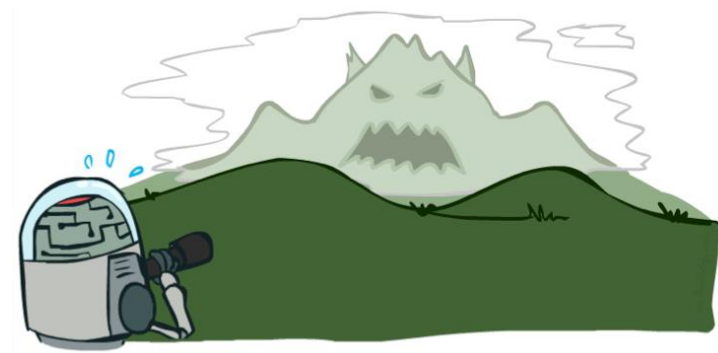
Ta sẽ **ước lượng giá trị minimax** của các trạng thái dưới cùng bằng một **hàm ước lượng** (evaluation function)

Rồi chạy thuật toán minimax để tính giá trị minimax của các trạng thái trước đó

- Thuật toán minimax có còn tối ưu (nếu đối thủ chơi tối ưu)?
 - Không, vì giá trị minimax của các trạng thái dưới cùng chỉ là giá trị ước lượng

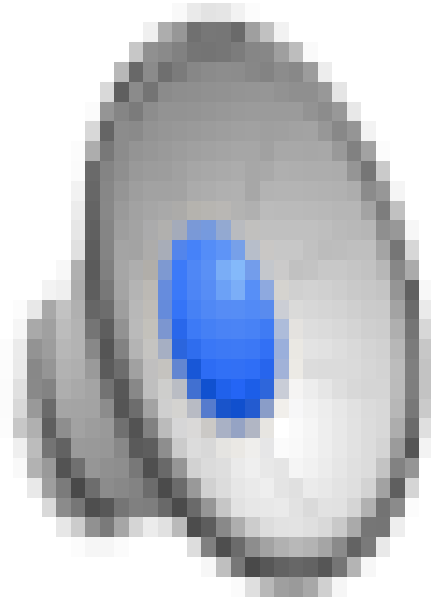
Nhìn càng xa thì sẽ càng tốt

- Nhìn càng xa thì sẽ càng tốn thời gian tính toán nhưng việc ước lượng giá trị minimax sẽ càng chính xác
 - Vd: trong cờ vua, với hàm ước lượng đơn giản $Eval(s) = \text{số lượng hậu trắng} - \text{số lượng hậu đen}$, nếu ta chỉ nhìn xa một số ít bước (khi mà cả 2 bên đều còn hậu) thì $Eval(s) = 0$ và sẽ không giúp ích được gì trong việc chọn hành-động/nước-đi
- Với một trạng thái bàn cờ, nếu bạn phải ra quyết định nước đi trong một khoảng thời gian t thì nên nhìn bao xa?
 - Nhìn xa một bước
 - Nếu vẫn còn thời gian, nhìn xa hai bước
 - Nếu vẫn còn thời gian, nhìn xa ba bước
 - ...

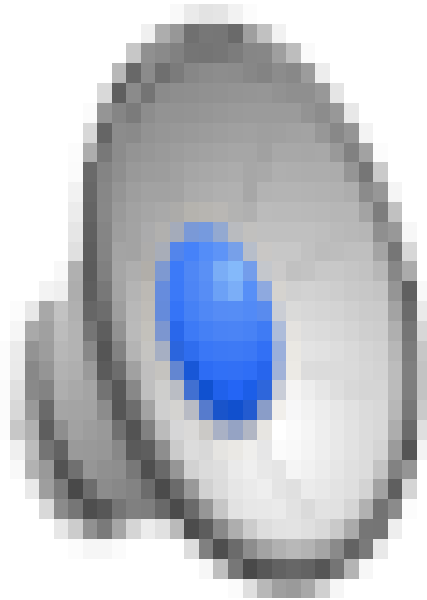


Ta luôn có được một quyết định, và quyết định này sẽ càng tốt khi thời gian có càng nhiều

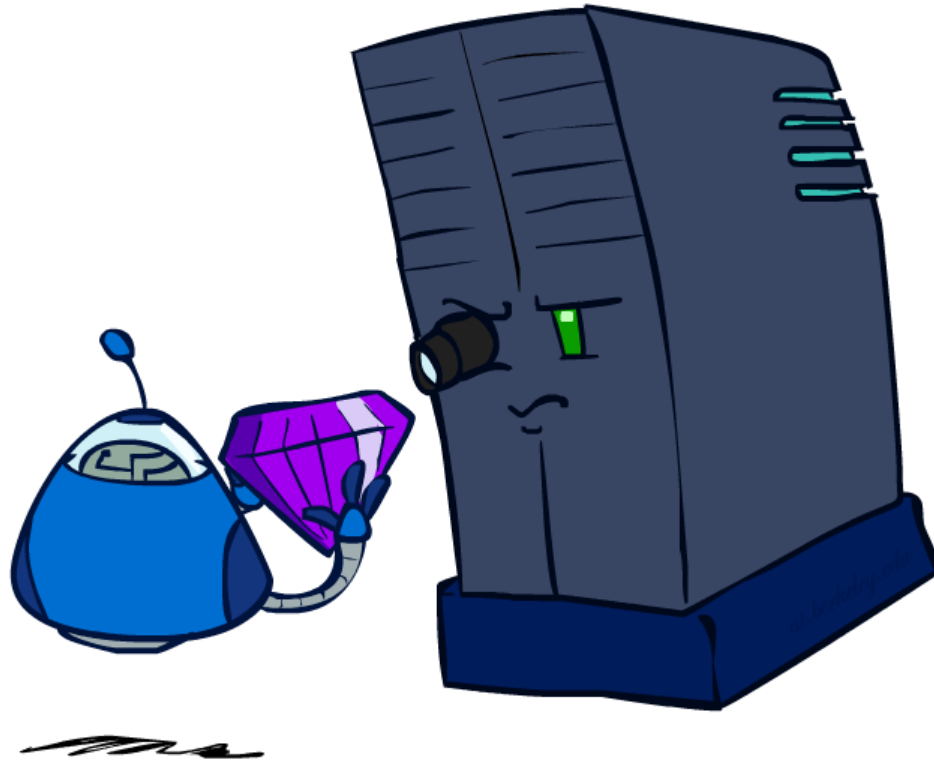
Video: Pacman nhìn không đủ xa



Video: Pacman nhìn siêu xa



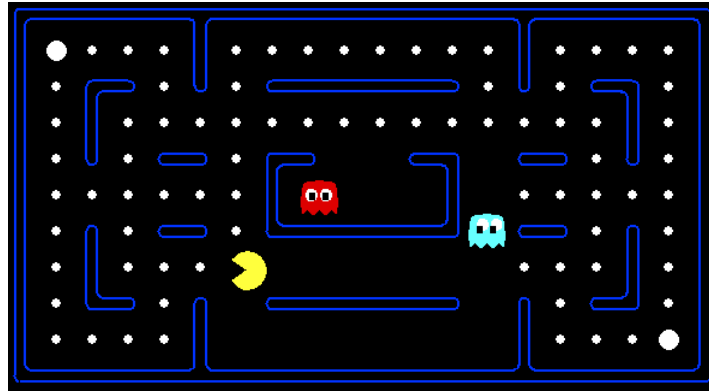
Nói kỹ hơn về hàm ước lượng



Hàm ước lượng

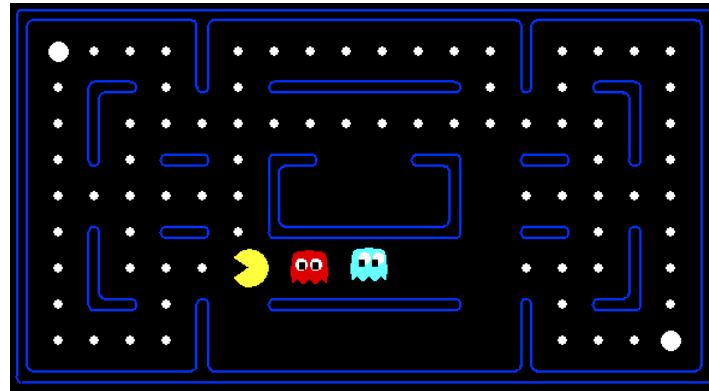
- Hàm ước lượng (evaluation function)
 - Đầu vào: một trạng thái (không phải trạng thái kết thúc)
 - Đầu ra: giá trị minimax ước lượng của trạng thái đó
- Với một game cụ thể, bạn phải thiết kế hàm ước lượng
- Thông thường, hàm ước lượng được thiết kế là tổng có trọng của nhiều đặc trưng
 - $Eval(s) = w_1 f_1(s) + w_2 f_2(s) + w_3 f_3(s) + \dots$
 - Vd: trong cờ vua, $f_1(s)$ = số lượng hậu trắng – số lượng hậu đen , $f_2(s)$ = số lượng mã trắng – số lượng mã đen, và ta nên cho trọng số w_1 của $f_1(s)$ lớn hơn trọng số w_2 của $f_2(s)$

Hàm ước lượng trong game Pacman



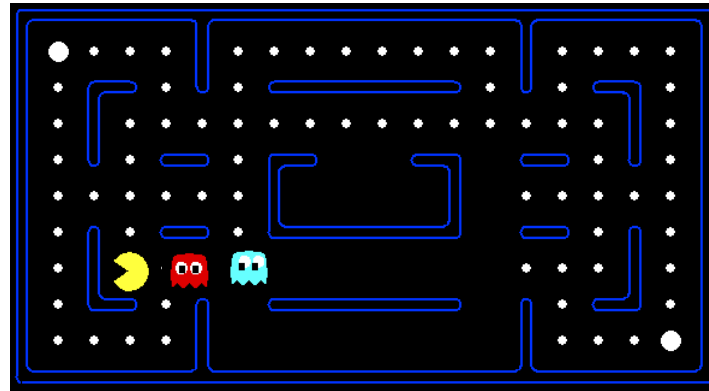
Với trạng thái này, bạn sẽ ước lượng giá trị minimax cao hay thấp?

Hàm ước lượng trong game Pacman



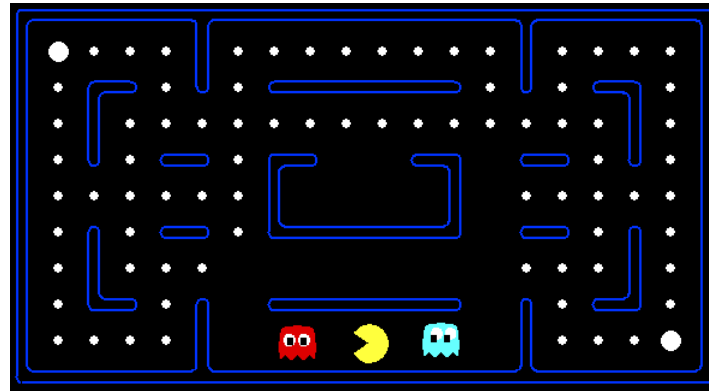
Với trạng thái này, bạn sẽ ước lượng giá trị minimax cao hay thấp?

Hàm ước lượng trong game Pacman



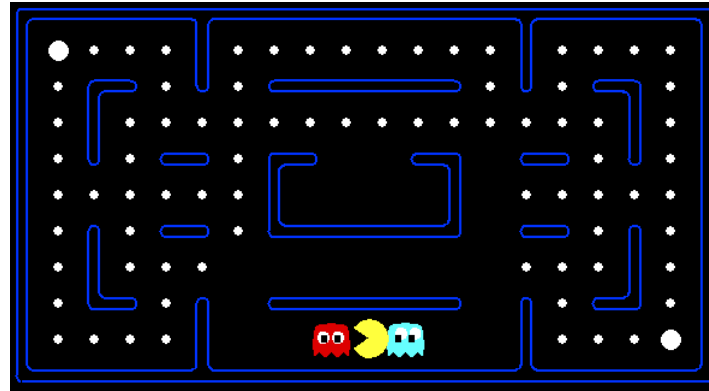
Với trạng thái này, bạn sẽ ước lượng giá trị minimax cao hay thấp?

Hàm ước lượng trong game Pacman



Với trạng thái này, bạn sẽ ước lượng giá trị minimax cao hay thấp?

Hàm ước lượng trong game Pacman



Với trạng thái này, bạn sẽ ước lượng giá trị minimax cao hay thấp?

Video: Paman bị “đi qua đi lại”

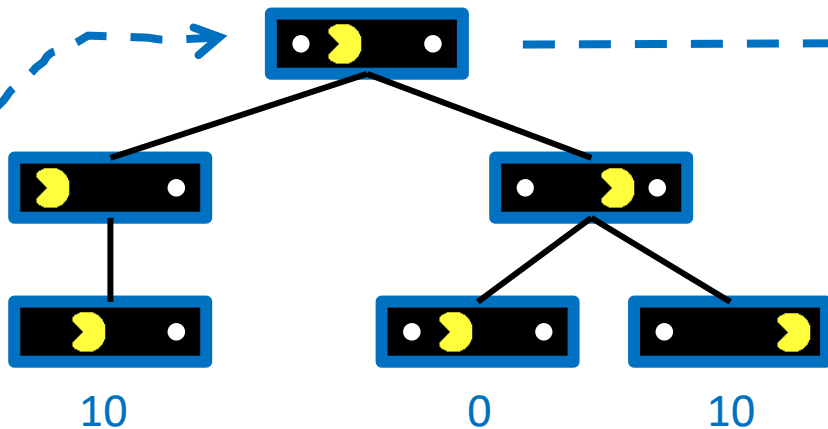
Xem “Thrashing.mp4”

- Nhìn xa 2 bước
- Hàm ước lượng: điểm số của game tại trạng thái đang xét

Tại sao Pacman bị “đi qua đi lại”?

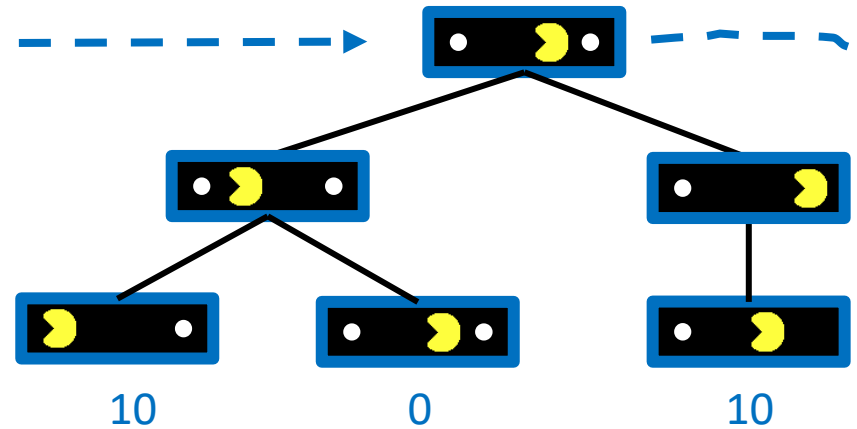
Nhìn xa hai bước từ trạng thái hiện tại

Hai trạng thái con ứng với hai hành động qua trái và qua phải có cùng giá trị \rightarrow chẳng hạn chọn qua phải



Nhìn xa hai bước từ trạng thái hiện tại

Hai trạng thái con ứng với hai hành động qua trái và qua phải có cùng giá trị \rightarrow chẳng hạn chọn qua trái



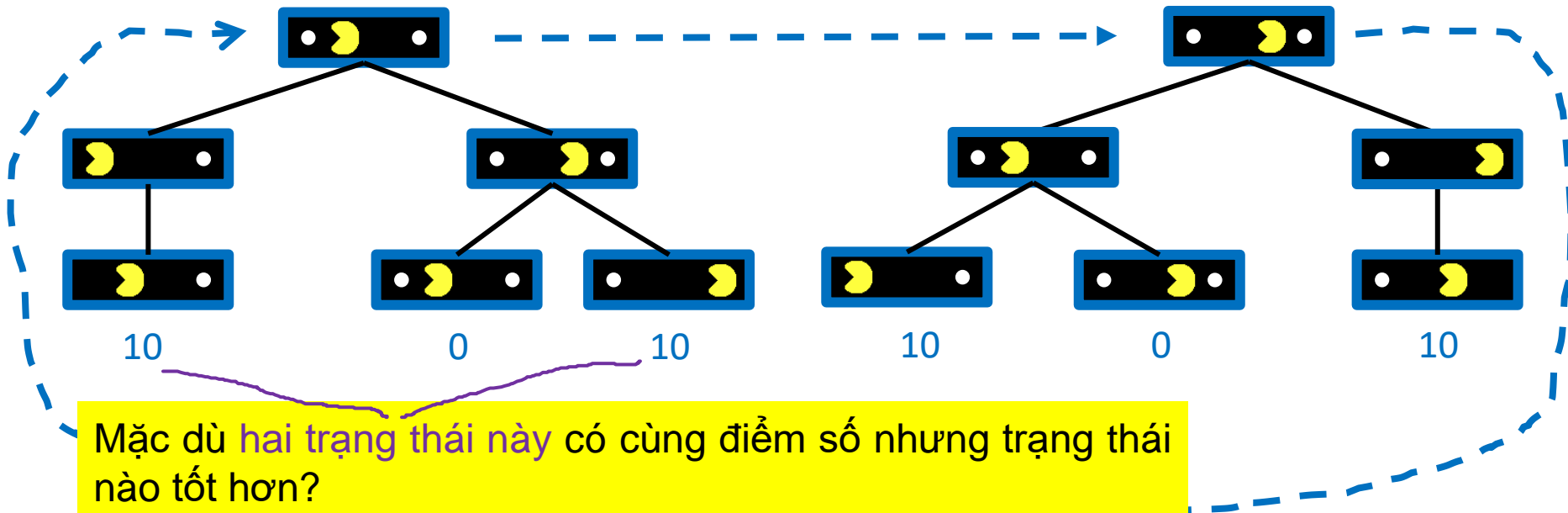
Tại sao Pacman bị “đi qua đi lại”?

Nhìn xa hai bước từ trạng thái hiện tại

Hai trạng thái con ứng với hai hành động qua trái và qua phải có cùng giá trị \rightarrow chẳng hạn chọn qua phải

Nhìn xa hai bước từ trạng thái hiện tại

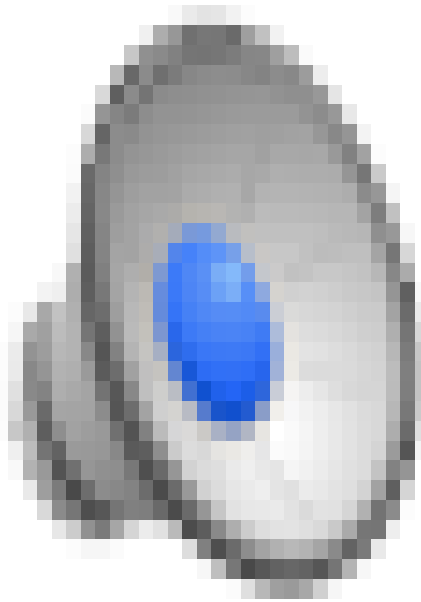
Hai trạng thái con ứng với hai hành động qua trái và qua phải có cùng giá trị \rightarrow chẳng hạn chọn qua trái



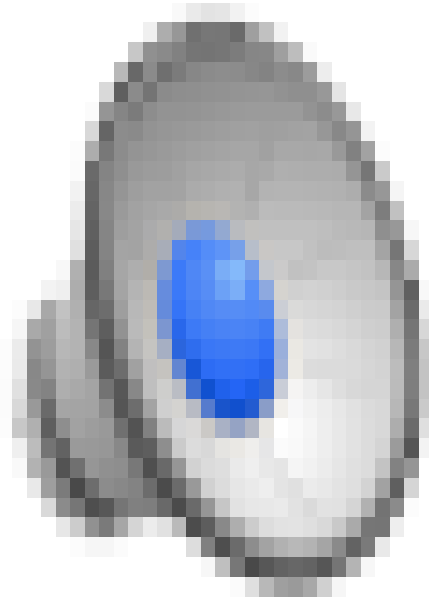
Mặc dù hai trạng thái này có cùng điểm số nhưng trạng thái nào tốt hơn?

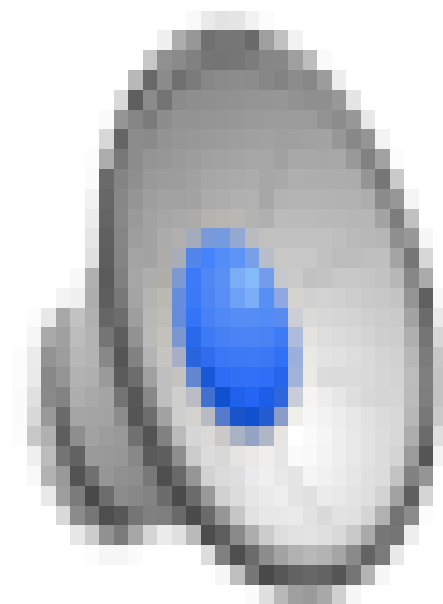
Ngoài điểm số, nếu ta dùng thêm thông tin về khoảng cách tới điểm thức ăn để ước lượng giá trị minimax thì Pacman sẽ hết bị “đi qua đi lại”

Video: Paman hết bị “đi qua đi lại”



Video: hiện tượng ma hợp tác với nhau

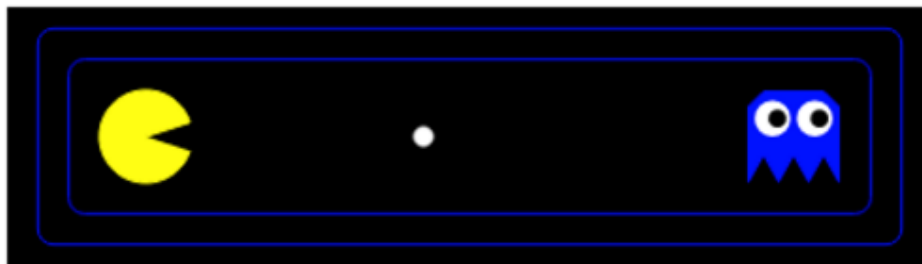




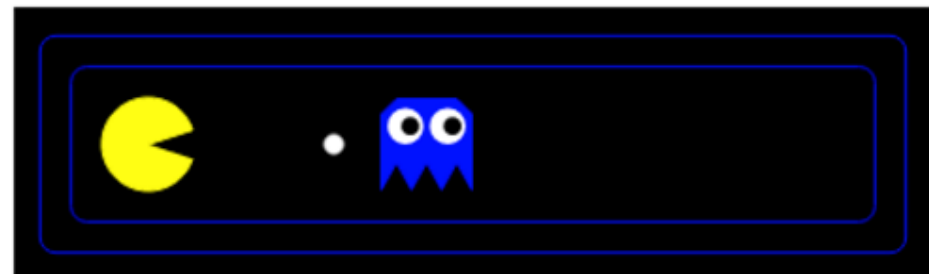
Bài tập: hàm ước lượng

Trong các hàm ước lượng dưới đây, hàm nào sẽ cho trạng thái bên trái giá trị cao hơn trạng thái bên phải?

- ☐ $1 / (\text{khoảng cách từ Pacman tới điểm thức ăn gần nhất})$
- ☐ Khoảng cách từ Pacman tới ma gần nhất
- ☐ Khoảng cách từ Pacman tới ma gần nhất + $1 / (\text{khoảng cách từ Pacman tới điểm thức ăn gần nhất})$
- ☐ Khoảng cách từ Pacman tới ma gần nhất + $1000 / (\text{khoảng cách từ Pacman tới điểm thức ăn gần nhất})$



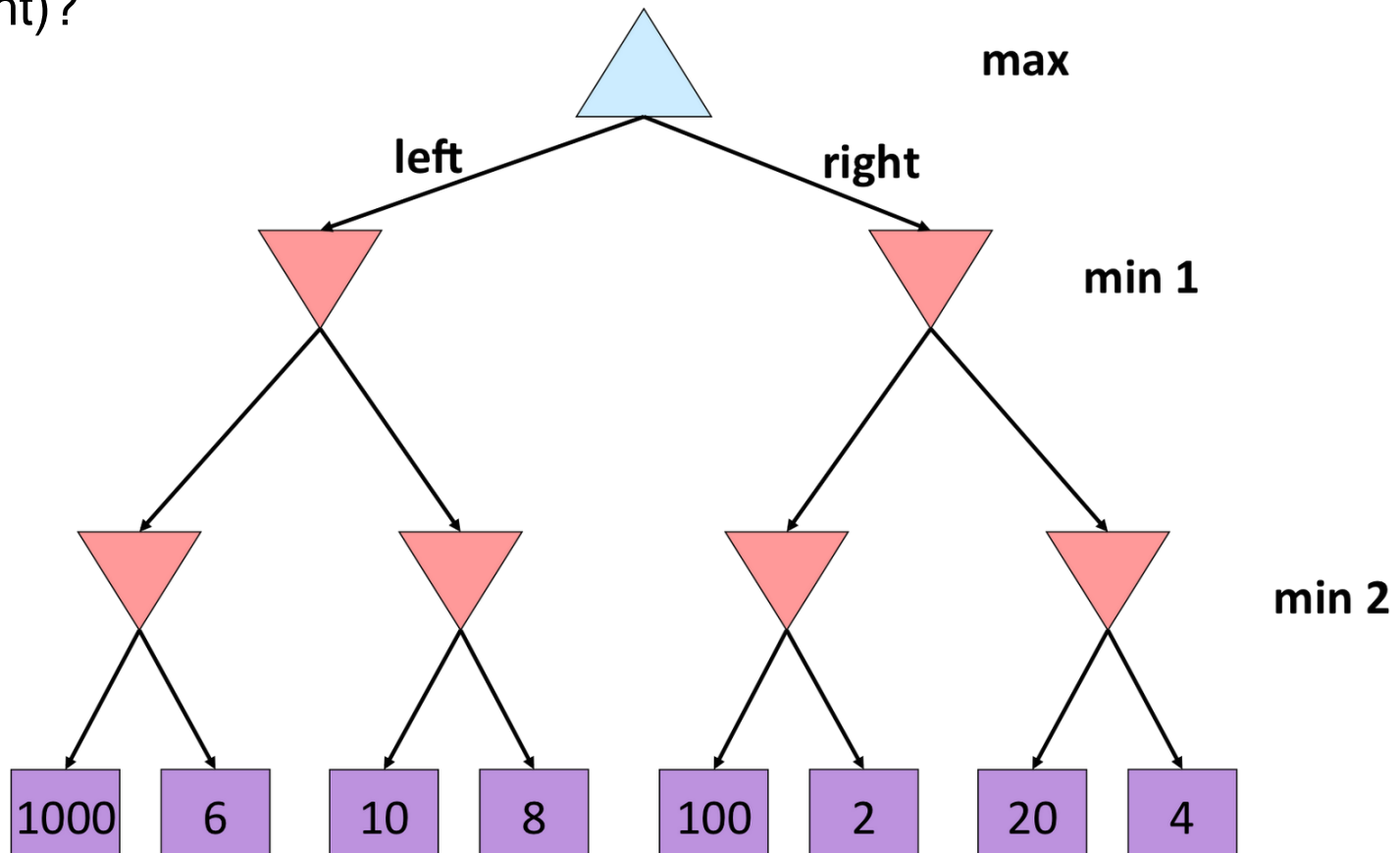
↑
1 2 3 4 5 6 7 8



↑ ↑ ↑
1 2 3 4 5 6 7 8

Bài tập: 2 người chơi min

- Tính giá trị minimax của các trạng thái?
- Từ trạng thái hiện tại (node gốc của cây), nên chọn thành động nào (left hay right)?



Tổng thể

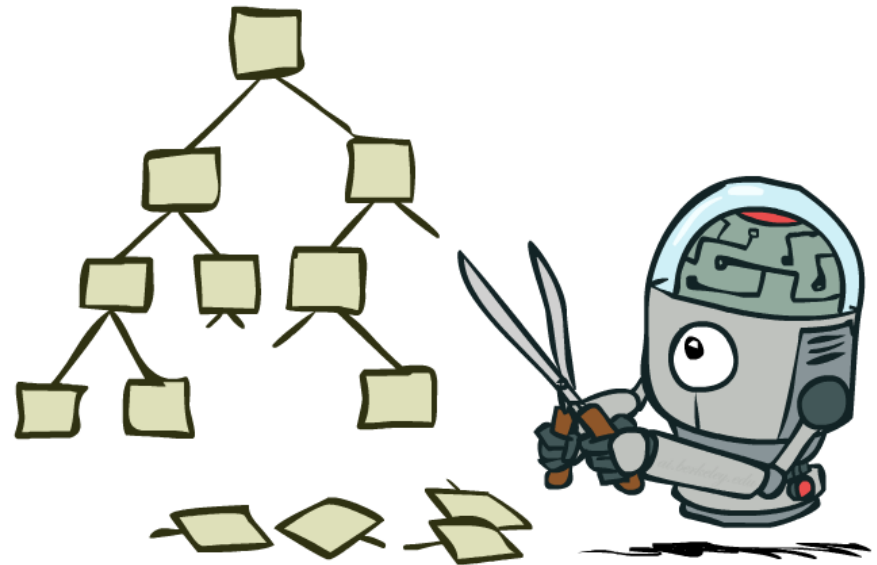
- Làm cho thuật toán minimax khả thi: giới hạn số bước nhìn về tương lai phía trước
- Làm cho thuật toán minimax tốt hơn nữa: tỉa alpha-beta

Tỉa alpha-beta

Tỉa alpha-beta sẽ giúp bỏ đi các nhánh mà không cần thiết phải duyệt

→ giảm thời gian tính toán lại

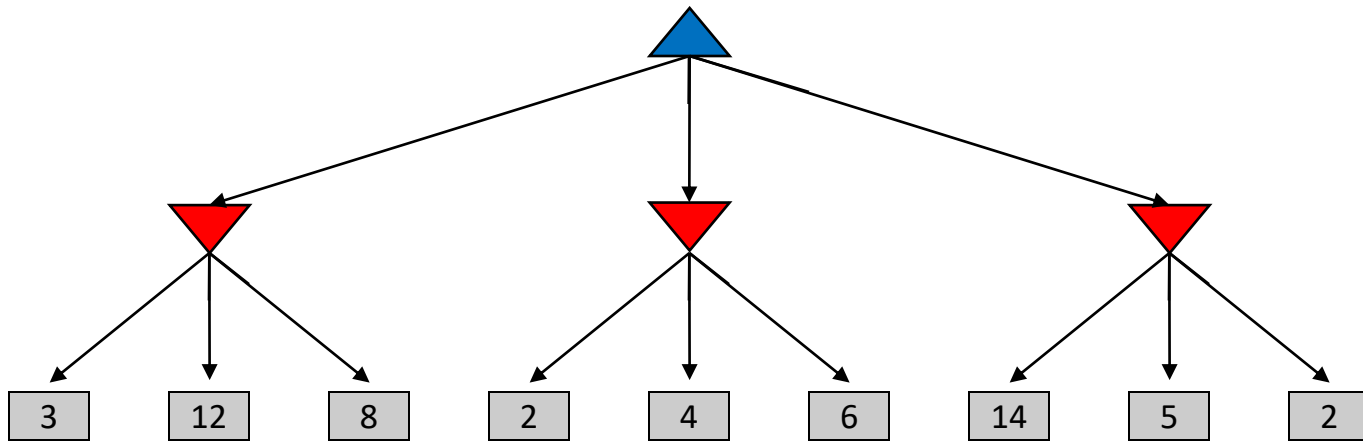
→ ta có thể đi xuống sâu hơn (nhìn xa hơn)



Tỉa alpha-beta: ví dụ 1

MAX

MIN

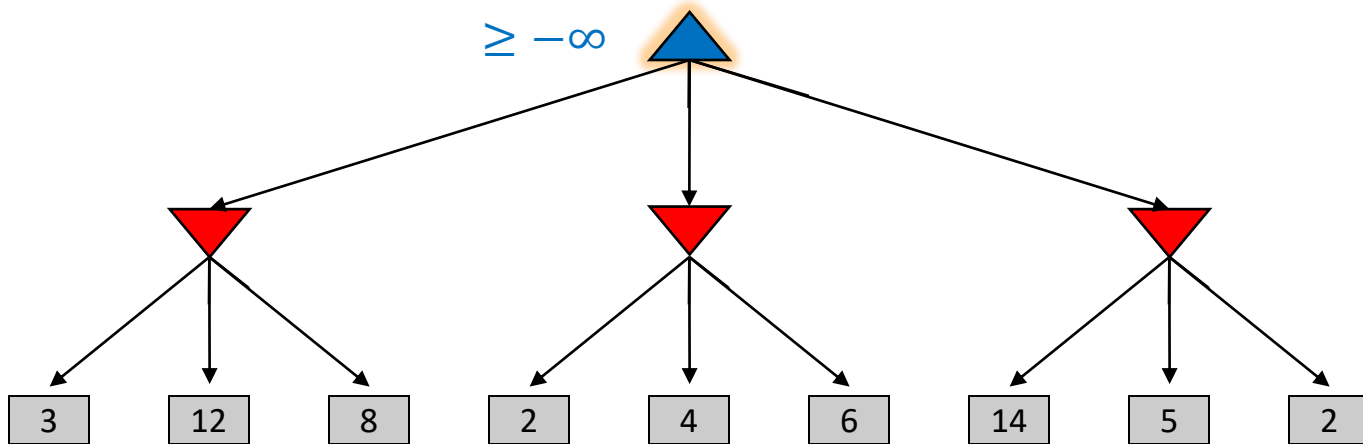


Tỉa alpha-beta: ví dụ 1

MAX

$\geq -\infty$

MIN



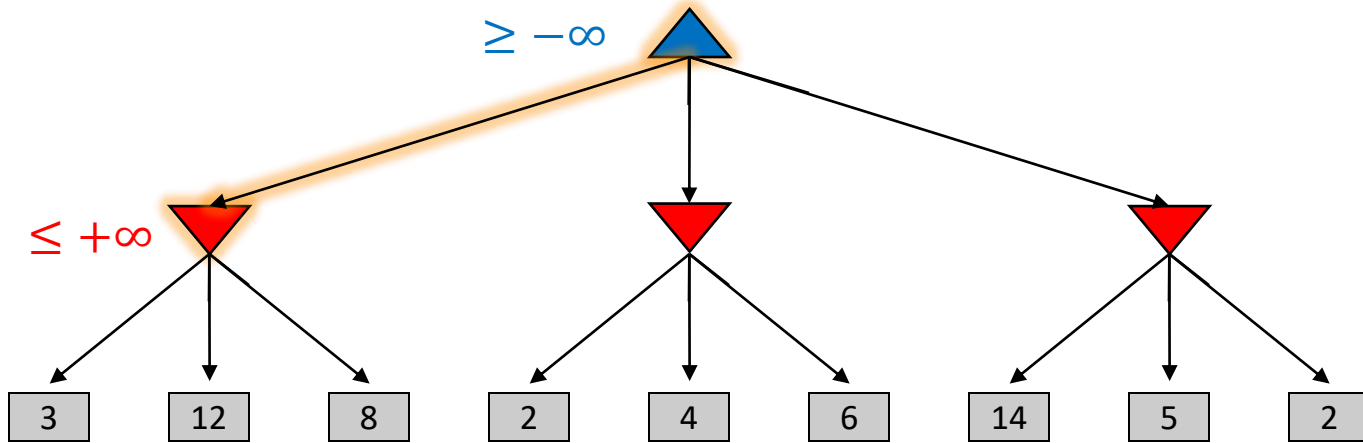
Tỉa alpha-beta: ví dụ 1

MAX

$\geq -\infty$

MIN

$\leq +\infty$



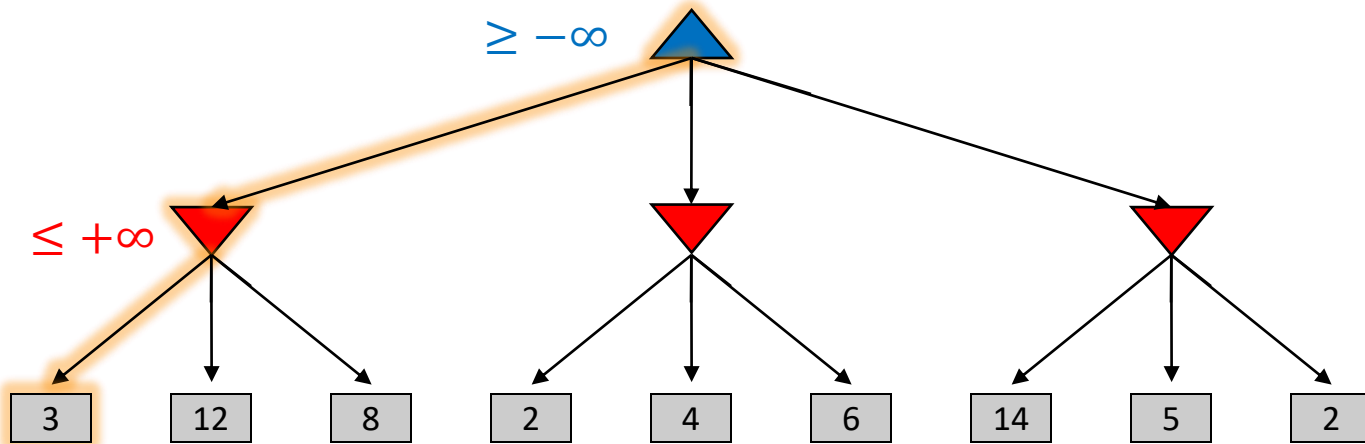
Tìm alpha-beta: ví dụ 1

MAX

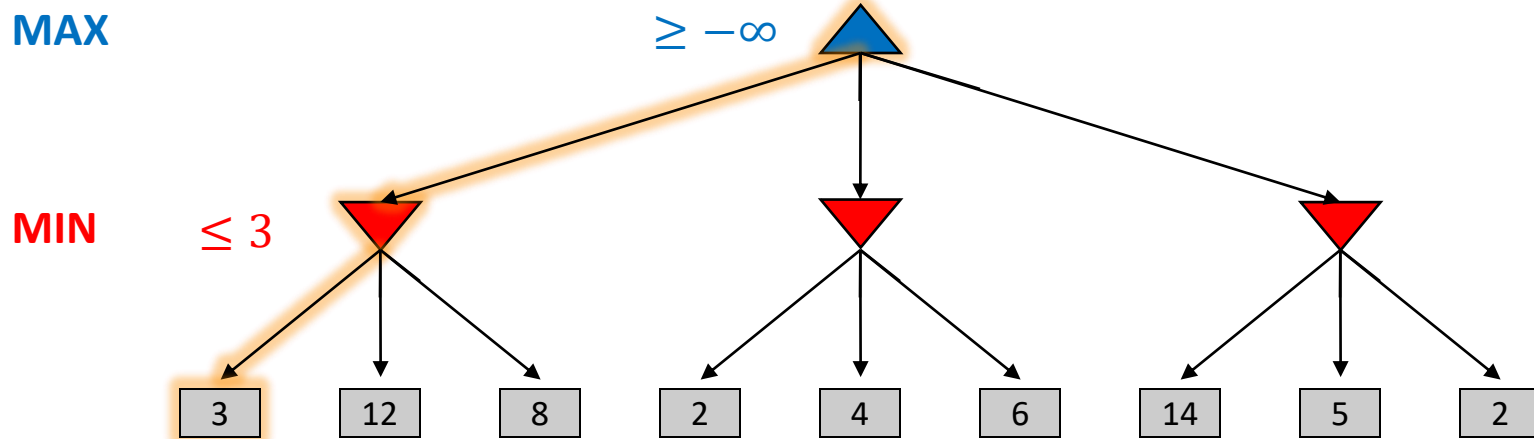
$\geq -\infty$

MIN

$\leq +\infty$



Tỉa alpha-beta: ví dụ 1



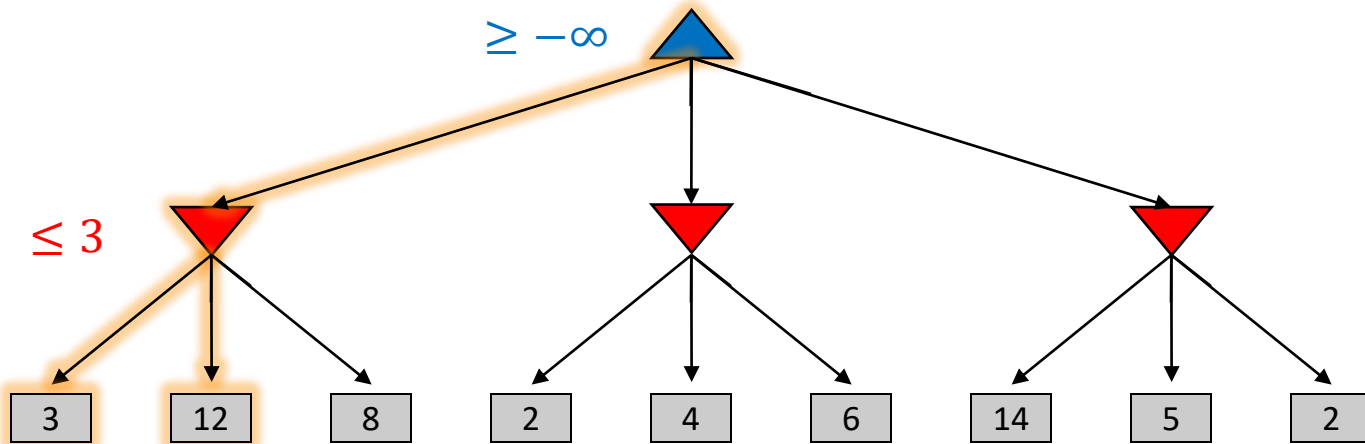
Tìm alpha-beta: ví dụ 1

MAX

$\geq -\infty$

MIN

≤ 3



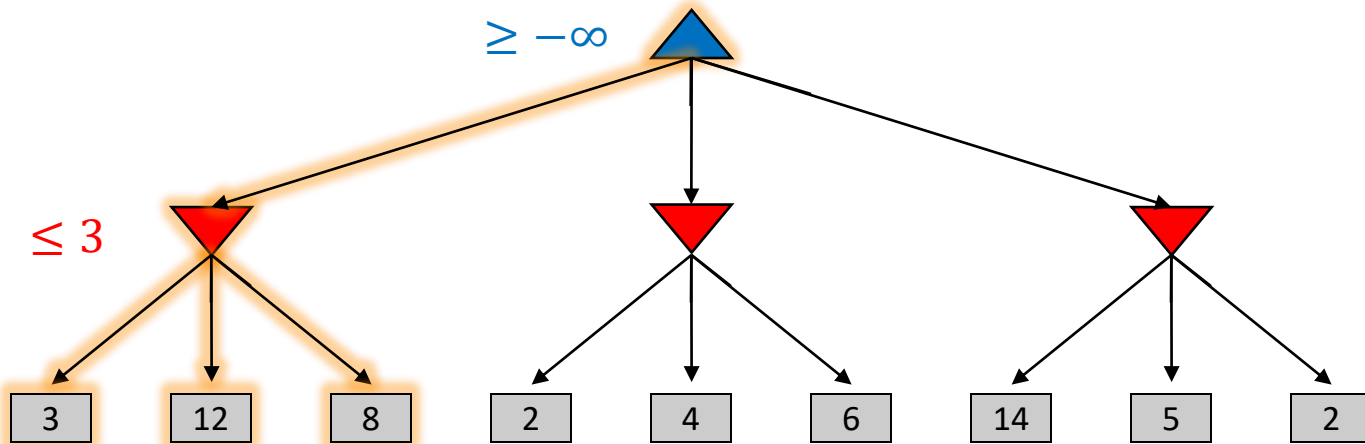
Tìm alpha-beta: ví dụ 1

MAX

$\geq -\infty$

MIN

≤ 3



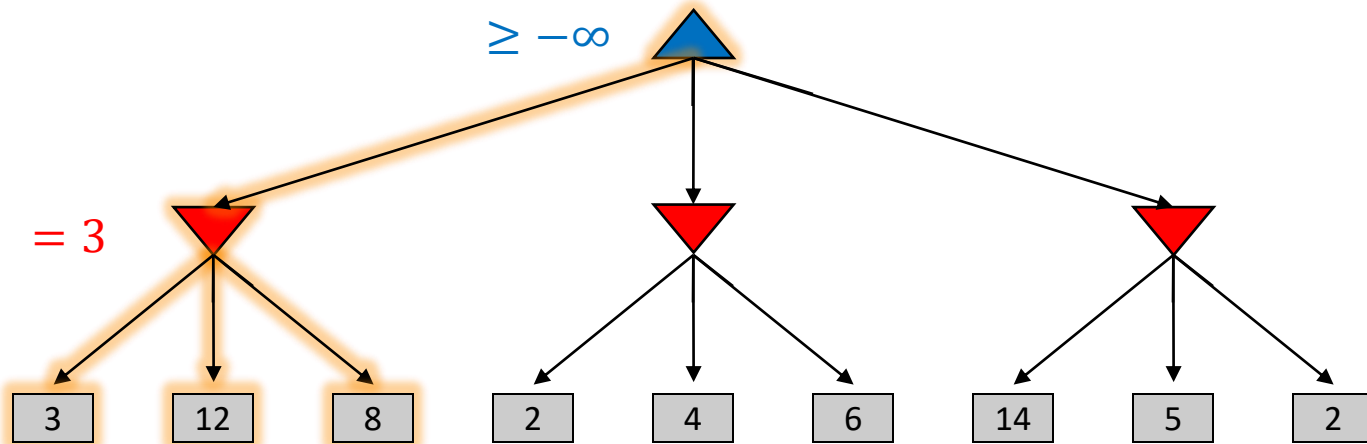
Tìm alpha-beta: ví dụ 1

MAX

$\geq -\infty$

MIN

$= 3$



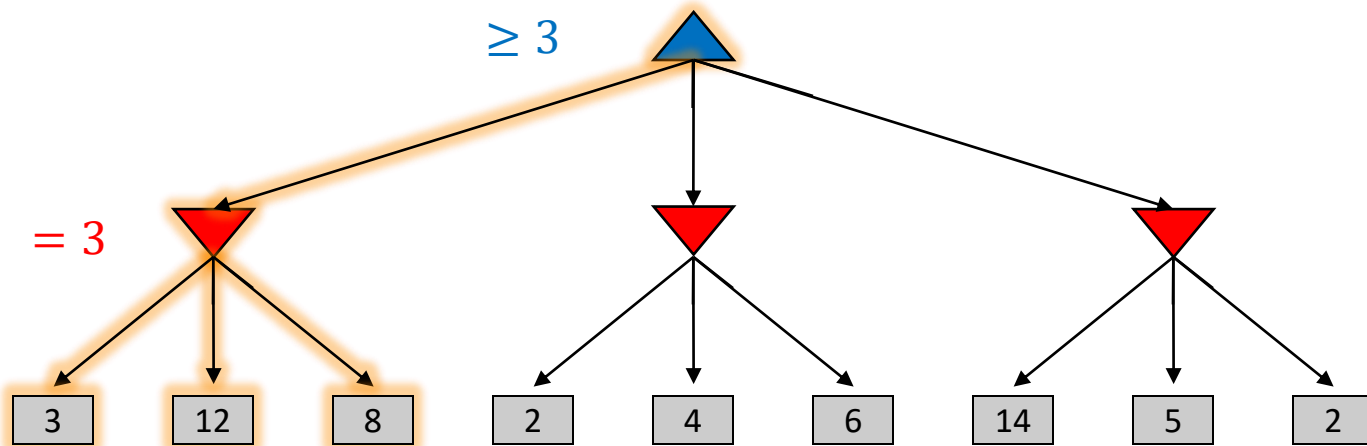
Tỉa alpha-beta: ví dụ 1

MAX

≥ 3

MIN

$= 3$



Tỉa alpha-beta: ví dụ 1

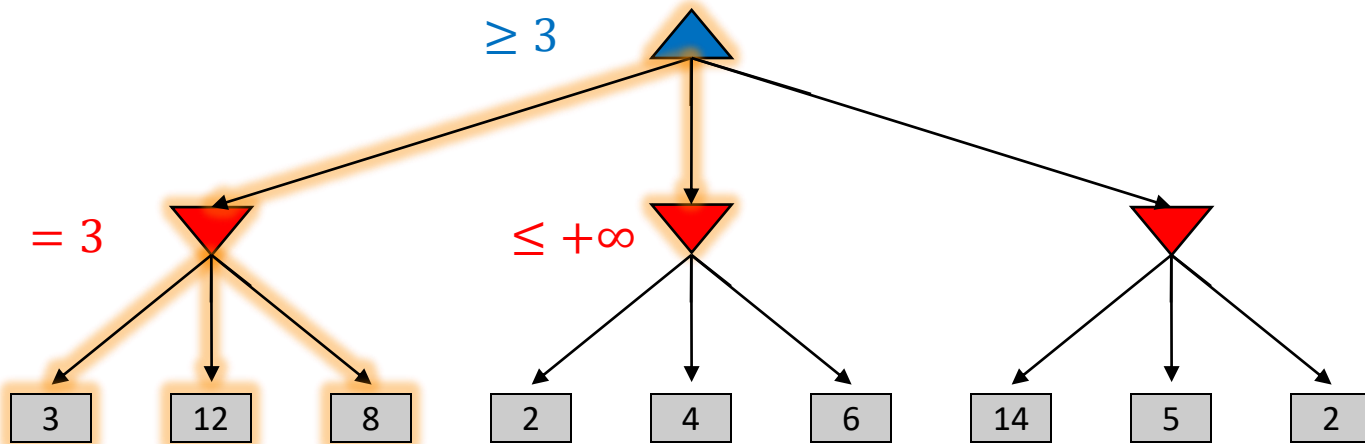
MAX

≥ 3

MIN

$= 3$

$\leq +\infty$



Tỉa alpha-beta: ví dụ 1

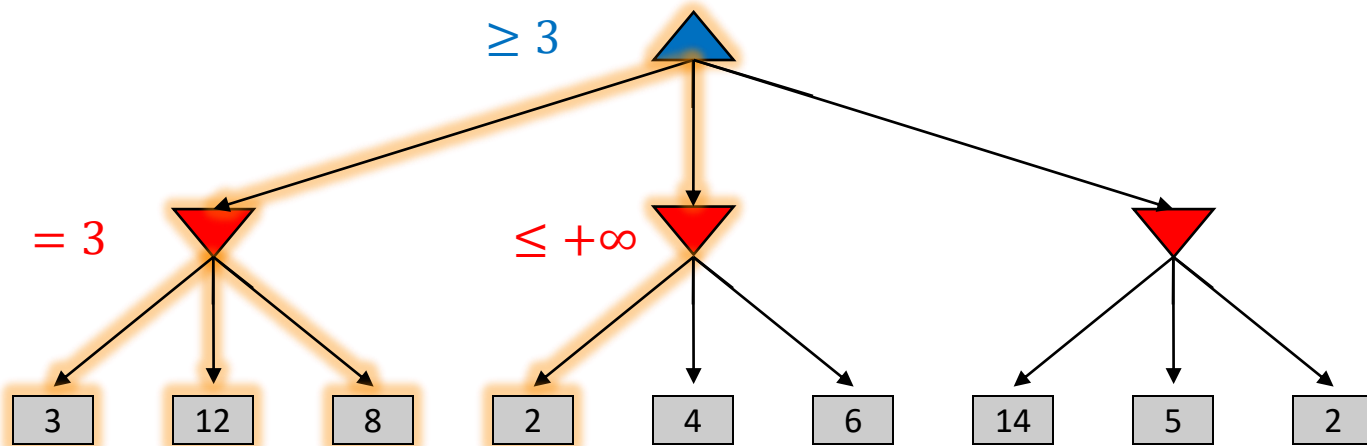
MAX

≥ 3

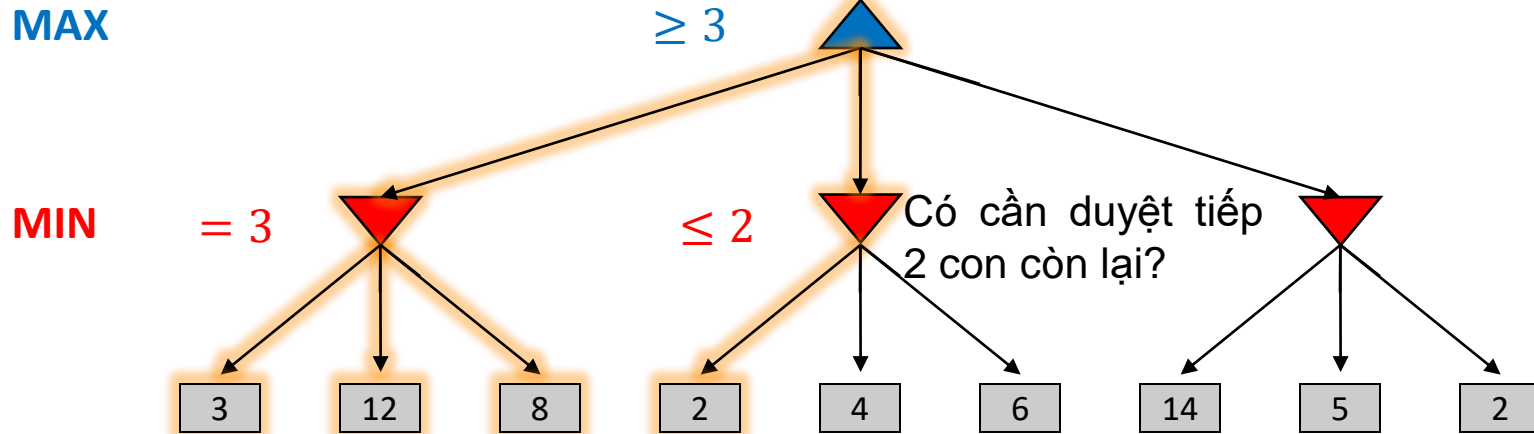
MIN

$= 3$

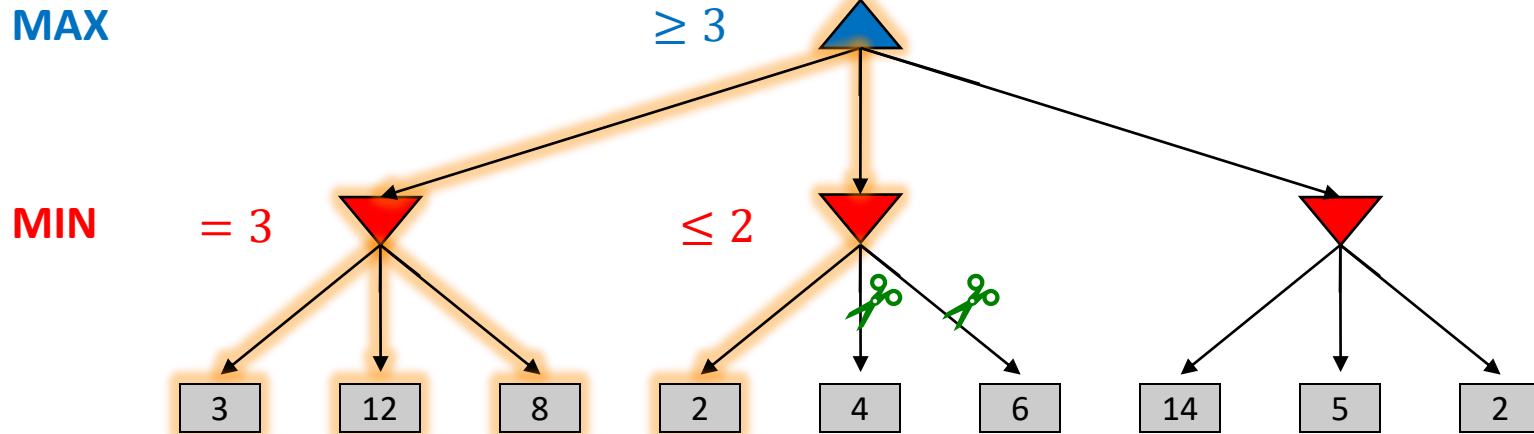
$\leq +\infty$



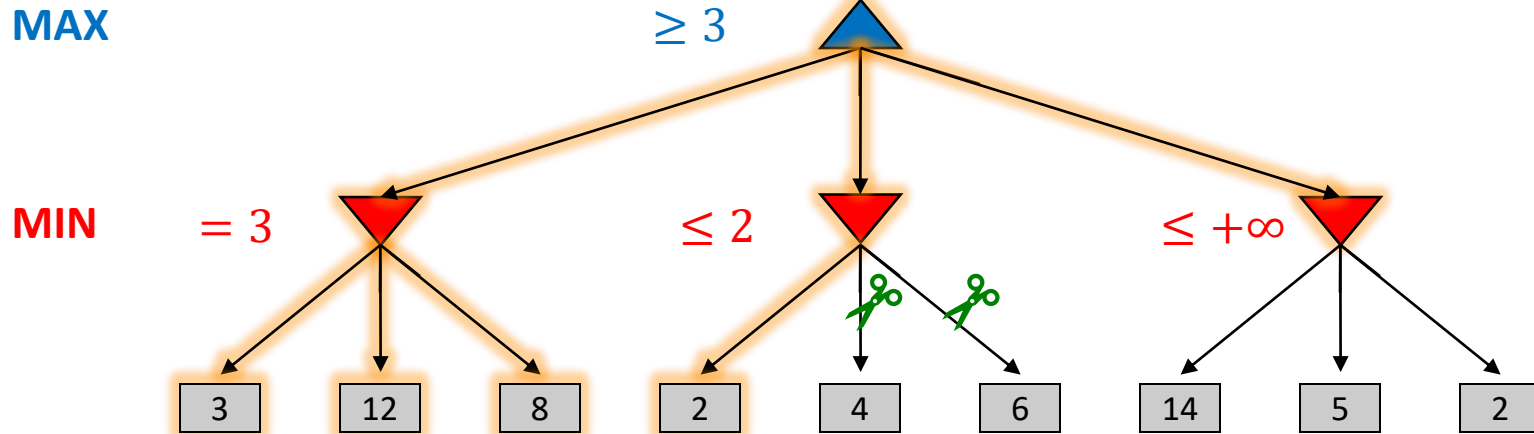
Tỉa alpha-beta: ví dụ 1



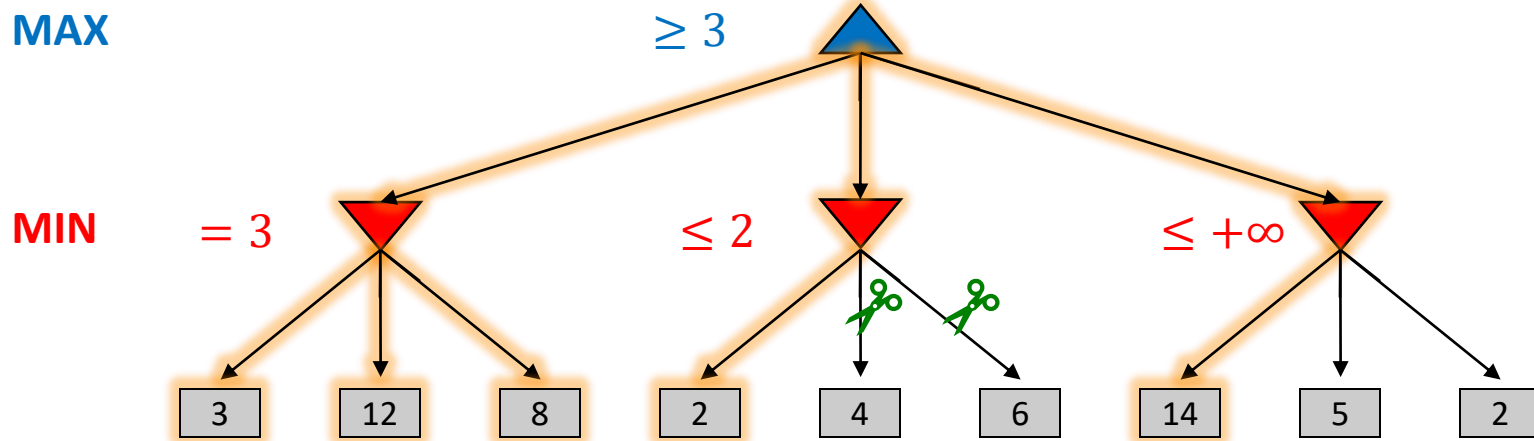
Tìm alpha-beta: ví dụ 1



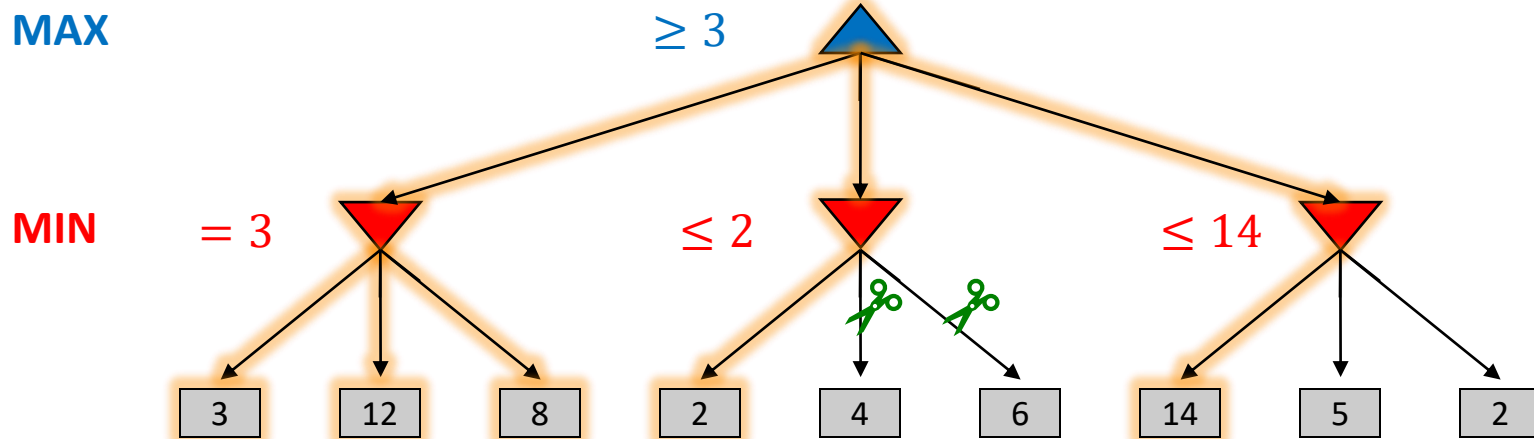
Tỉa alpha-beta: ví dụ 1



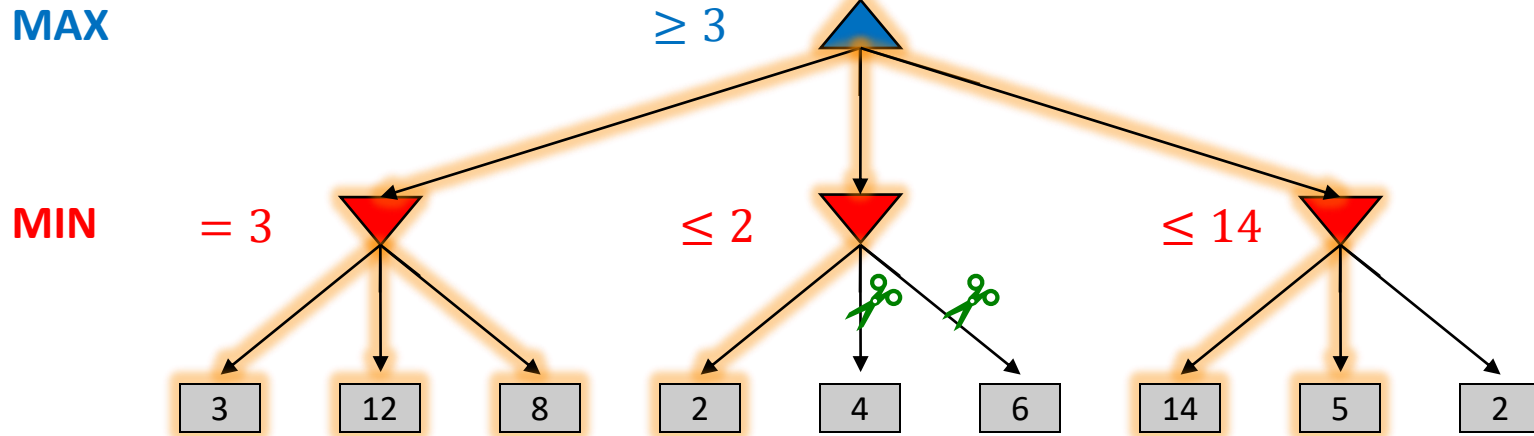
Tỉa alpha-beta: ví dụ 1



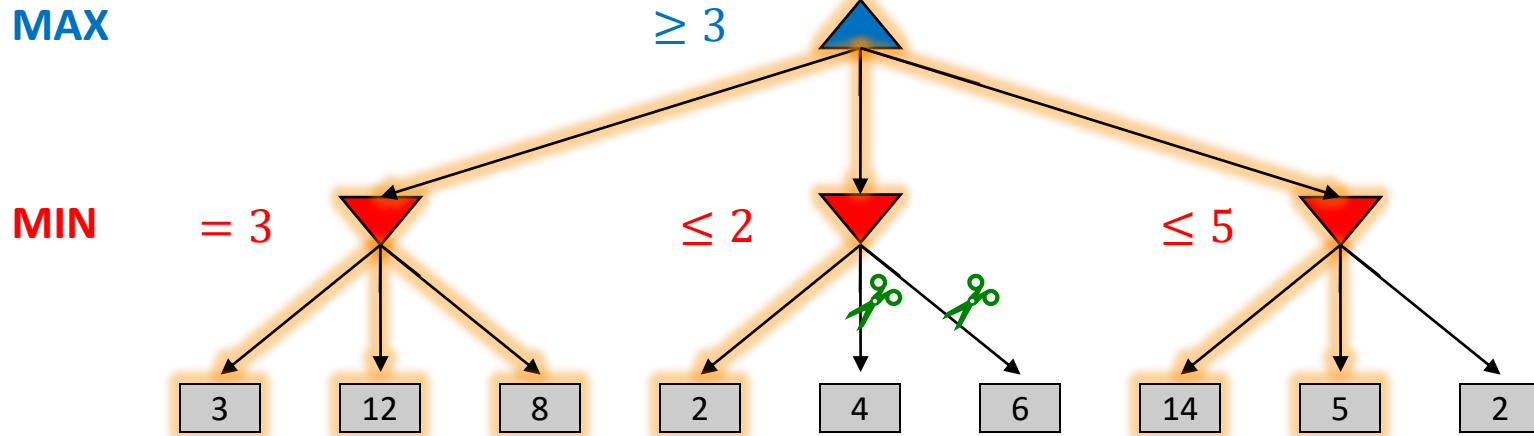
Tỉa alpha-beta: ví dụ 1



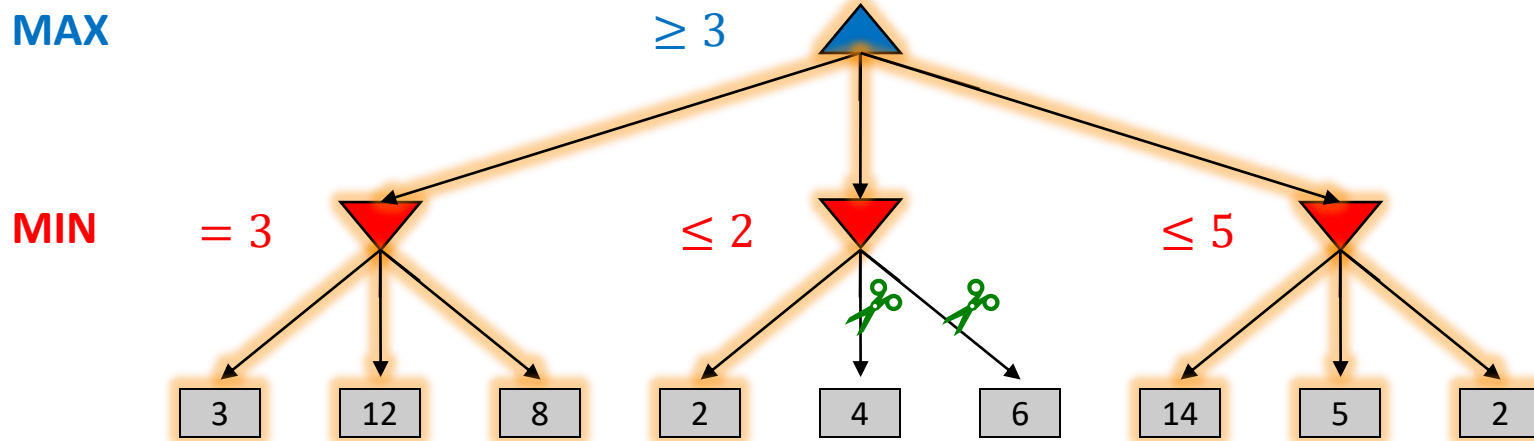
Tỉa alpha-beta: ví dụ 1



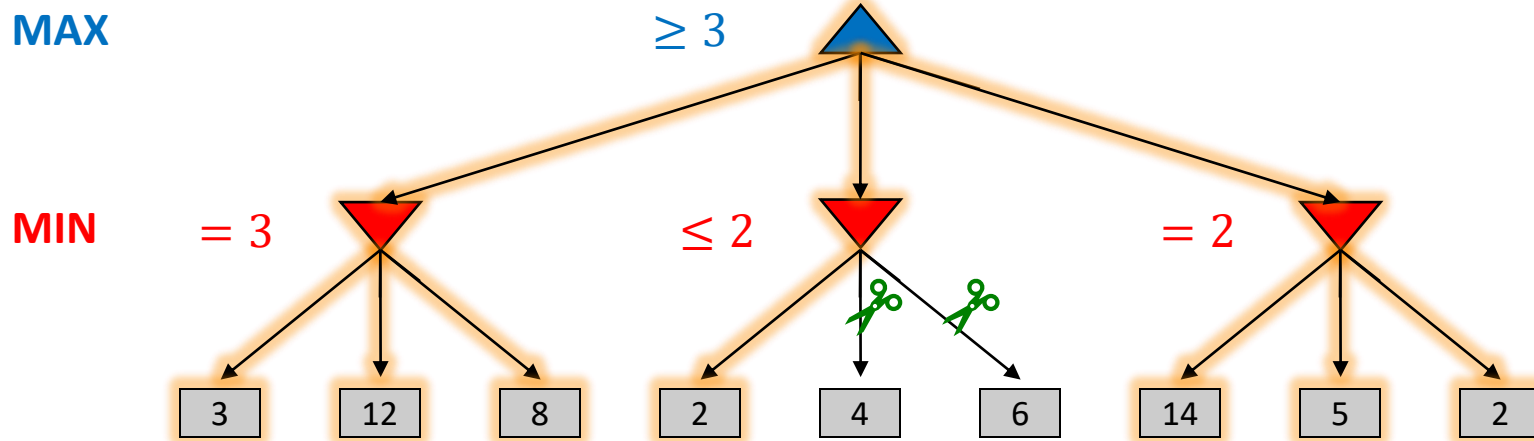
Tỉa alpha-beta: ví dụ 1



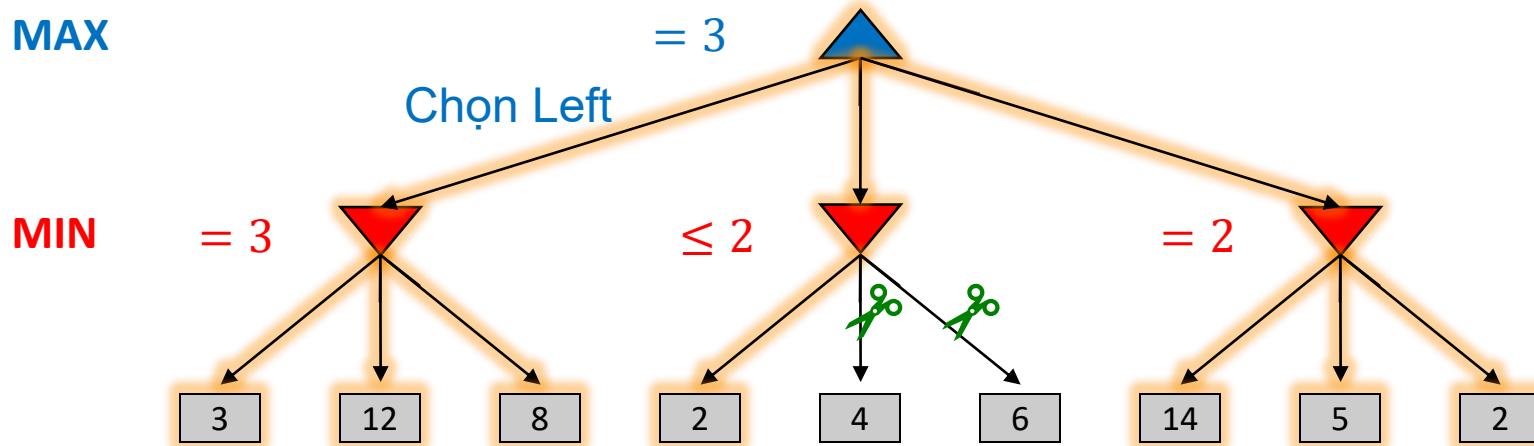
Tỉa alpha-beta: ví dụ 1



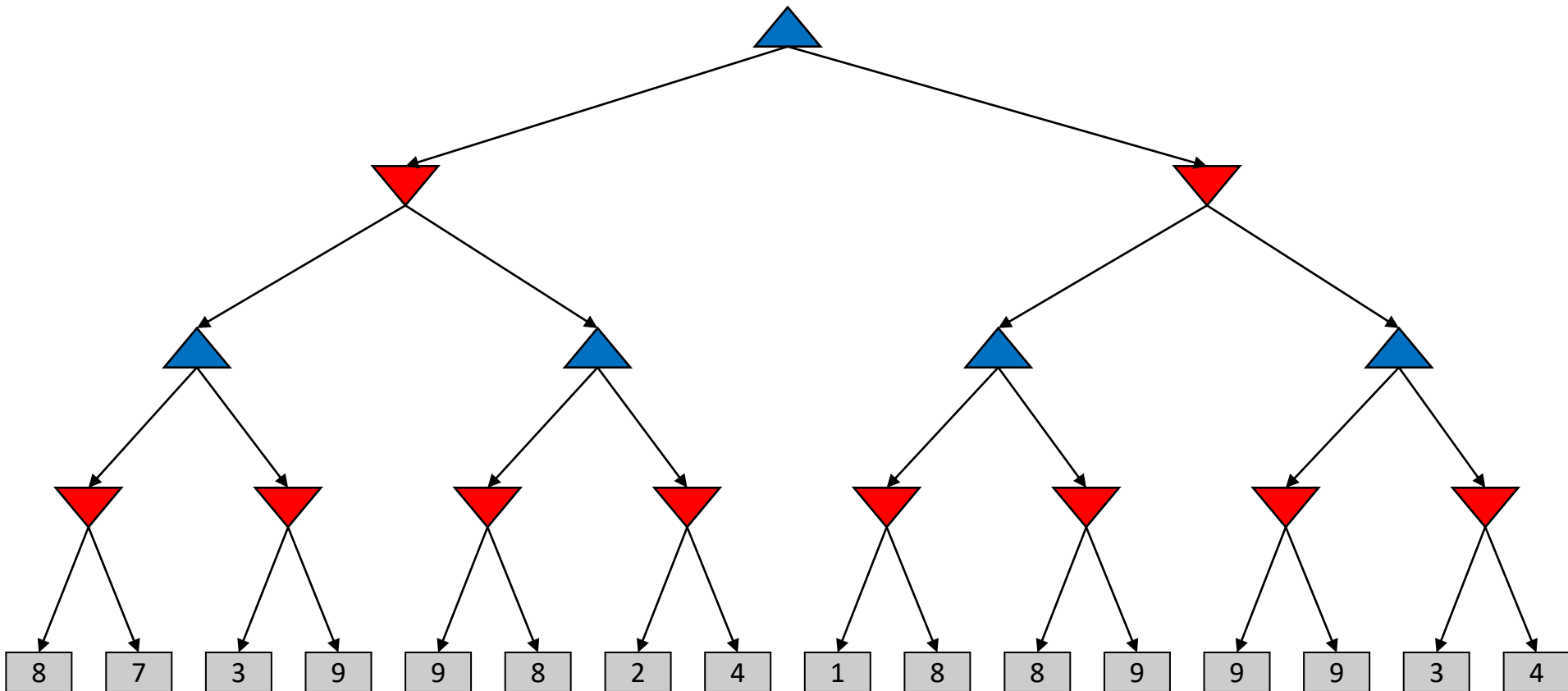
Tỉa alpha-beta: ví dụ 1



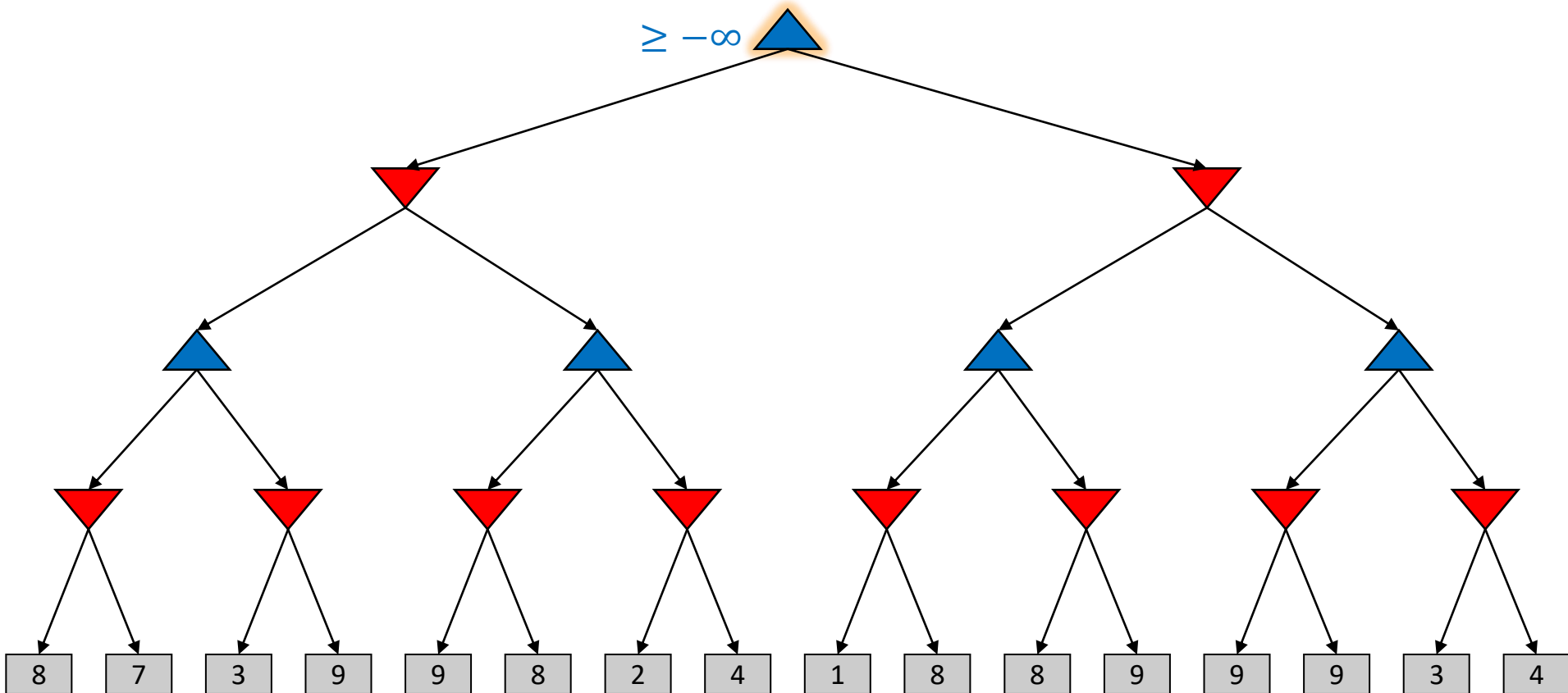
Tìm alpha-beta: ví dụ 1



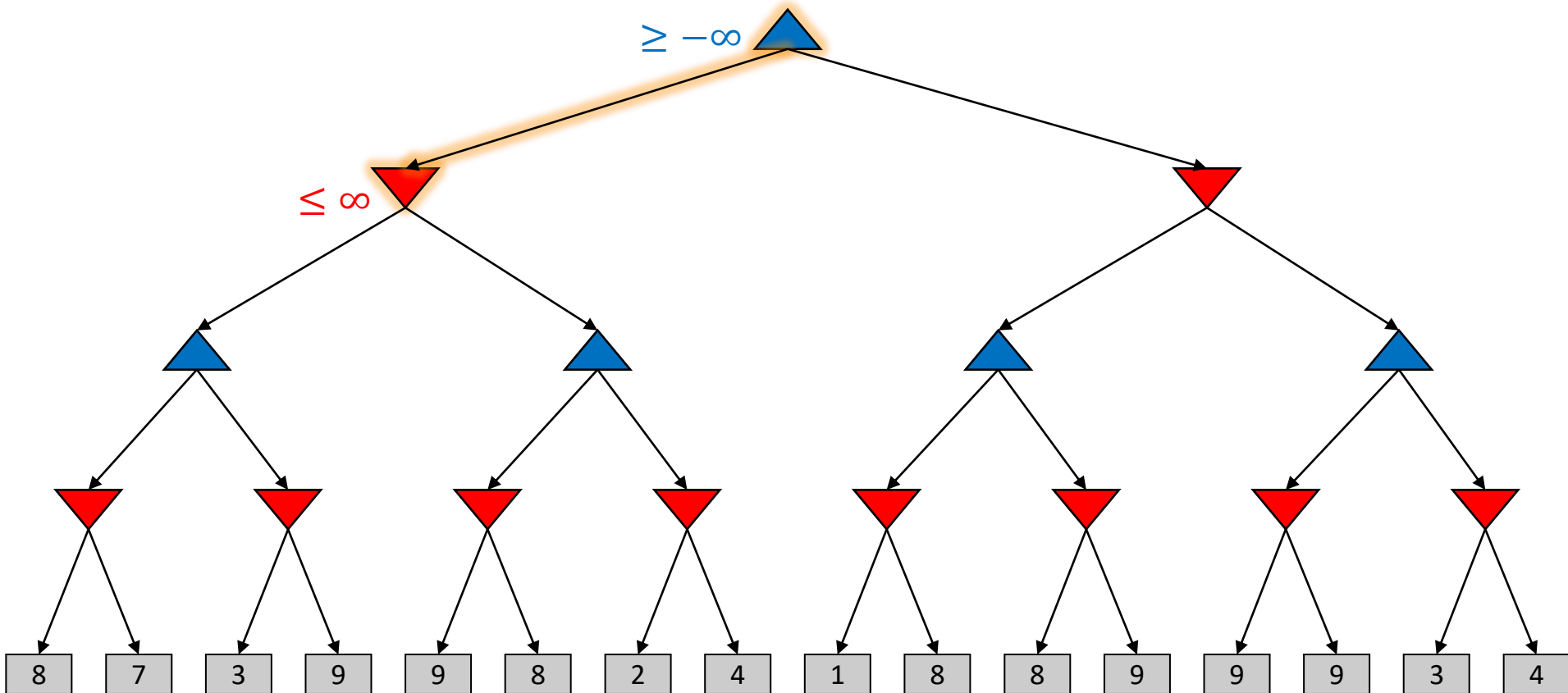
Tỉa alpha-beta: ví dụ 2



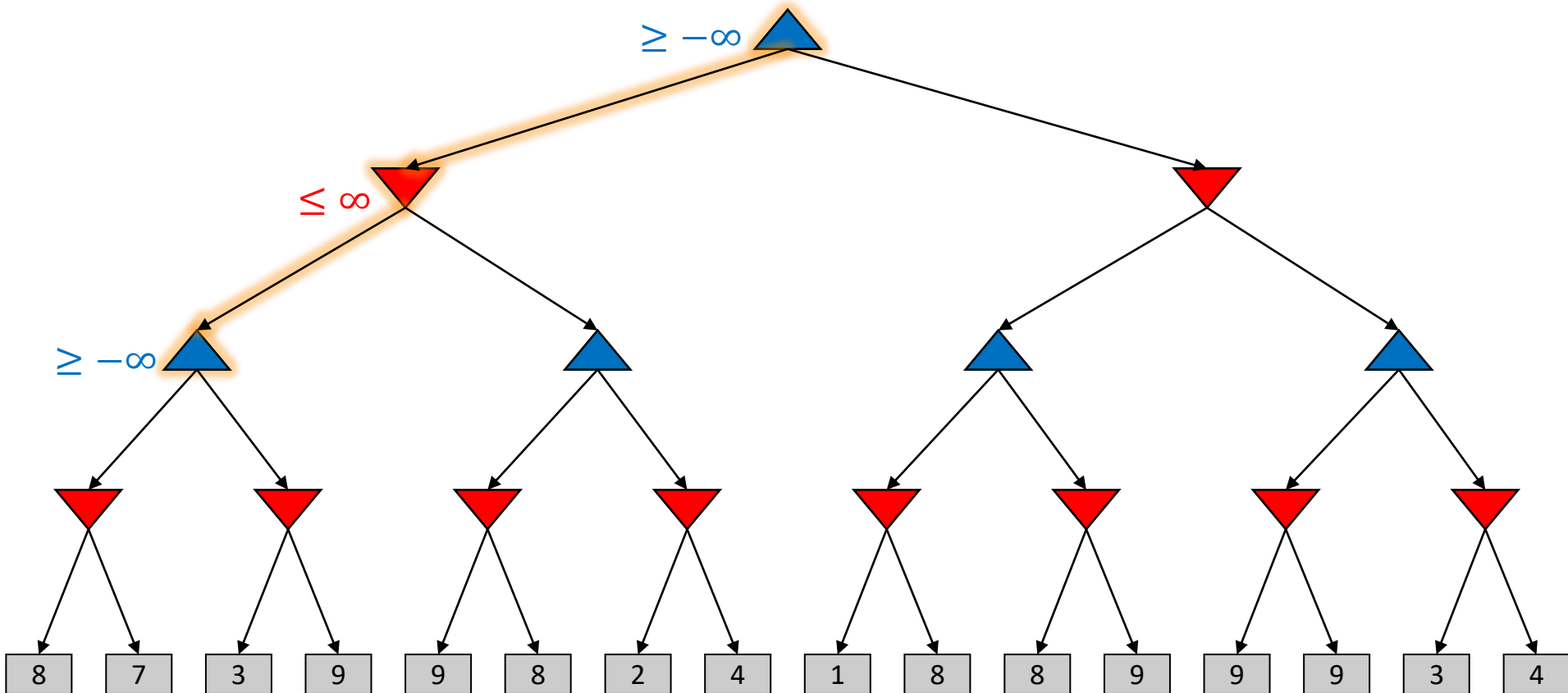
Tỉa alpha-beta: ví dụ 2



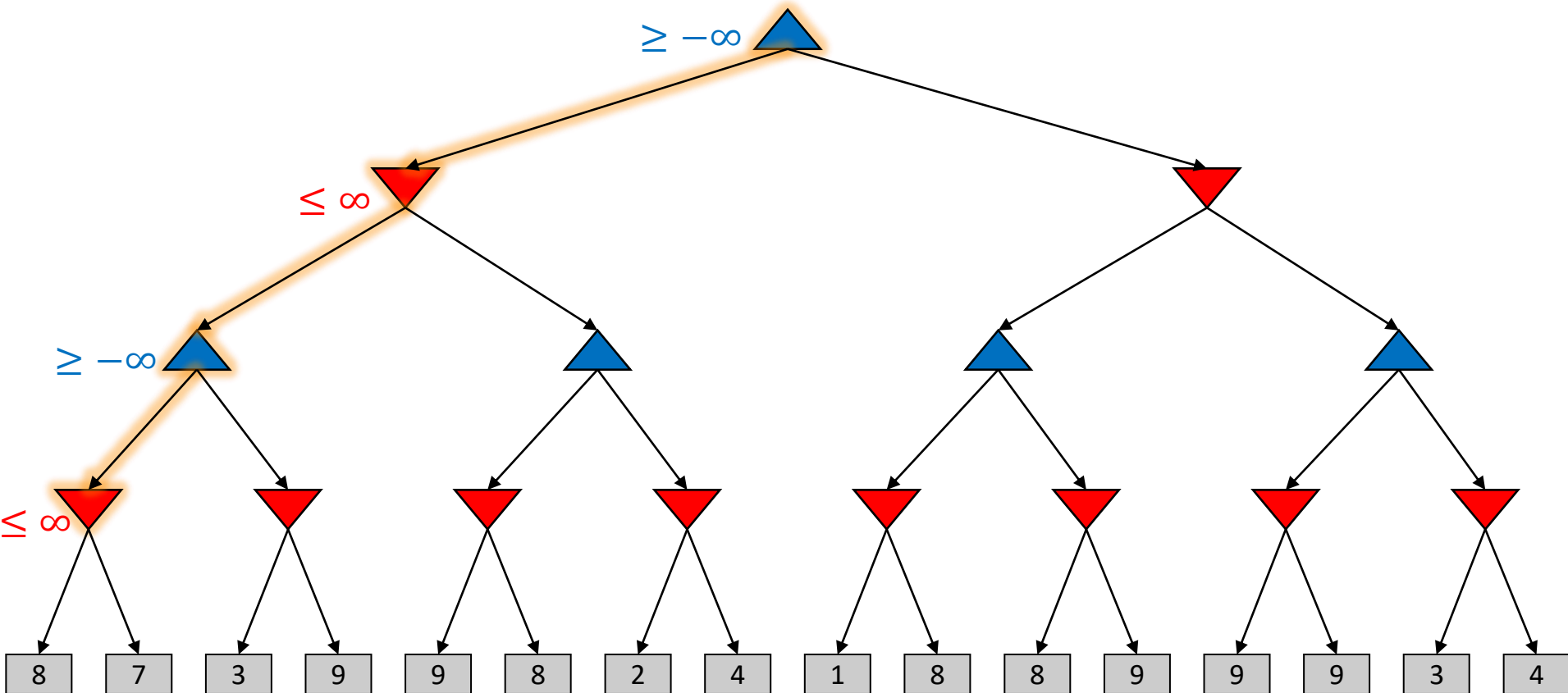
Tỉa alpha-beta: ví dụ 2



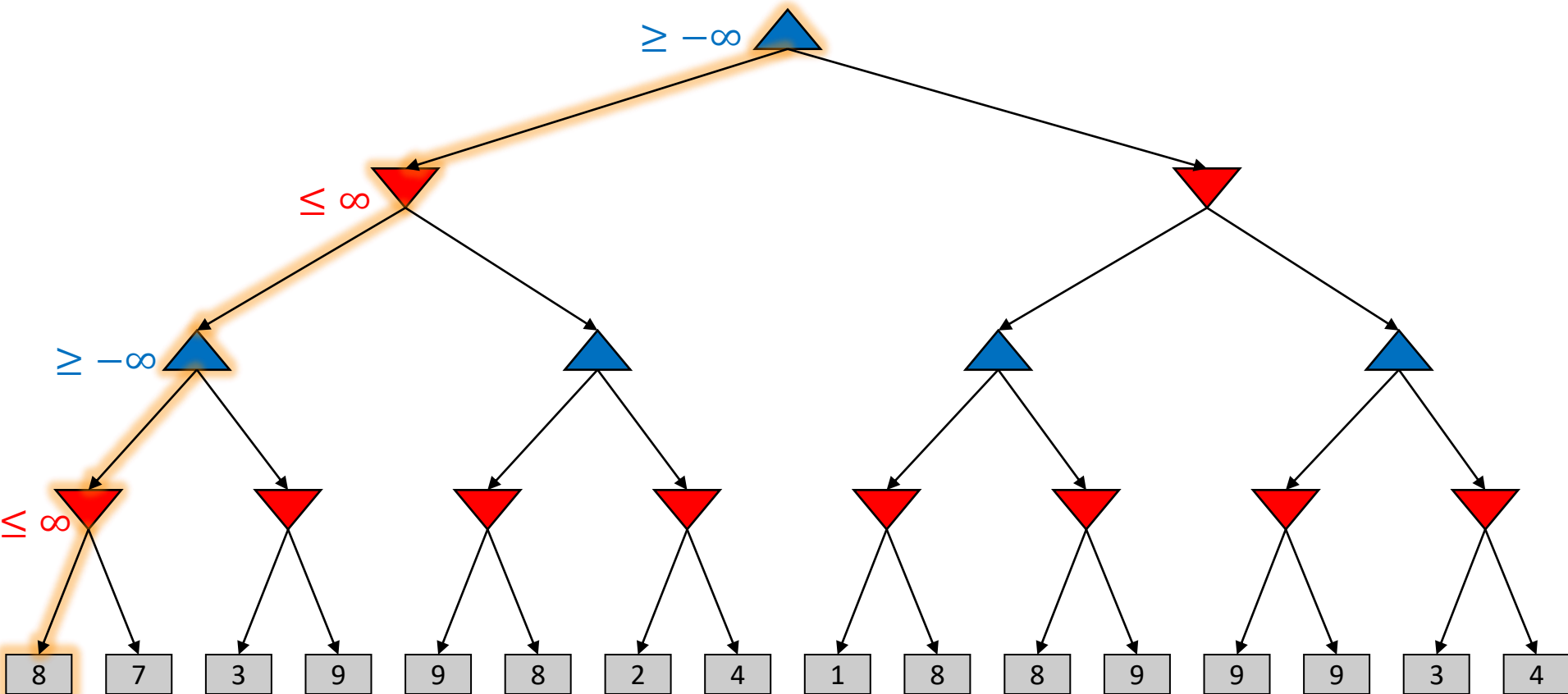
Tìm alpha-beta: ví dụ 2



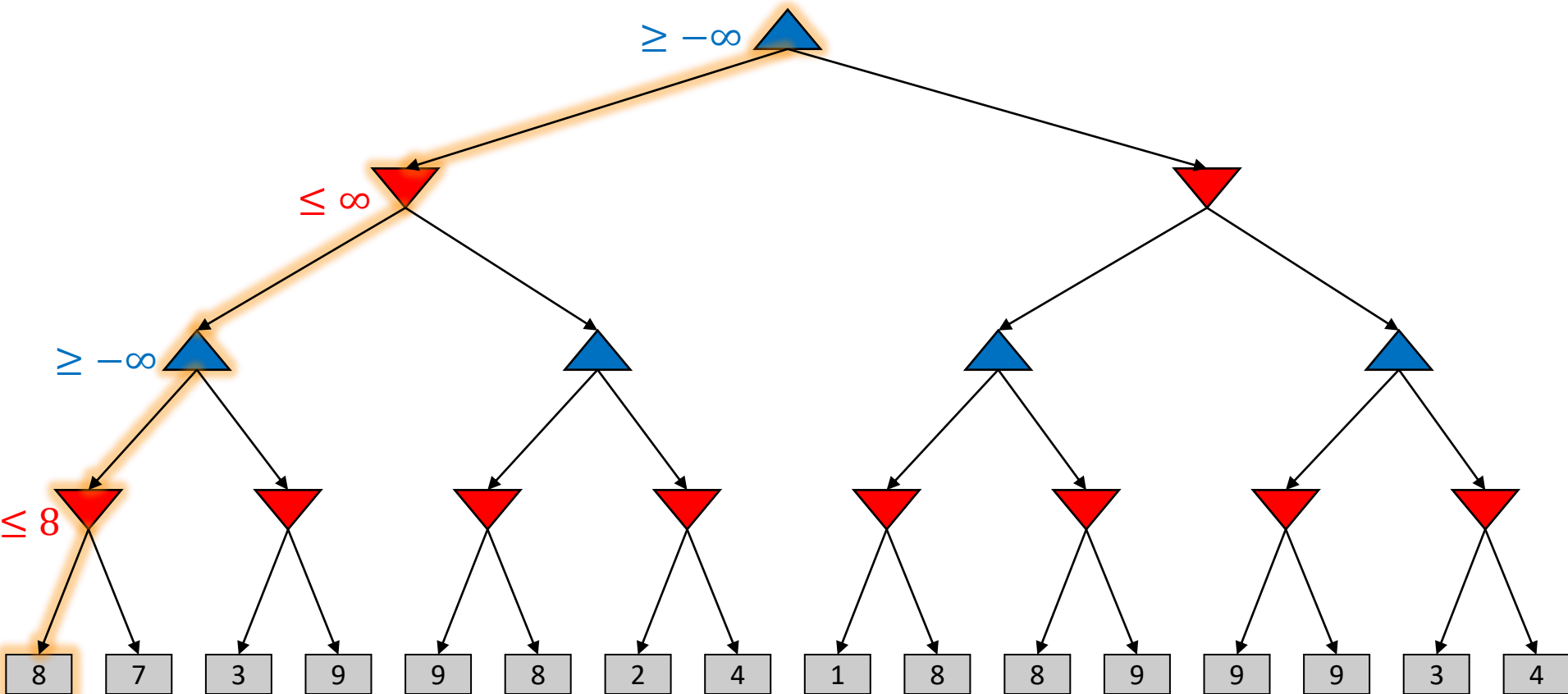
Tìm alpha-beta: ví dụ 2



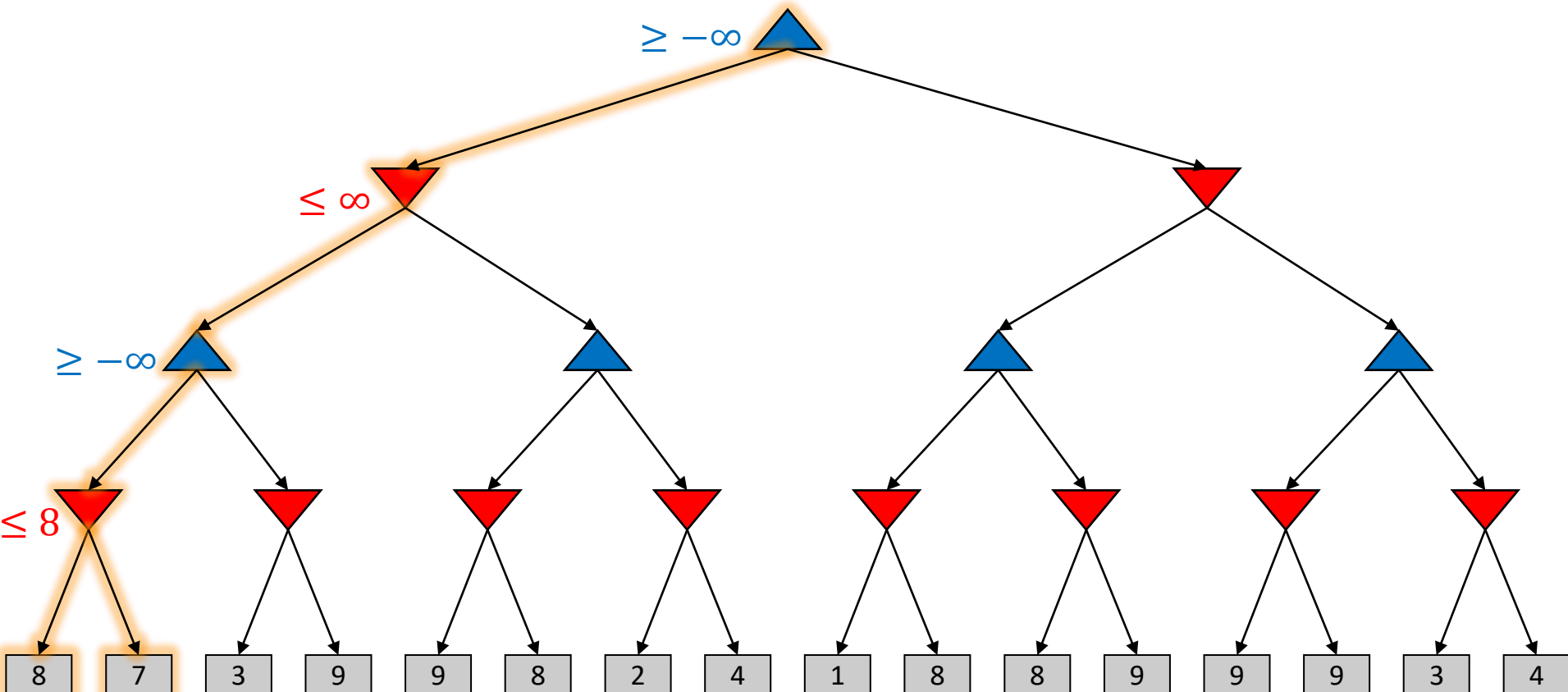
Tìm alpha-beta: ví dụ 2



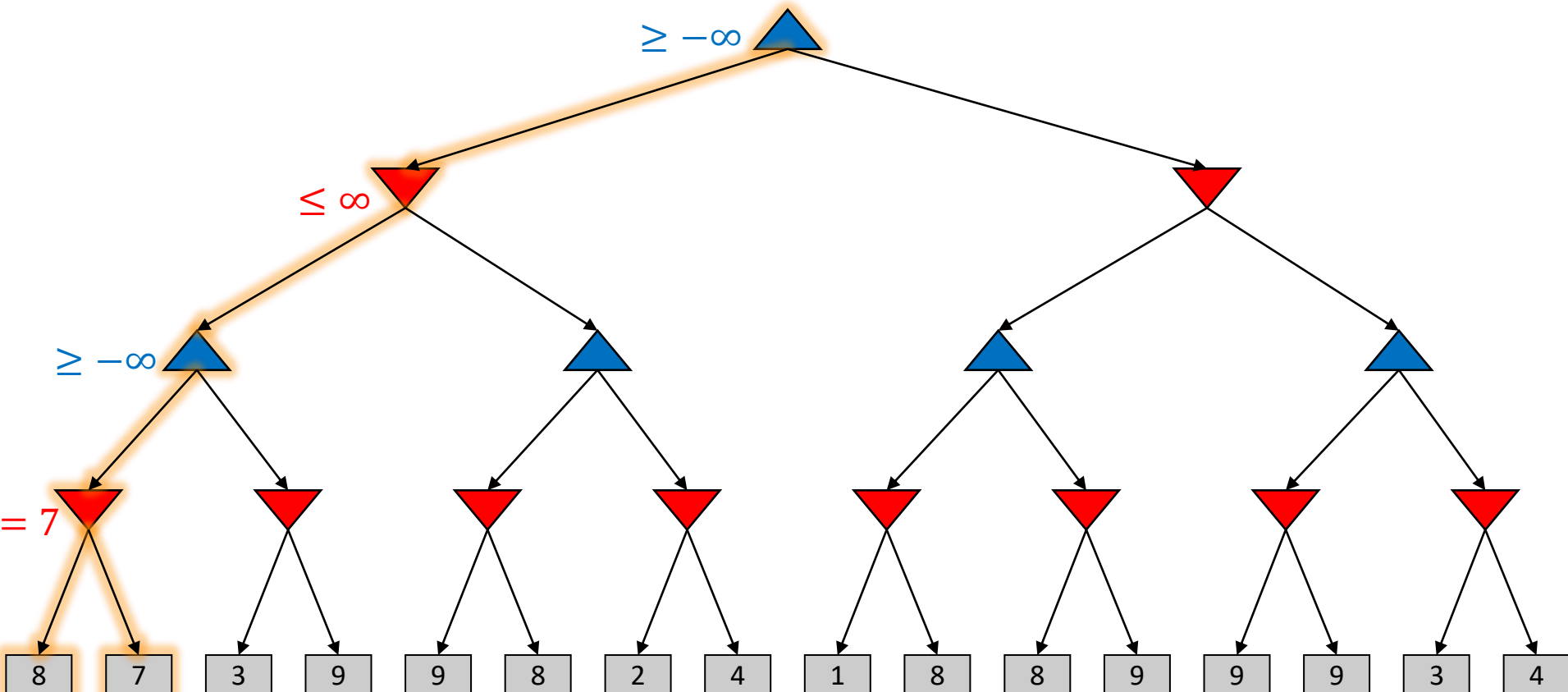
Tìm alpha-beta: ví dụ 2



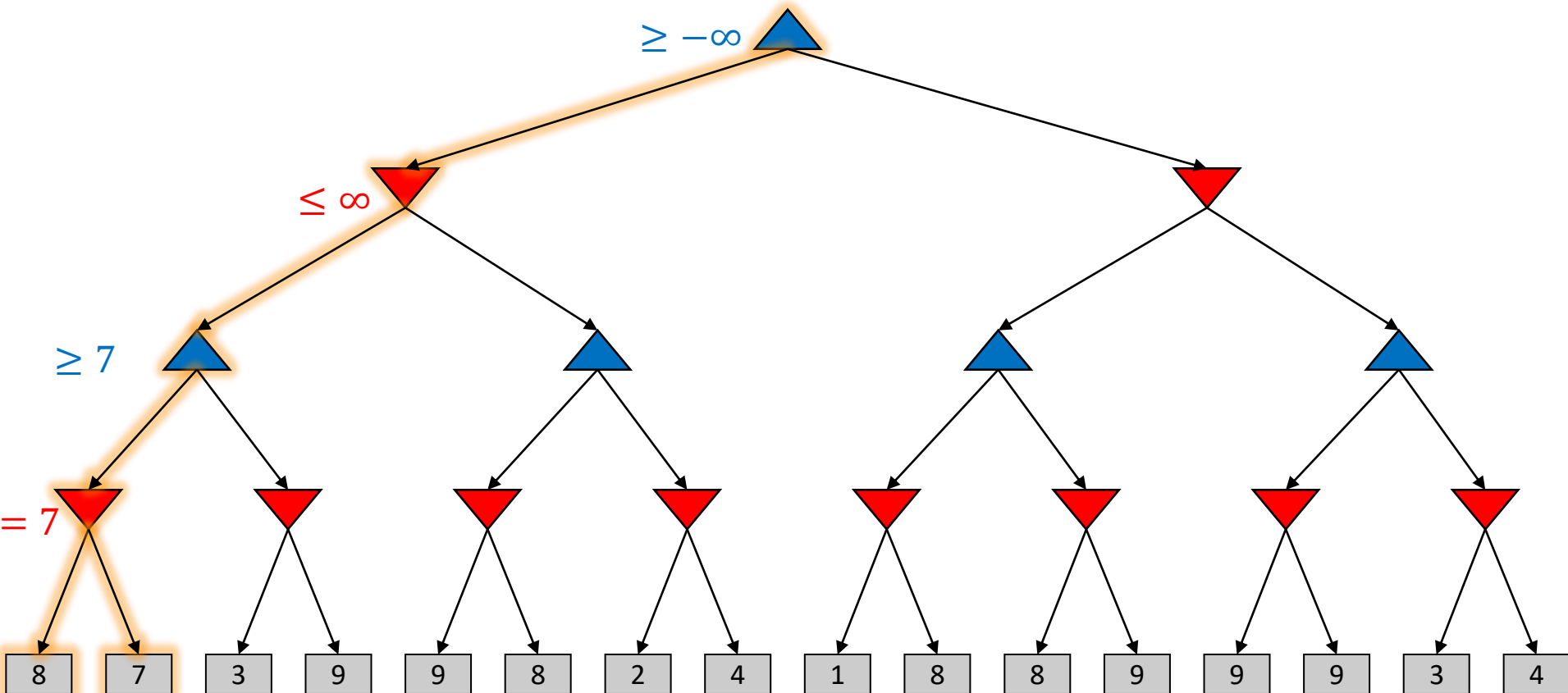
Tìm alpha-beta: ví dụ 2



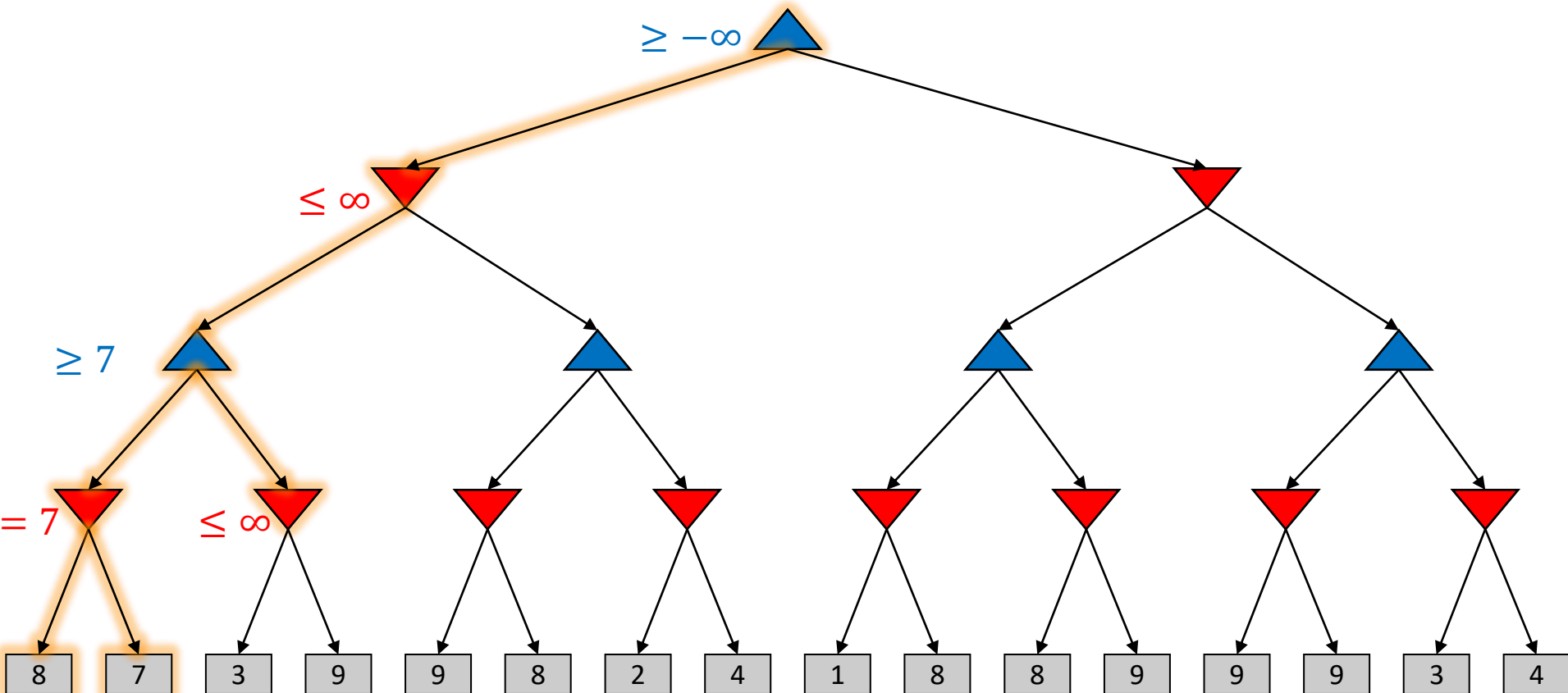
Tỉa alpha-beta: ví dụ 2



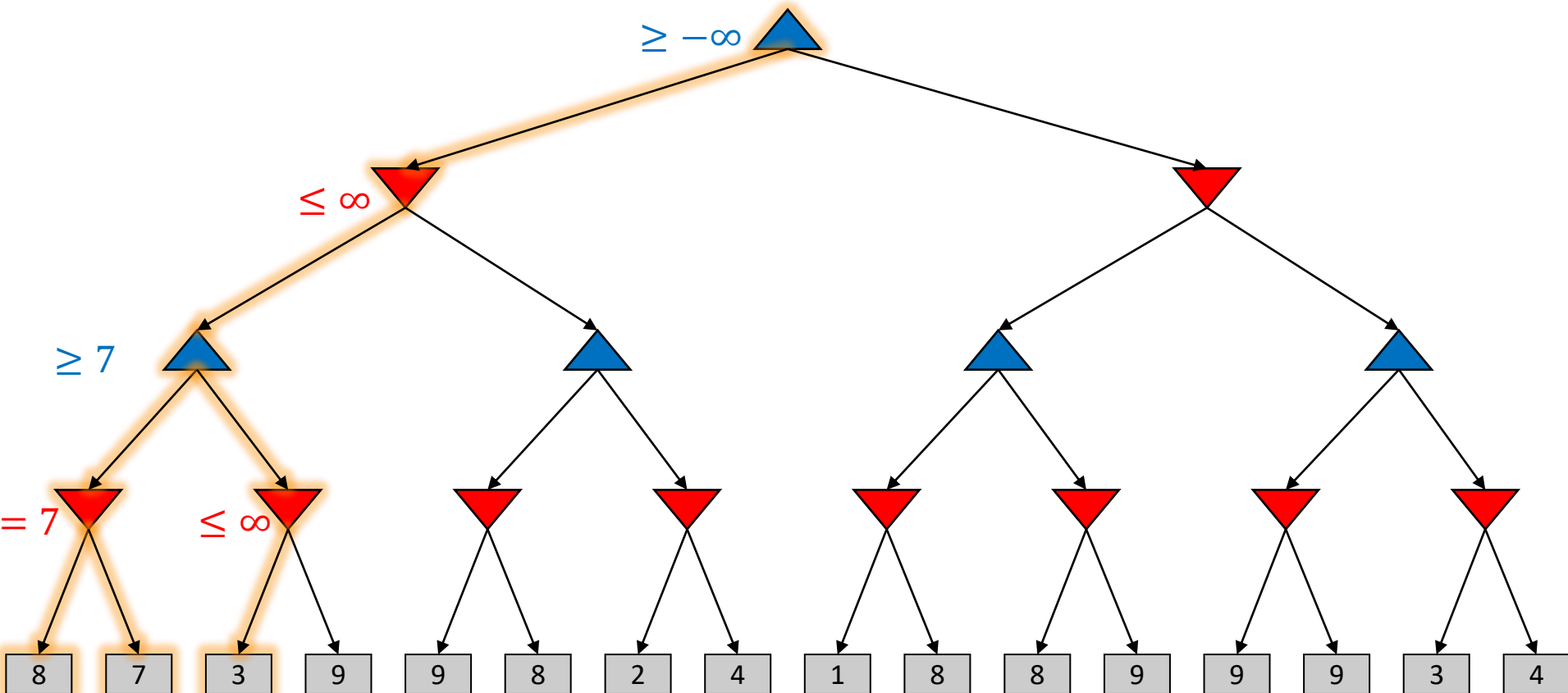
Tỉa alpha-beta: ví dụ 2



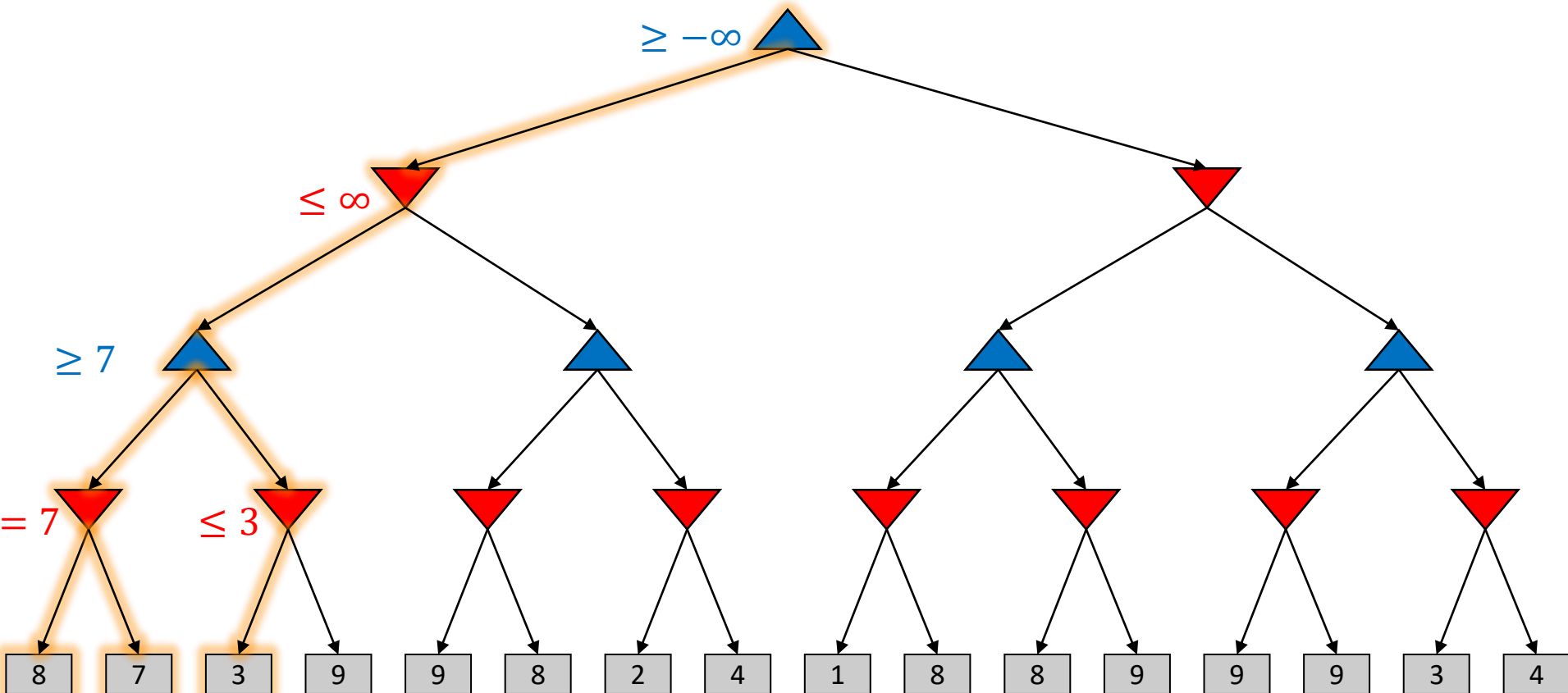
Tỉa alpha-beta: ví dụ 2



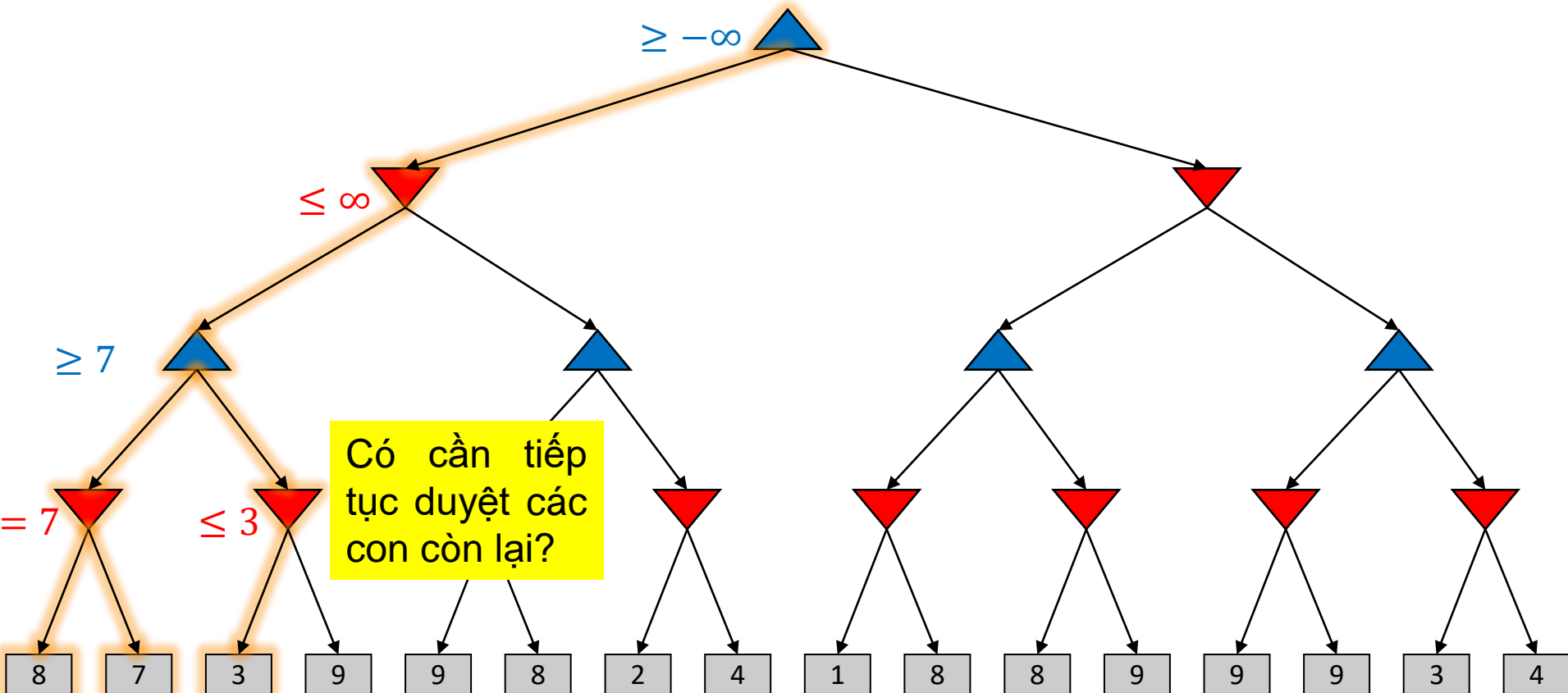
Tỉa alpha-beta: ví dụ 2



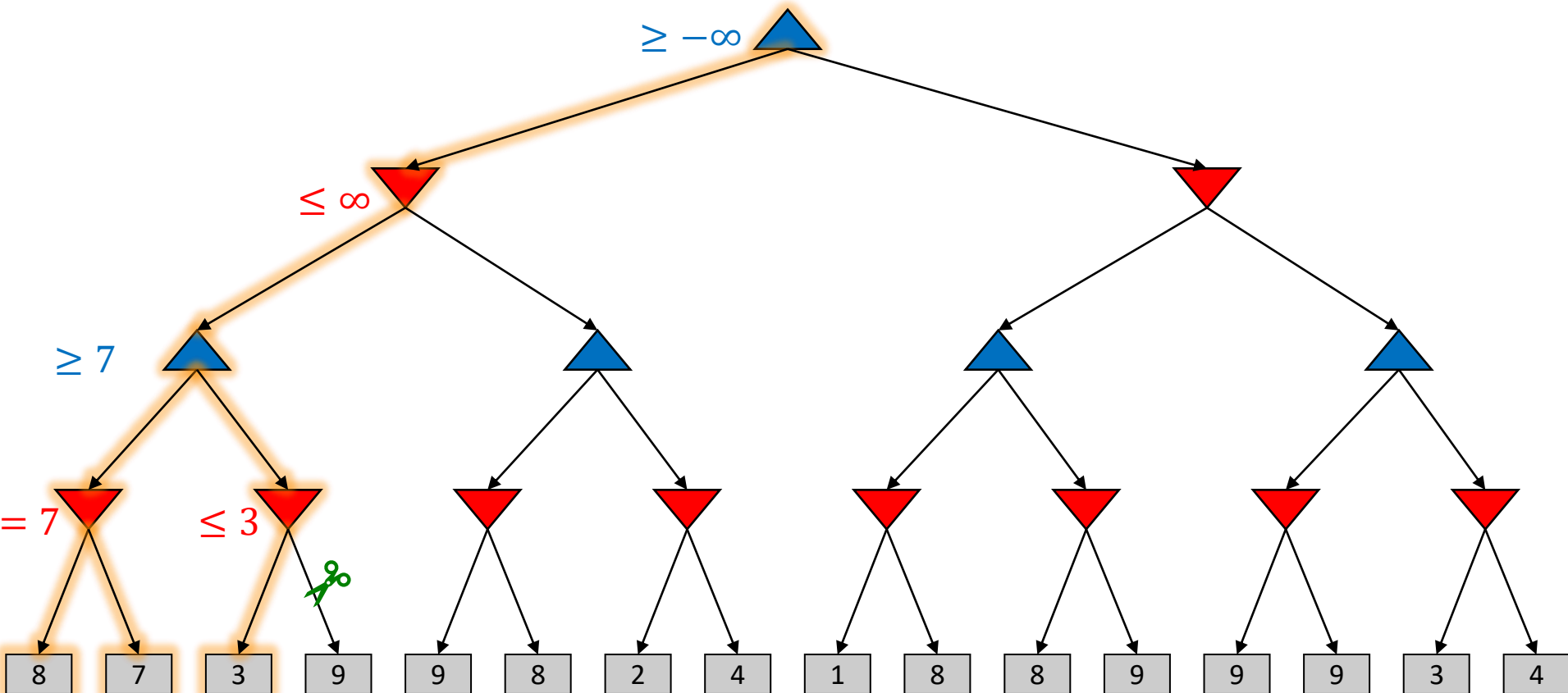
Tỉa alpha-beta: ví dụ 2



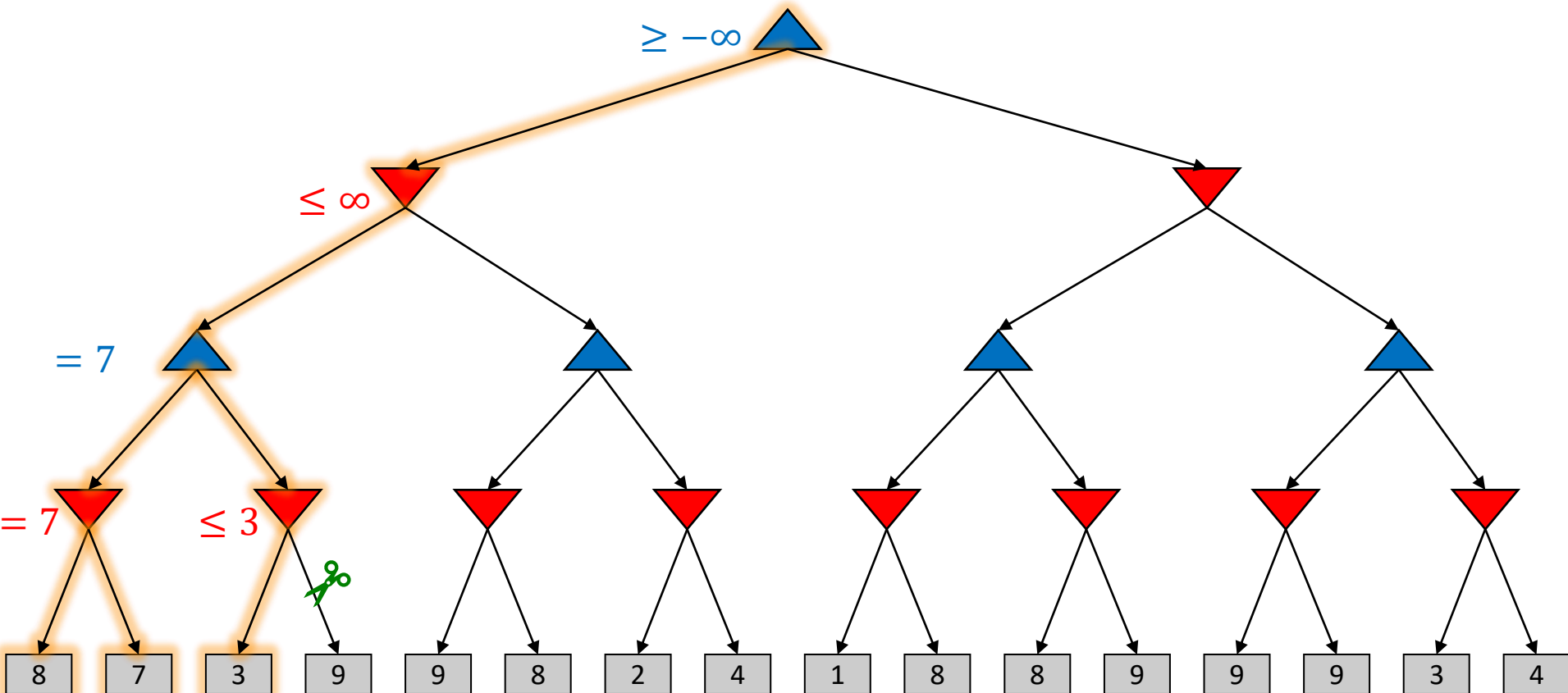
Tỉa alpha-beta: ví dụ 2



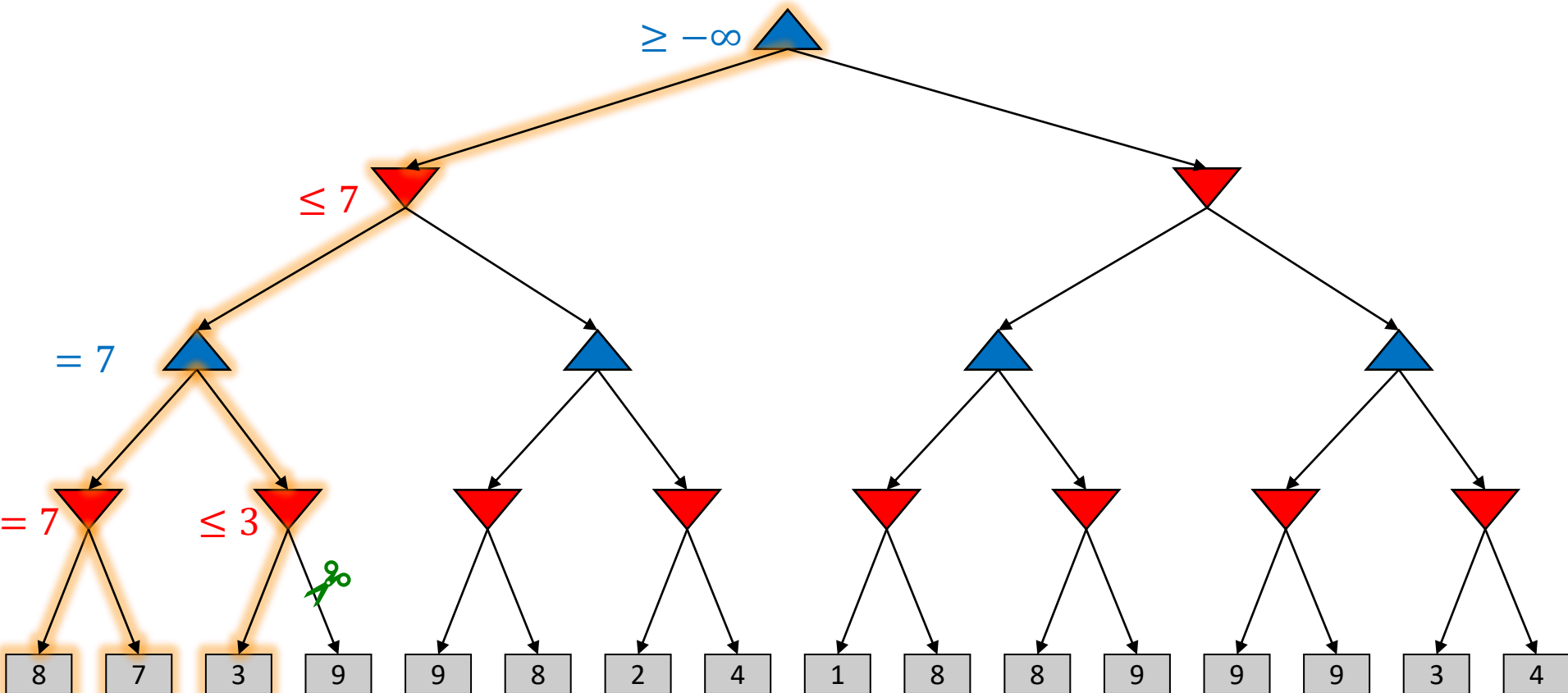
Tỉa alpha-beta: ví dụ 2



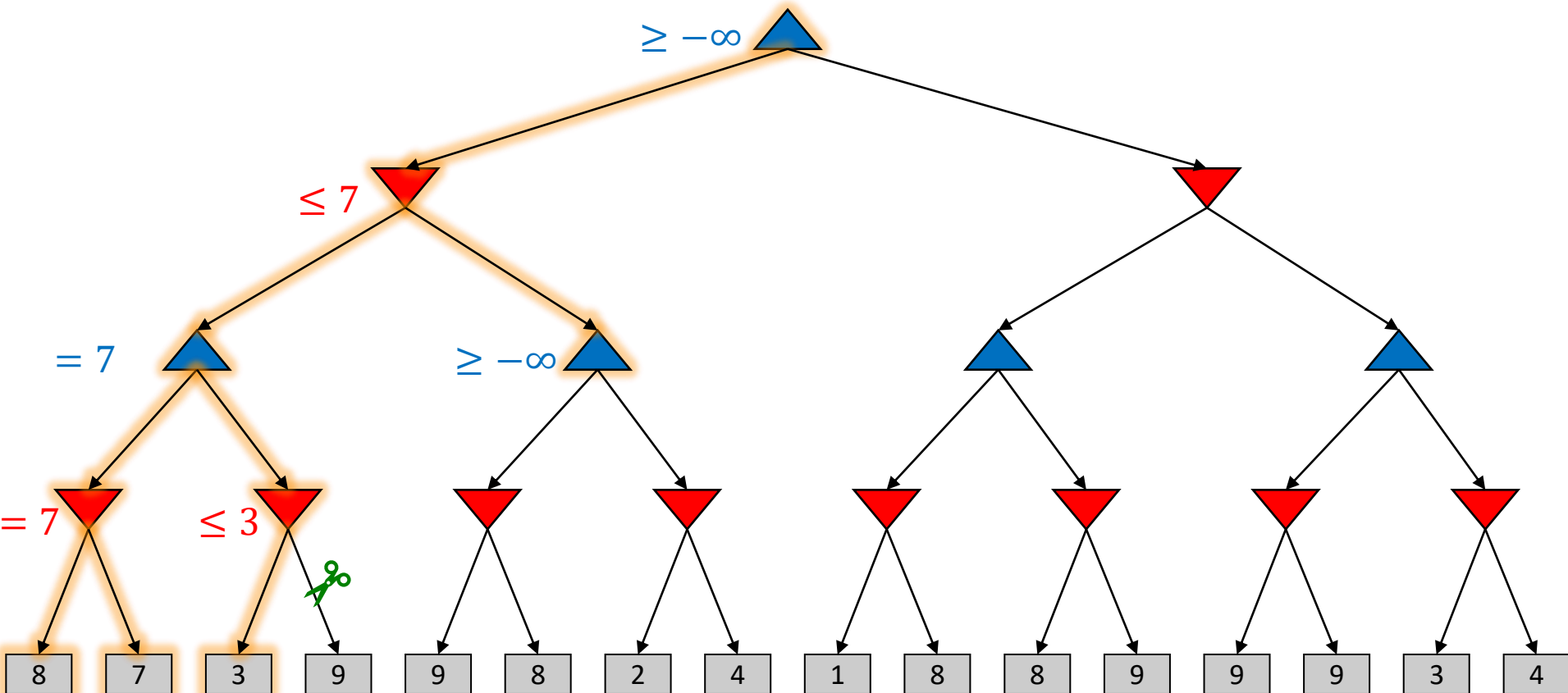
Tỉa alpha-beta: ví dụ 2



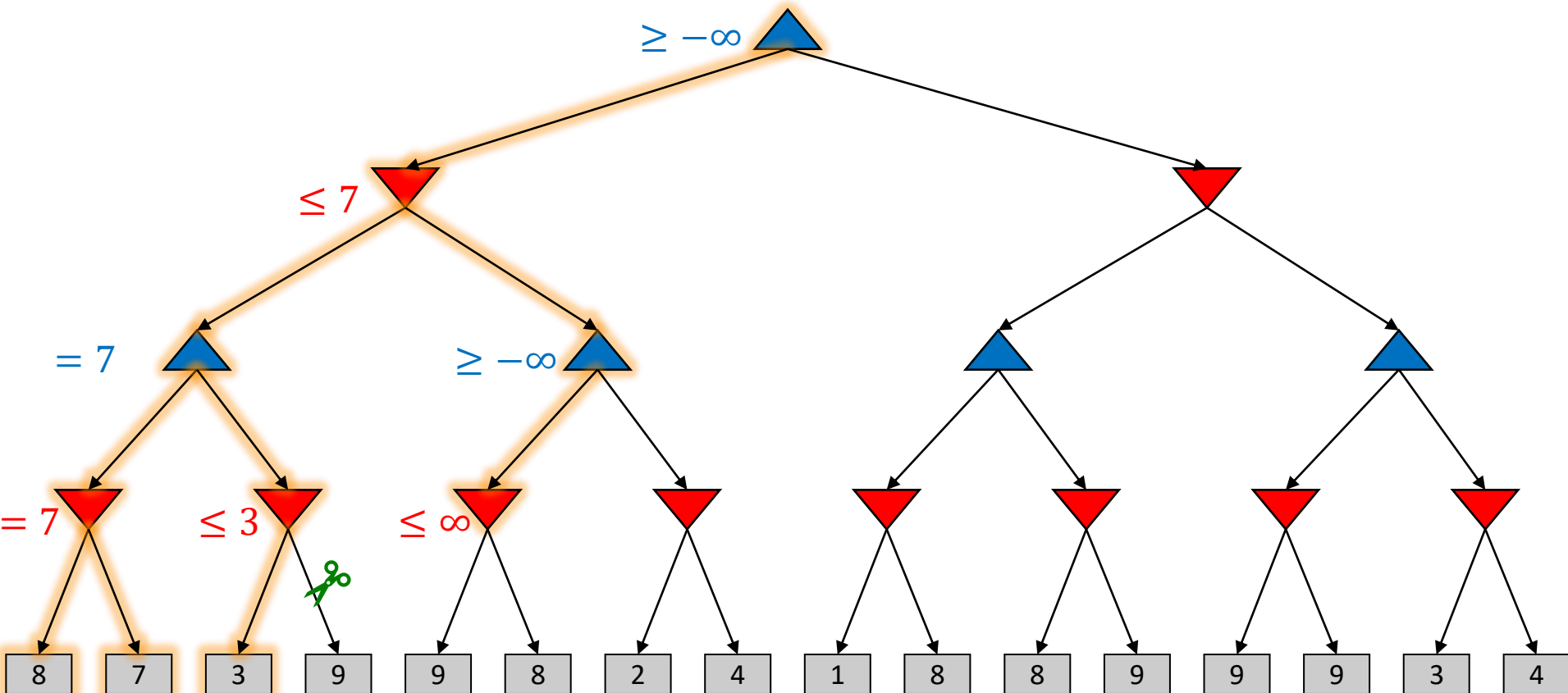
Tỉa alpha-beta: ví dụ 2



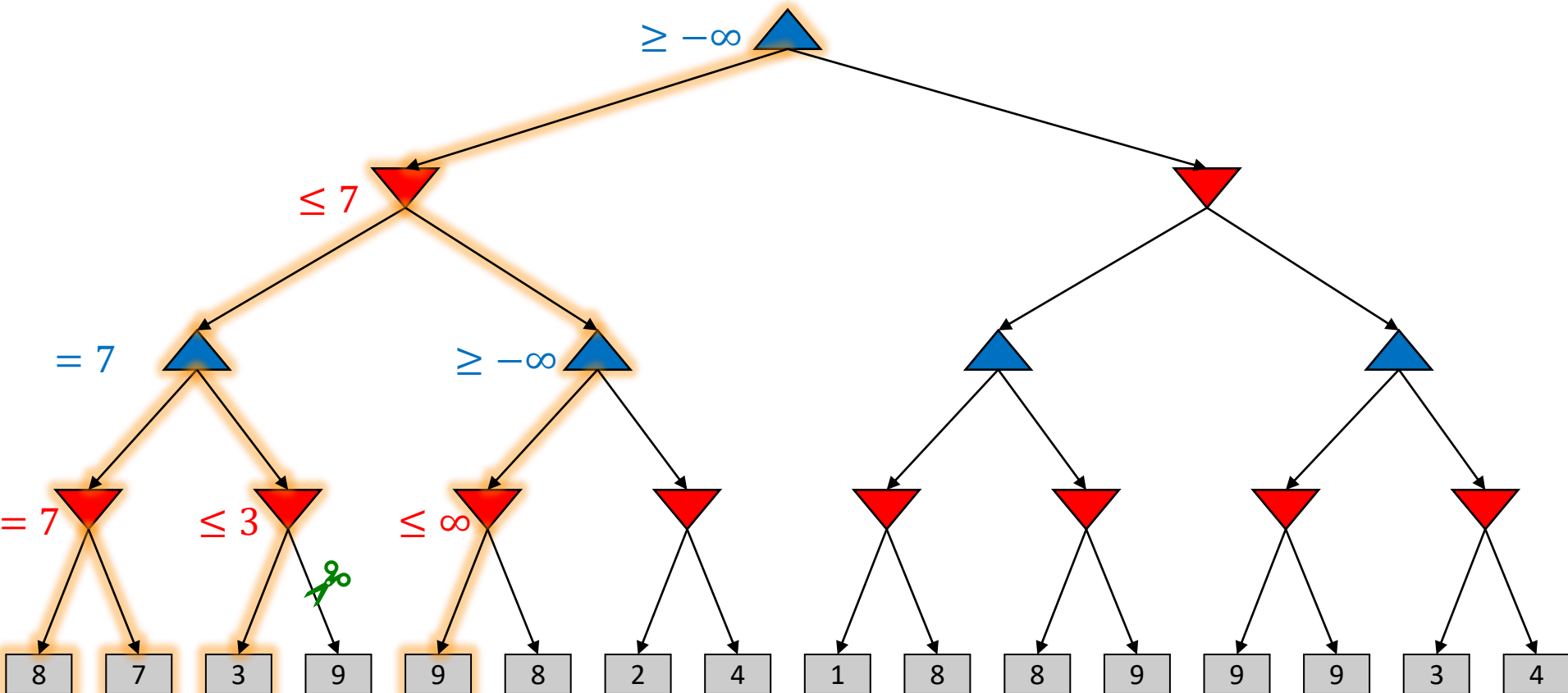
Tỉa alpha-beta: ví dụ 2



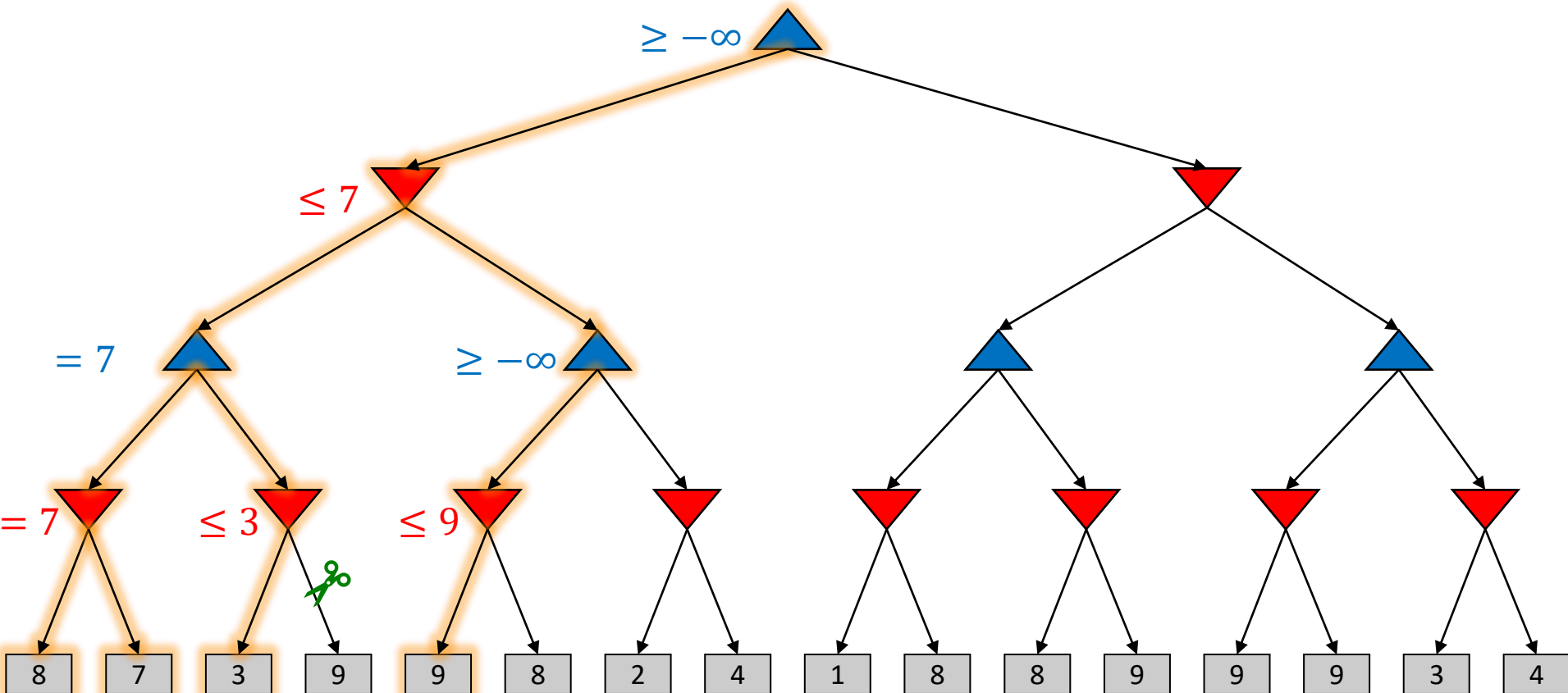
Tỉa alpha-beta: ví dụ 2



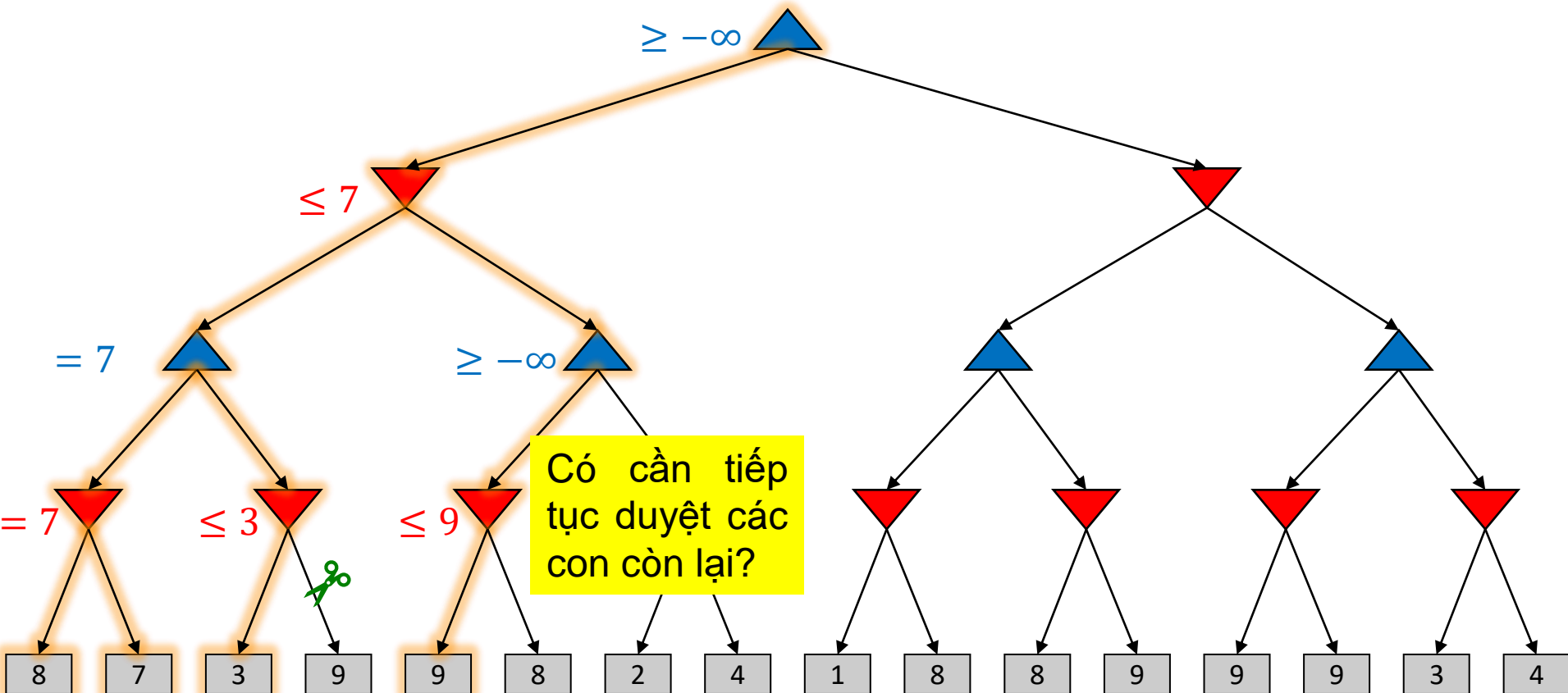
Tỉa alpha-beta: ví dụ 2



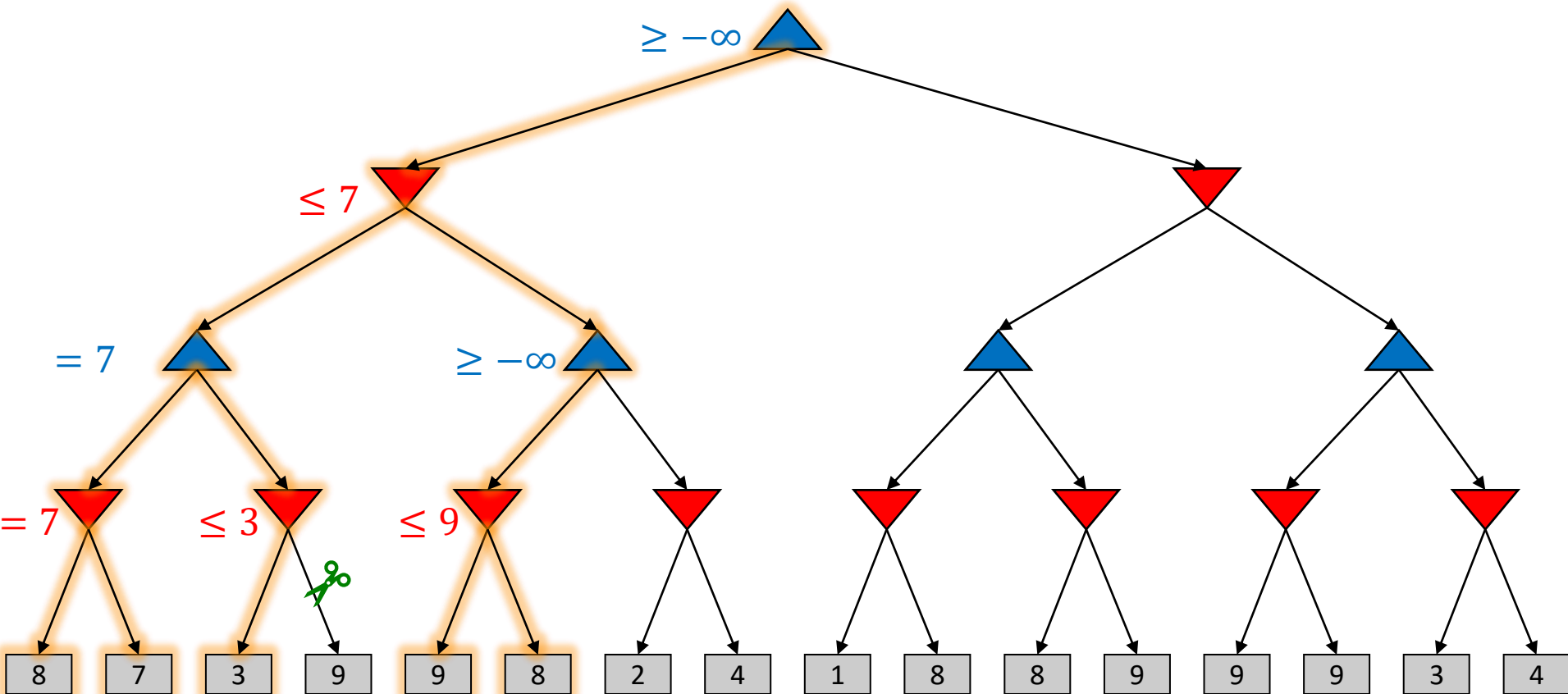
Tỉa alpha-beta: ví dụ 2



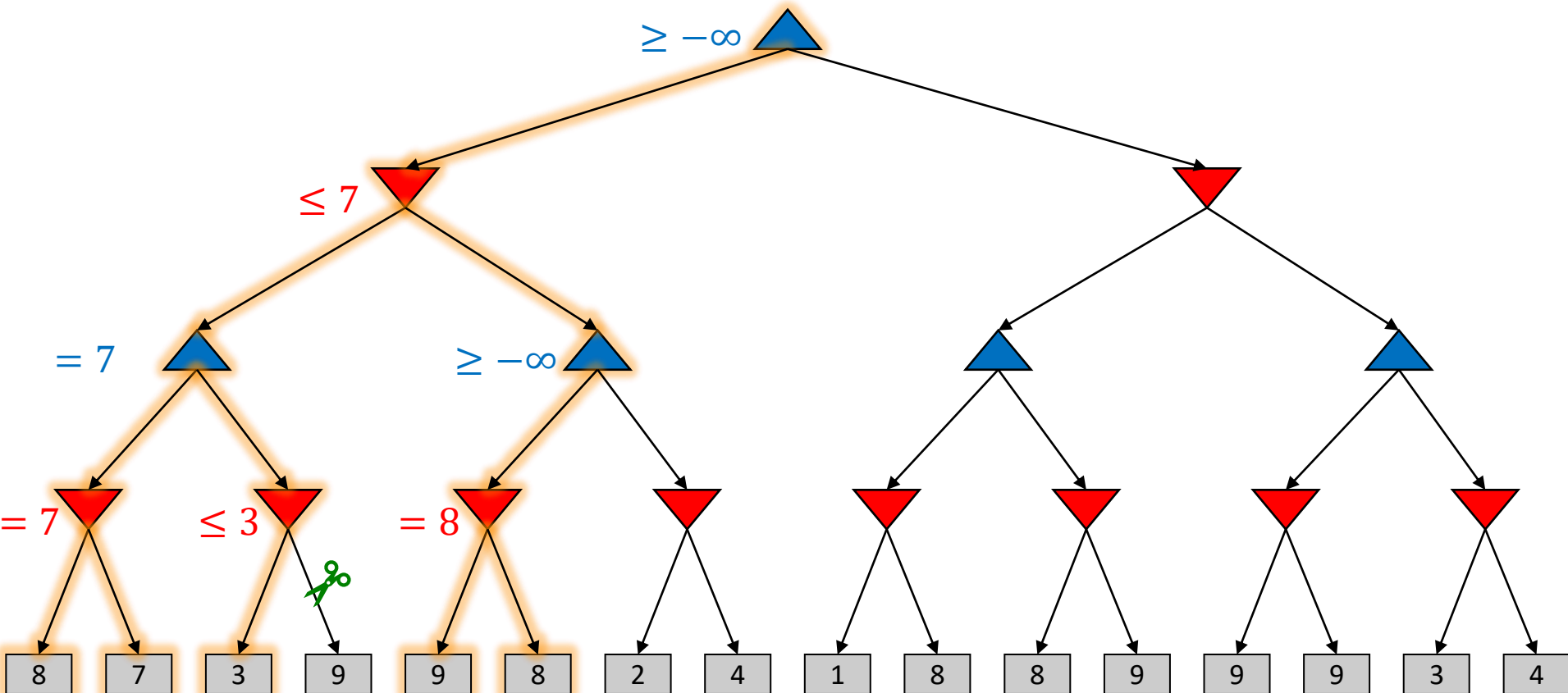
Tỉa alpha-beta: ví dụ 2



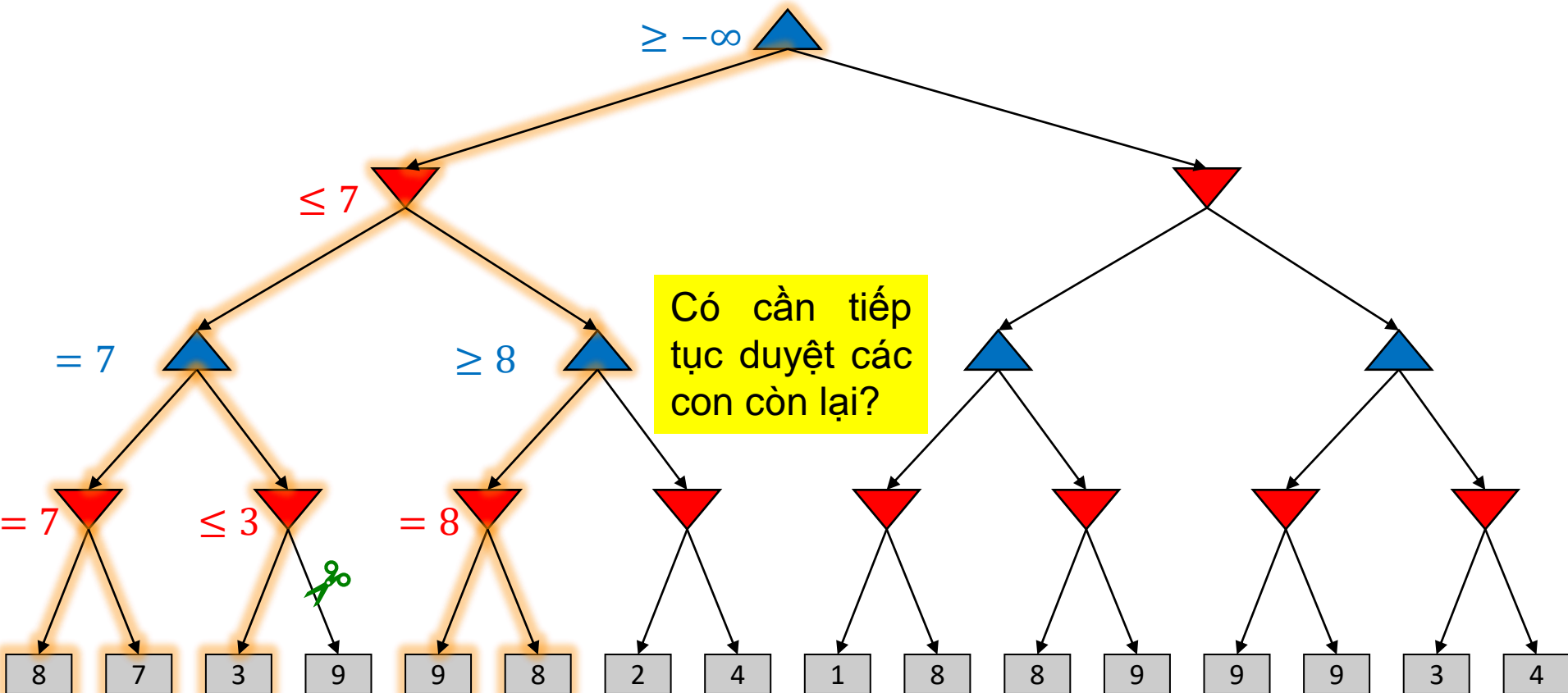
Tìm alpha-beta: ví dụ 2



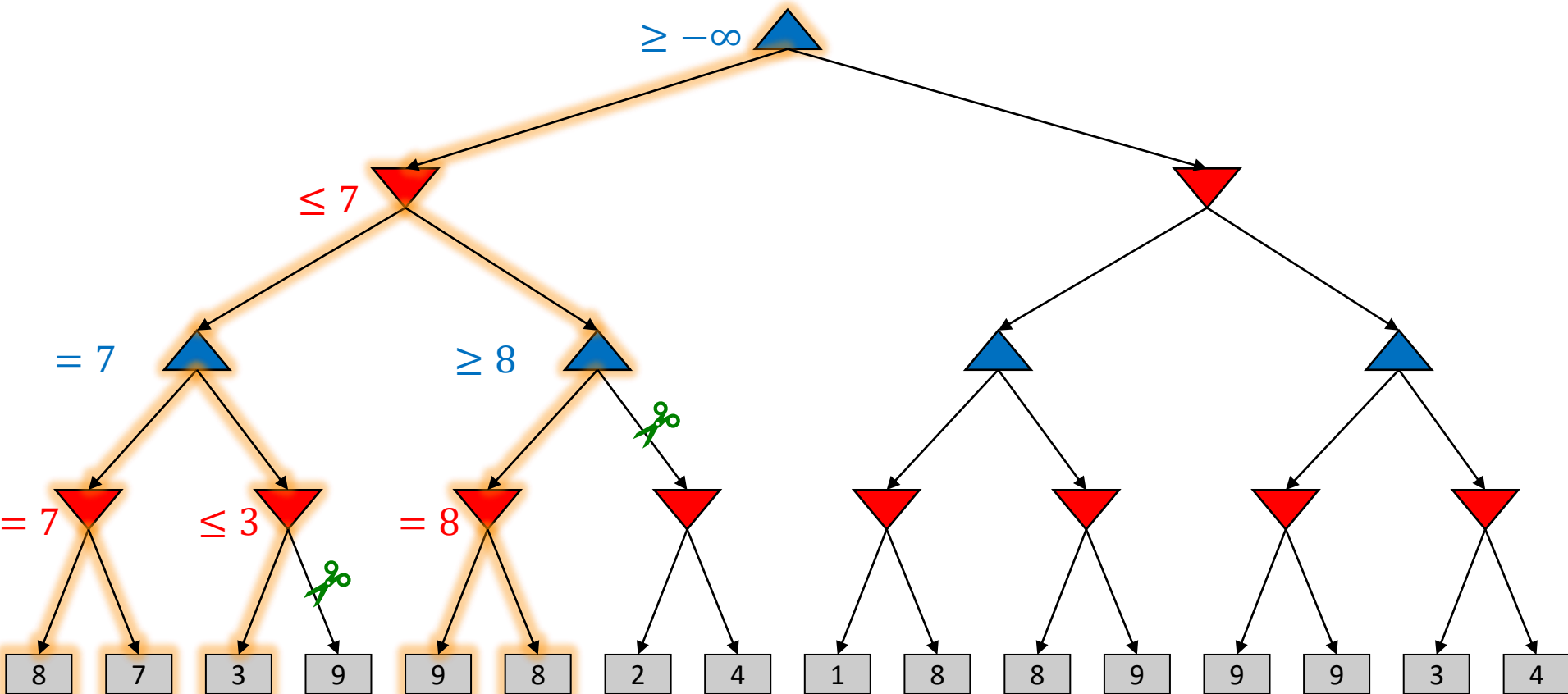
Tỉa alpha-beta: ví dụ 2



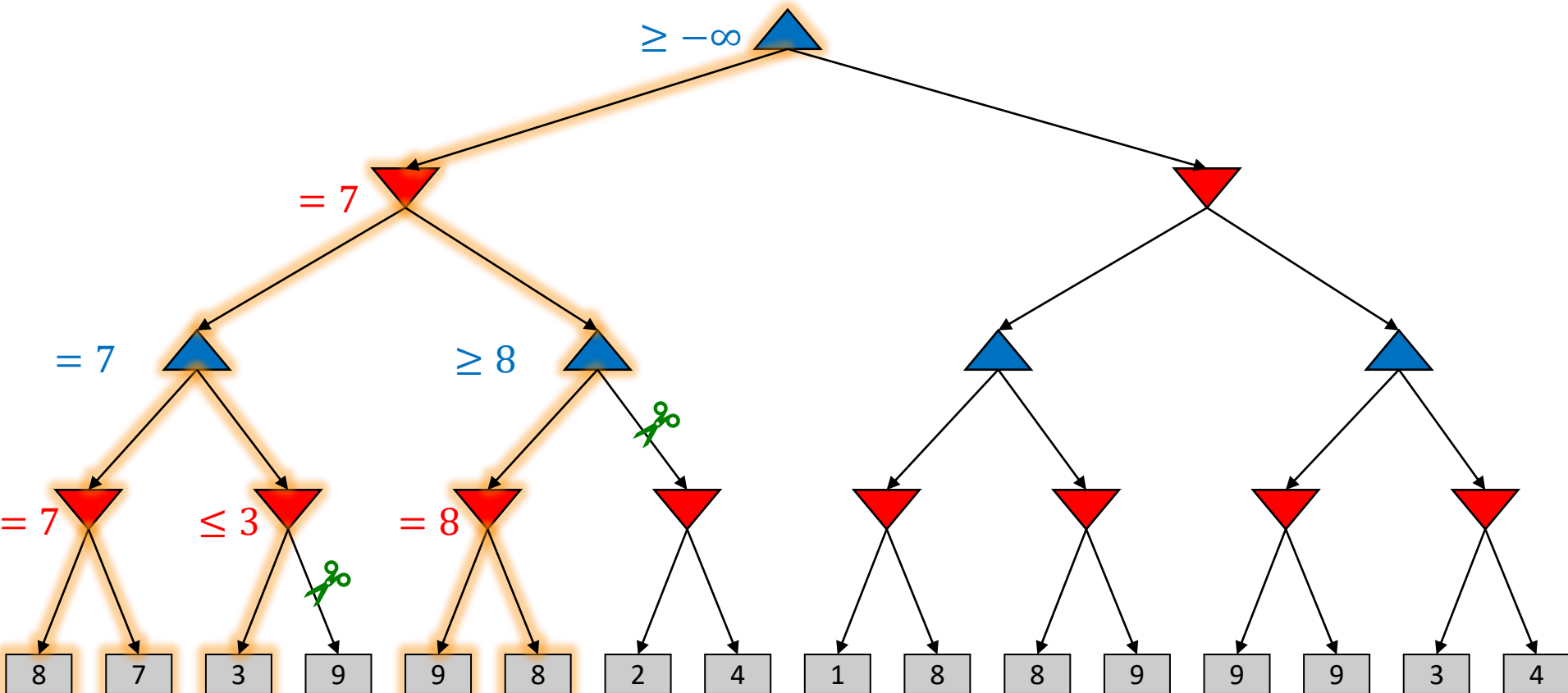
Tỉa alpha-beta: ví dụ 2



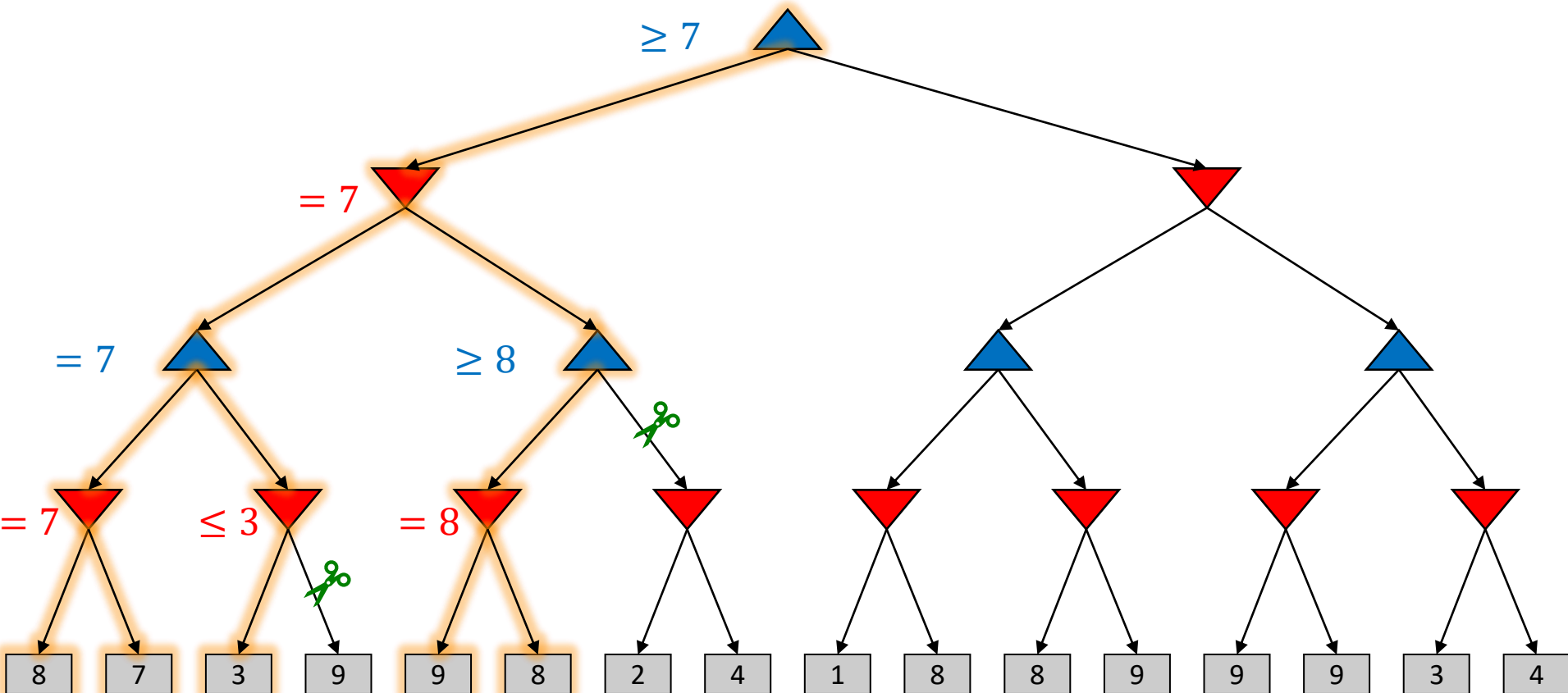
Tỉa alpha-beta: ví dụ 2



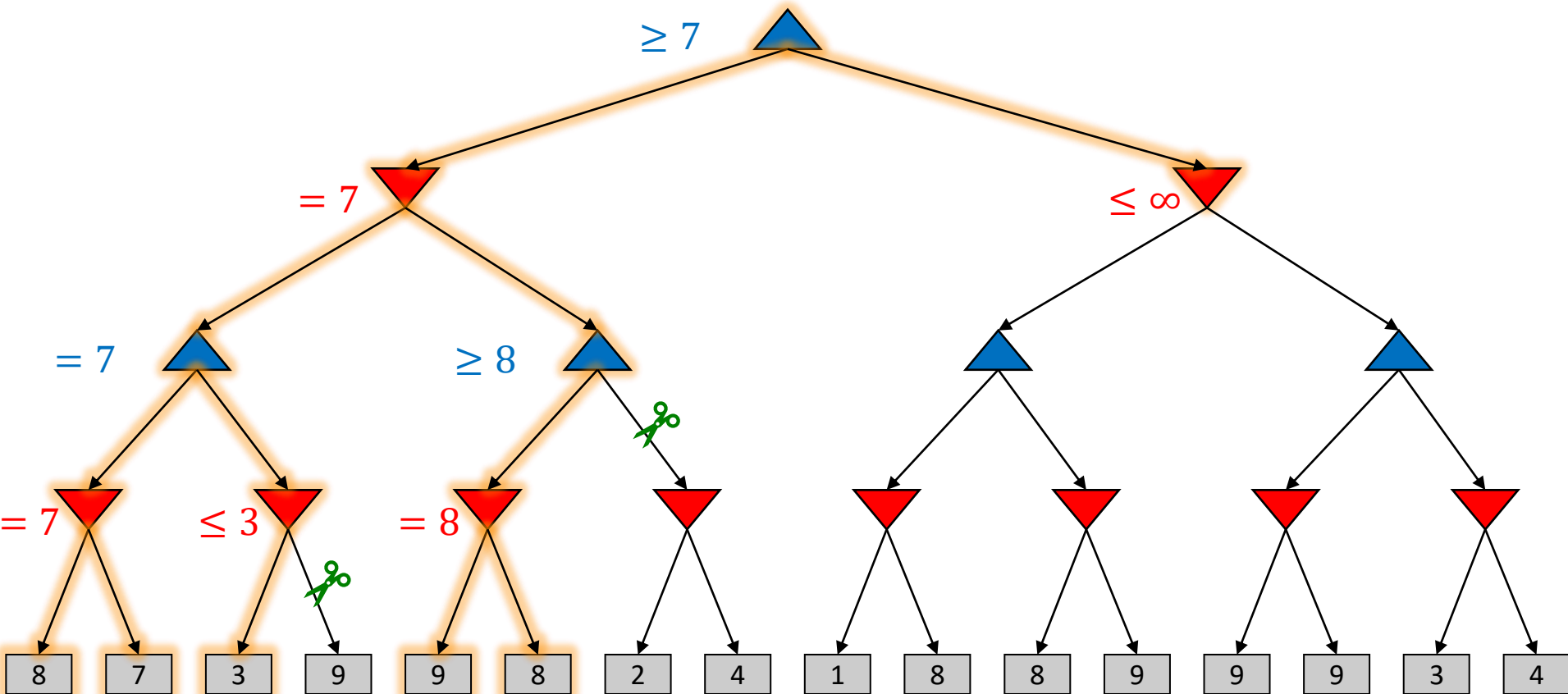
Tỉa alpha-beta: ví dụ 2



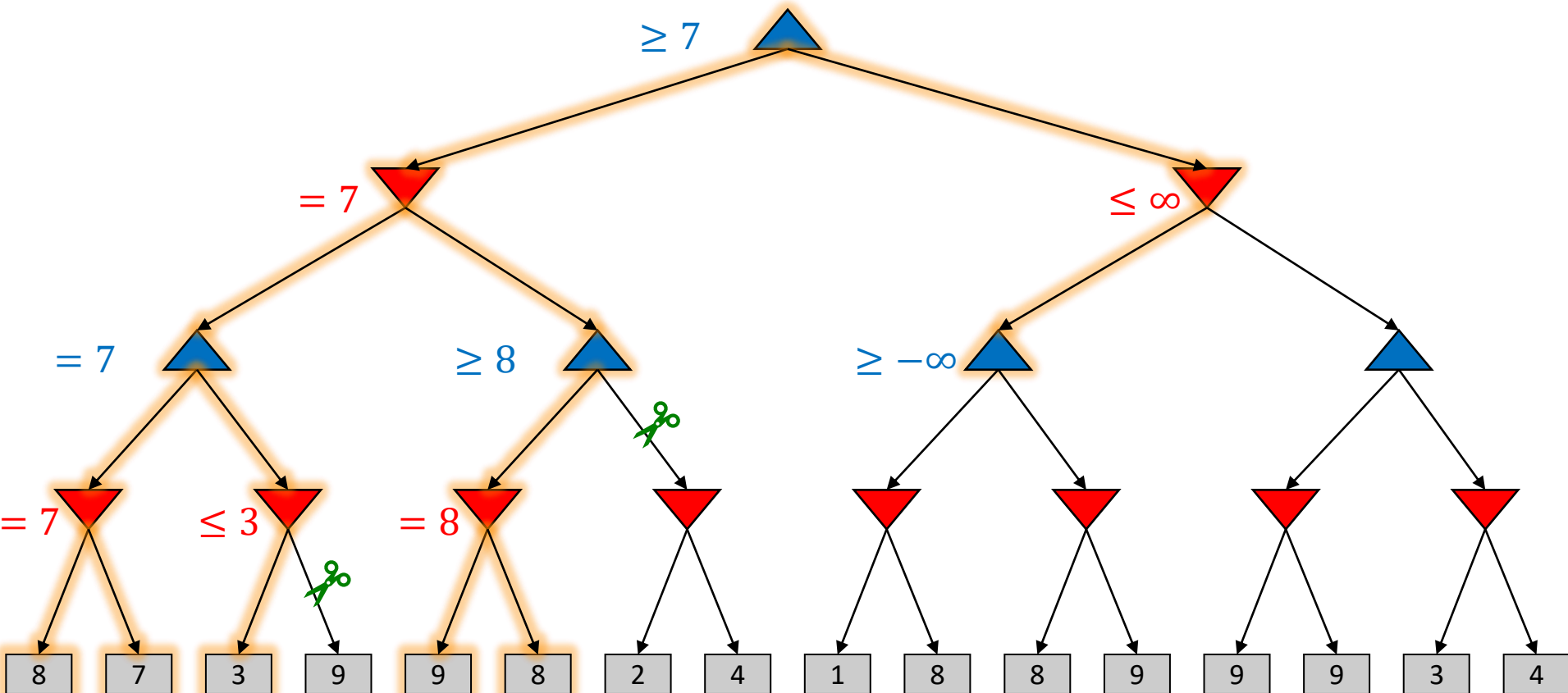
Tỉa alpha-beta: ví dụ 2



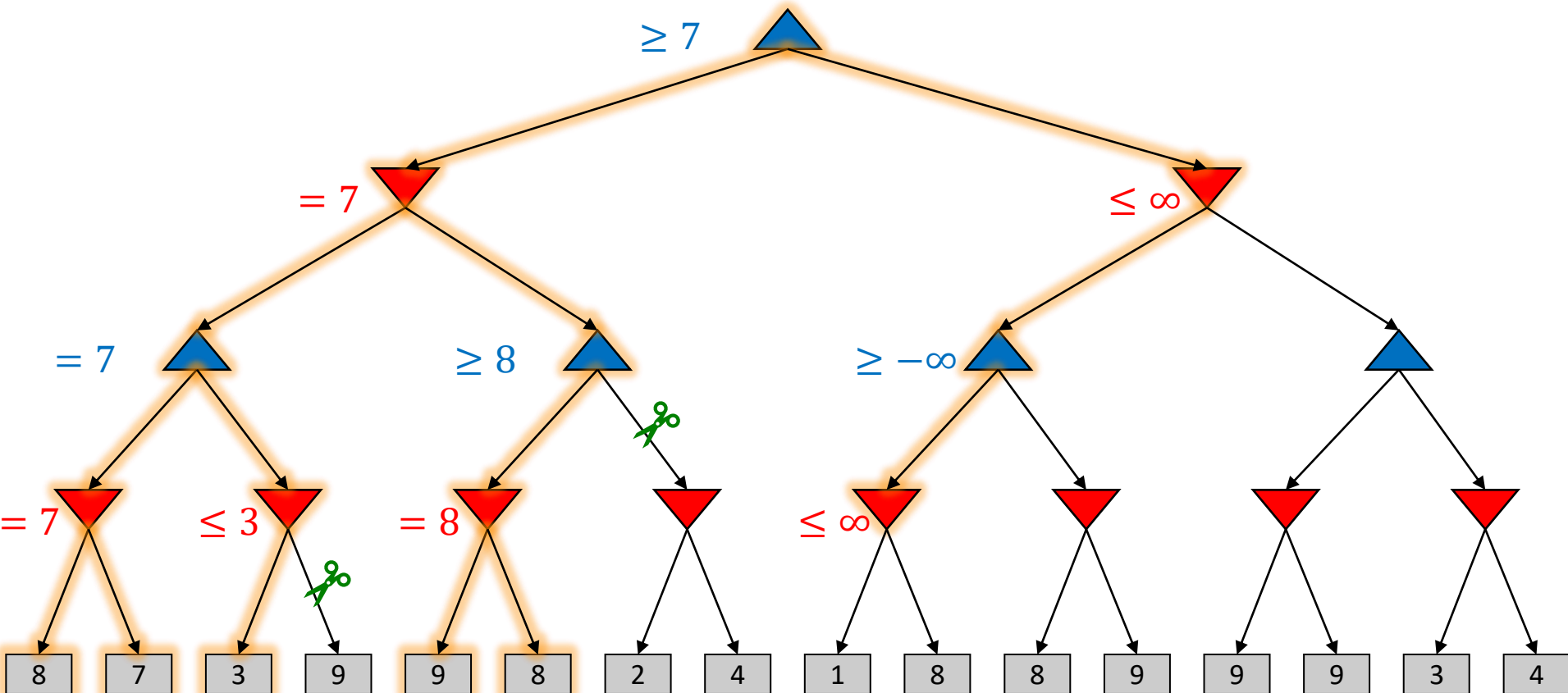
Tỉa alpha-beta: ví dụ 2



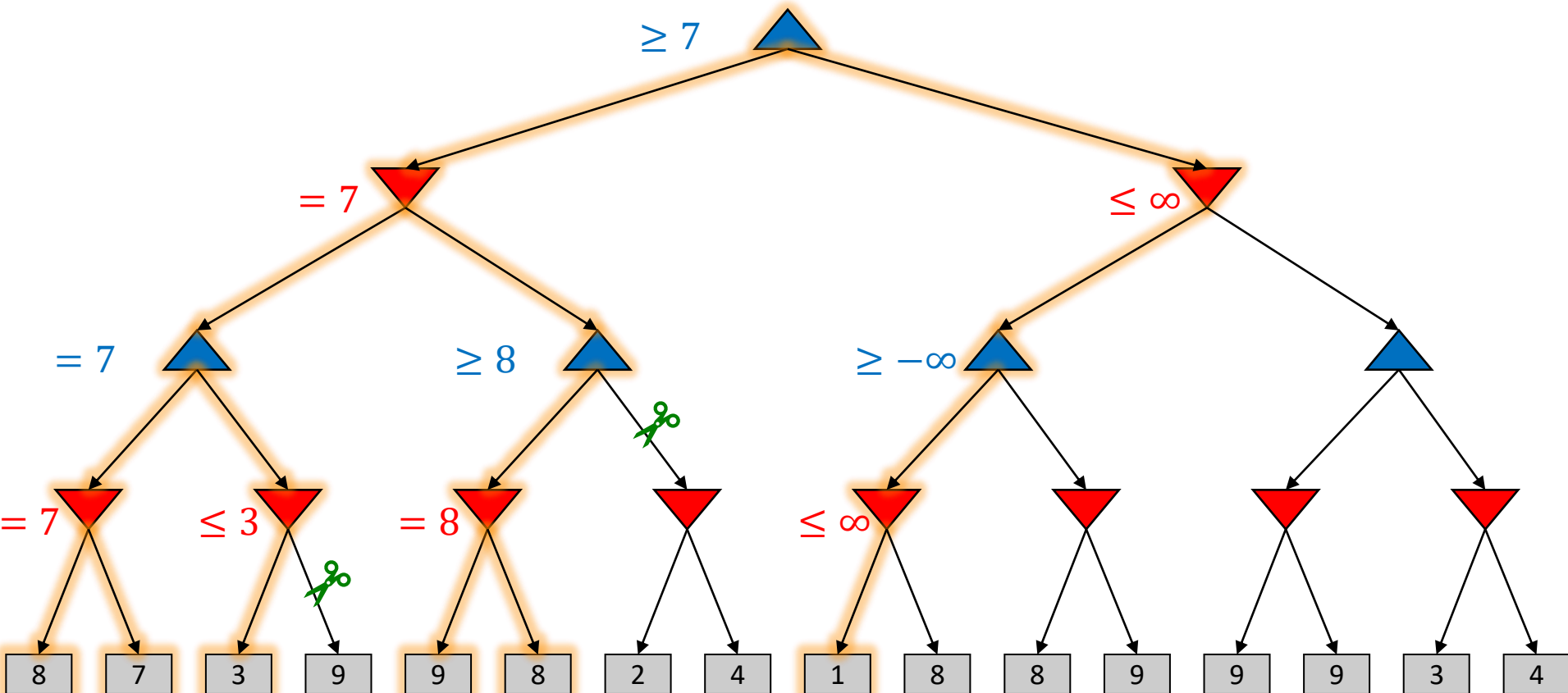
Tỉa alpha-beta: ví dụ 2



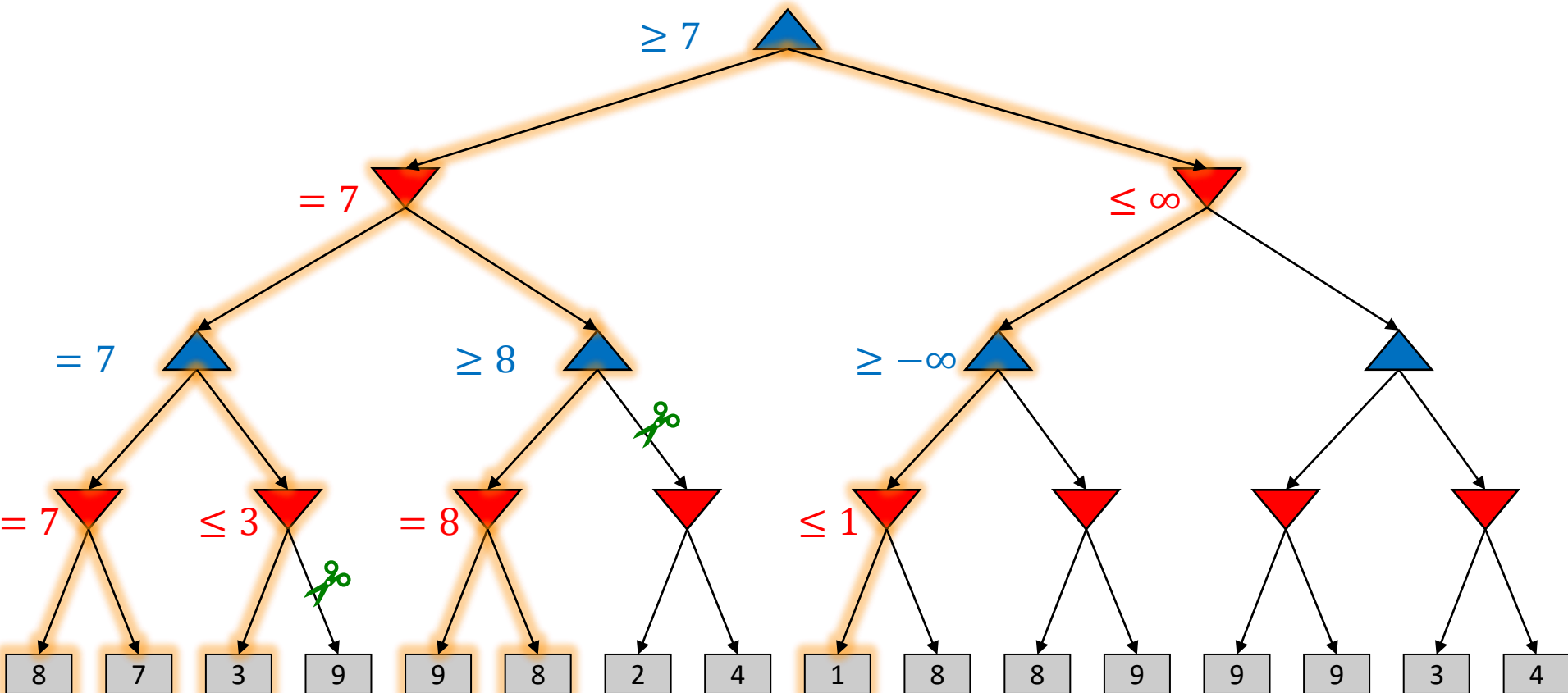
Tỉa alpha-beta: ví dụ 2



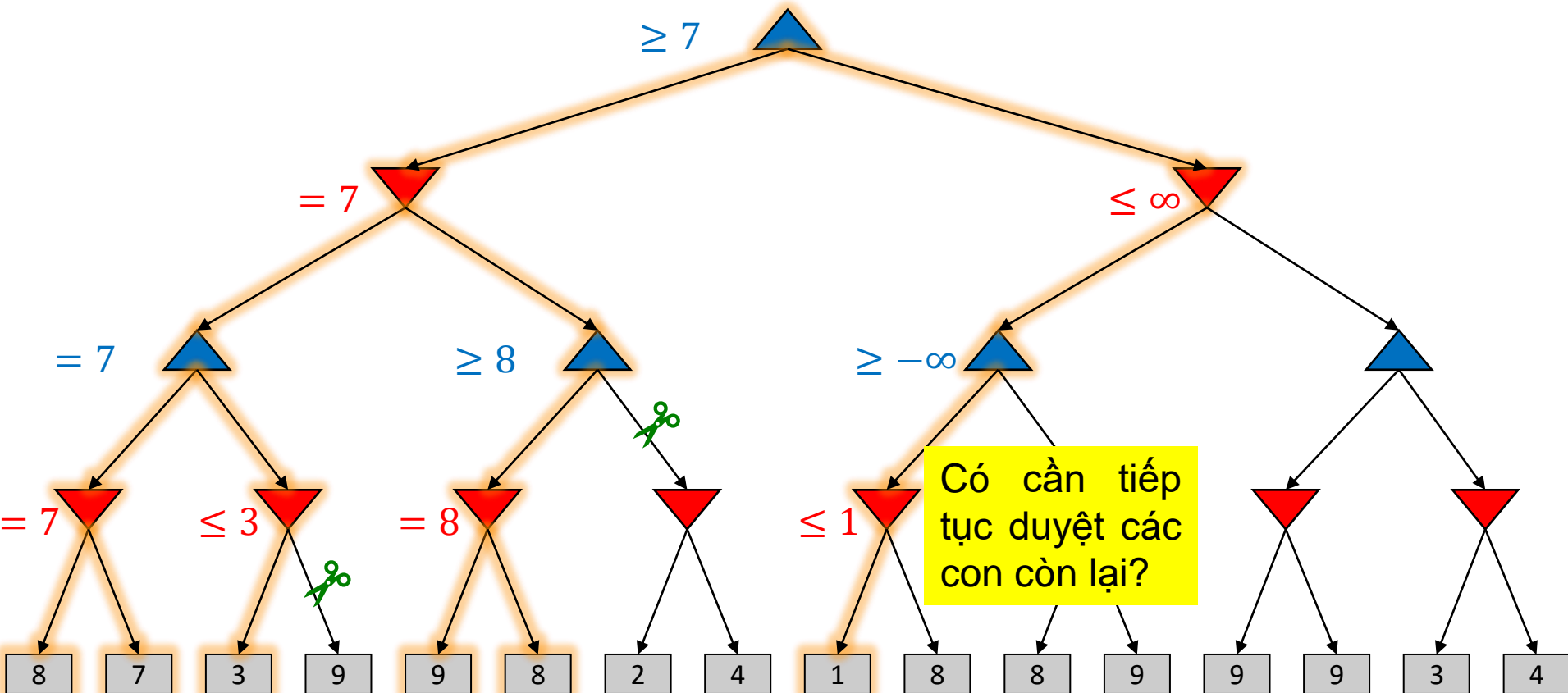
Tỉa alpha-beta: ví dụ 2



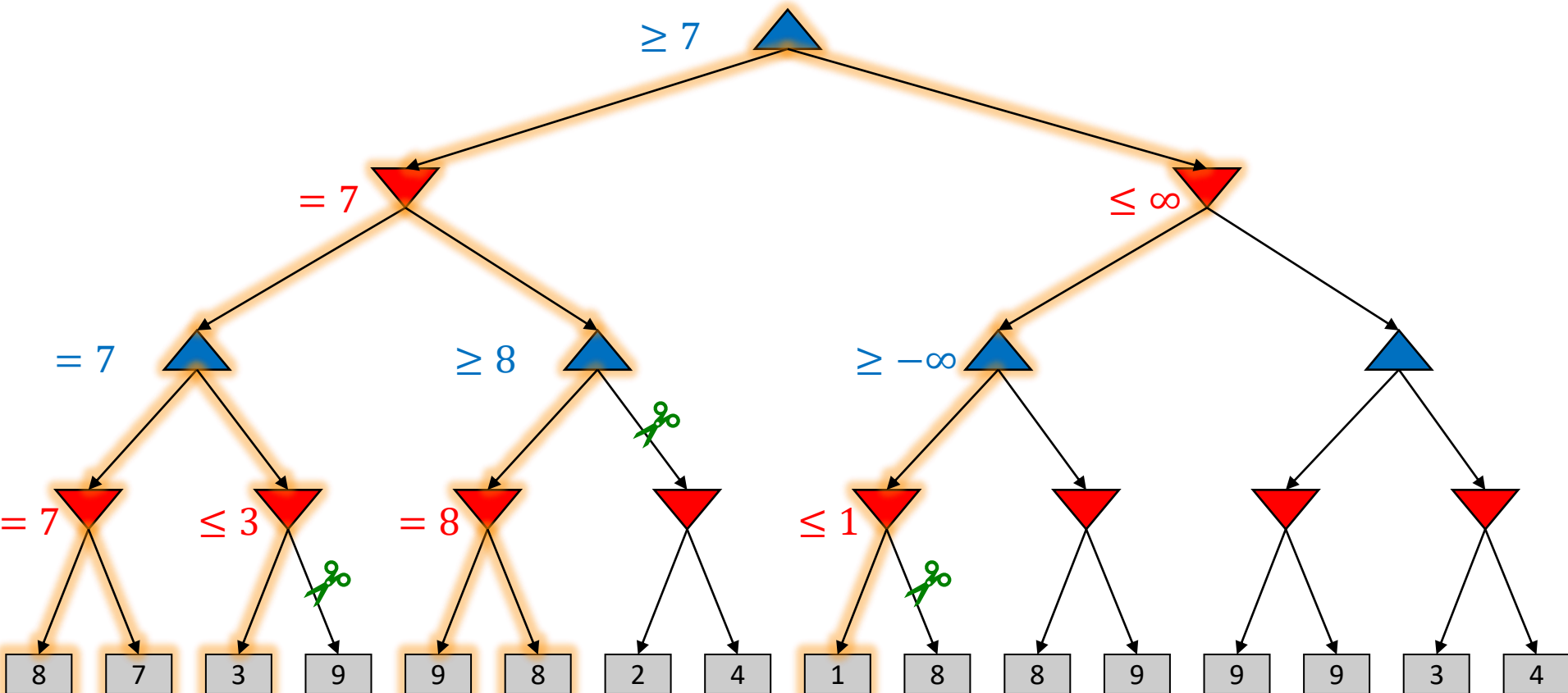
Tỉa alpha-beta: ví dụ 2



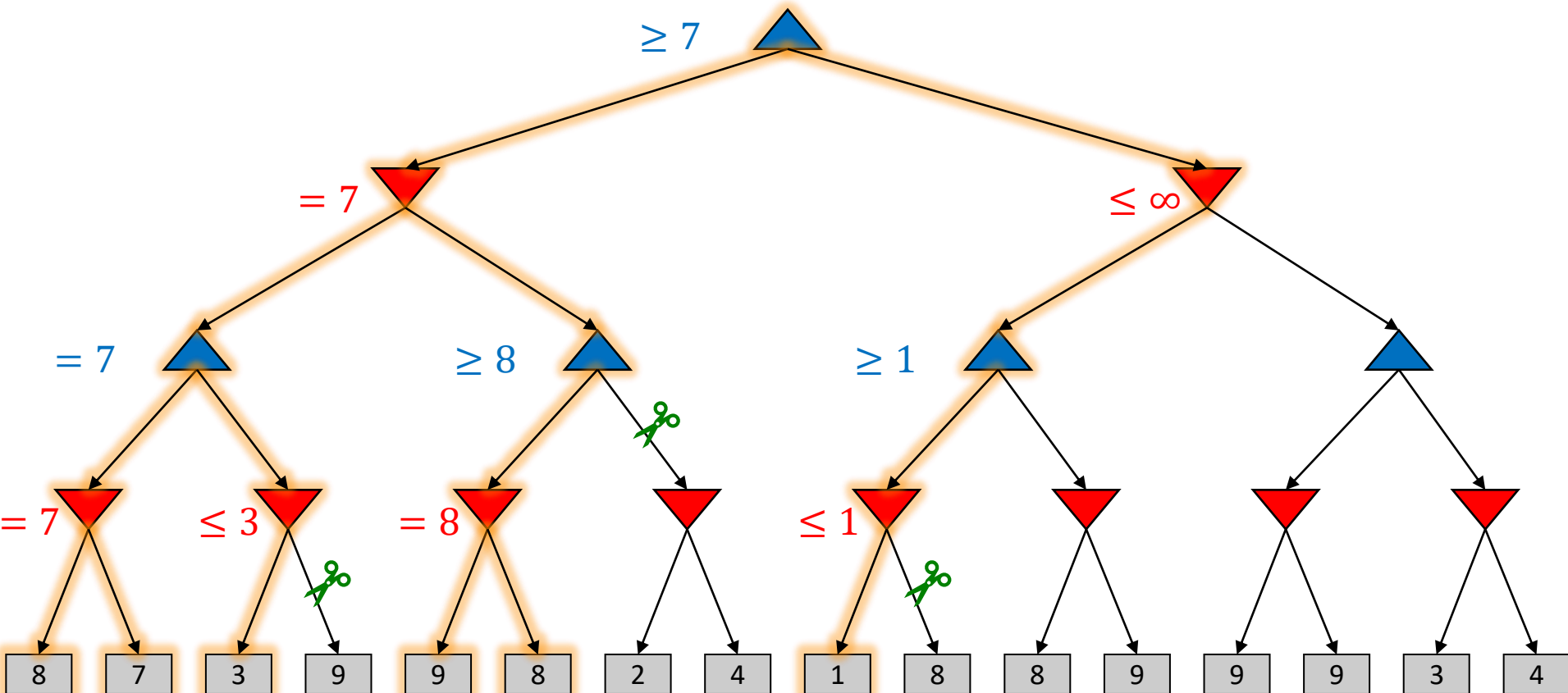
Tỉa alpha-beta: ví dụ 2



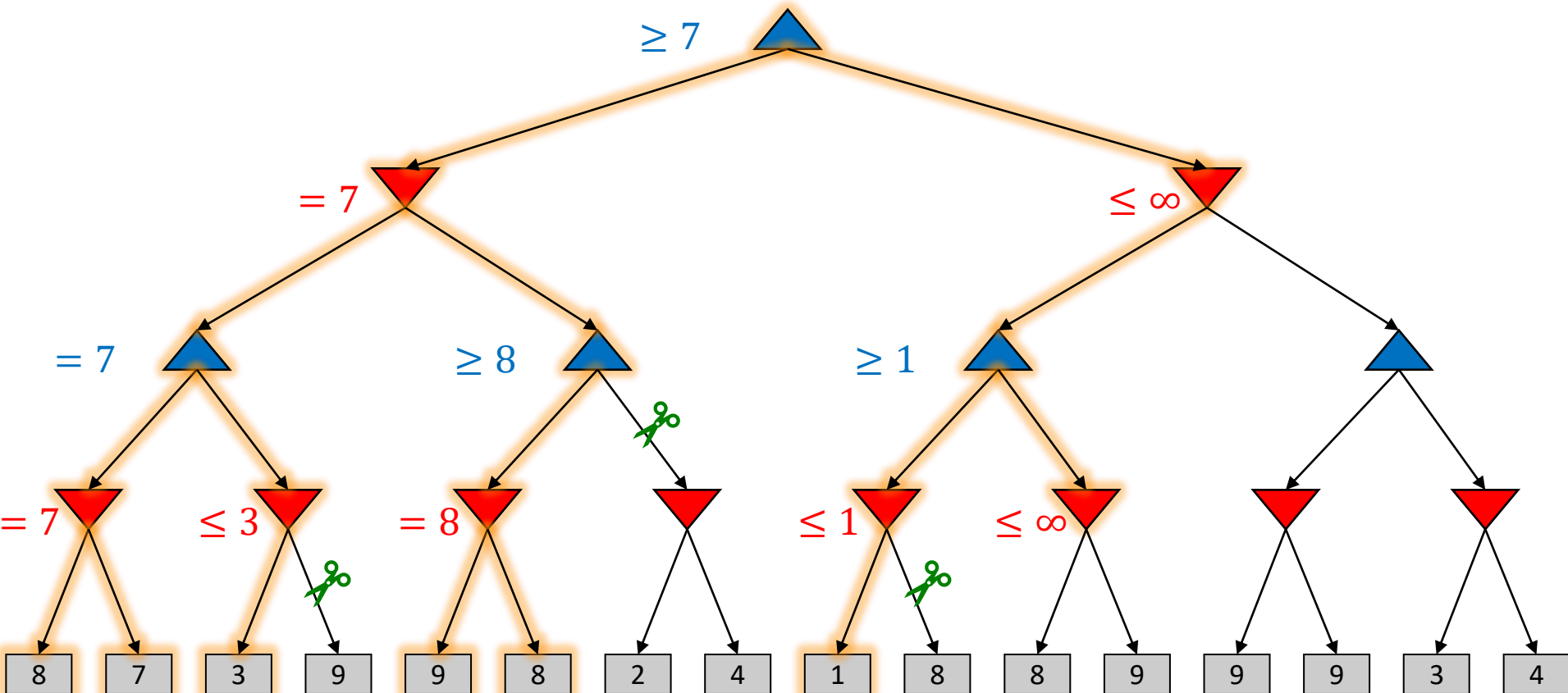
Tỉa alpha-beta: ví dụ 2



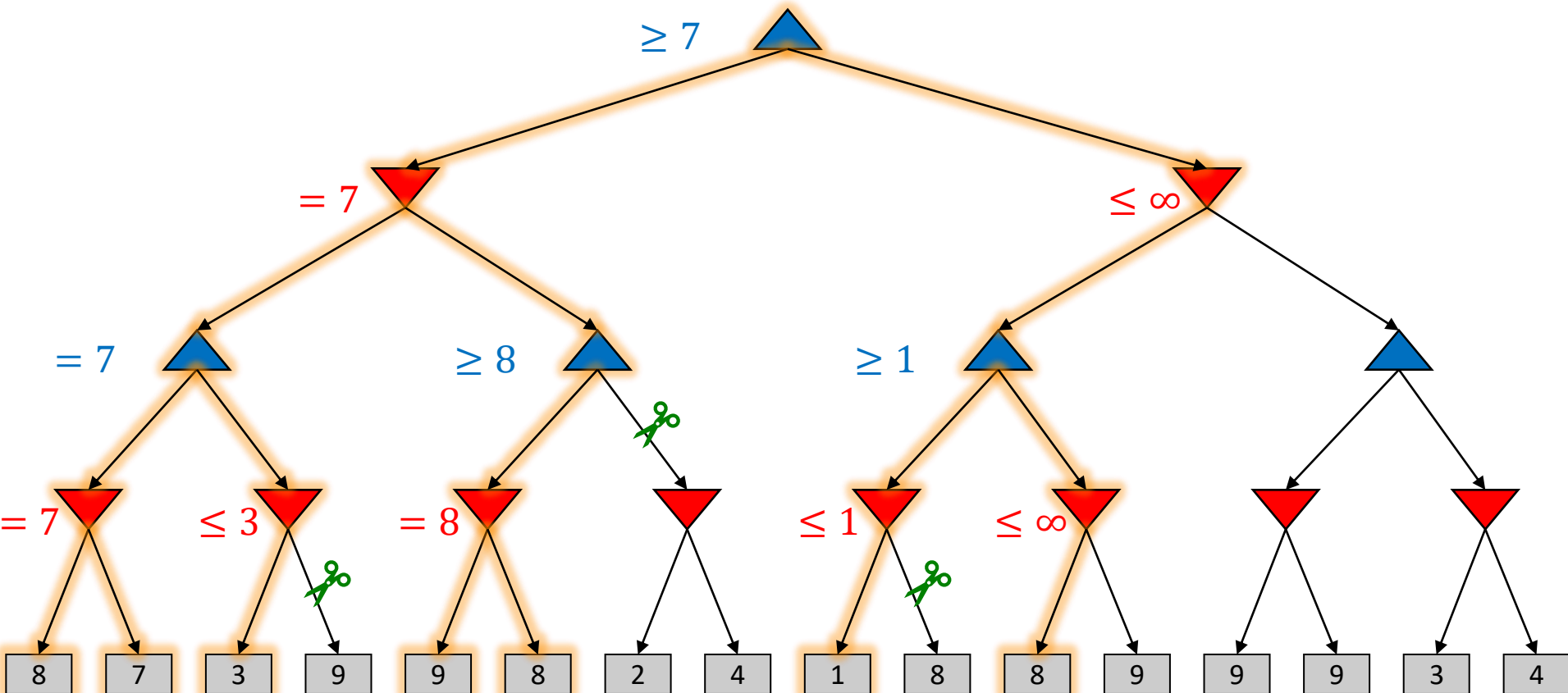
Tỉa alpha-beta: ví dụ 2



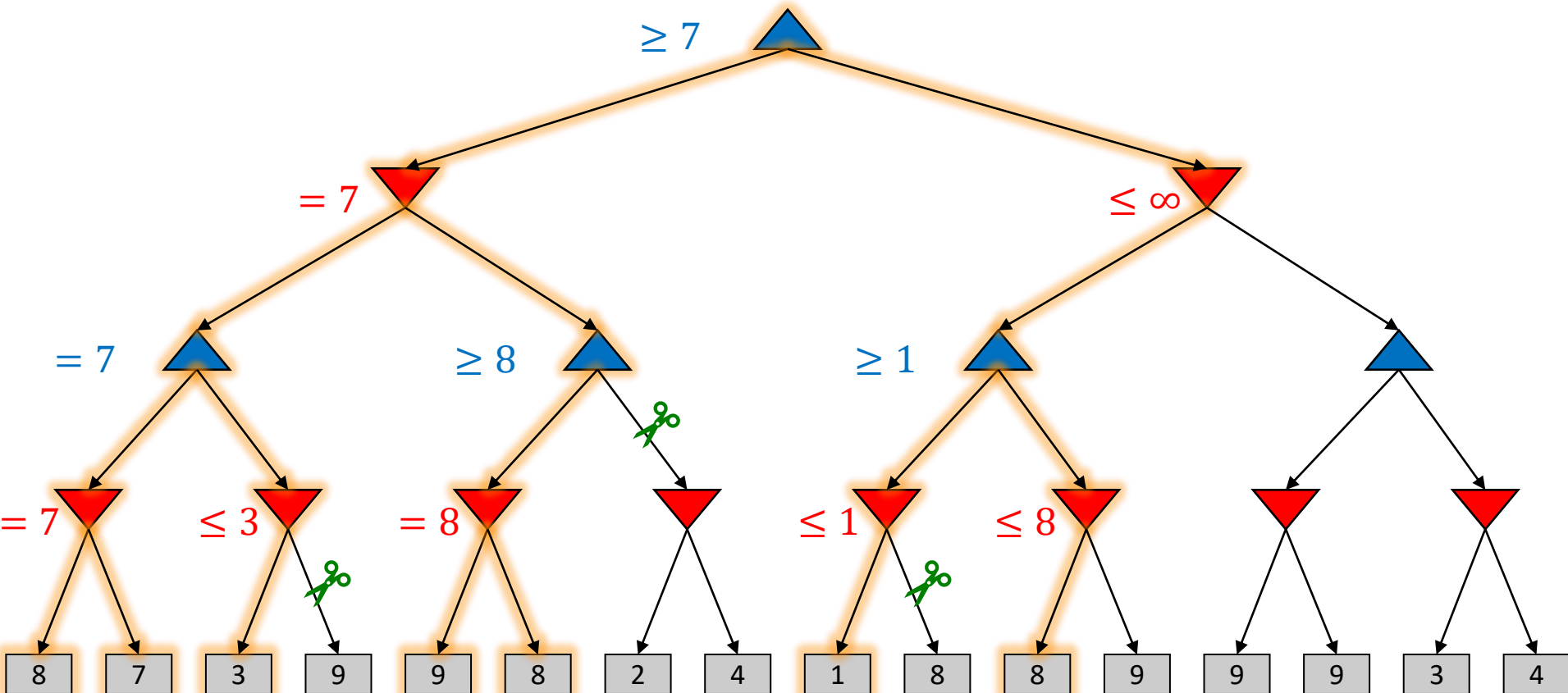
Tỉa alpha-beta: ví dụ 2



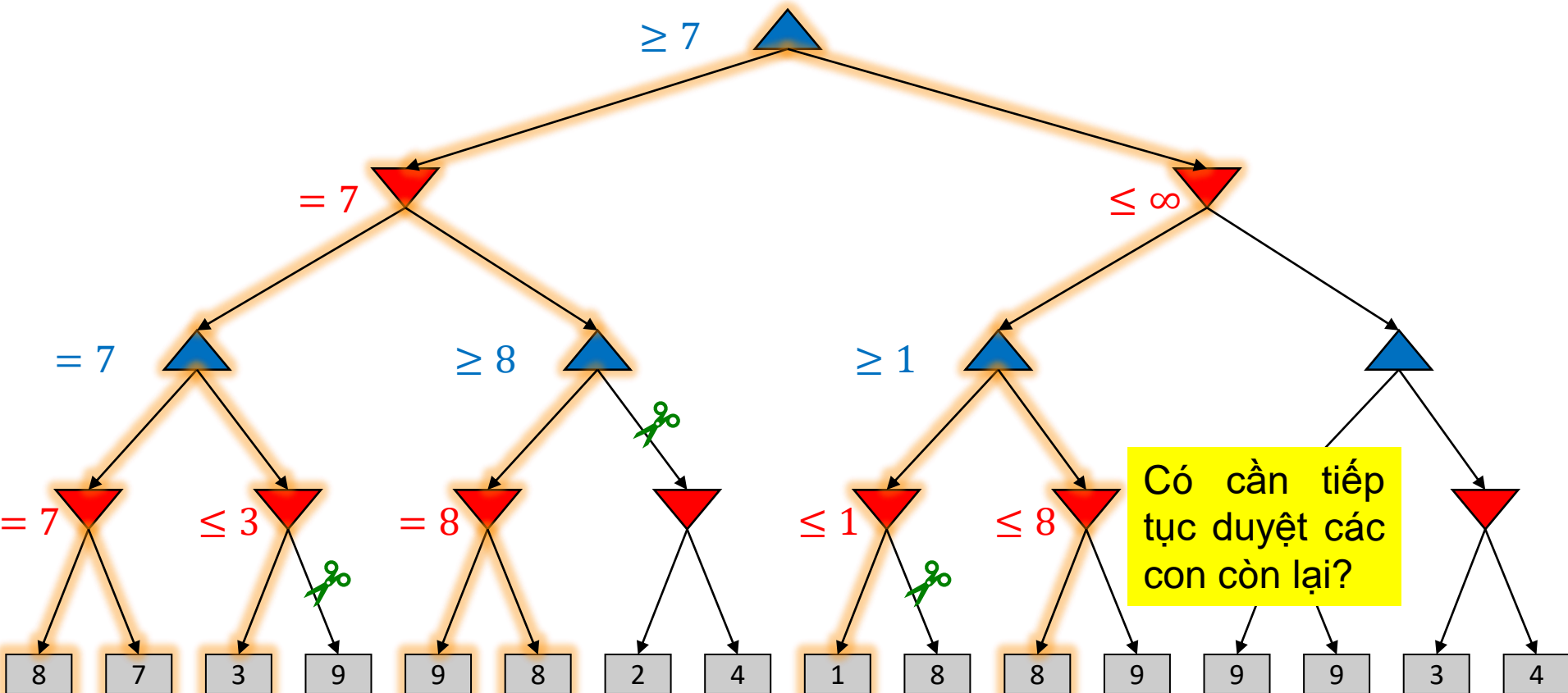
Tỉa alpha-beta: ví dụ 2



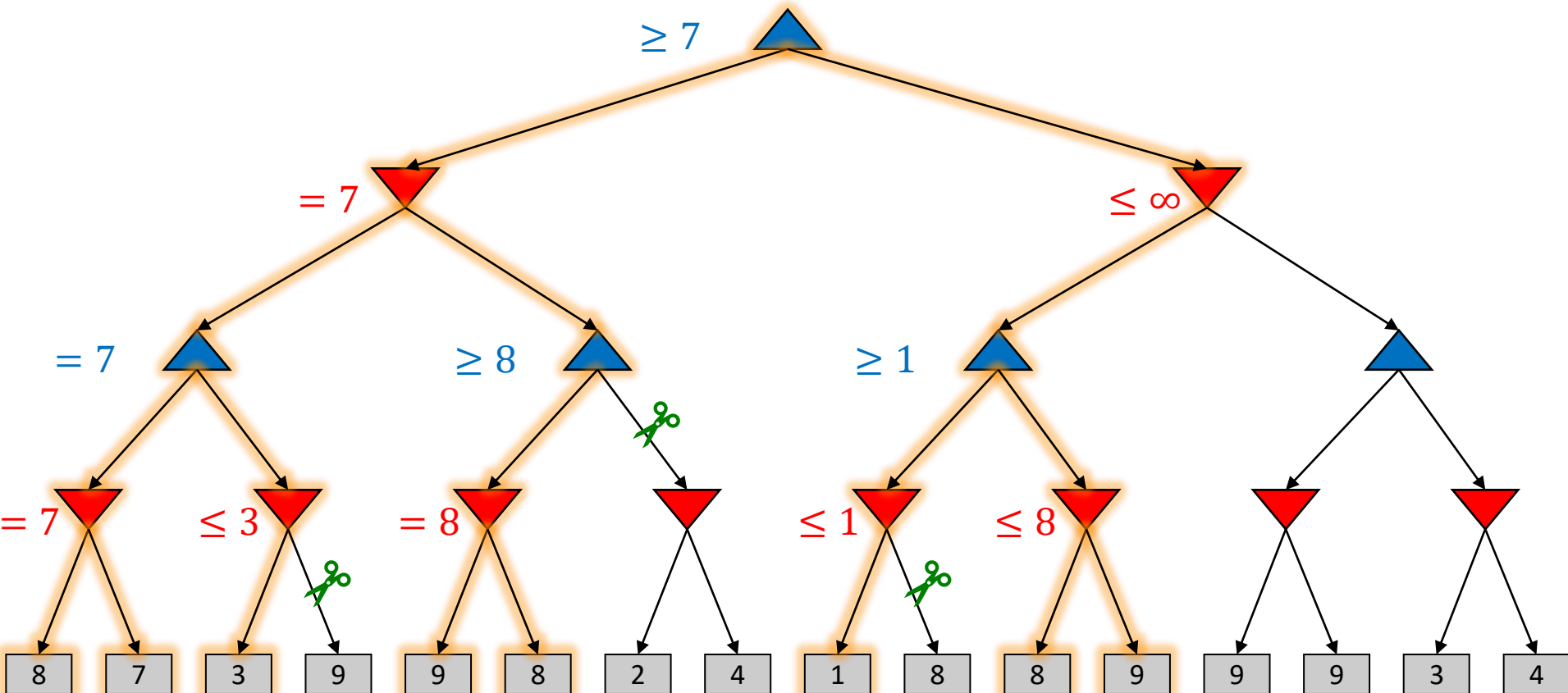
Tỉa alpha-beta: ví dụ 2



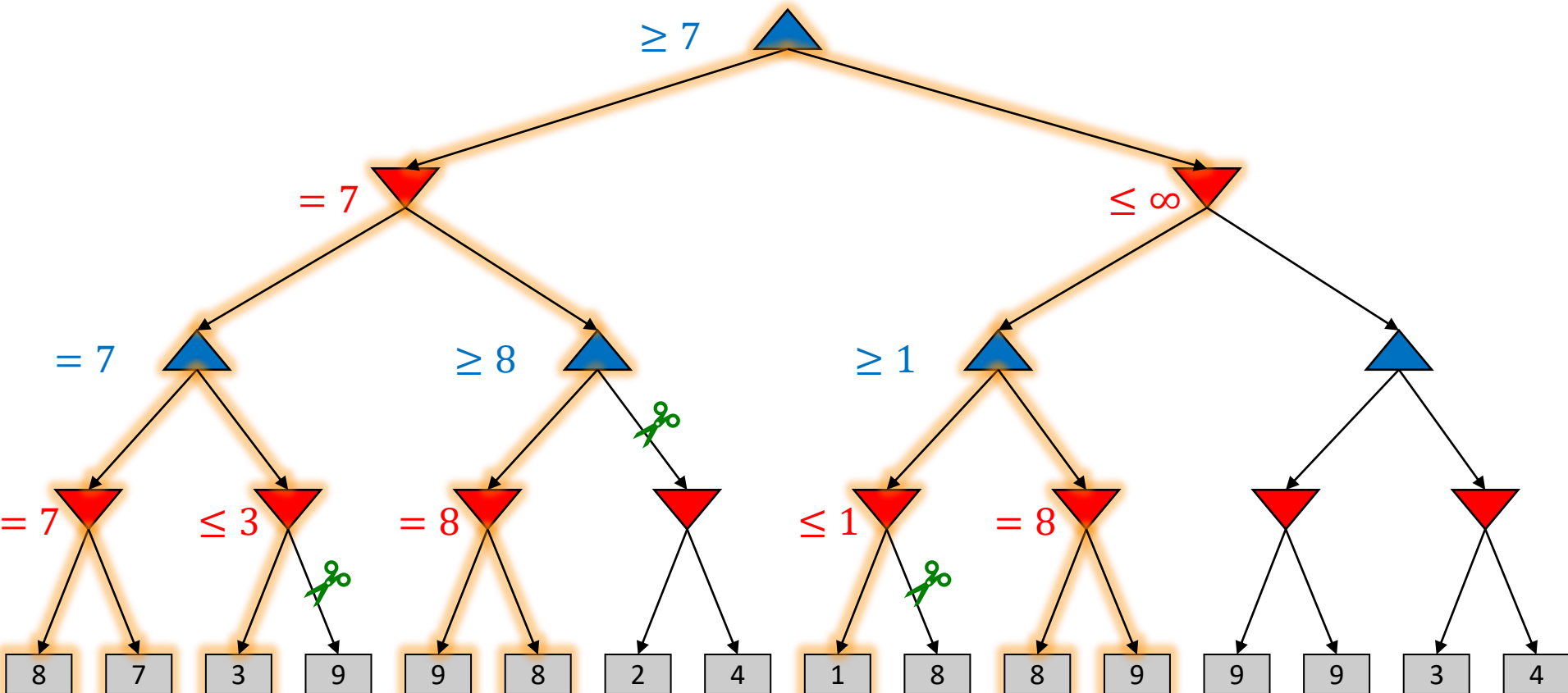
Tỉa alpha-beta: ví dụ 2



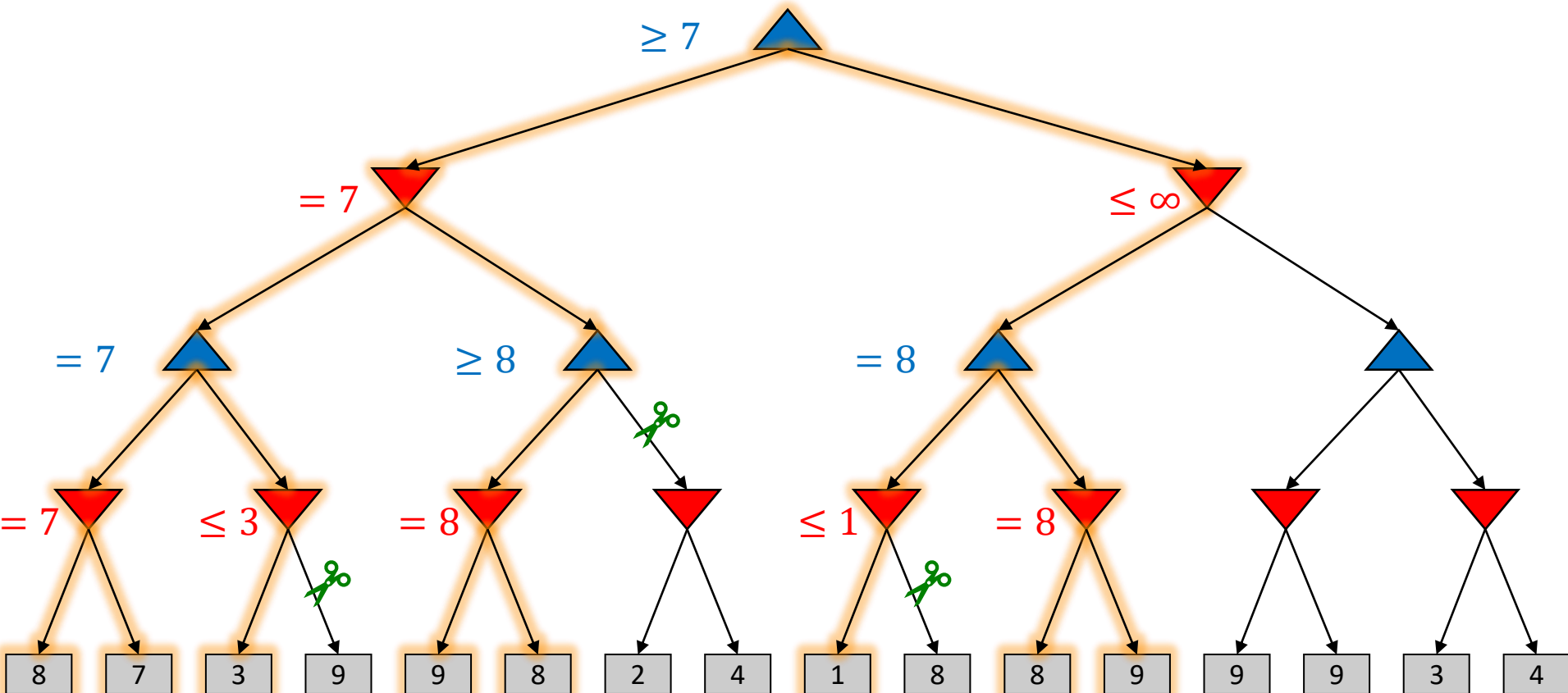
Tỉa alpha-beta: ví dụ 2



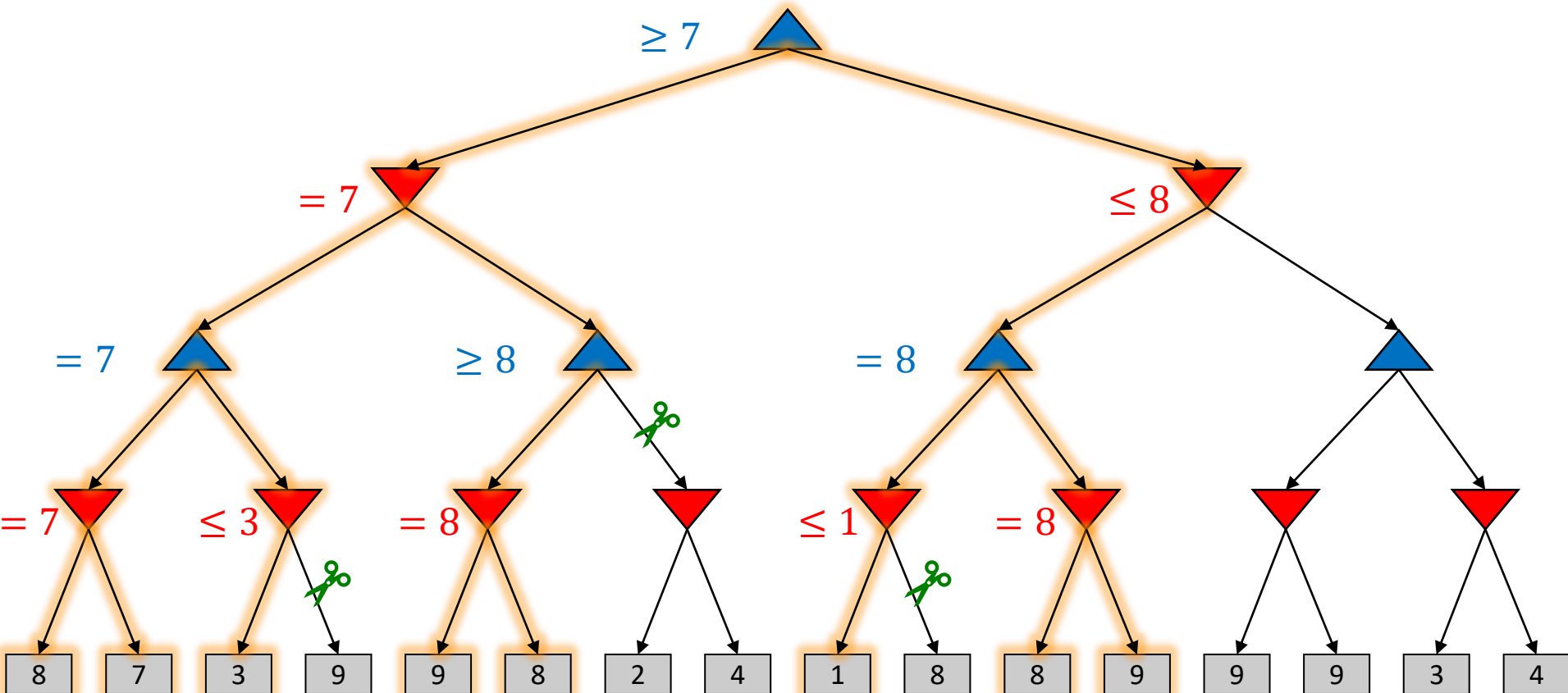
Tỉa alpha-beta: ví dụ 2



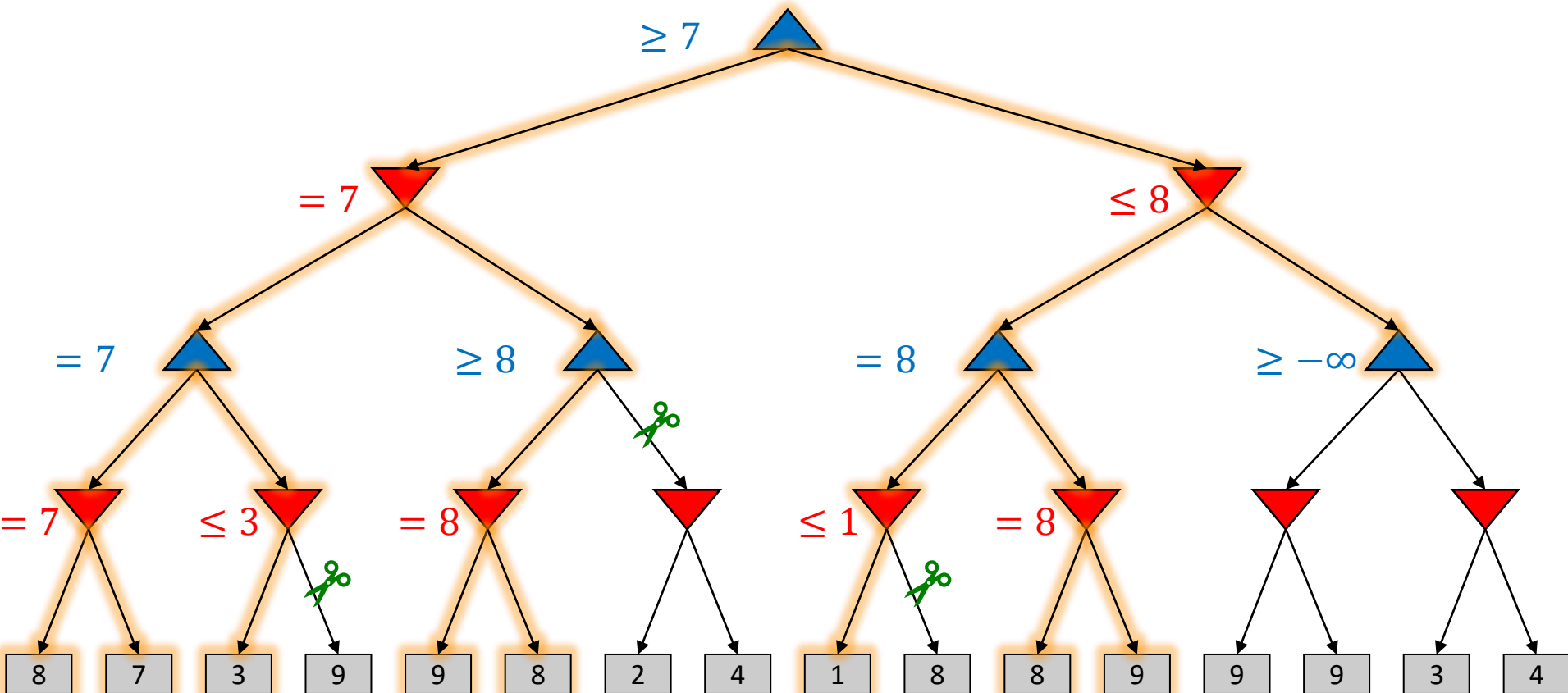
Tỉa alpha-beta: ví dụ 2



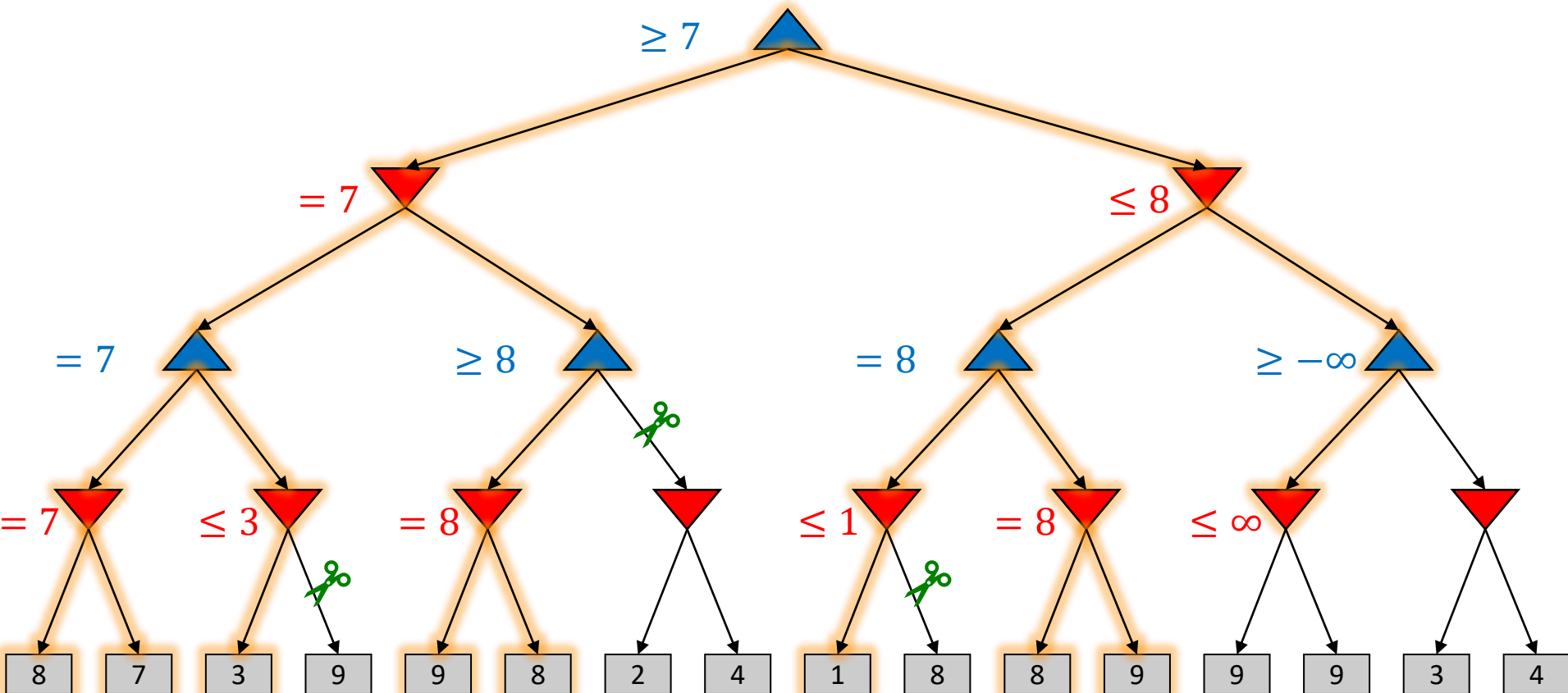
Tỉa alpha-beta: ví dụ 2



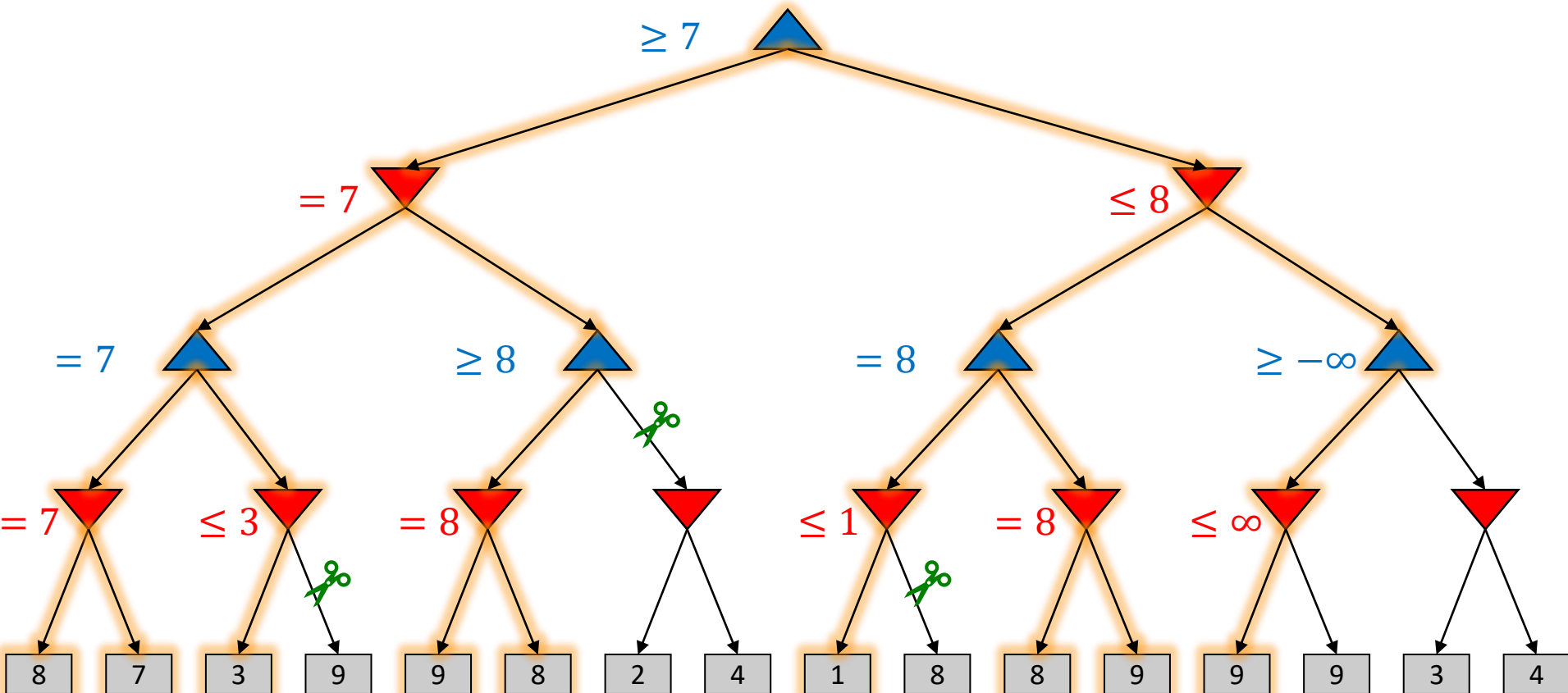
Tỉa alpha-beta: ví dụ 2



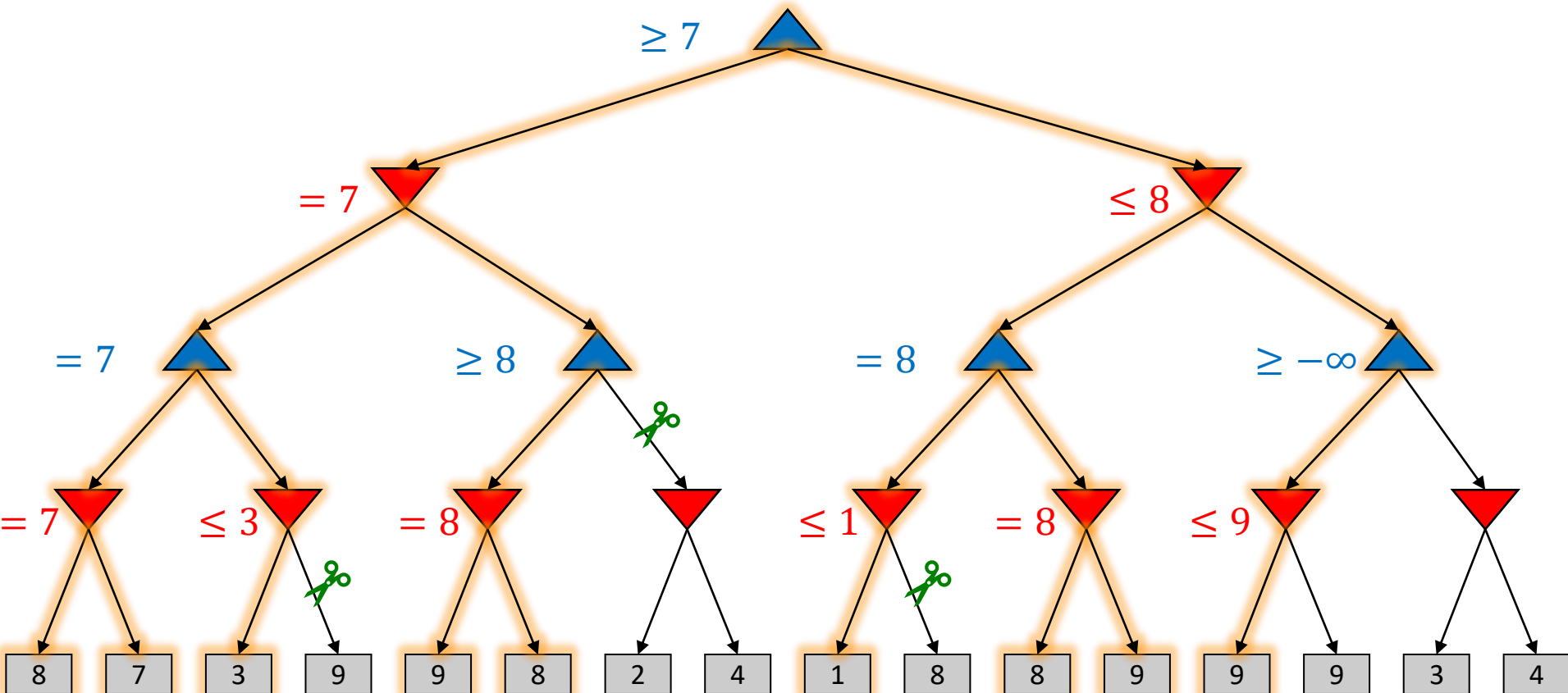
Tỉa alpha-beta: ví dụ 2



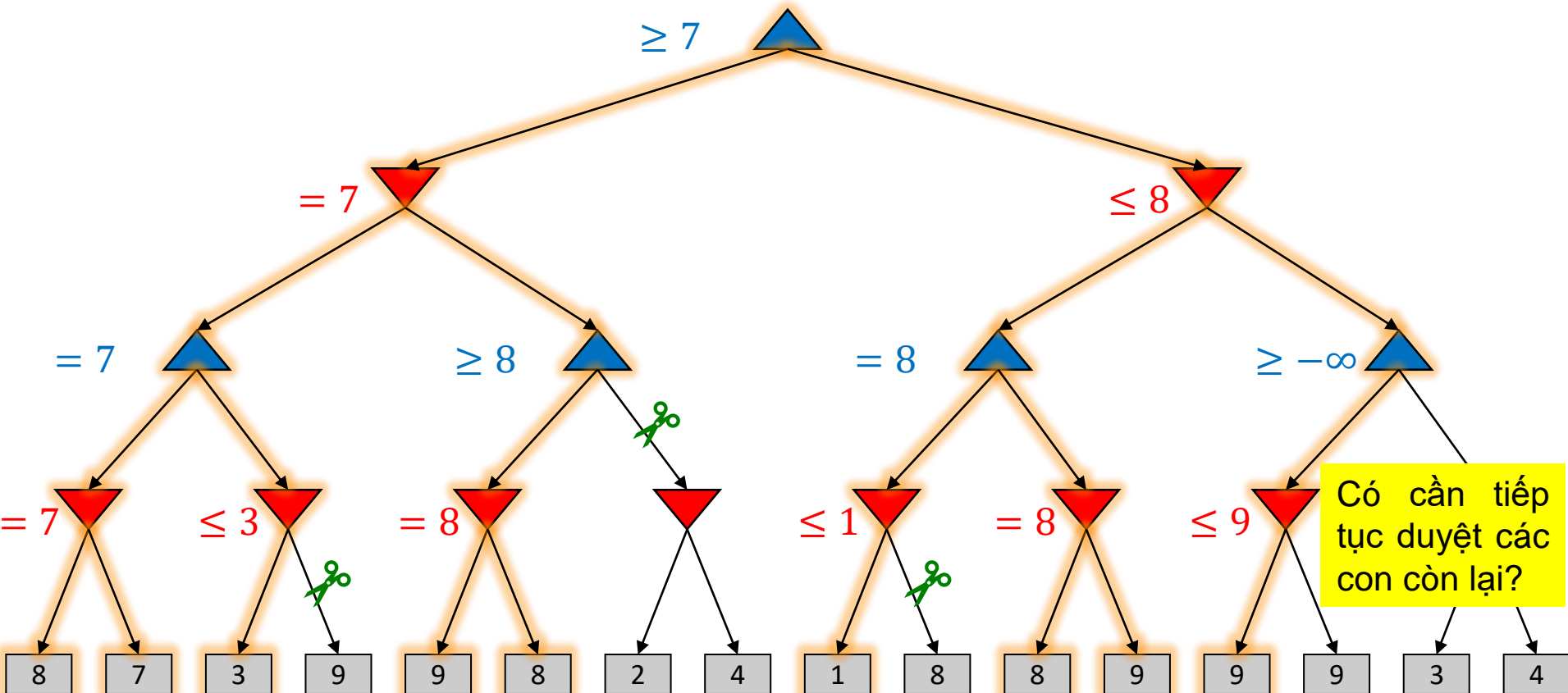
Tỉa alpha-beta: ví dụ 2



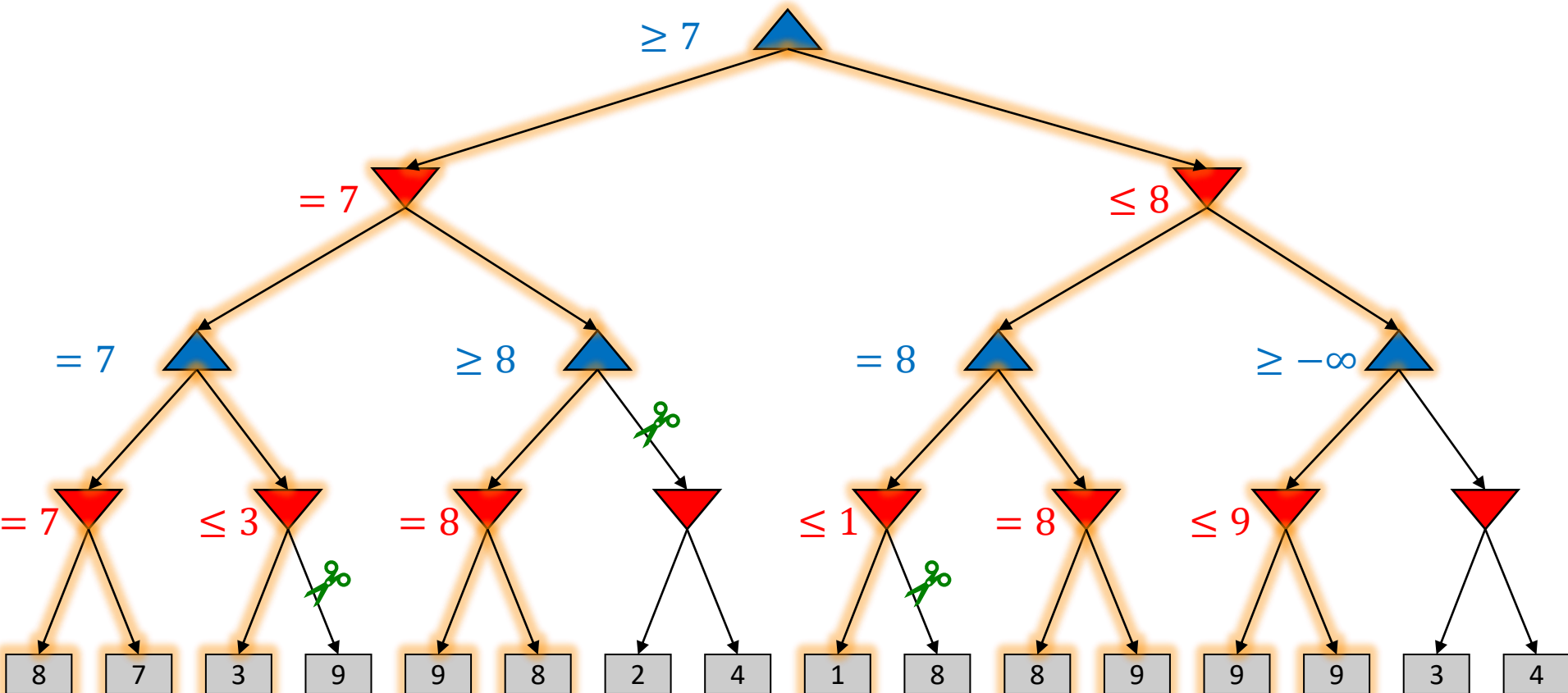
Tỉa alpha-beta: ví dụ 2



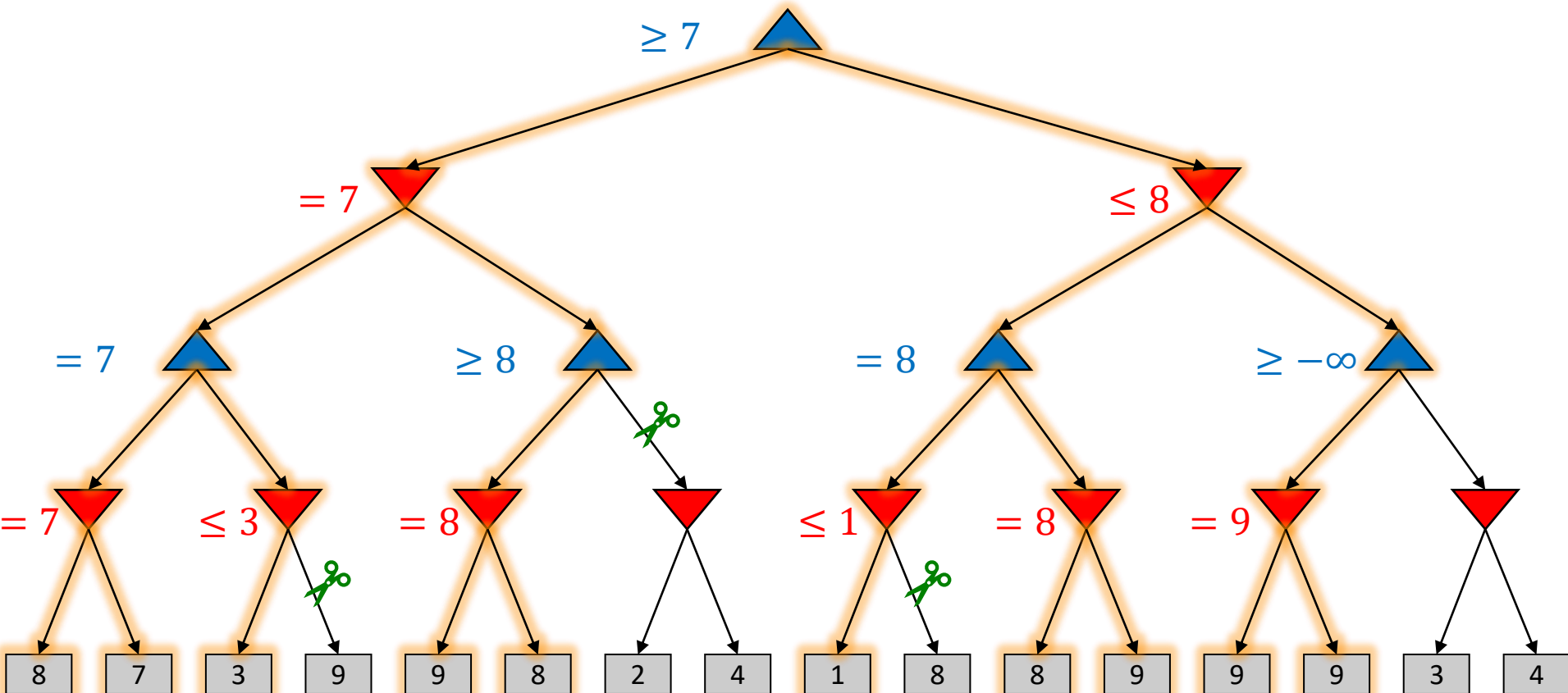
Tỉa alpha-beta: ví dụ 2



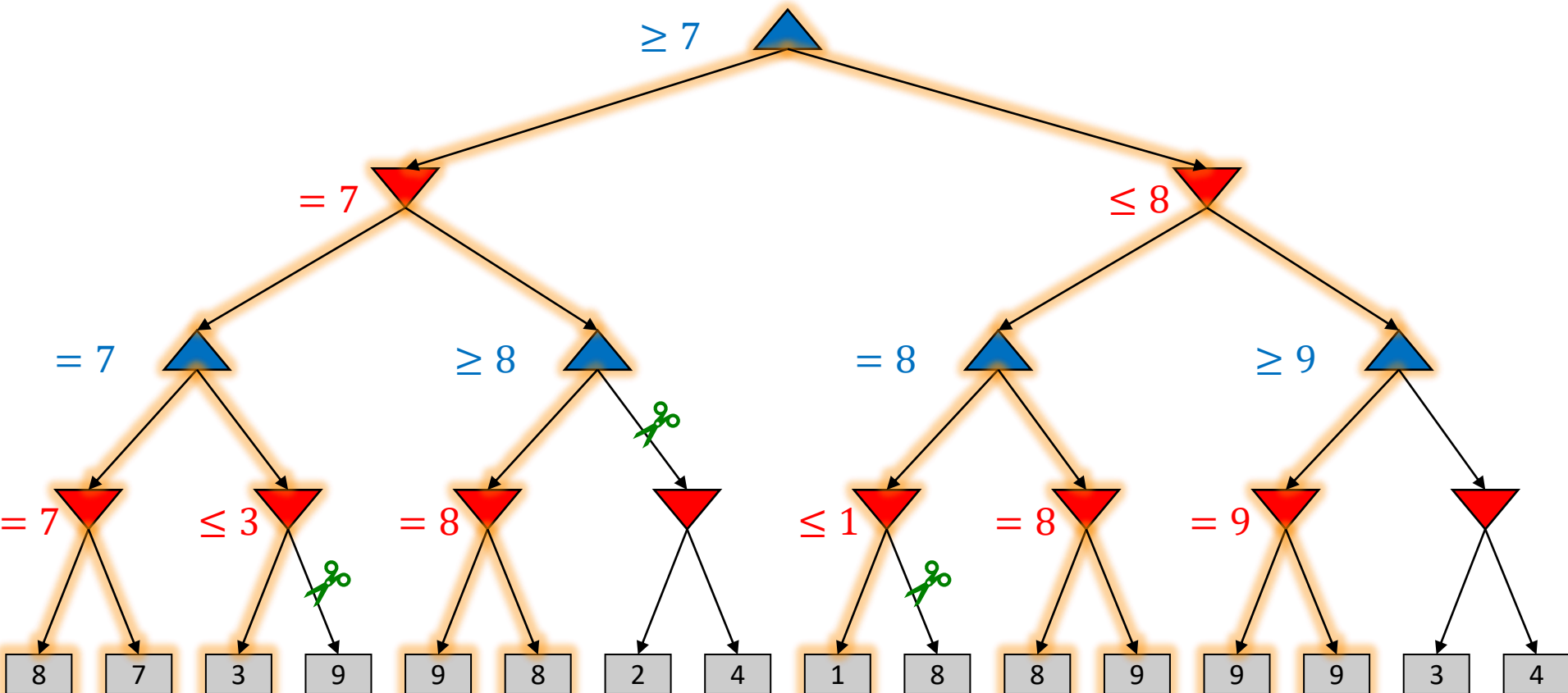
Tỉa alpha-beta: ví dụ 2



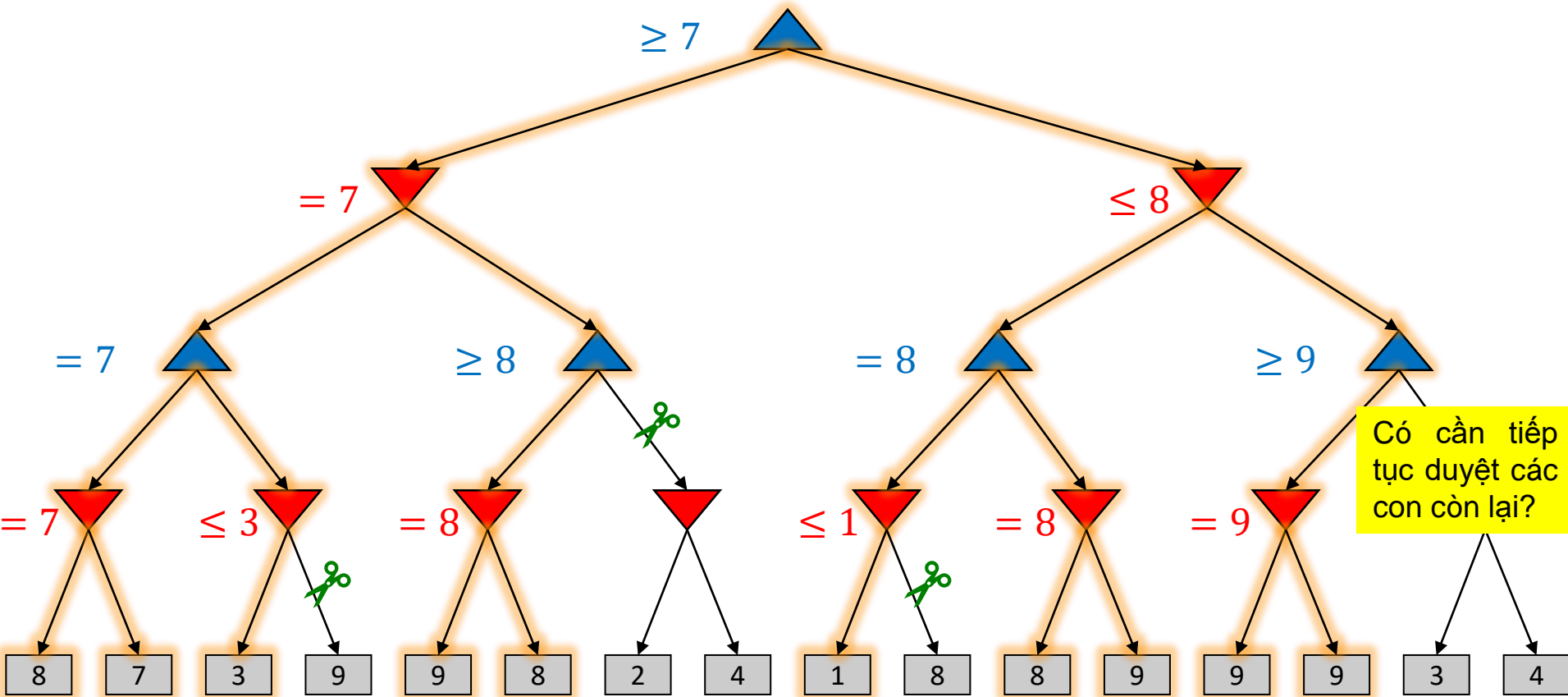
Tỉa alpha-beta: ví dụ 2



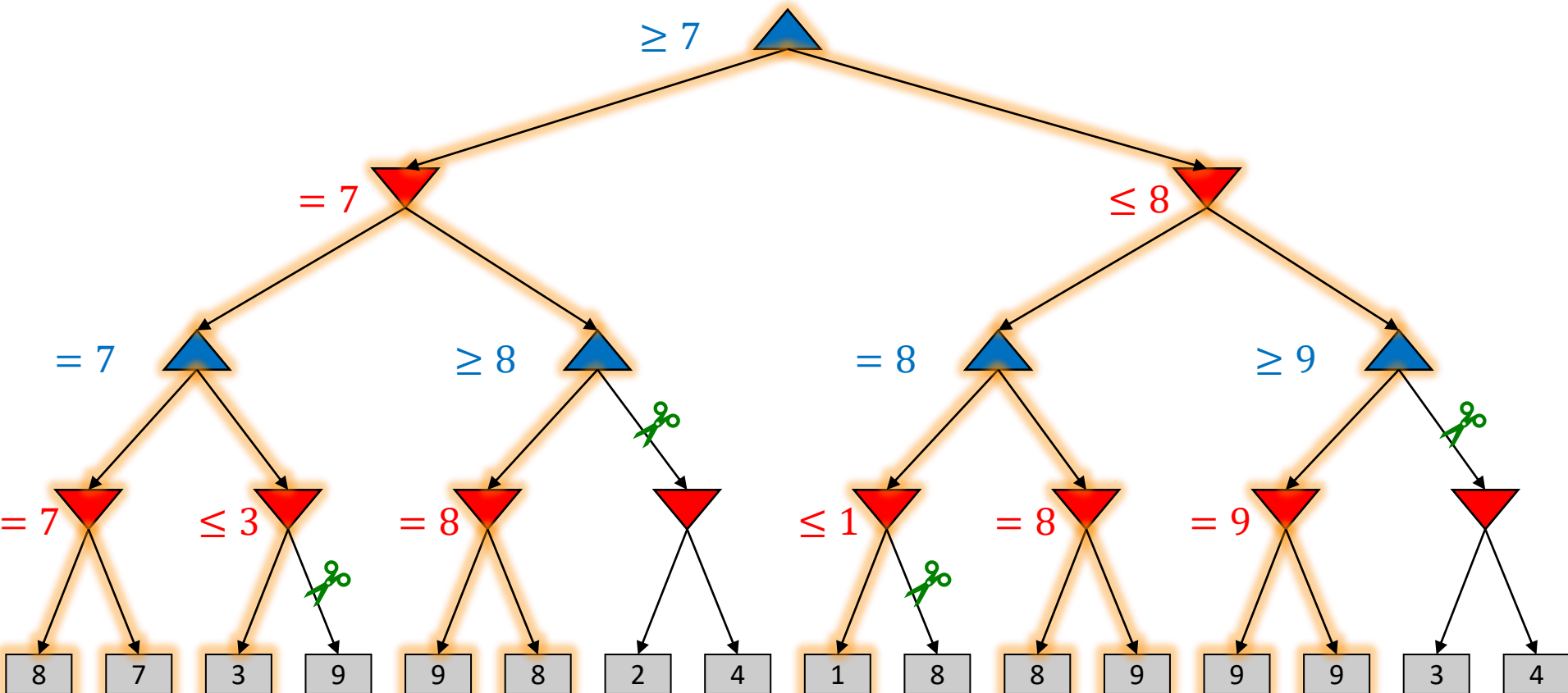
Tỉa alpha-beta: ví dụ 2



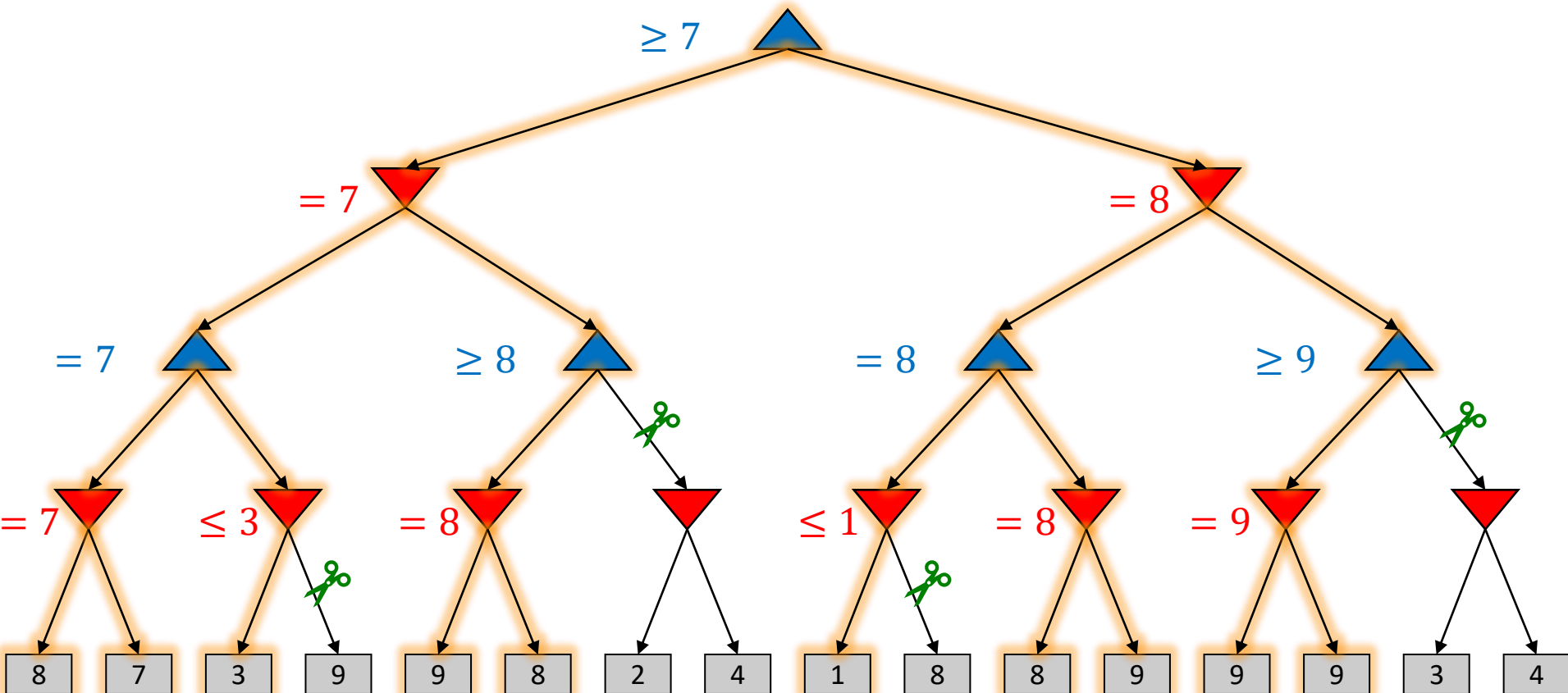
Tỉa alpha-beta: ví dụ 2



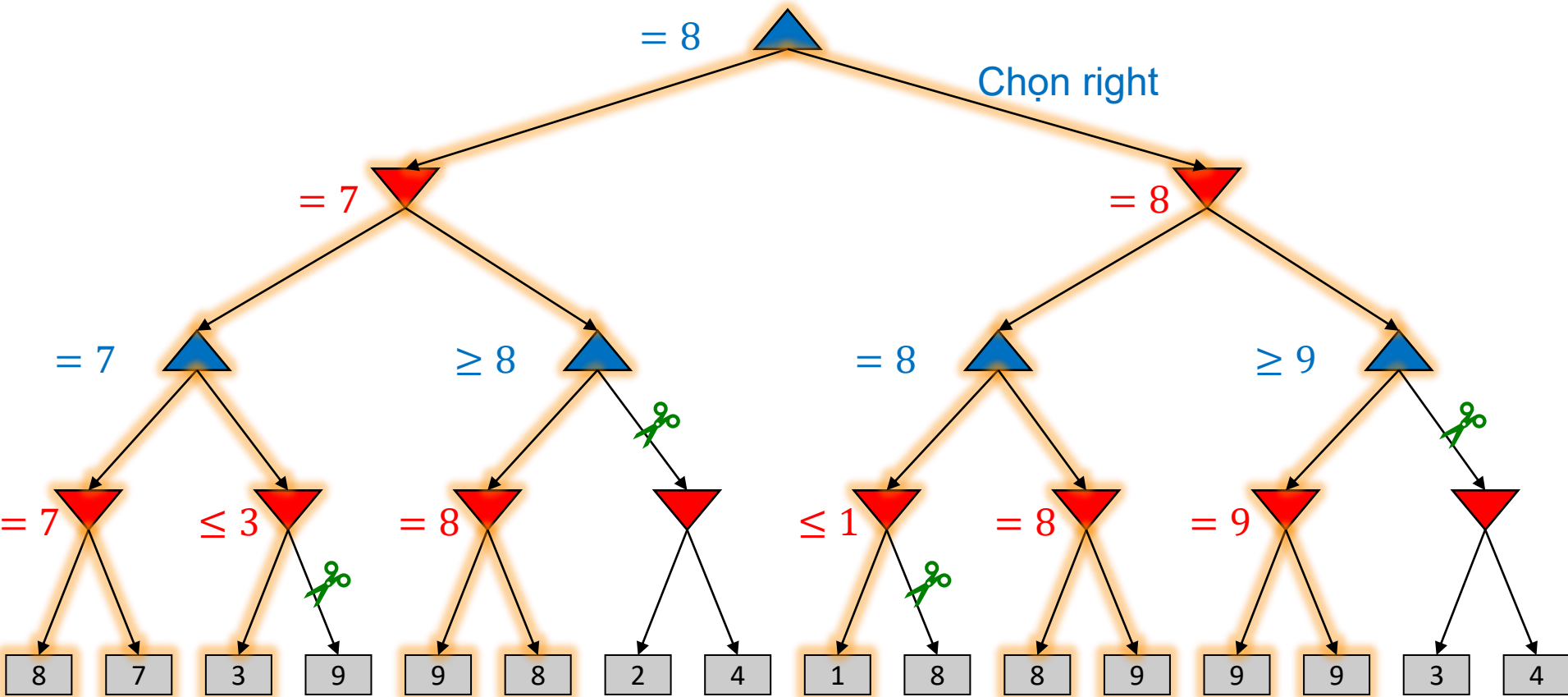
Tỉa alpha-beta: ví dụ 2



Tỉa alpha-beta: ví dụ 2



Tỉa alpha-beta: ví dụ 2



Mã giả thuật toán minimax + tỉa alpha-beta

Ở một trạng thái, khi tính giá trị minimax, để biết có cần phải duyệt các trạng thái con còn lại hay không, ta sẽ có hai biến α và β :

- α : giá trị tốt nhất (lớn nhất) hiện tại của người chơi MAX tính từ trạng thái đang xét cho đến trạng thái gốc
- β : giá trị tốt nhất (nhỏ nhất) hiện tại của người chơi MIN tính từ trạng thái đang xét cho đến trạng thái gốc

```
def max-value(state,  $\alpha$ ,  $\beta$ ):
```

```
    if the state is a terminal state:
```

```
        return the state's utility
```

```
    initialize  $v = -\infty$ 
```

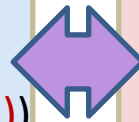
```
    for each successor of state:
```

```
         $v = \max(v, \text{min-value}(\text{successor}, \alpha, \beta))$ 
```

```
        if  $v \geq \beta$  return  $v$ 
```

```
         $\alpha = \max(\alpha, v)$ 
```

```
    return  $v$ 
```



```
def min-value(state,  $\alpha$ ,  $\beta$ ):
```

```
    if the state is a terminal state:
```

```
        return the state's utility
```

```
    initialize  $v = +\infty$ 
```

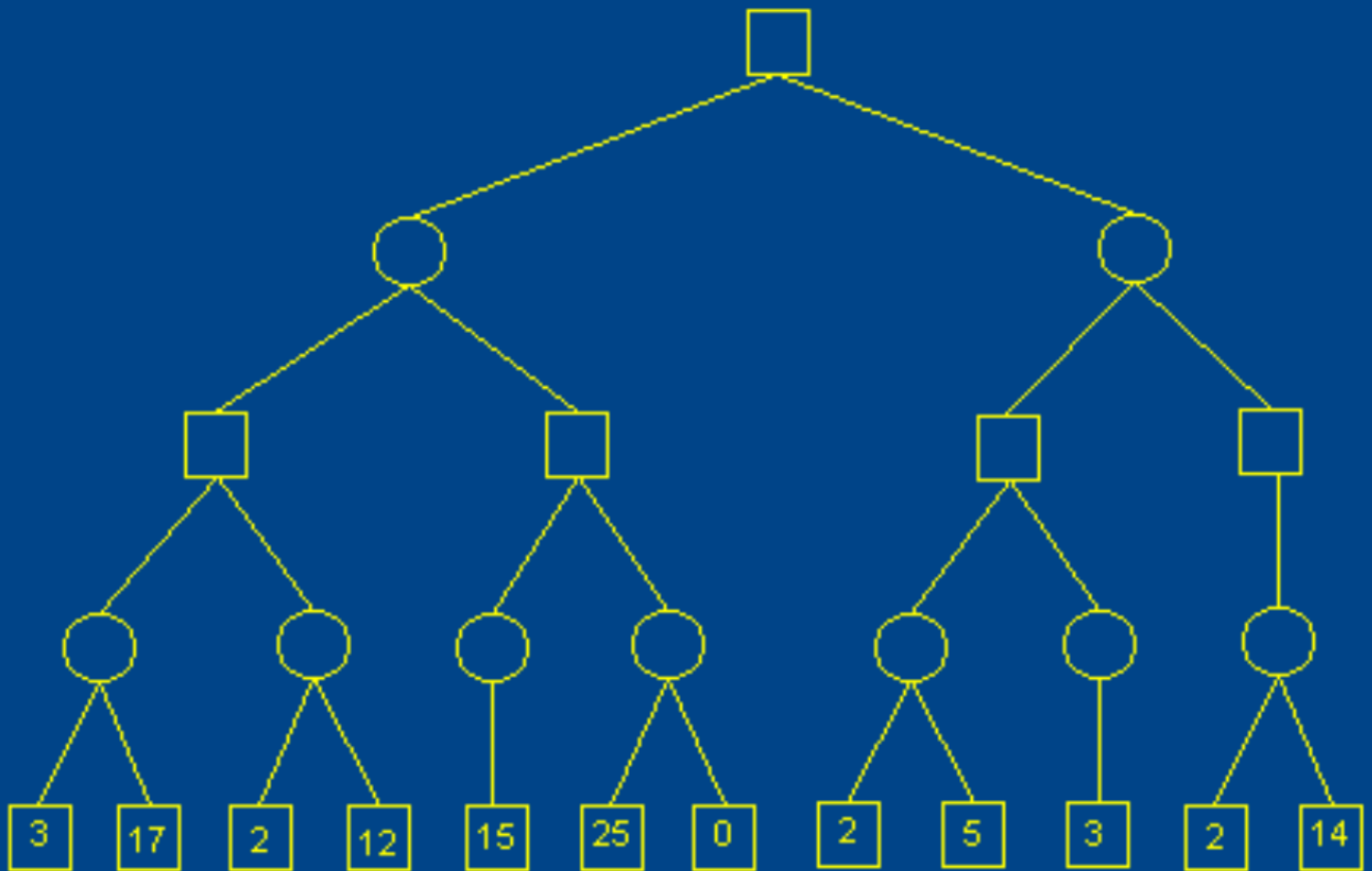
```
    for each successor of state:
```

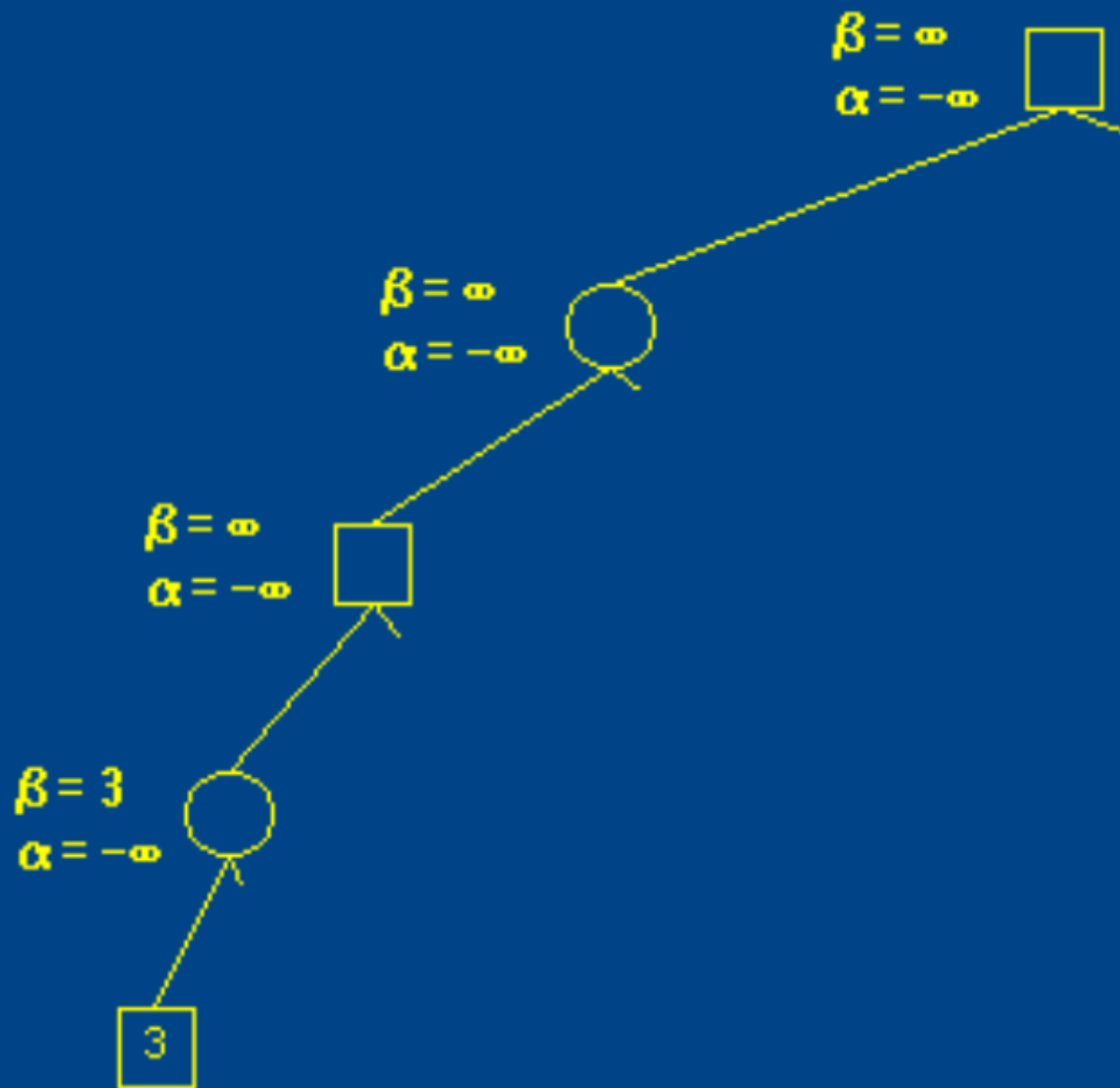
```
         $v = \min(v, \text{max-value}(\text{successor}, \alpha, \beta))$ 
```

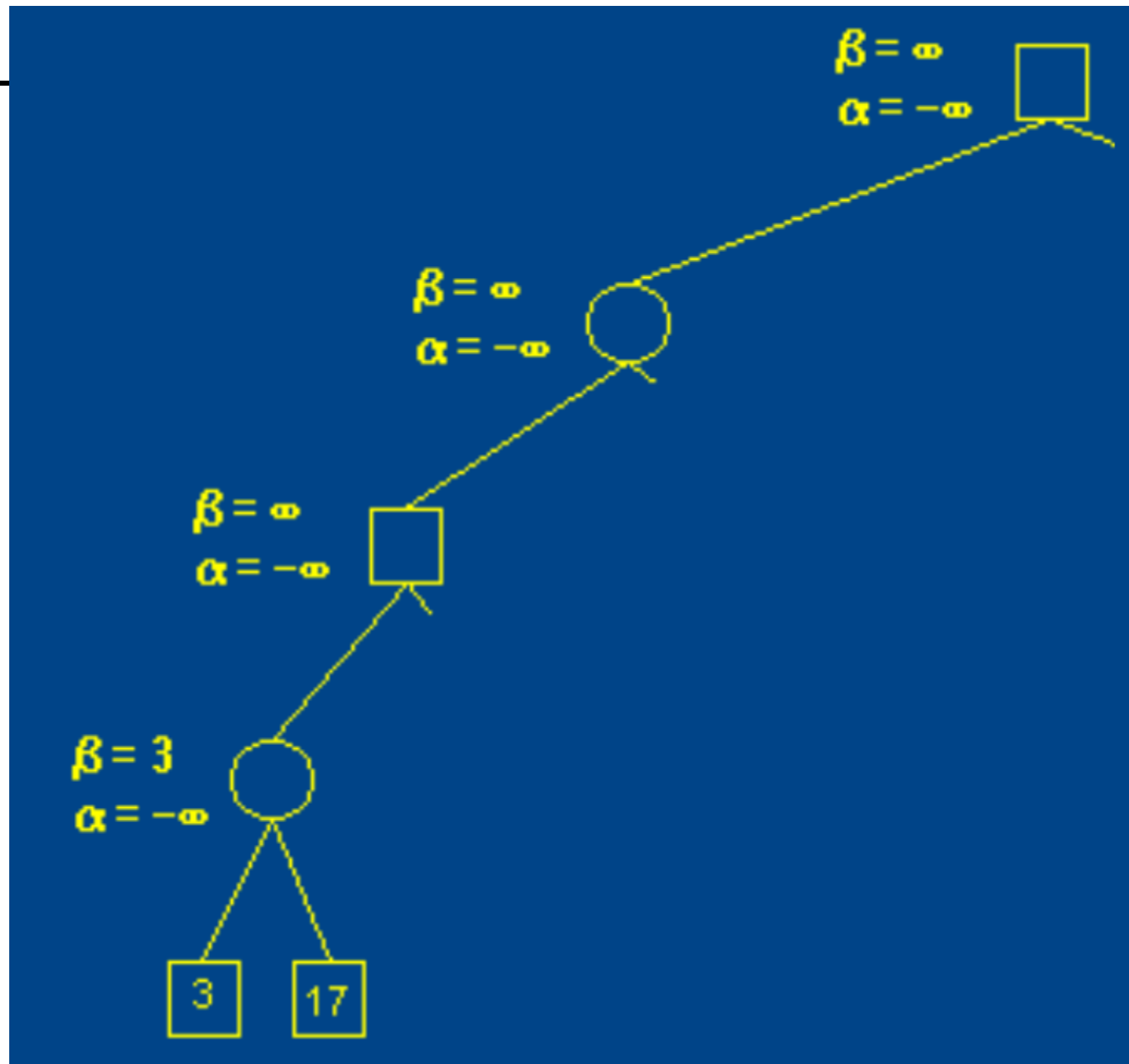
```
        if  $v \leq \alpha$  return  $v$ 
```

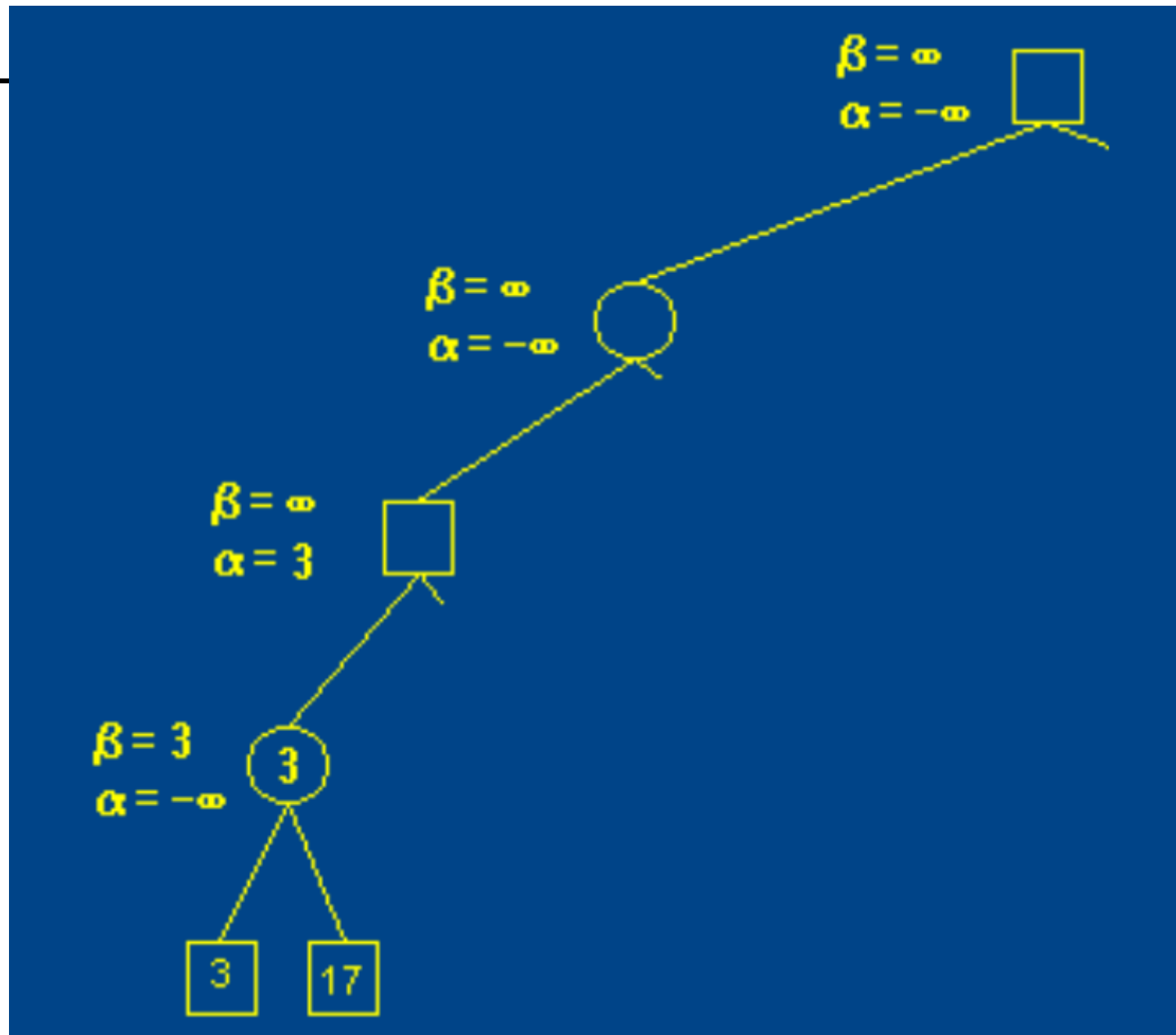
```
         $\beta = \min(\beta, v)$ 
```

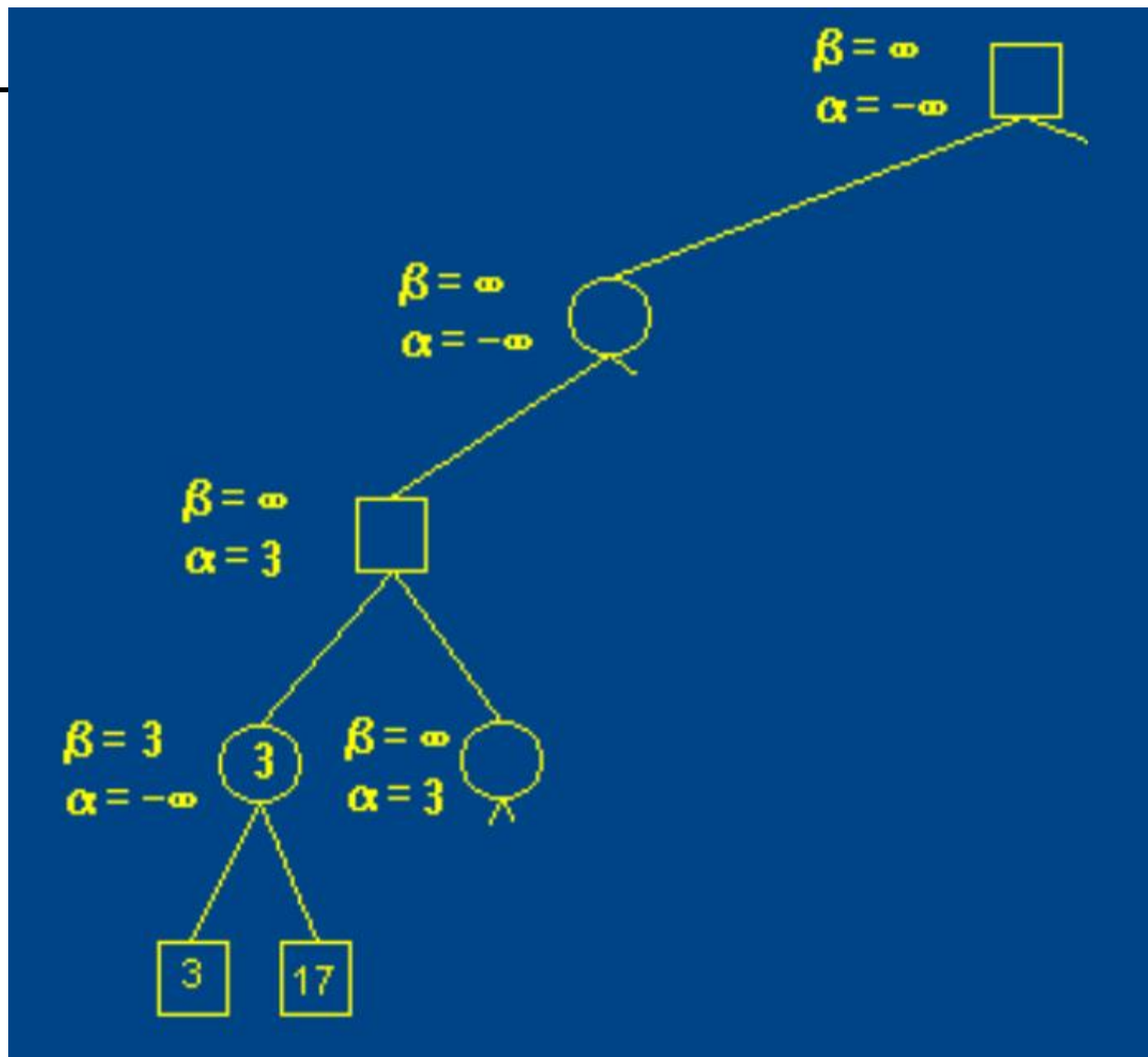
```
    return  $v$ 
```

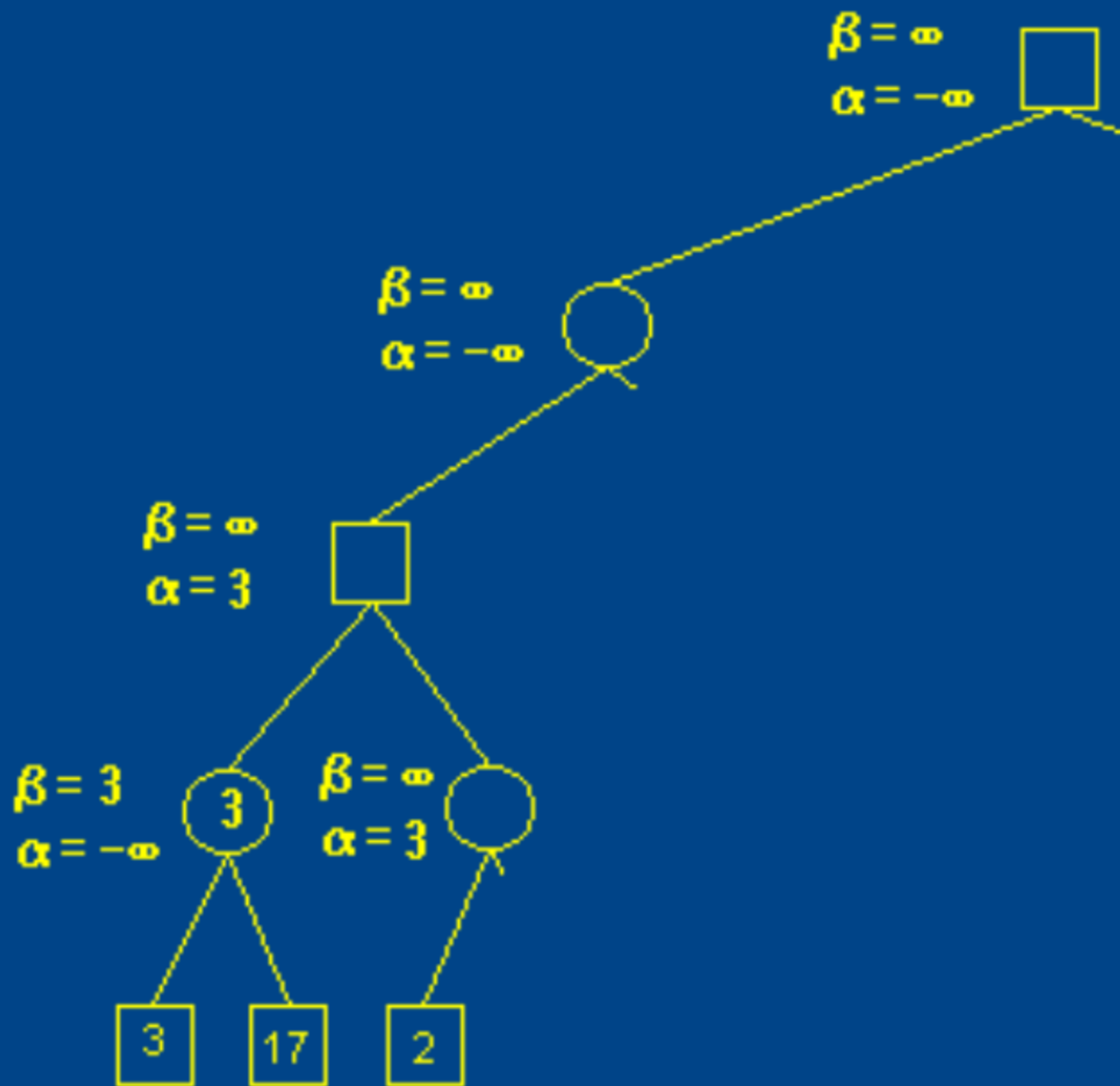


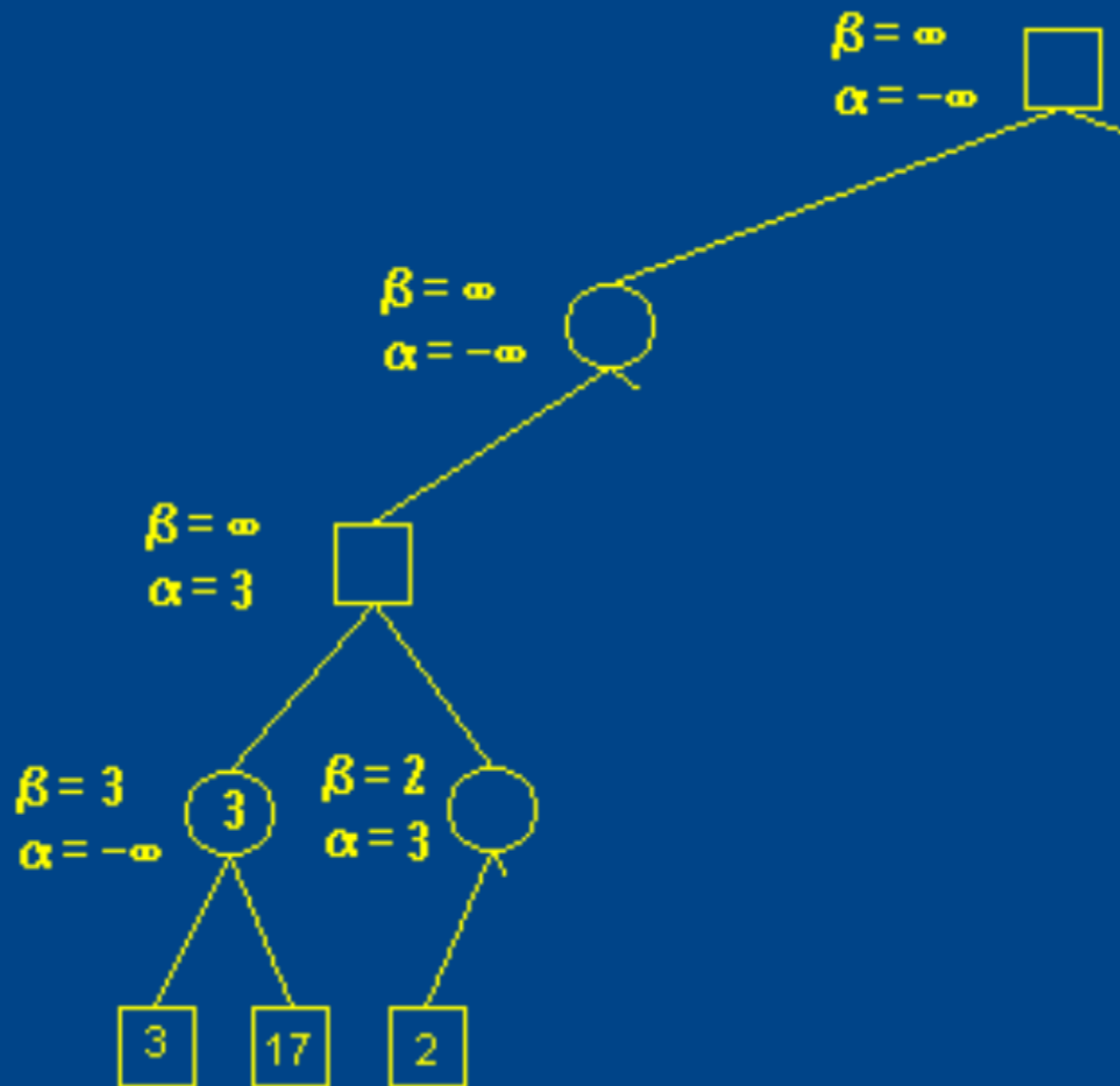


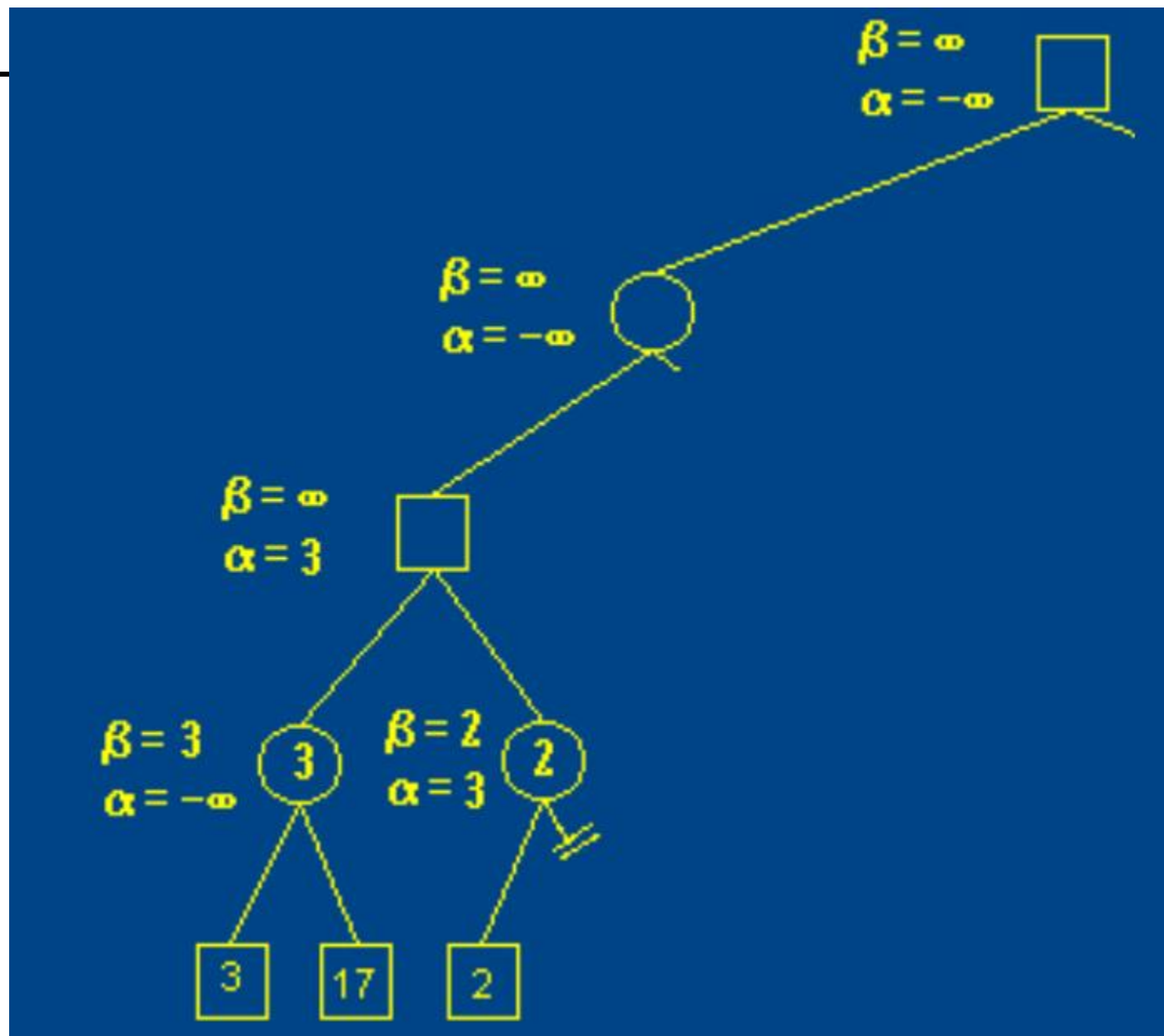


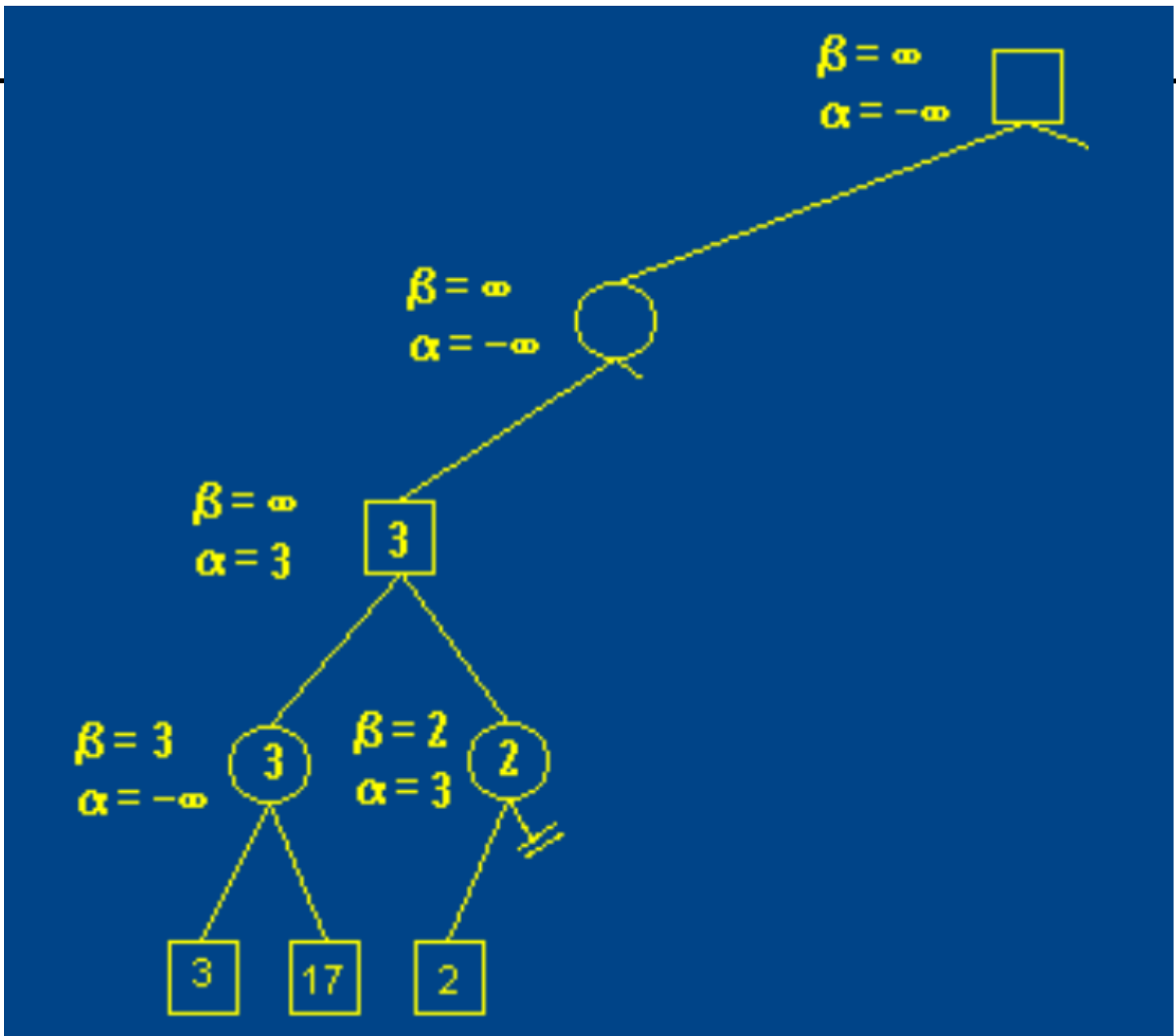


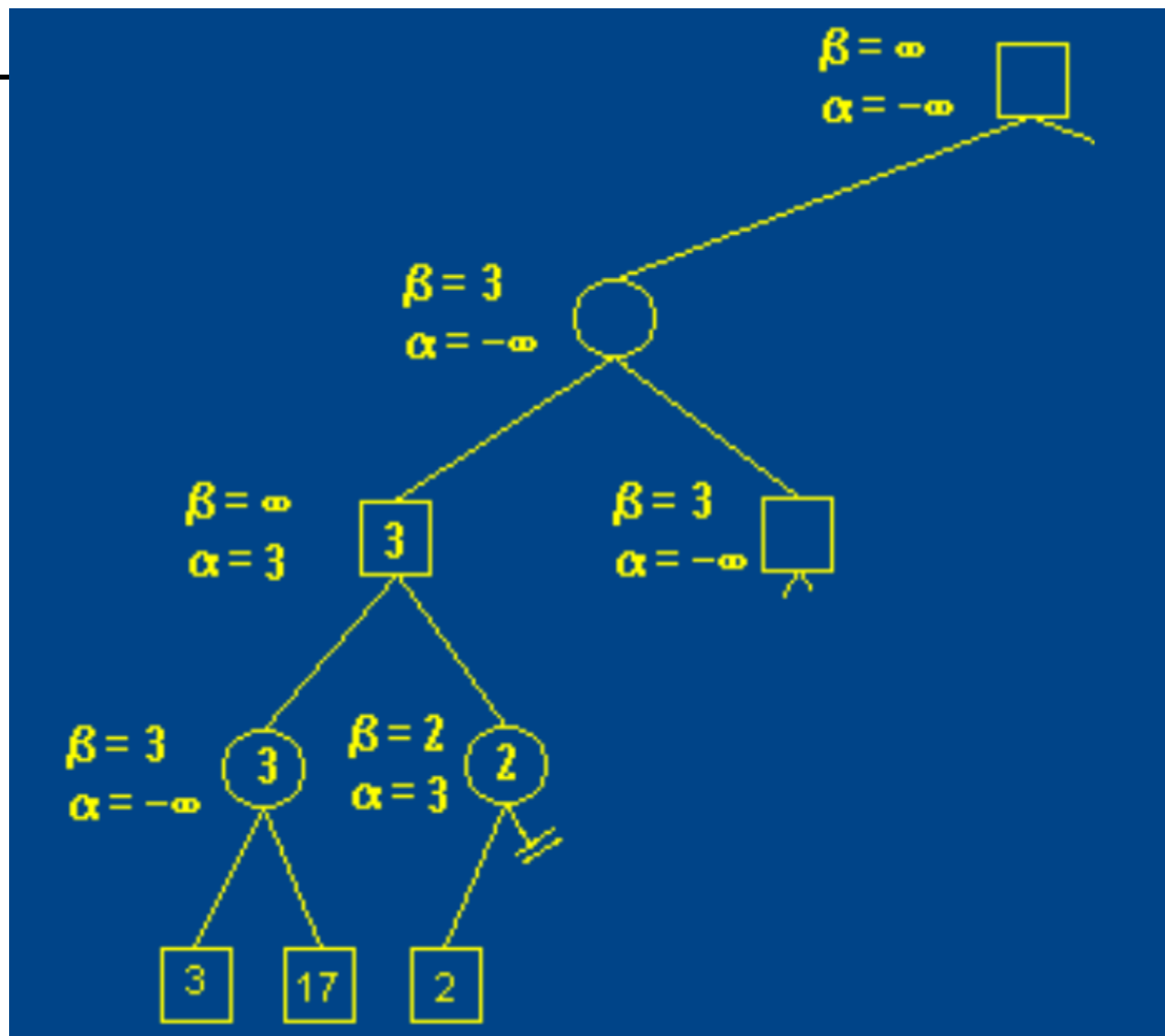


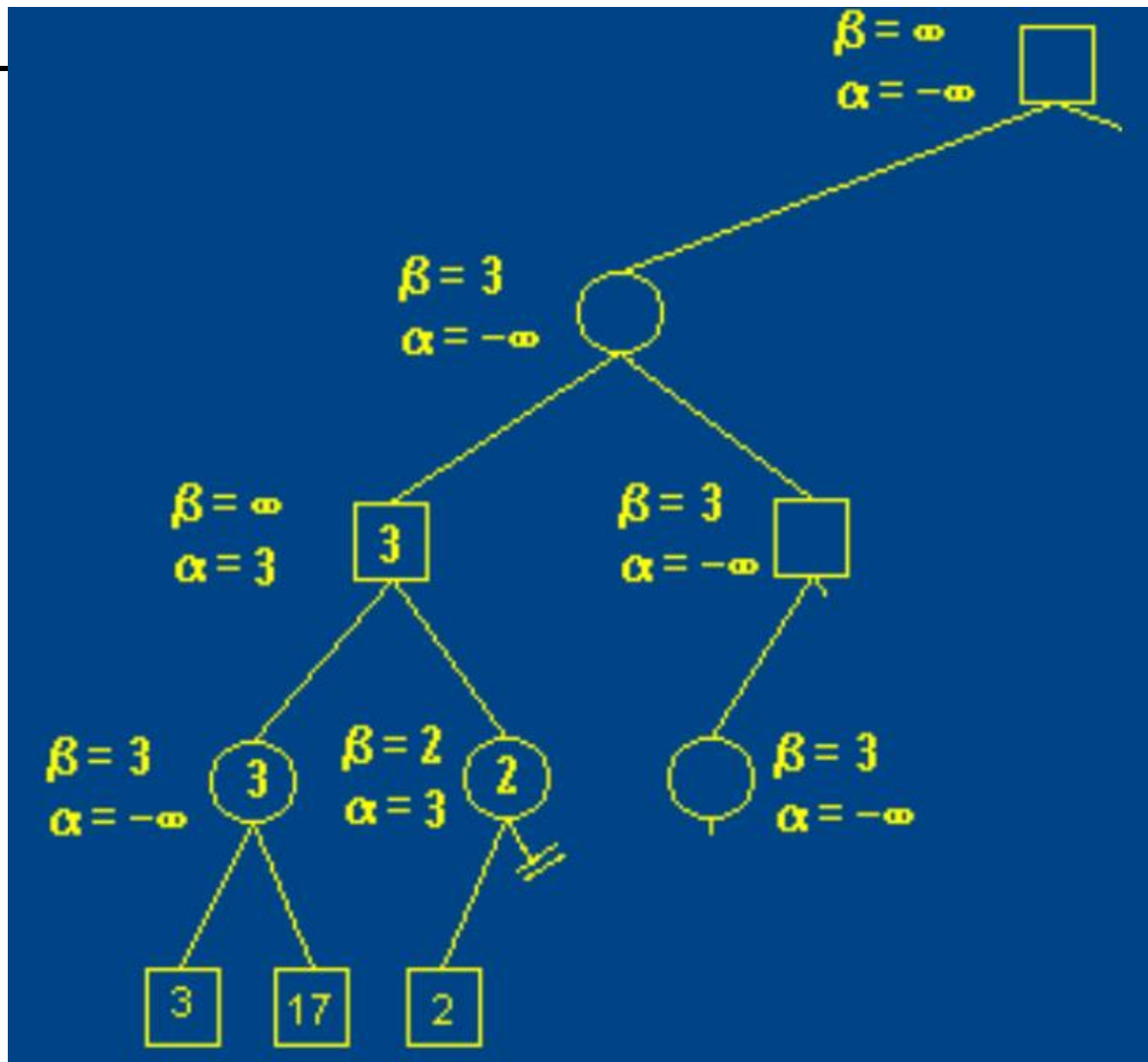


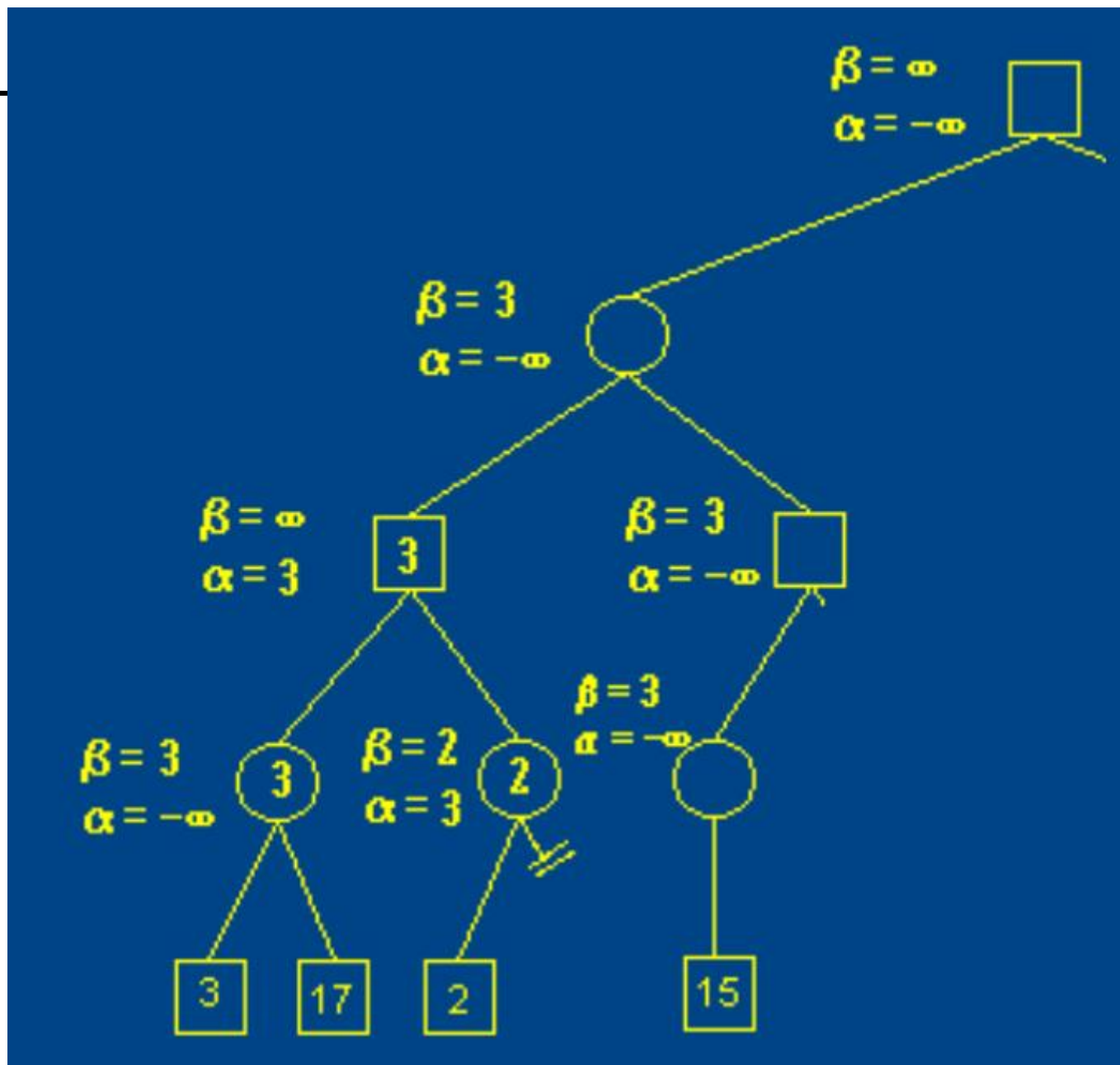


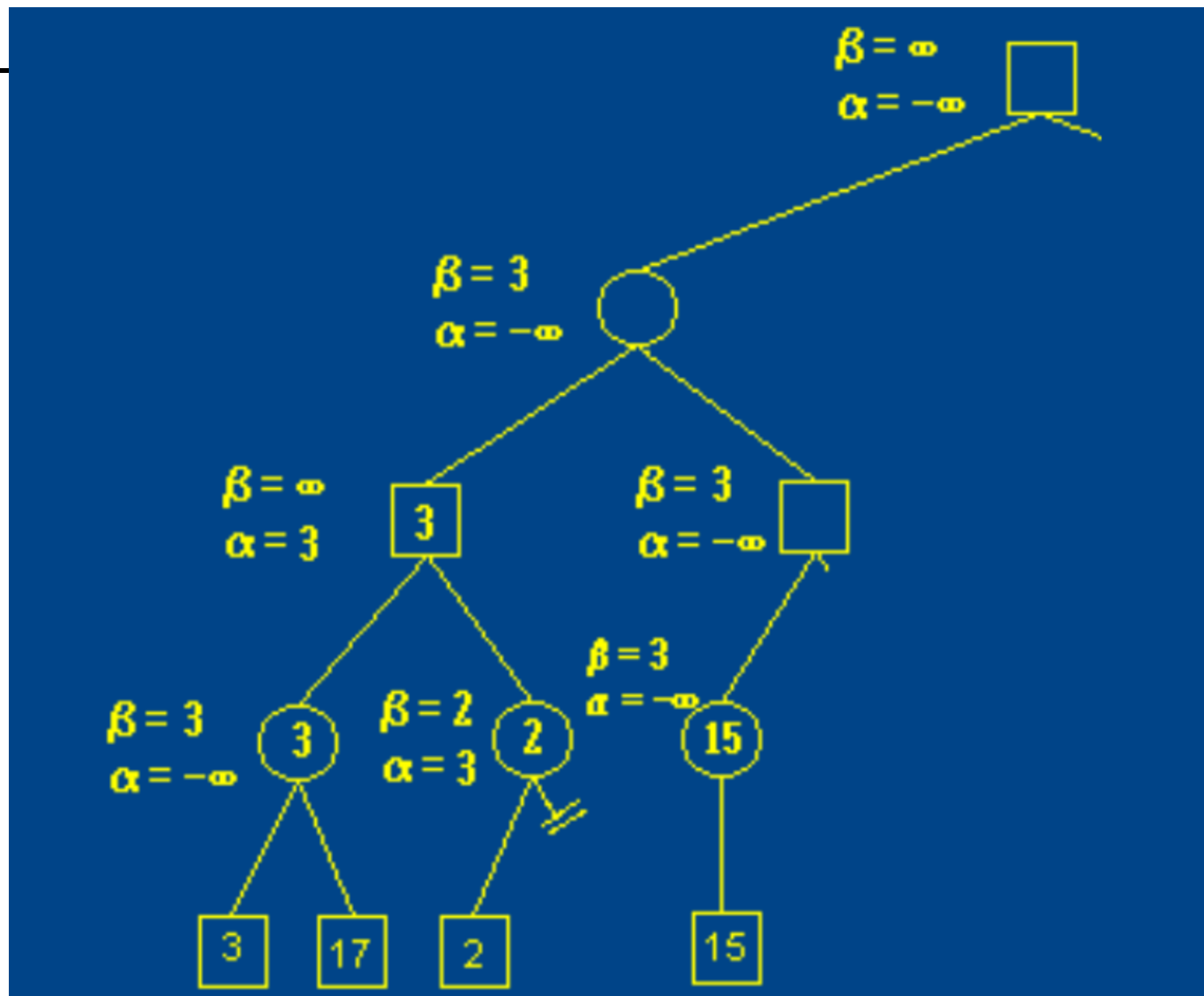


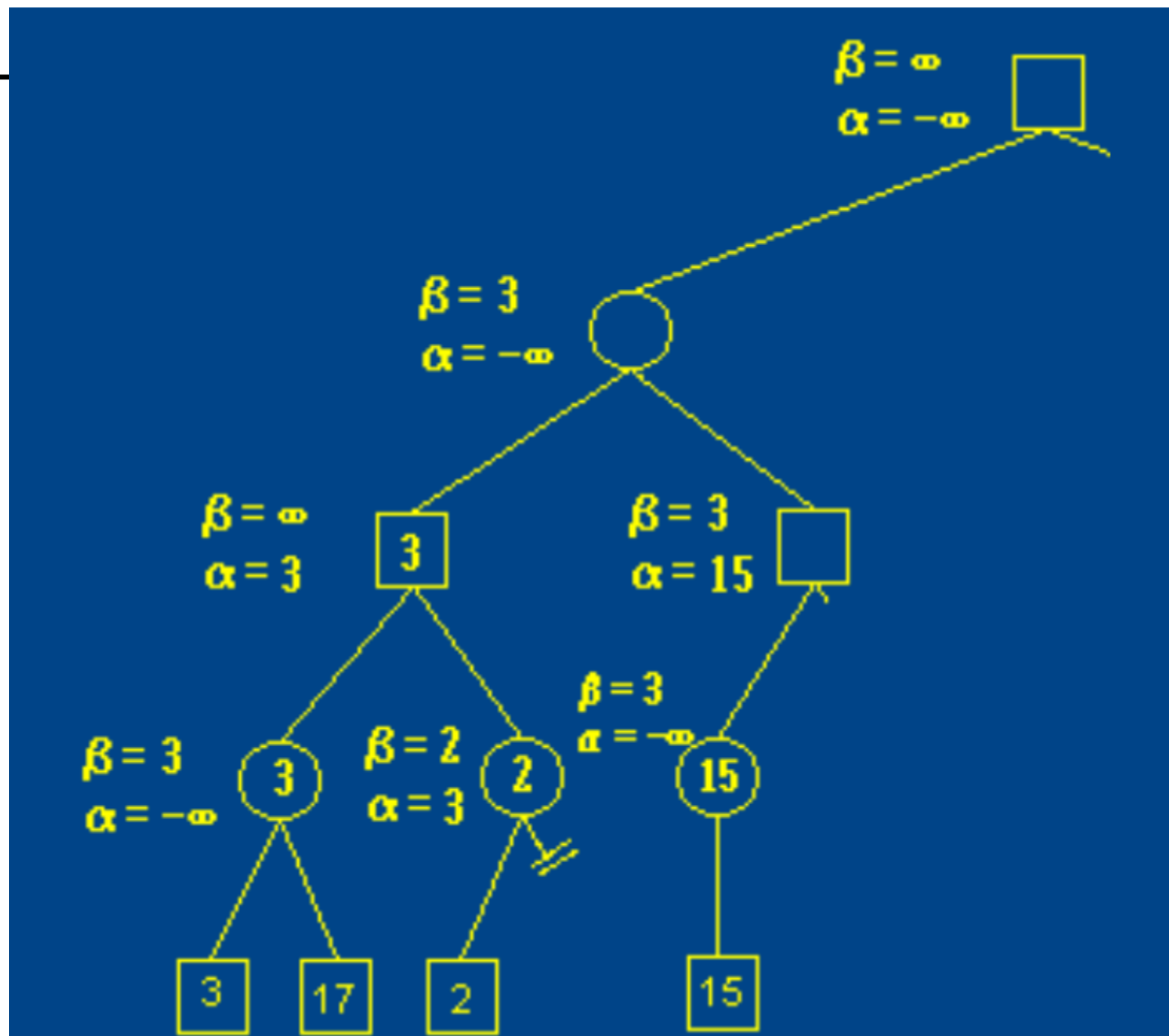


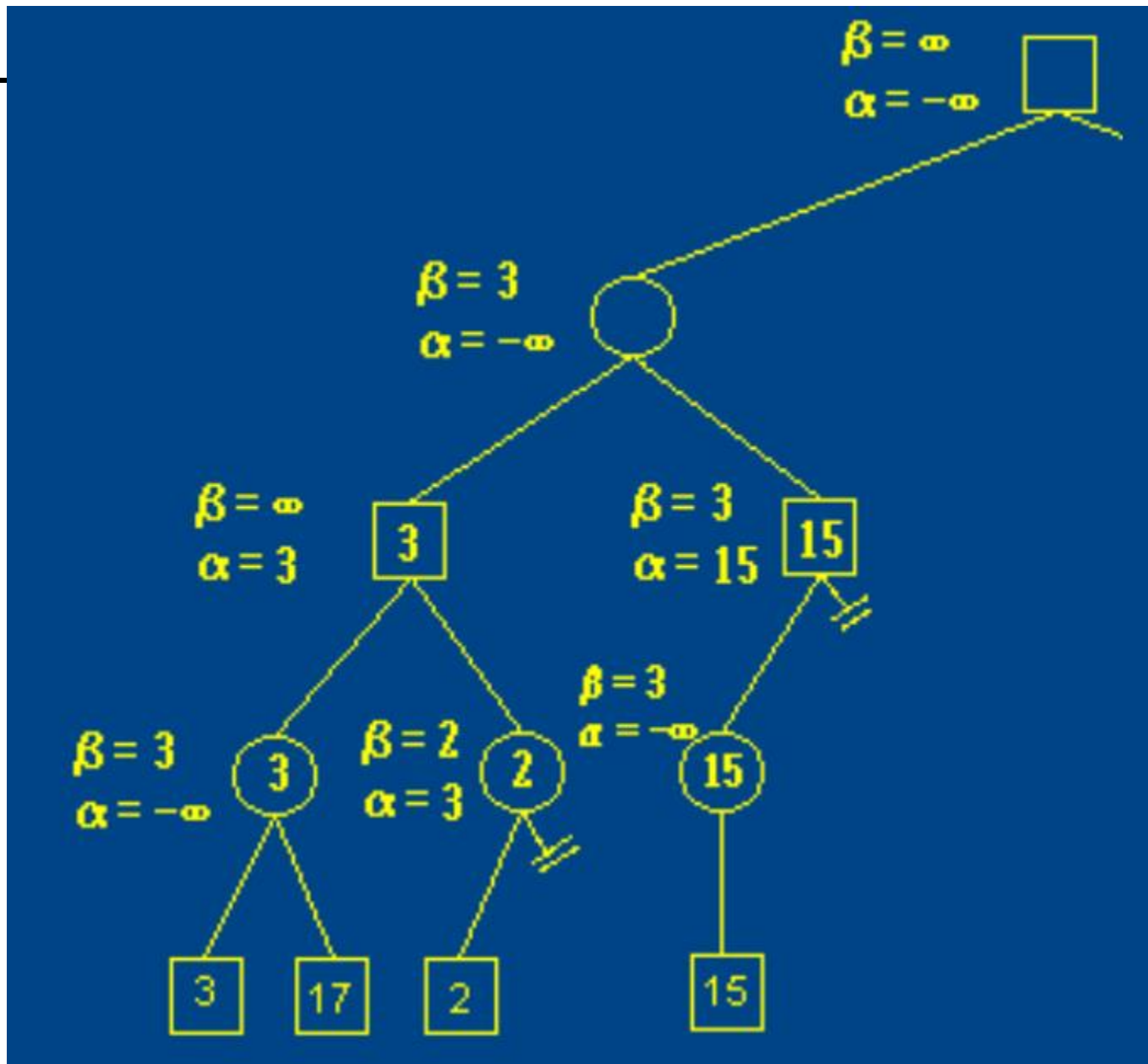


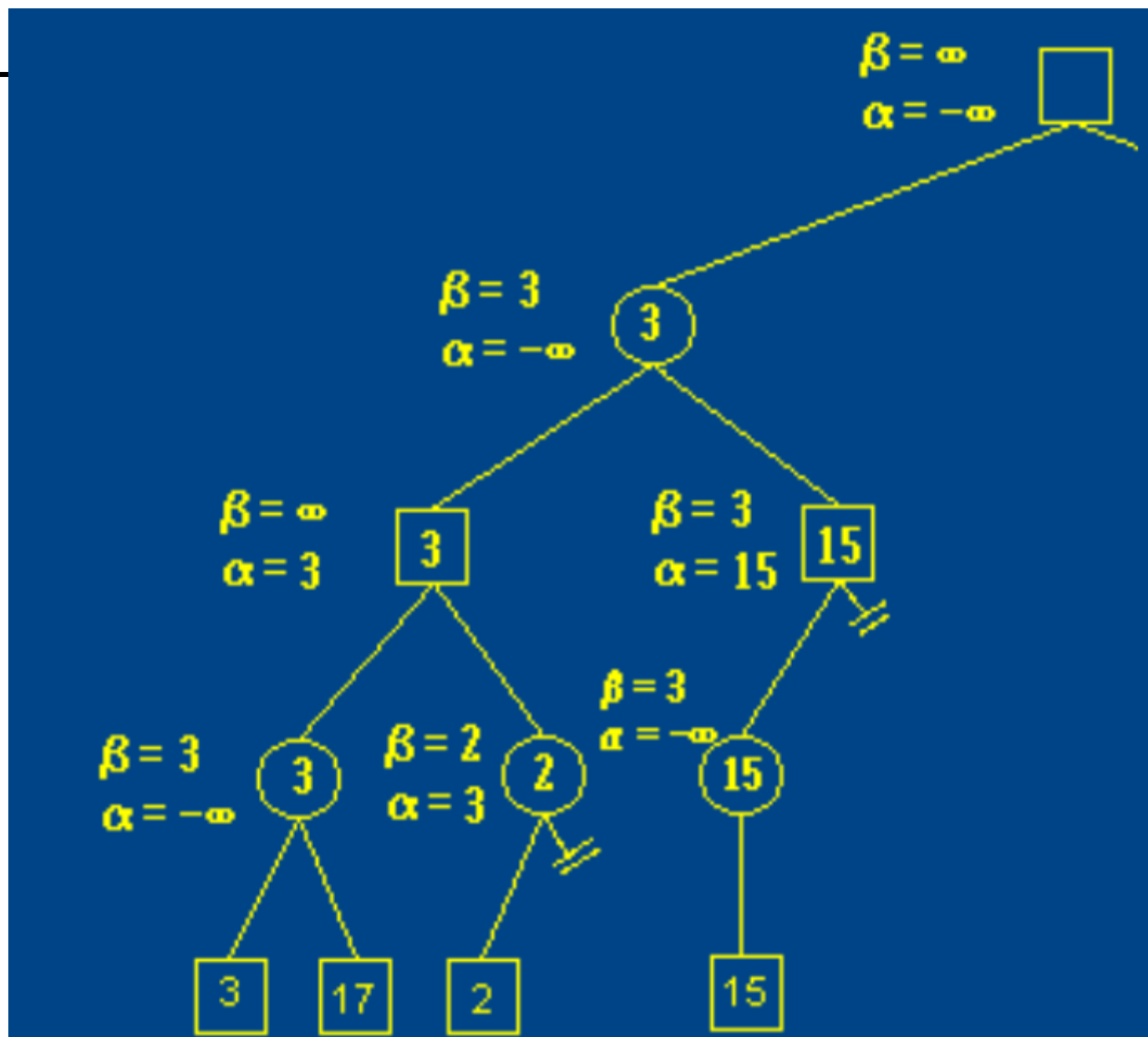


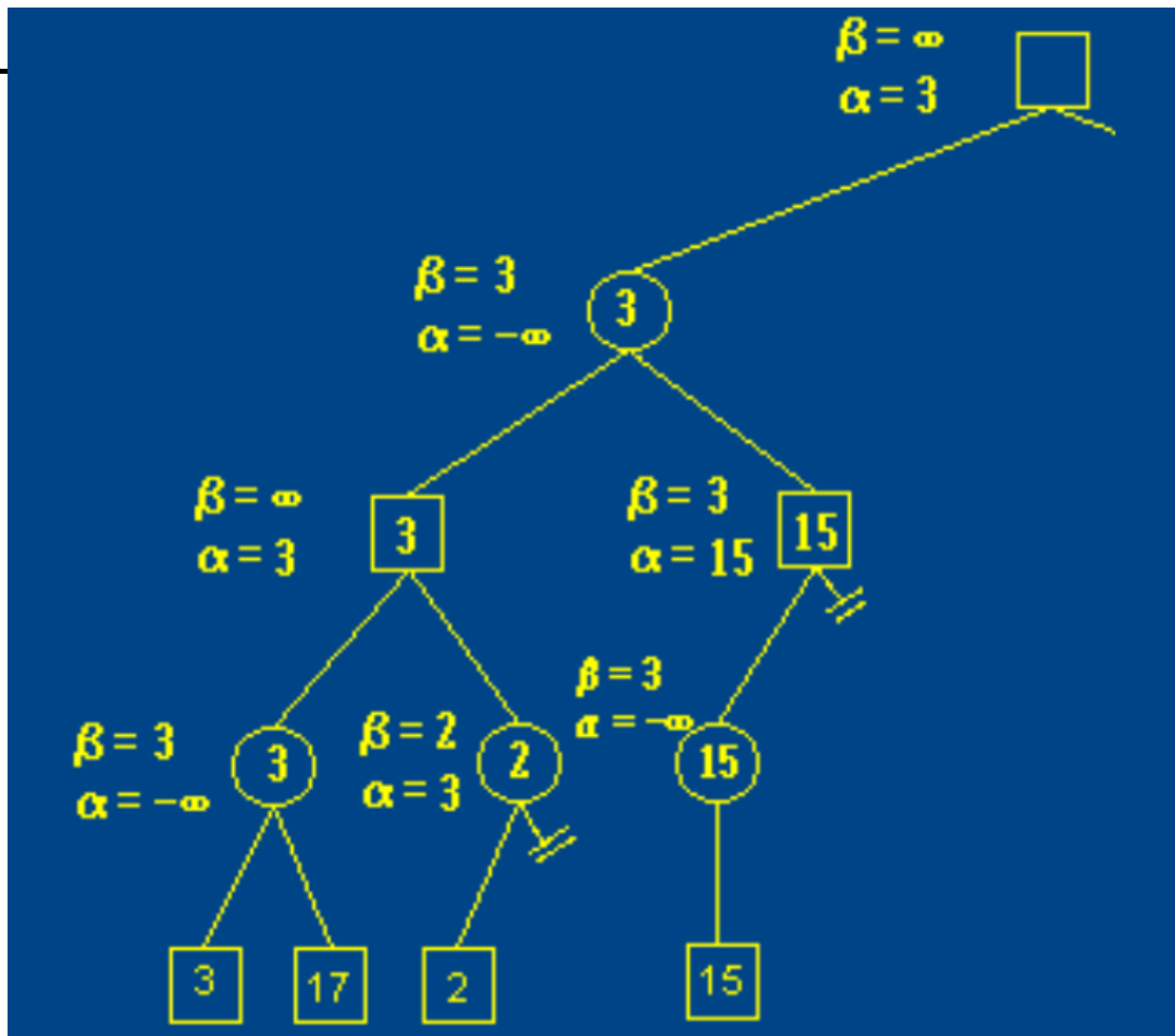


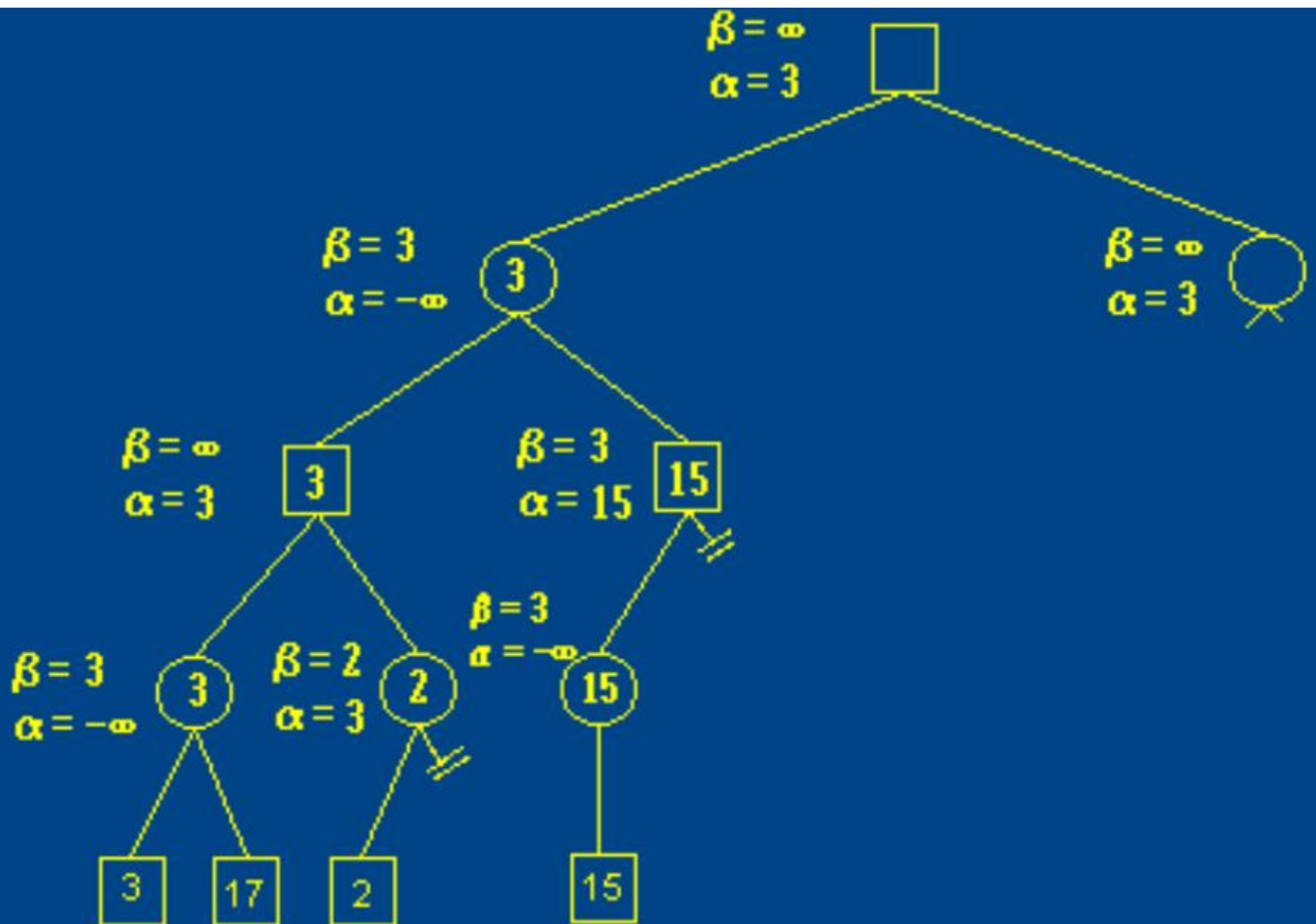


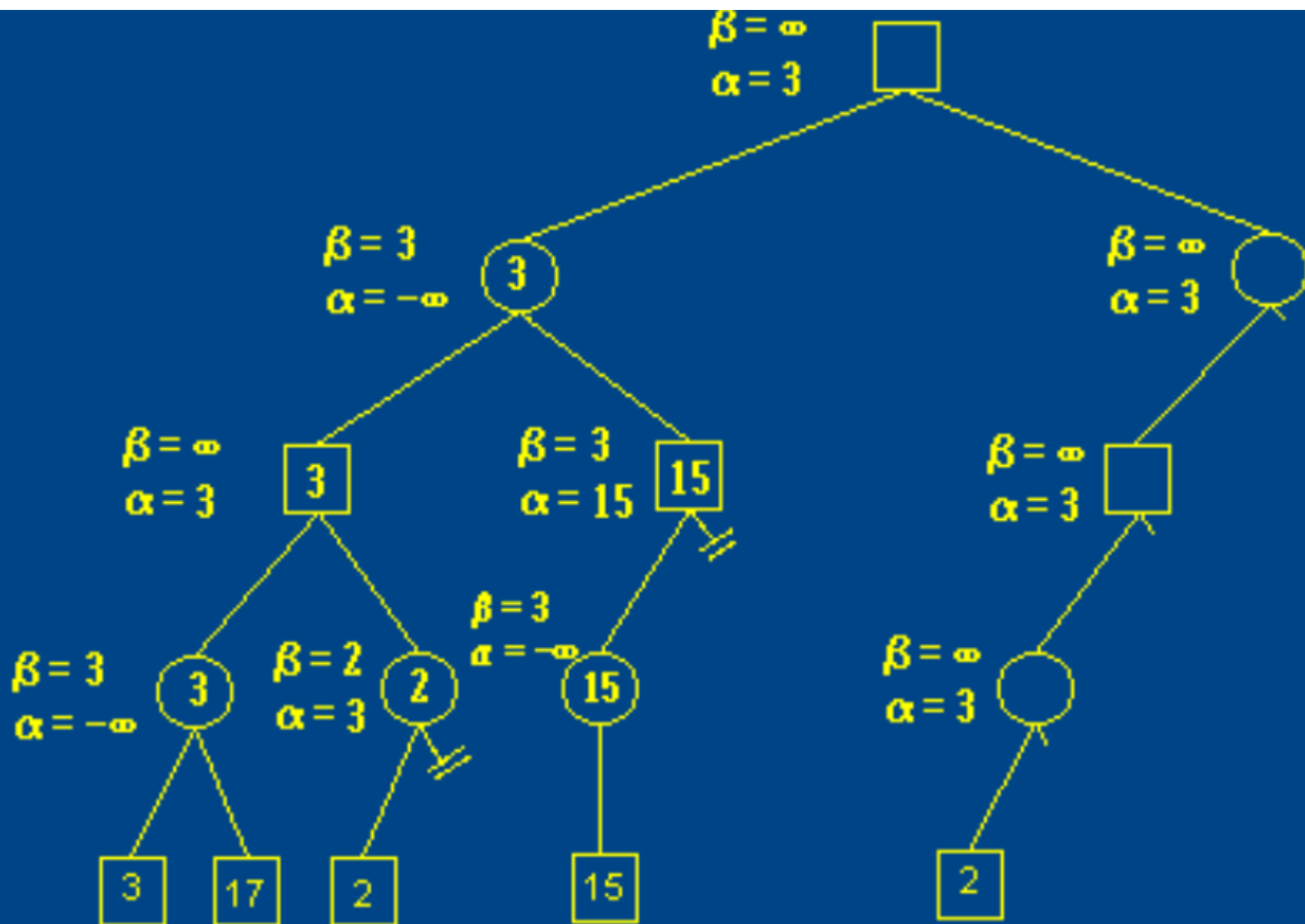




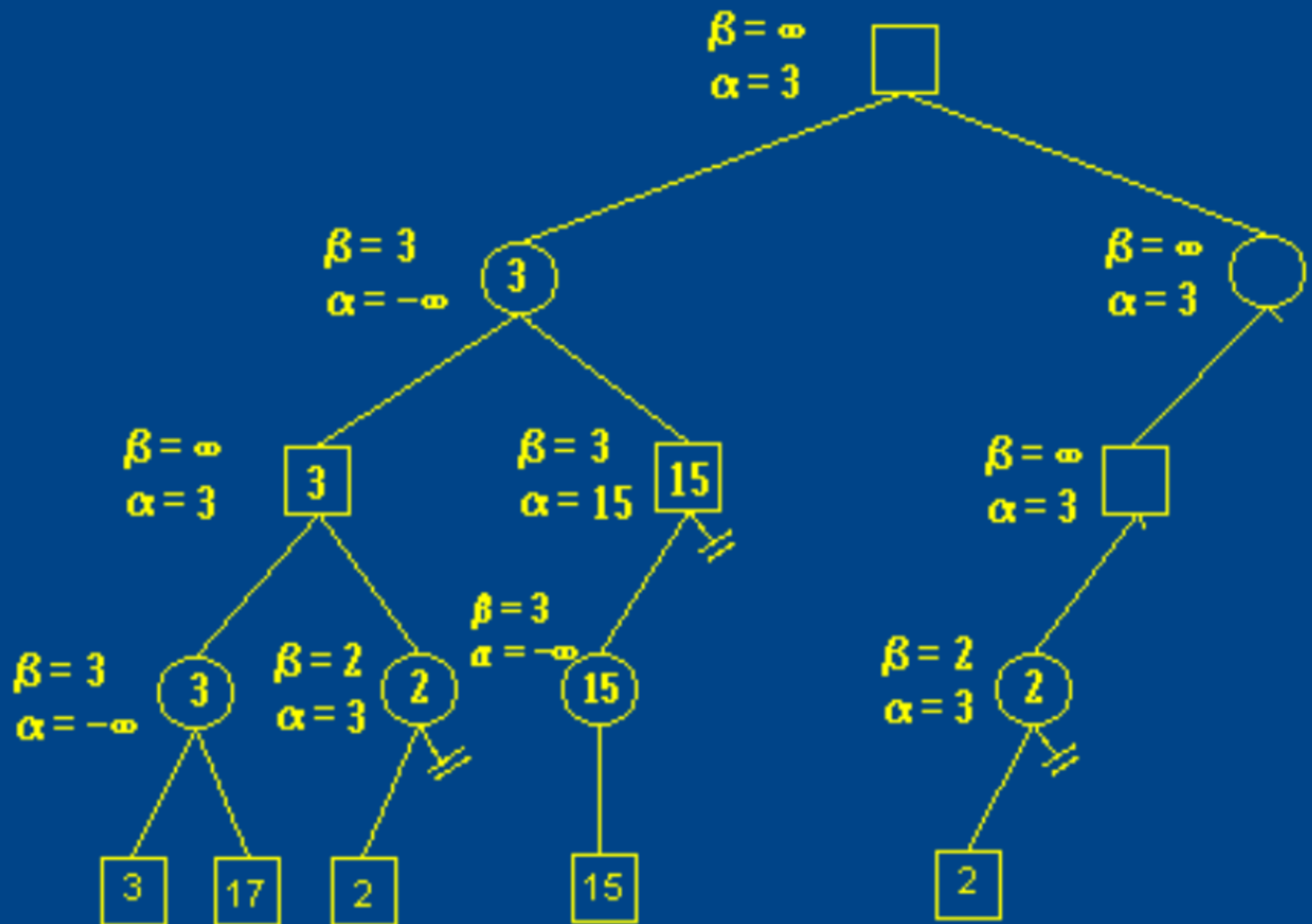


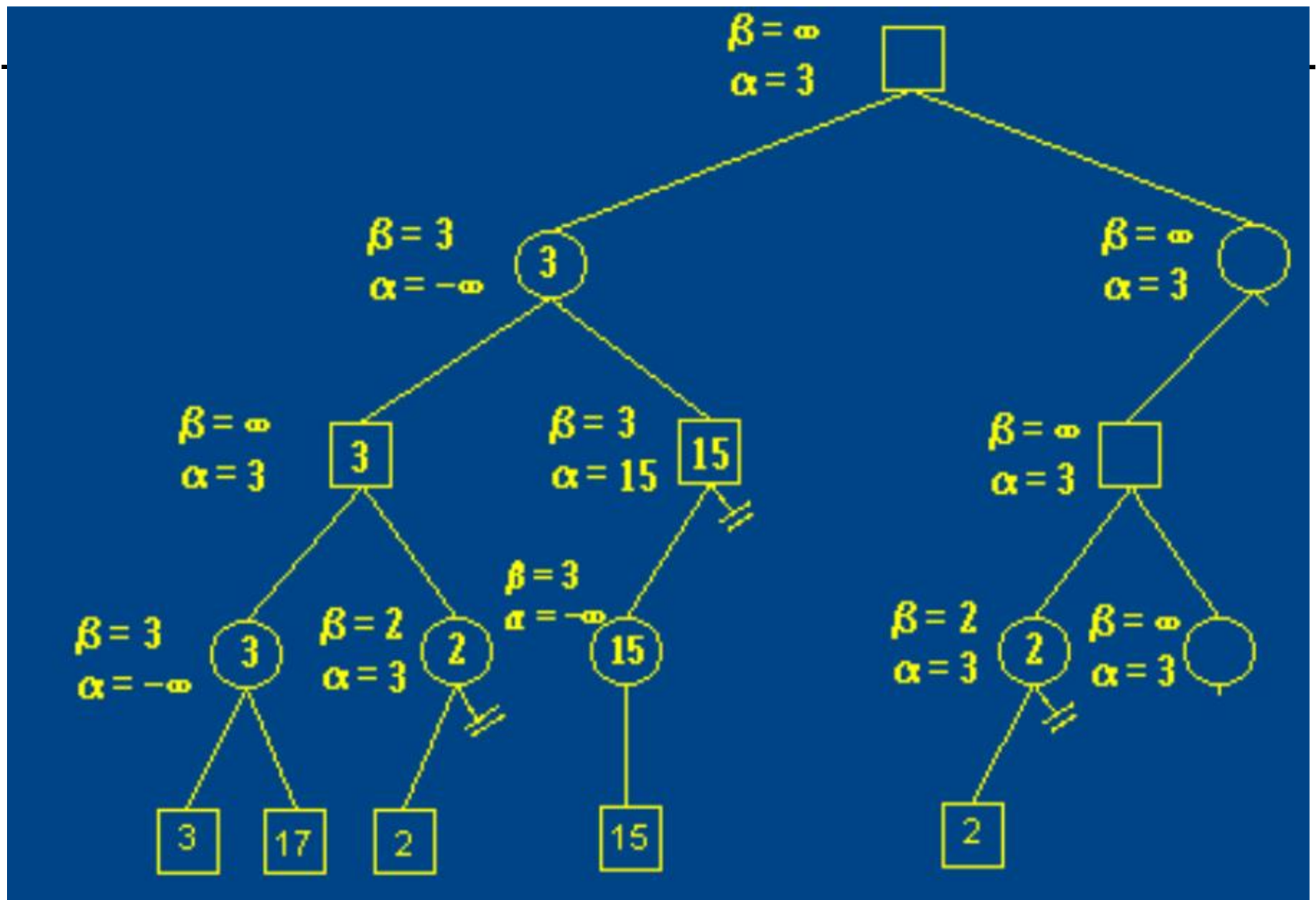






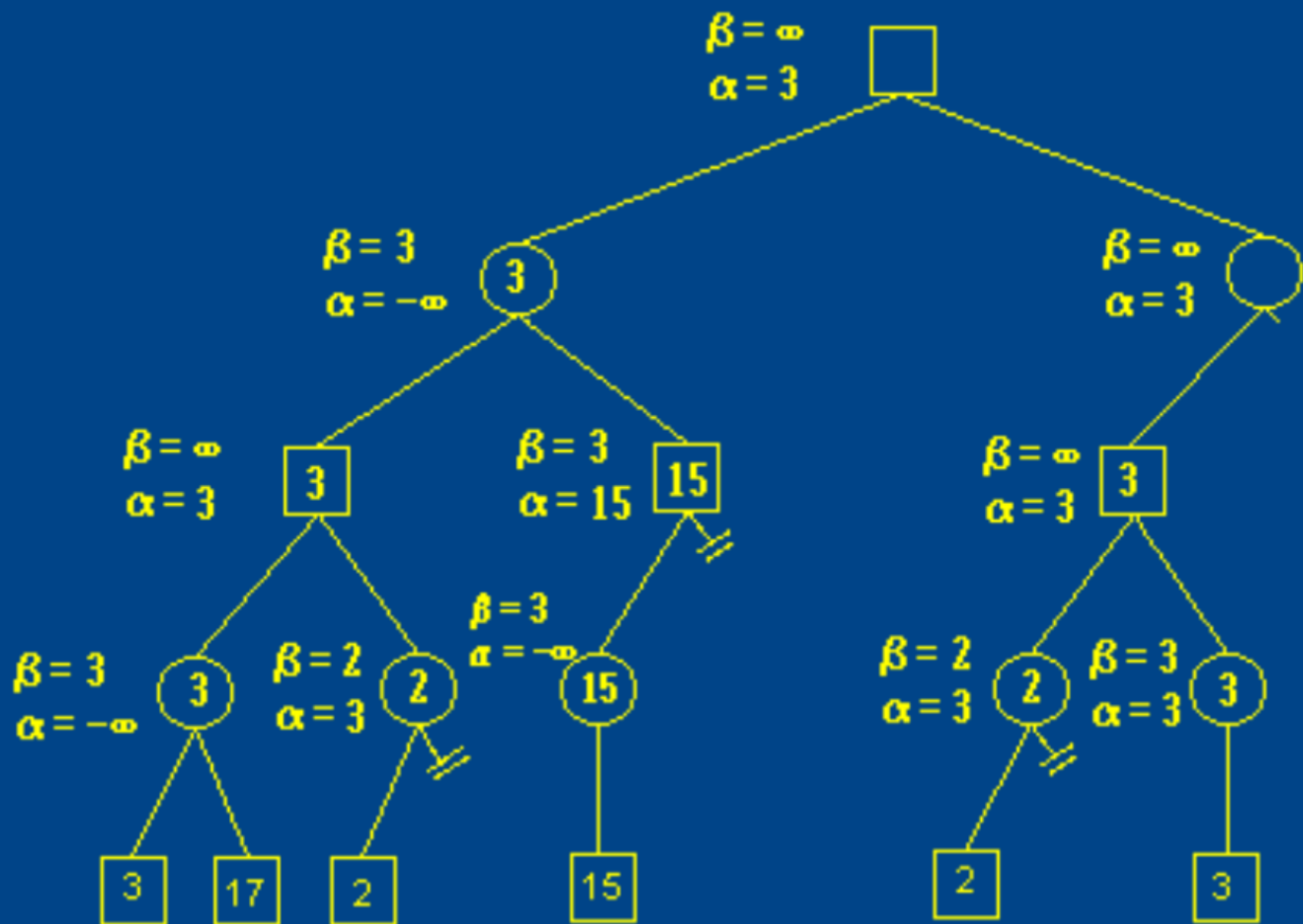


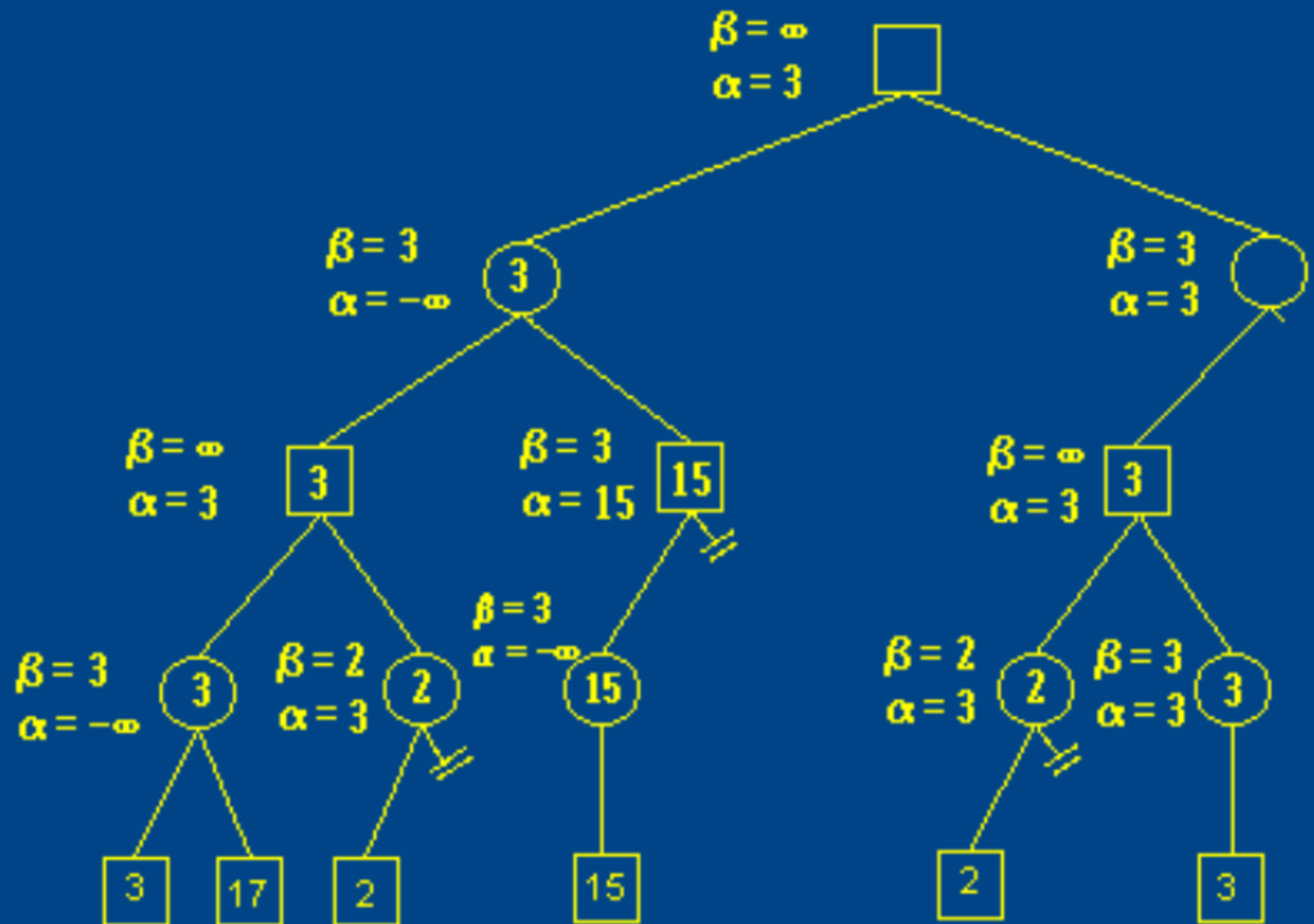






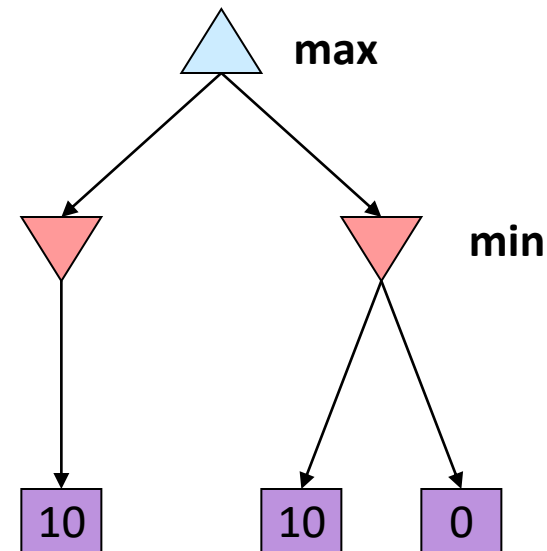




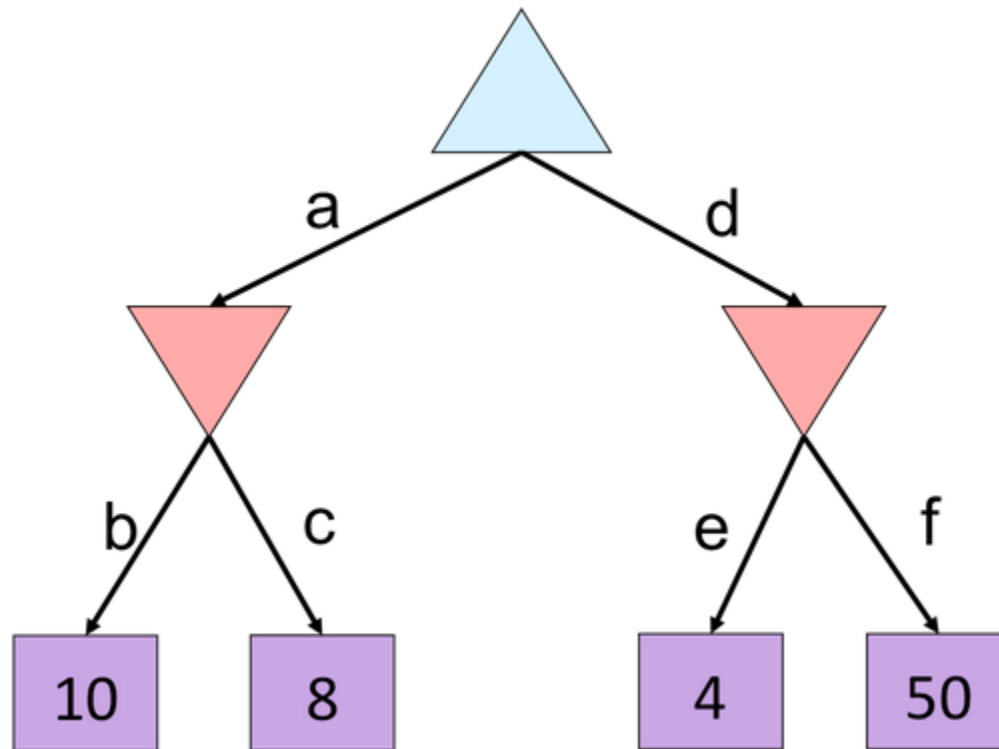


Bình luận về tỉa alpha-beta

- Việc tỉa sẽ không ảnh hưởng gì đến giá trị minimax của trạng thái gốc, nhưng **sẽ ảnh hưởng tới giá trị minimax của các trạng thái bên dưới** → các trạng thái con của trạng thái gốc có thể sẽ có giá trị minimax bị sai → làm việc chọn hành động từ trạng thái gốc có thể bị sai
 - Chạy thử với ví dụ ở bên ...
- Giải pháp?
 - Một giải pháp là tính riêng giá trị minimax cho từng trạng thái con



Bài tập: tìm alpha-beta



Bài tập: tìm alpha-beta

