

# TRÍ TUỆ NHÂN TẠO (ARTIFICIAL INTELLIGENCE)

## CHƯƠNG 3

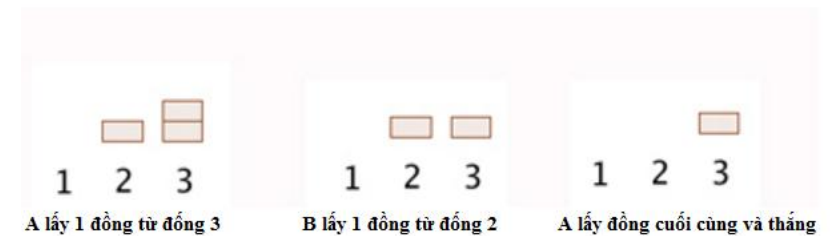
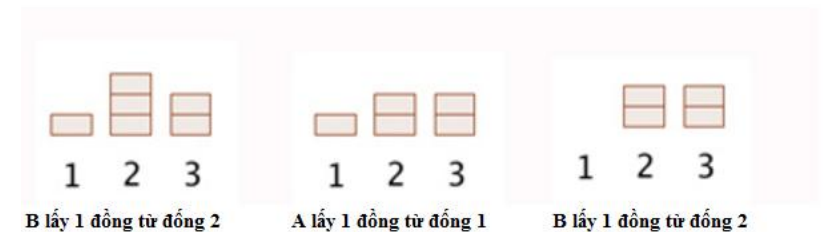
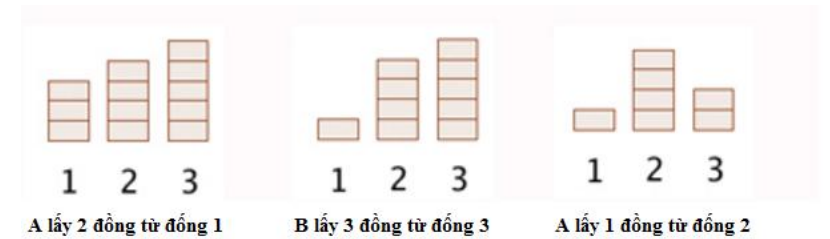
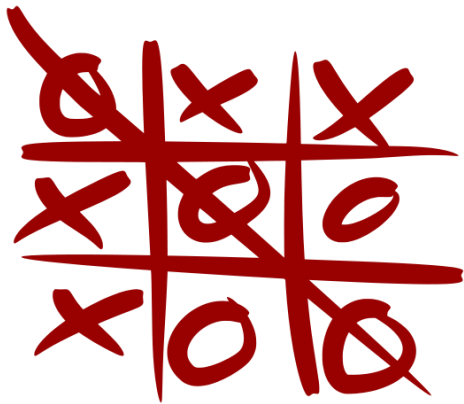
# GIẢI QUYẾT VẤN ĐỀ BẰNG TÌM KIẾM

Trần Nguyễn Minh Thư  
tnmthu@ctu.edu.vn

# Nội dung

- Biểu diễn bài toán trong KGTT
- Tìm kiếm mù (uninformed search)
- Tìm kiếm heuristic (informed search)
- **Tìm kiếm đối kháng (cây trò chơi)**

# Sự khác biệt giữa các bài toán?



# Ứng dụng Heuristic trong các trò chơi

- Sử dụng không gian trạng thái để giải quyết bài toán
  - *Tìm kiếm mù*
  - *Tìm kiếm heuristic – có thông tin bổ sung*
  - *Sử dụng heuristic cho trò chơi*
- *Có 2 người* tham gia vào quá trình sinh trạng thái
- *Bạn* tạo ra trạng thái này, *đối thủ* của bạn sẽ tạo ra trạng thái kế tiếp với mong muốn đánh bại bạn

# Cây trò chơi Tic-Tac-Toe



MAX (X)



MIN (O)



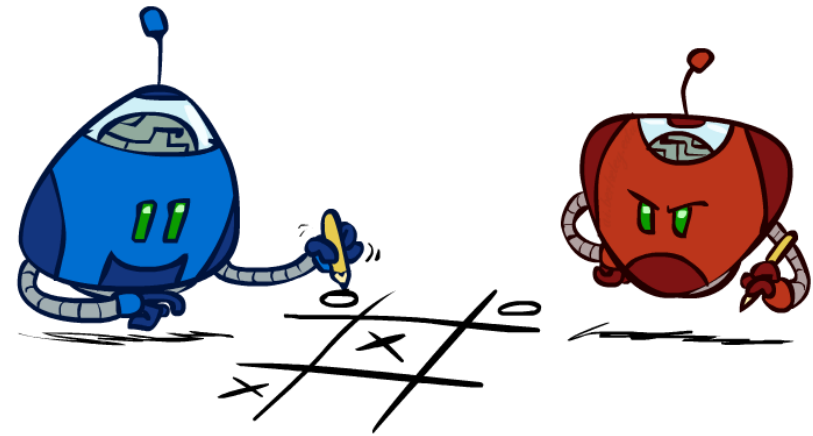
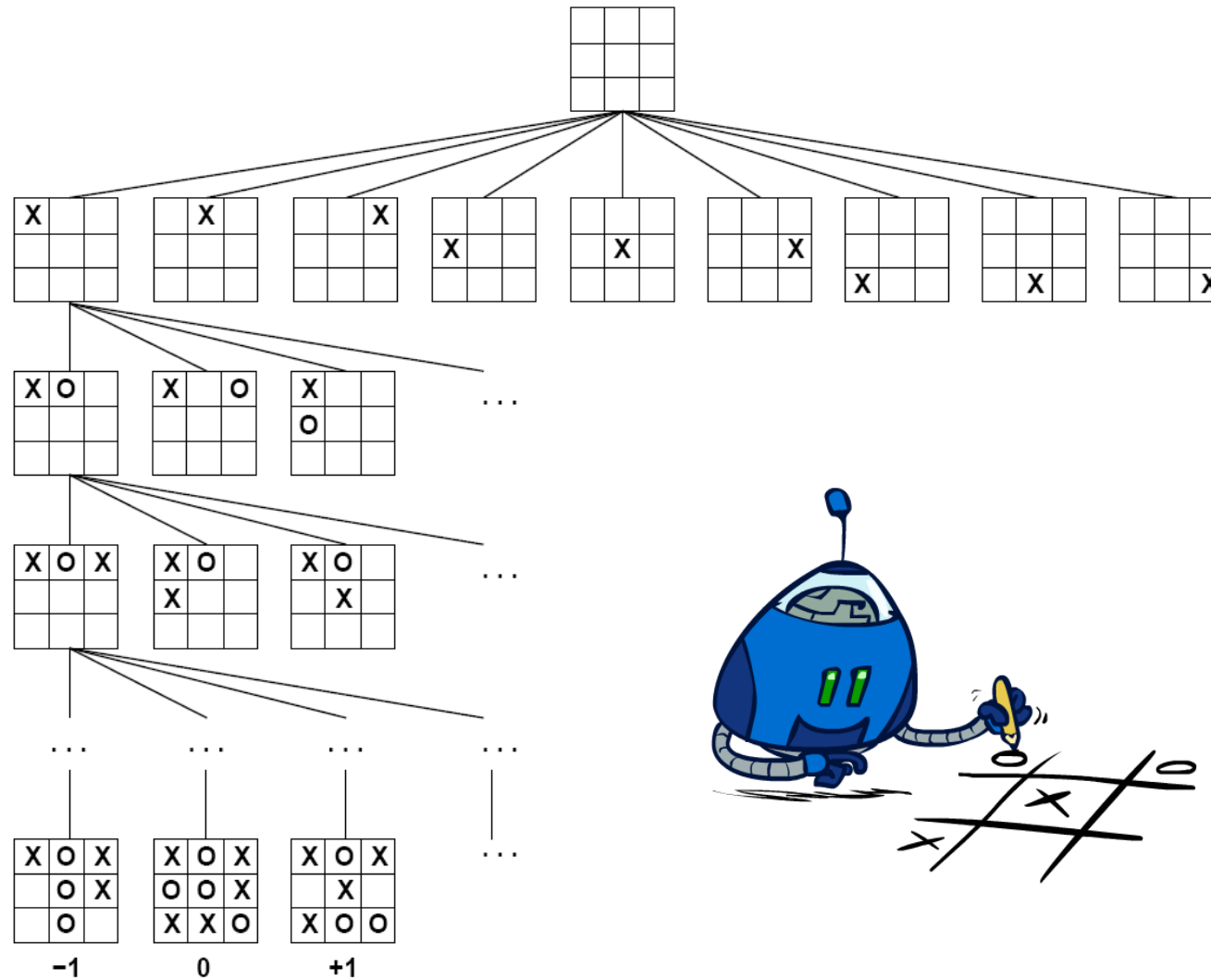
MAX (X)



MIN (O)

TERMINAL

Utility

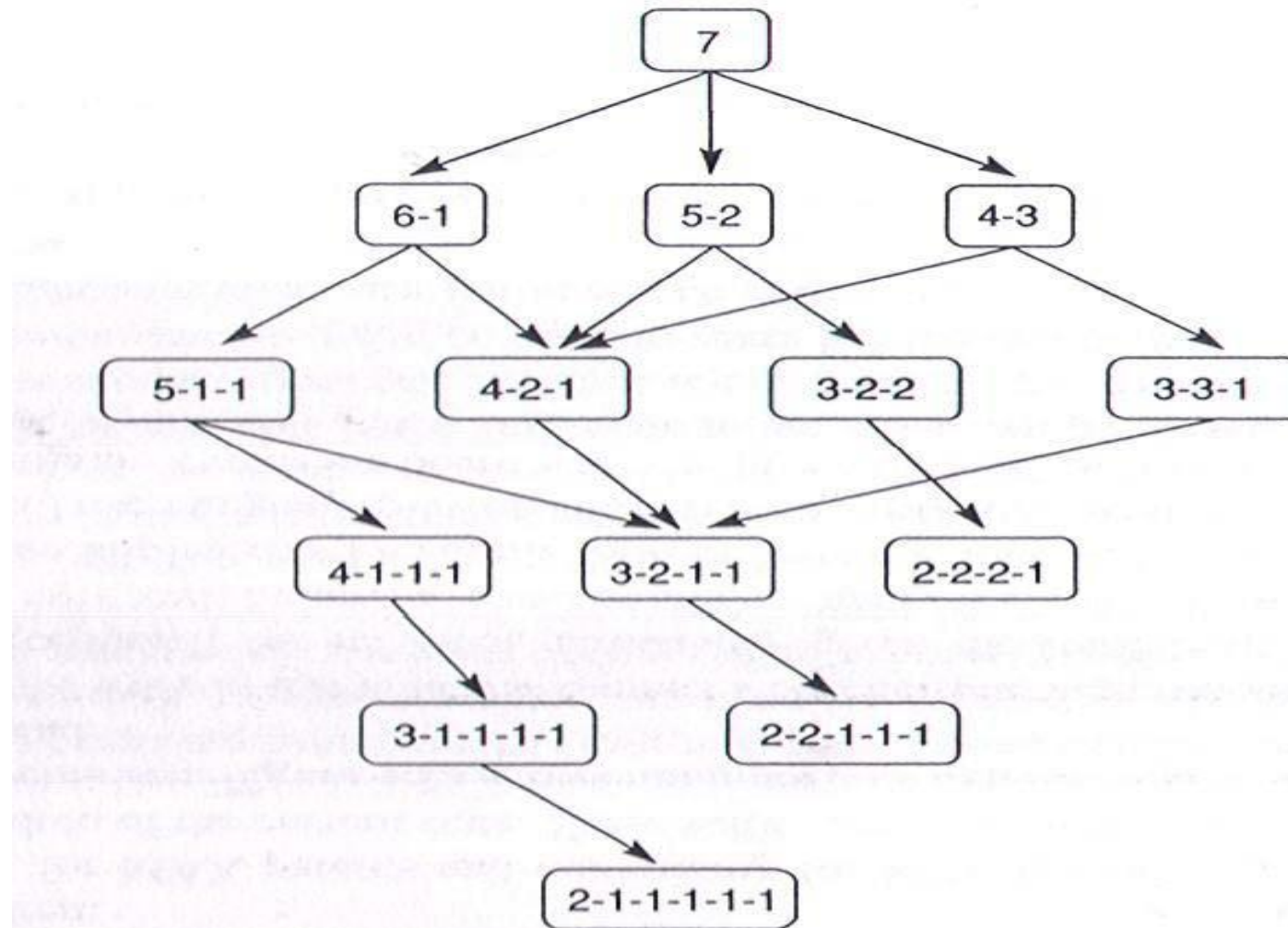


# Ứng dụng Heuristic trong các trò chơi

## ■ Trò chơi Nim:

- Một số token (đồng xu, mảnh gỗ...) được đặt giữa 2 đối thủ.
- Ở mỗi lượt đi, người chơi phải chia các token **thành 2 phần (không rỗng) với số lượng khác nhau**. VD: 6 token có thể được chia thành 5 - 1, 4 - 2 (trường hợp 3 - 3 là không hợp lệ)
- Khi các token không thể được chia một cách hợp lệ ở lượt chơi kế tiếp, người chơi thuộc về lượt đi đó sẽ thua cuộc.

# Không gian trạng thái trò chơi Nim



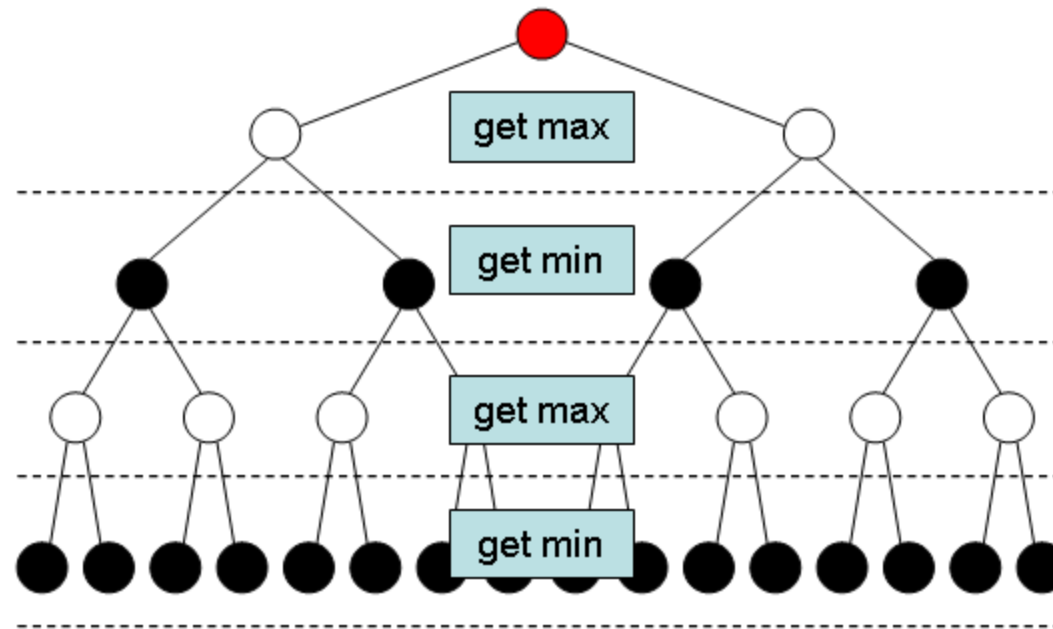
# Giải thuật minimax (1)

## ■ Giải thuật minimax:

- Một đấu thủ trong trò chơi được gọi là **MIN** và đấu thủ còn lại là **MAX**. Max đại diện cho người chơi luôn muốn chiến thắng, Min đại diện cho người chơi cố gắng cho người max giành số điểm càng thấp càng tốt
- Giá trị của nút lá:
  - 1 nếu là MAX thắng,
  - 0 nếu hòa
  - -1 nếu là MIN thắng (MAX thua)
- Minimax sẽ truyền các giá trị này lên cao dần trên đồ thị, qua các nút cha kế tiếp theo các luật sau:
  - Nếu trạng thái cha là **MAX**, gán cho nó giá trị **lớn nhất** có trong các trạng thái con.
  - Nếu trạng thái cha là **MIN**, gán cho nó giá trị **nhỏ nhất** có trong các trạng thái con.

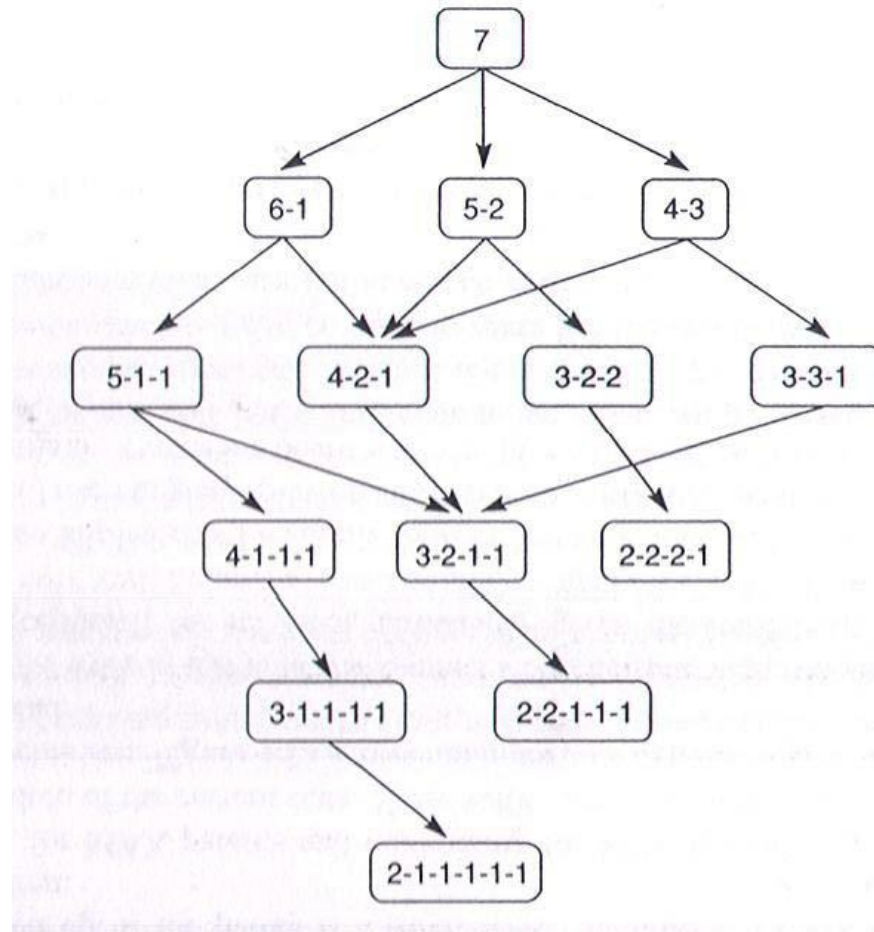


# Giải thuật minimax (2)



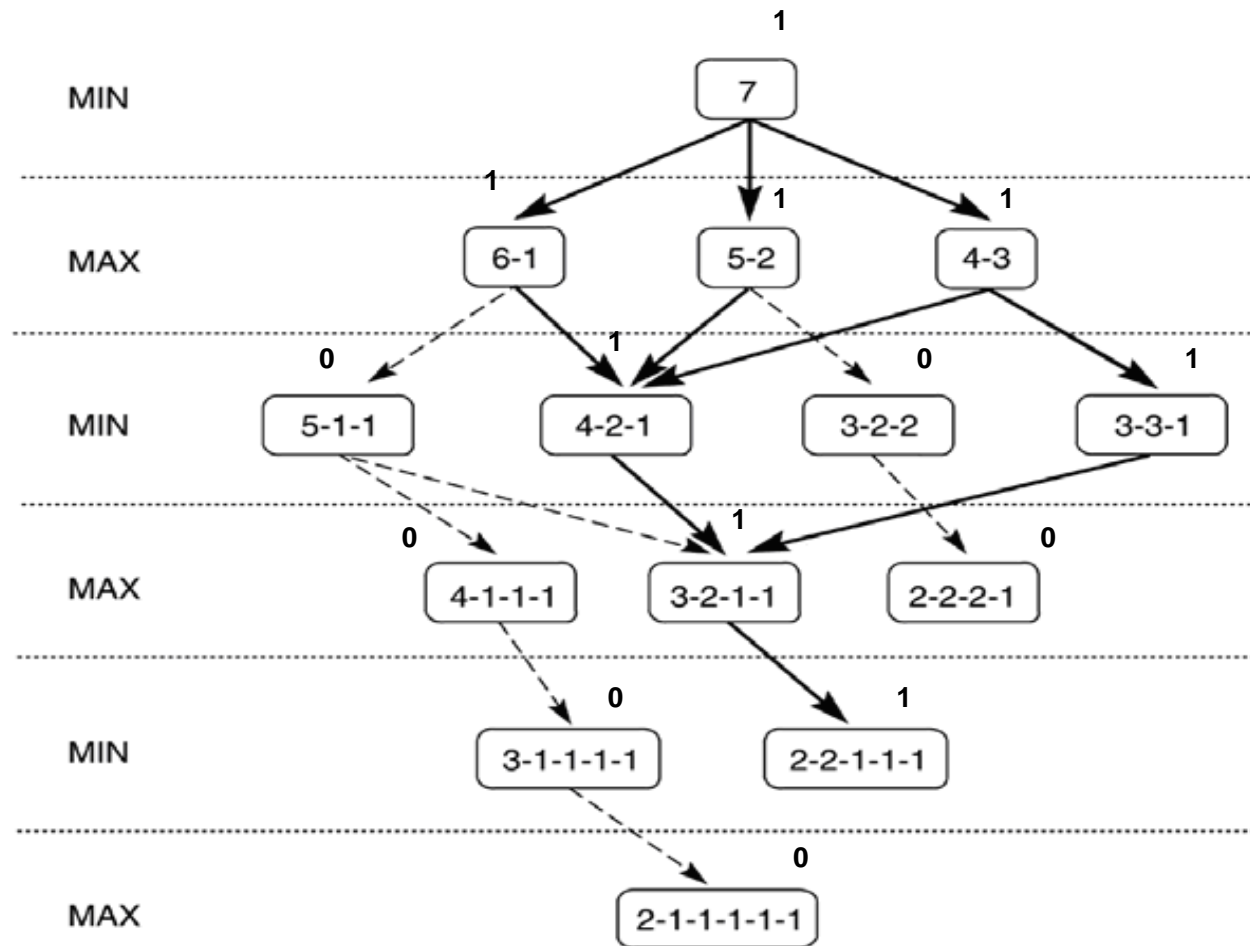
# Giải thuật minimax

- Không gian trạng thái trò chơi Nim



# Giải thuật minimax

- Áp dụng giải thuật minimax vào trò chơi Nim



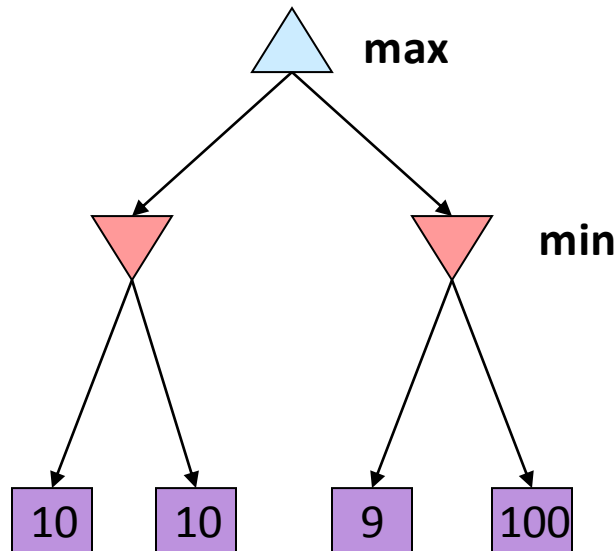
# Giải thuật minimax

## ■ Minimax đến độ sâu lớp cố định

- Đối với **các trò chơi phức tạp**, đồ thị KGTT có khả năng không được triển khai đến các nút lá
- KGTT chỉ có thể được triển khai đến một số mức xác định (tùy vào tiềm năng về thời gian và bộ nhớ) => **tính trước n nước đi**
- Vì các nút lá của đồ thị con này không phải là trạng thái kết thúc của trò chơi => không xác định được các giá trị thắng - thua (1 hoặc 0)
- Cần sử dụng một **hàm đánh giá Heuristic** nào đó
- Giá trị nút lá là các giá trị Heuristic đạt được sau n nước đi kể từ nút xuất phát (tùy vào hàm đánh giá Heuristic cụ thể)
- Các giá trị này sẽ được truyền ngược về nút gốc tương tự như trong trò chơi Nim, và chỉ là giá trị của trạng thái tốt nhất có thể.

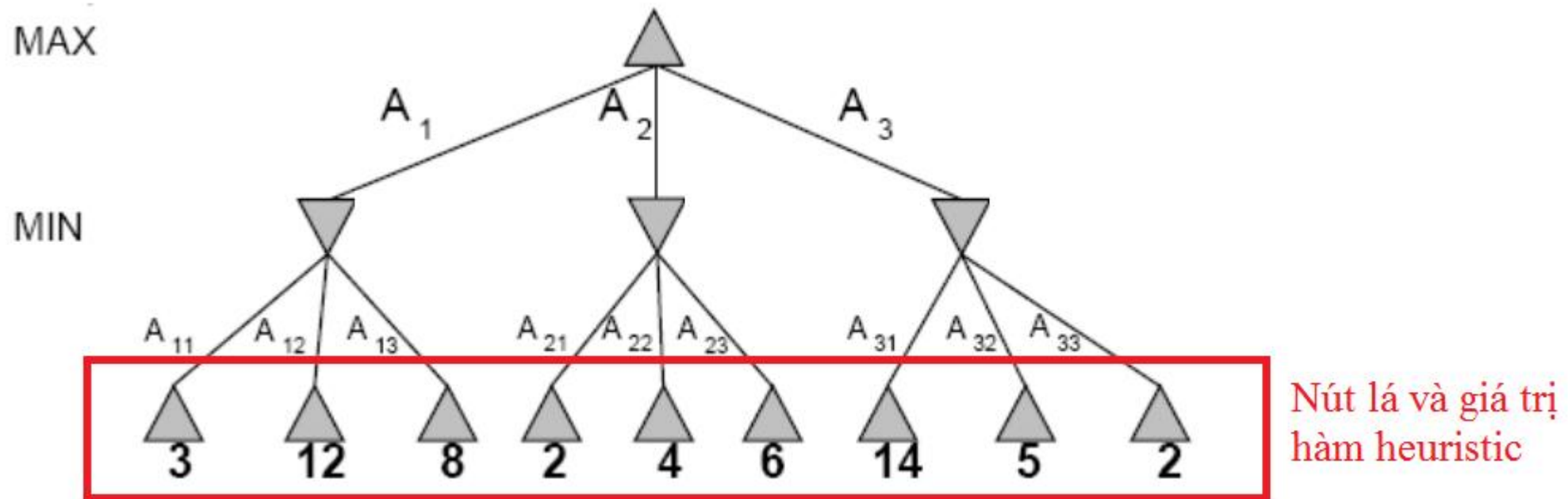
# Minimax với độ sâu lớp cố định

- Các **nút lá** được gán các giá trị ***heuristic***
- Còn giá trị tại các nút trong là các giá trị nhận được dựa trên giải thuật Minimax



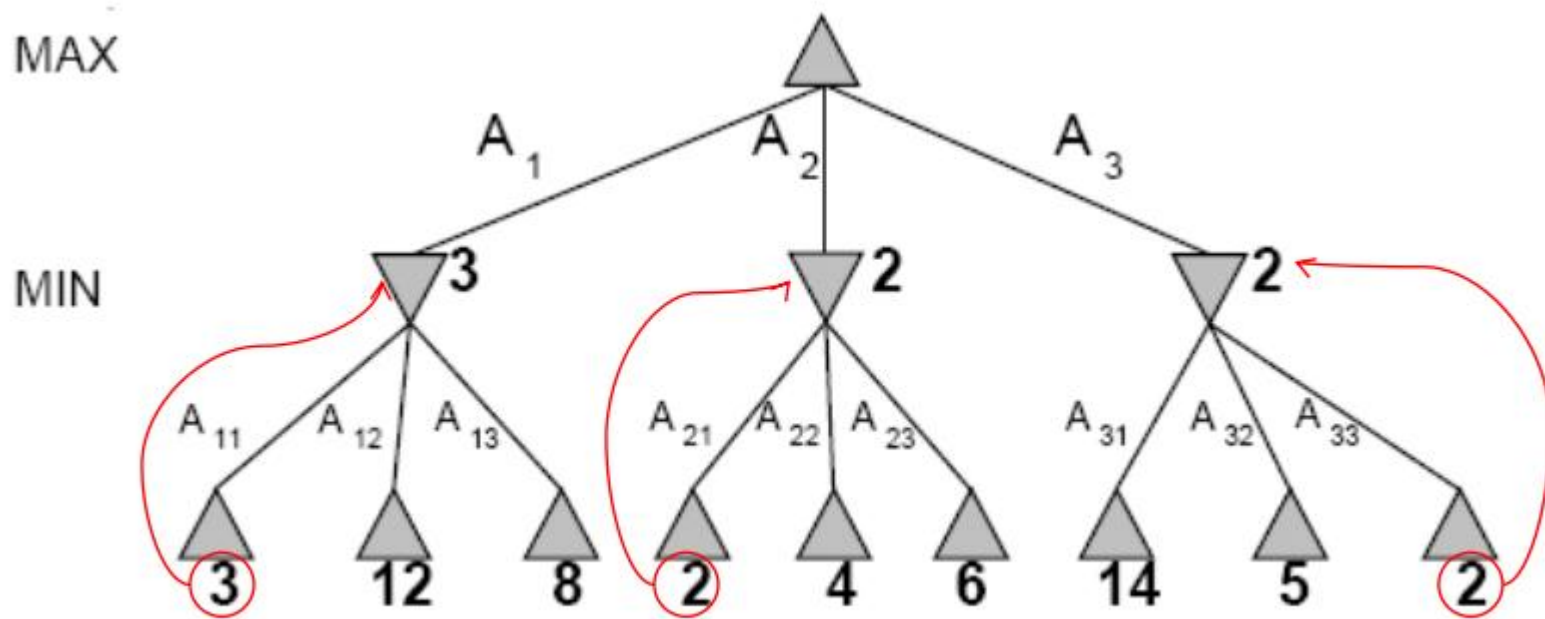
# Minimax với độ sâu lớp cố định

- Ví dụ 1. Cho cây trò chơi có độ sâu bằng 2 và giá trị Heuristic của các nút lá. Hãy tính giá trị của các nút còn lại và xác định nước chơi tốt nhất.



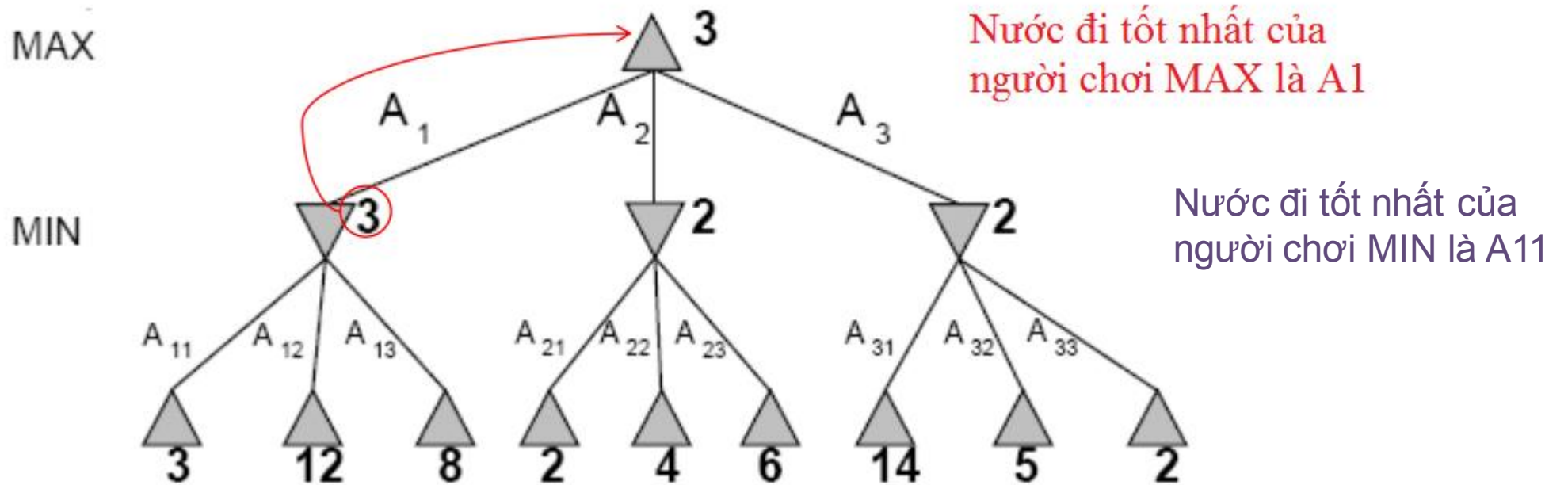
# Minimax với độ sâu lớp cố định

## ■ Ví dụ 1.



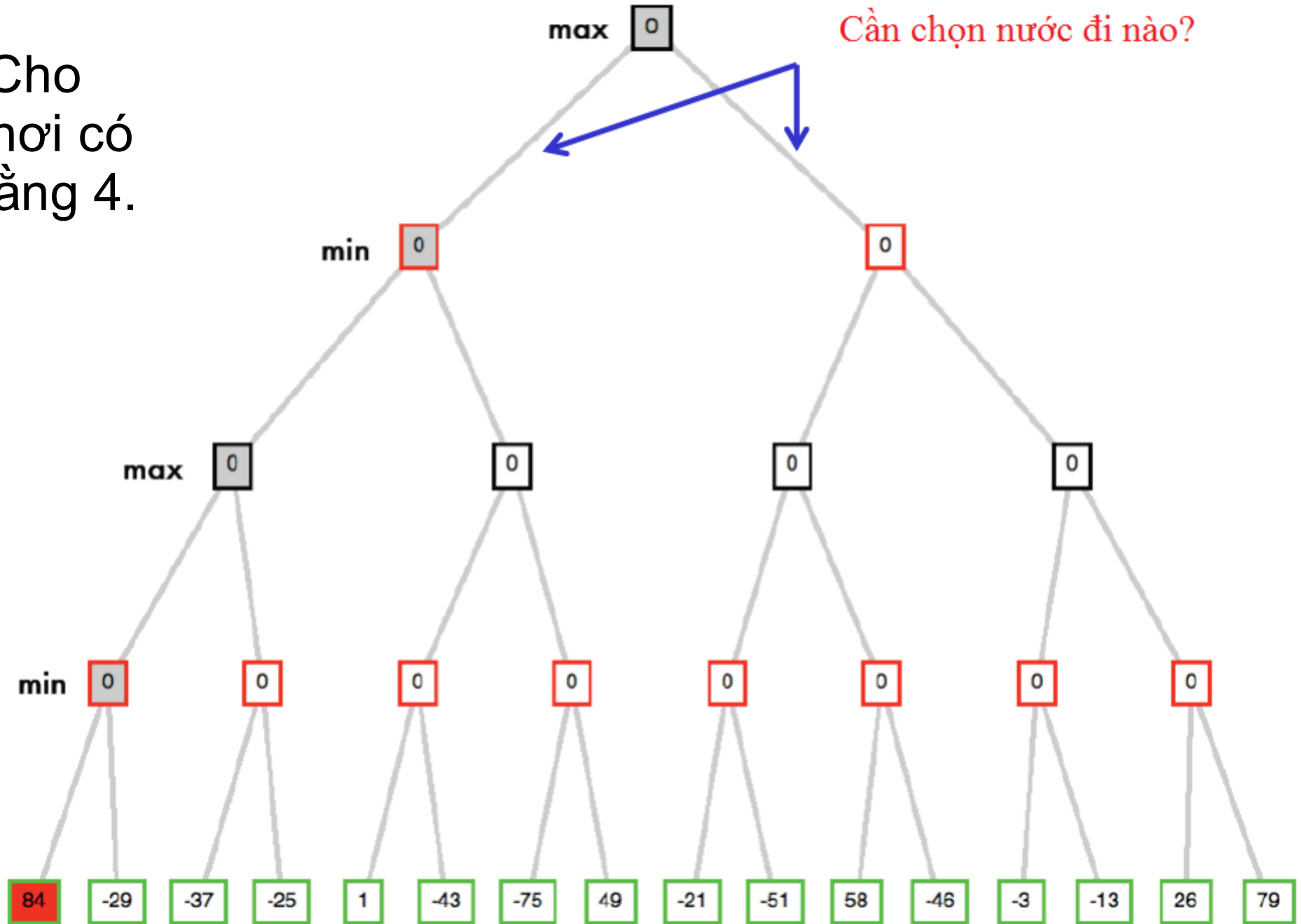
# Minimax với độ sâu lớp cố định

## ■ Ví dụ 1.

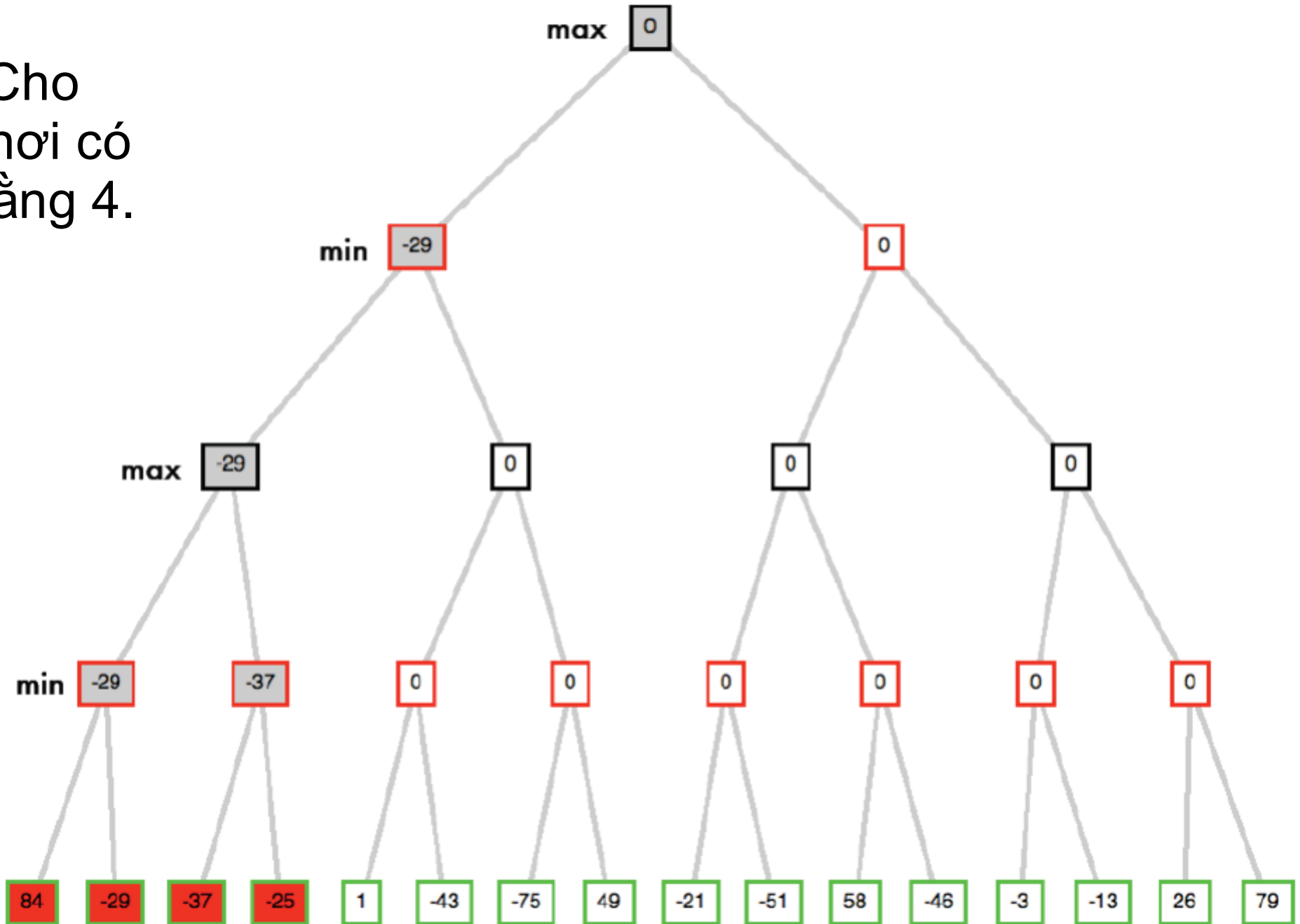




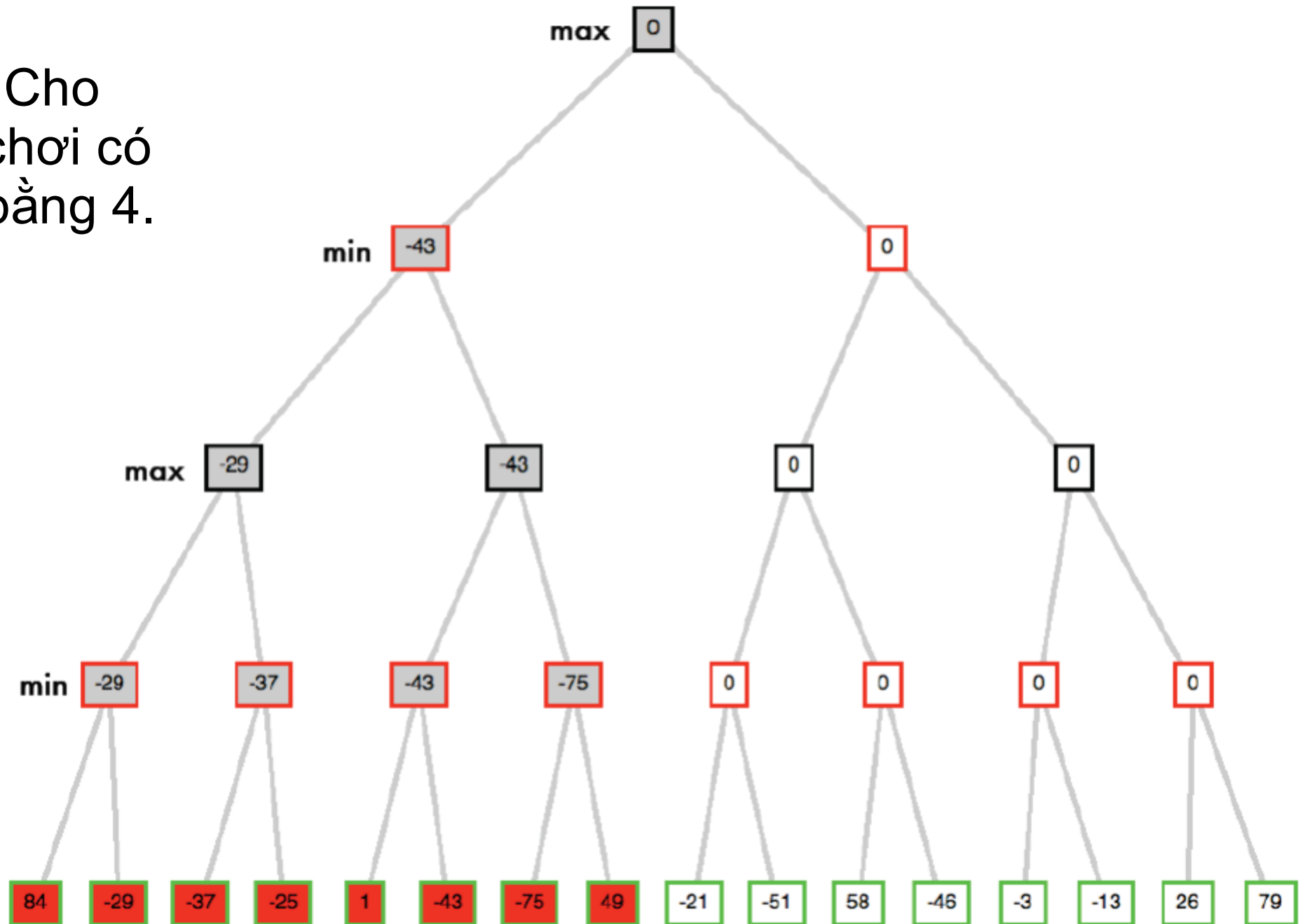
- Ví dụ 2. Cho cây trò chơi có độ sâu bằng 4.



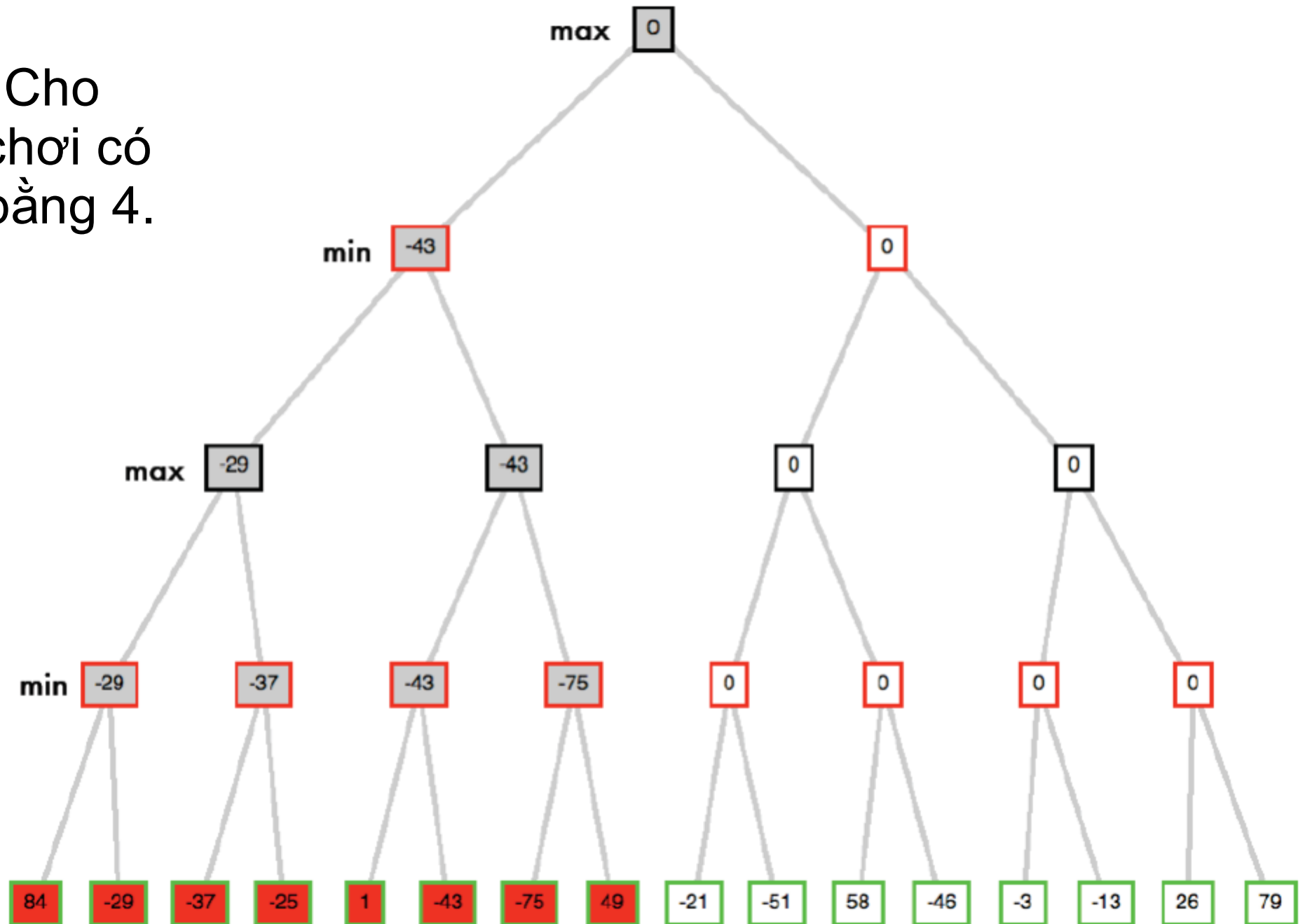
- Ví dụ 2. Cho cây trò chơi có độ sâu bằng 4.



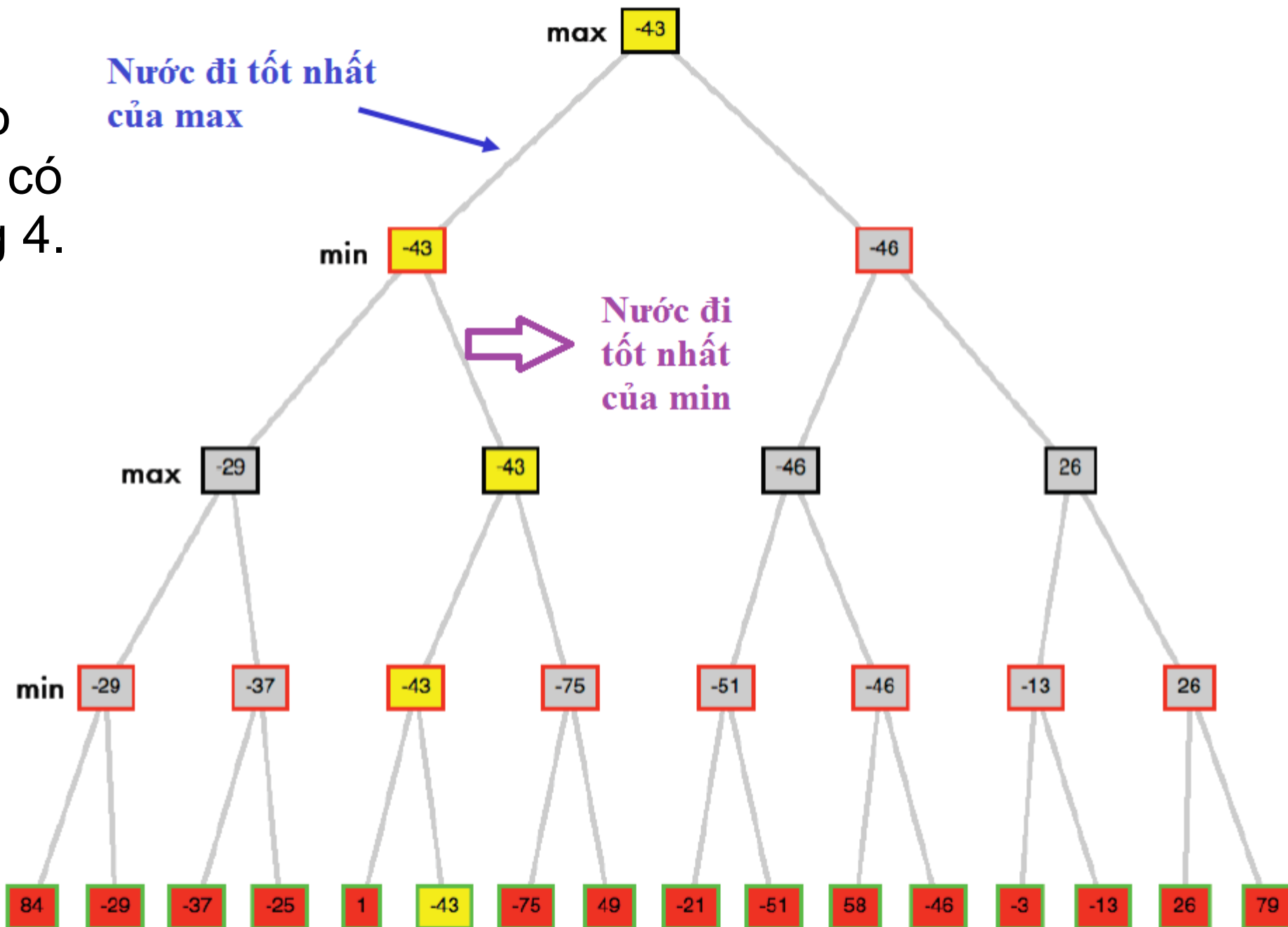
- Ví dụ 2. Cho cây trò chơi có độ sâu bằng 4.



- Ví dụ 2. Cho cây trò chơi có độ sâu bằng 4.

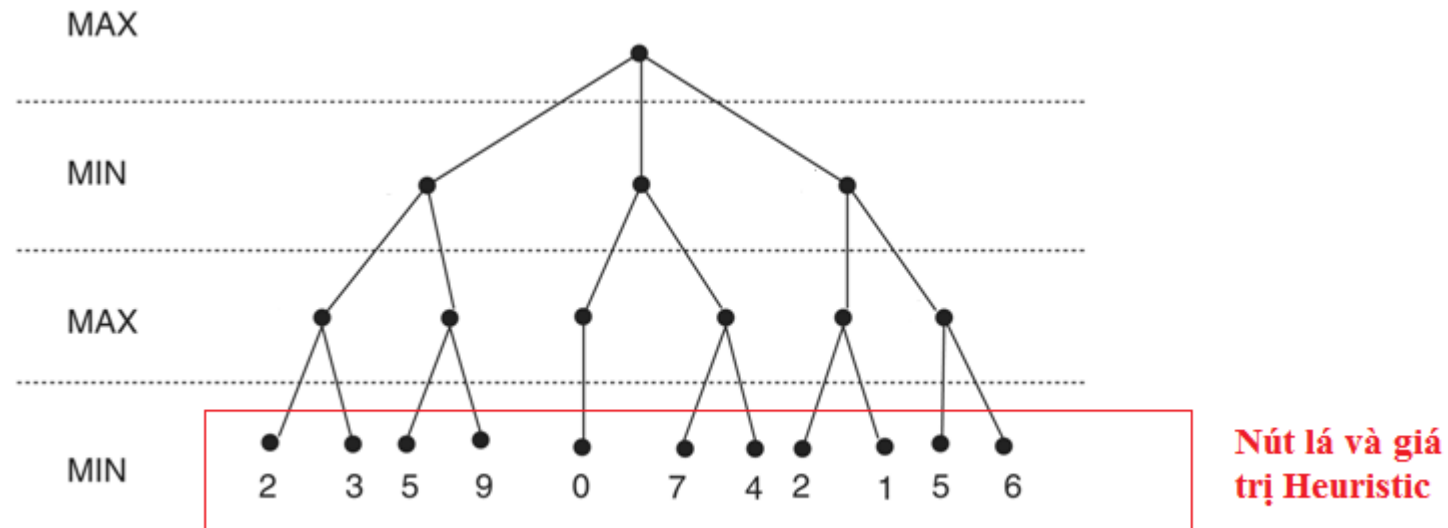


- Ví dụ 2. Cho cây trò chơi có độ sâu bằng 4.



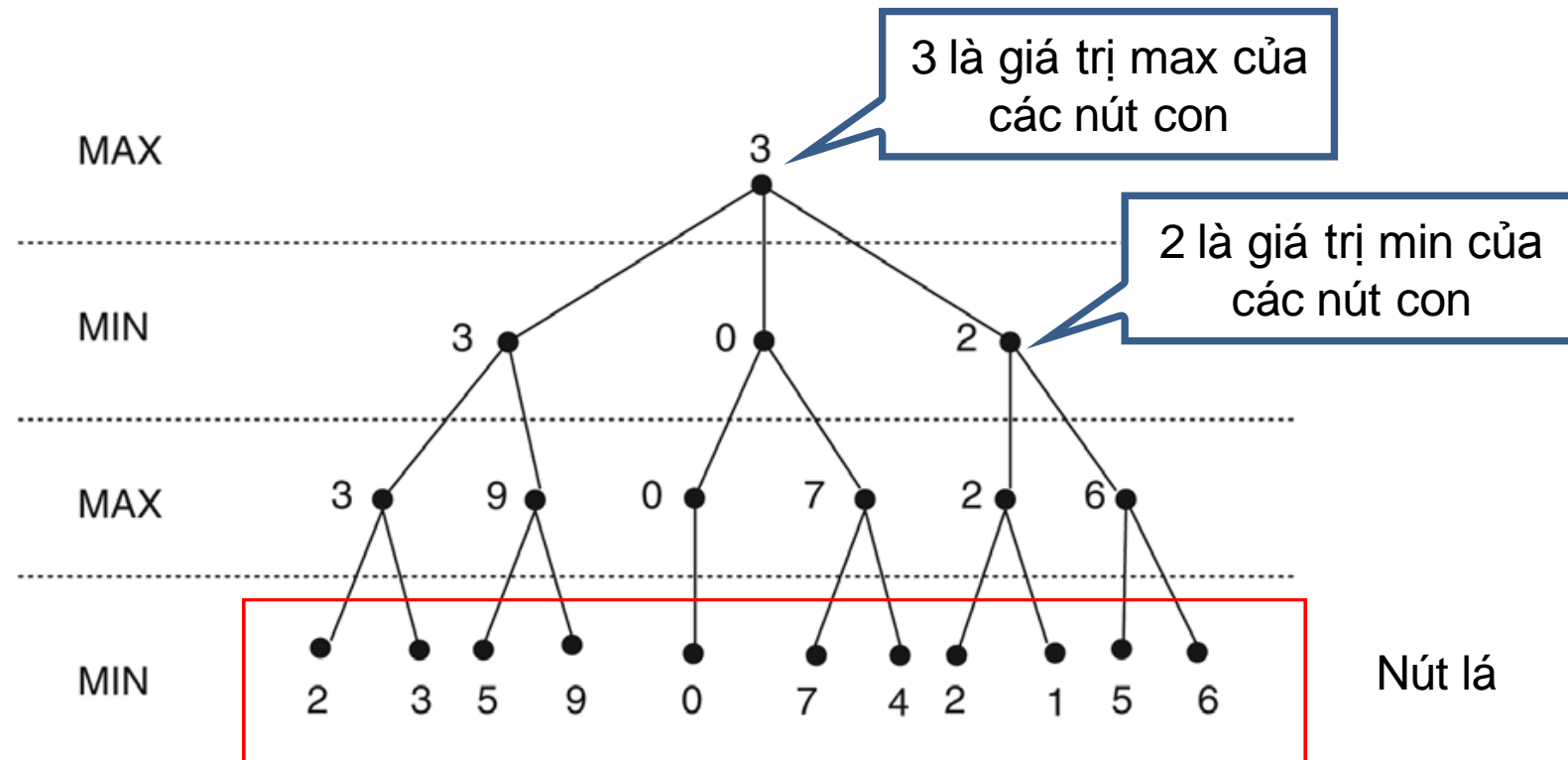
# Minimax với độ sâu lớp cố định

- Ví dụ 3. Cho KGTT giả định và giá trị Heuristic của các nút lá. Hãy tính giá trị của các nút còn lại

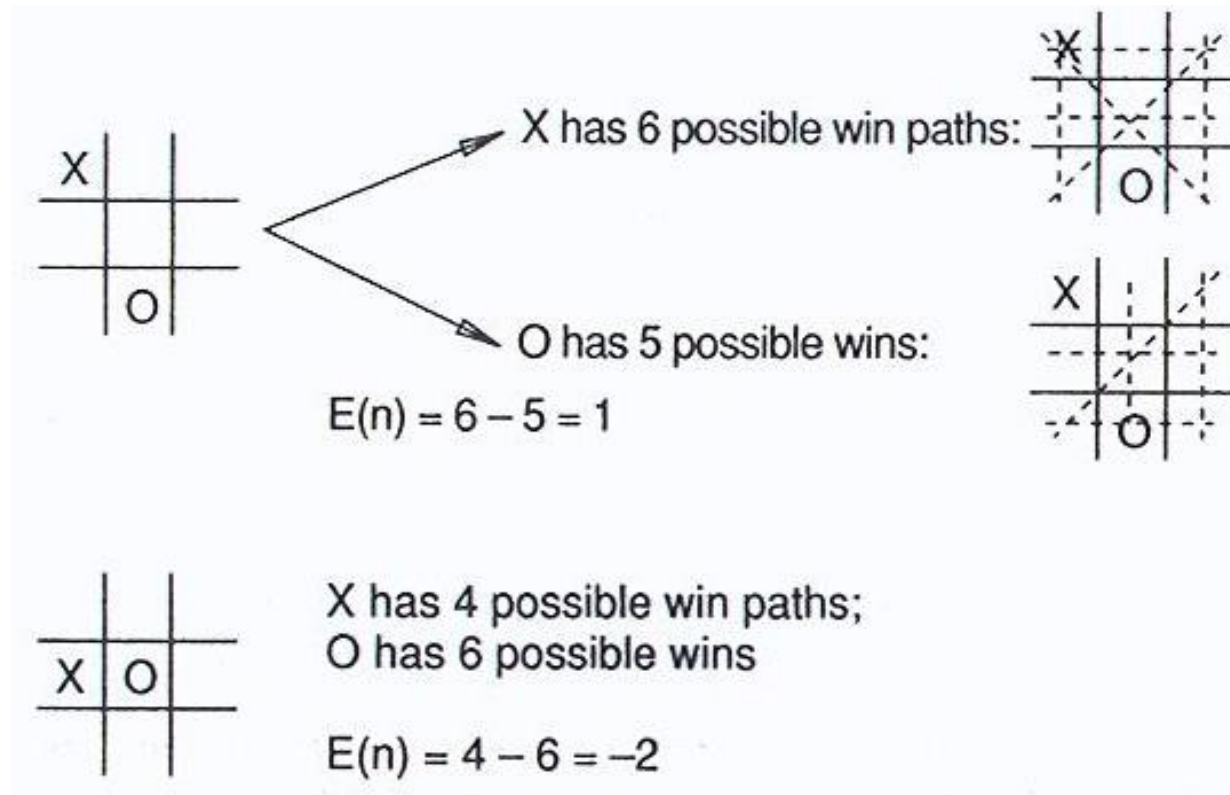


# Giải thuật minimax

## ■ Ví dụ 3:



# Heuristic trong trò chơi tic-tac-toe



**Hàm Heuristic:**  $E(n) = M(n) - O(n)$

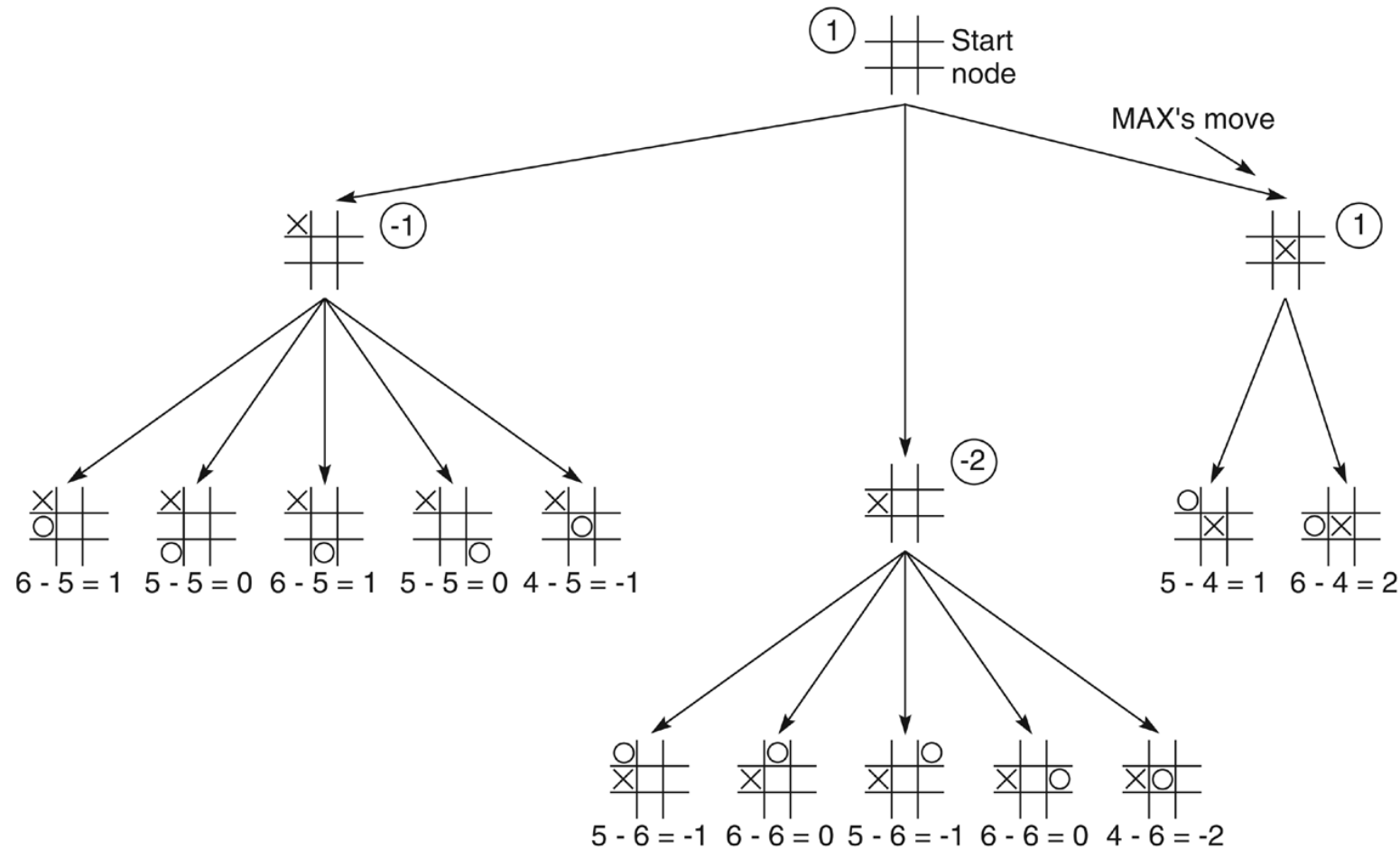
Trong đó:  $M(n)$  là tổng số đường thắng có thể của tôi

$O(n)$  là tổng số đường thắng có thể của đối thủ

$E(n)$  là trị số đánh giá tổng cộng cho trạng thái  $n$



# Minimax 2 lớp được áp dụng vào nước đi mở đầu trong tic-tac-toe



Trích từ Nilsson (1971).

## Hàm Heuristic:

$$E(n) = M(n) - O(n)$$

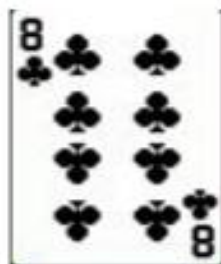
Trong đó:  $M(n)$  là tổng số đường thắng có thể của tôi

$O(n)$  là tổng số đường thắng có thể của đối thủ

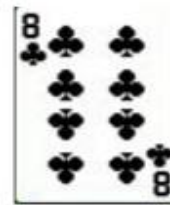
$E(n)$  là trị số đánh giá tổng cộng cho trạng thái  $n$

# Trò chơi bắt lá bài

- Mỗi người chơi chọn 1 lá bài trong 4 lá bài cho trước
- Với 4 lá bài được cho sẵn, mỗi người sẽ chọn 2 lá bài
- Cộng điểm 2 lá bài đã chọn, nếu người chơi nào có tổng điểm là chẵn và cao nhất sẽ thắng

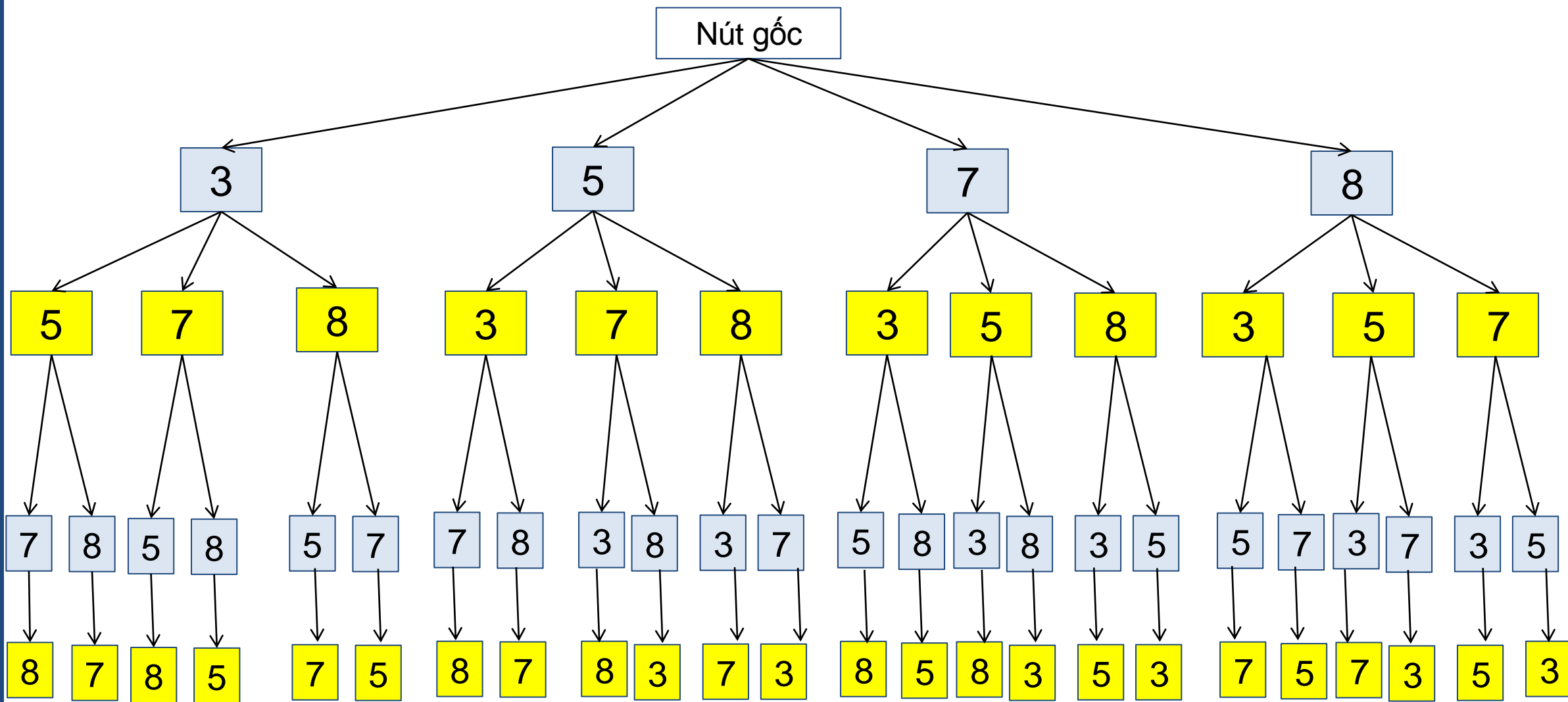
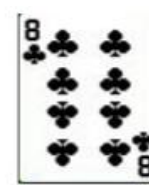


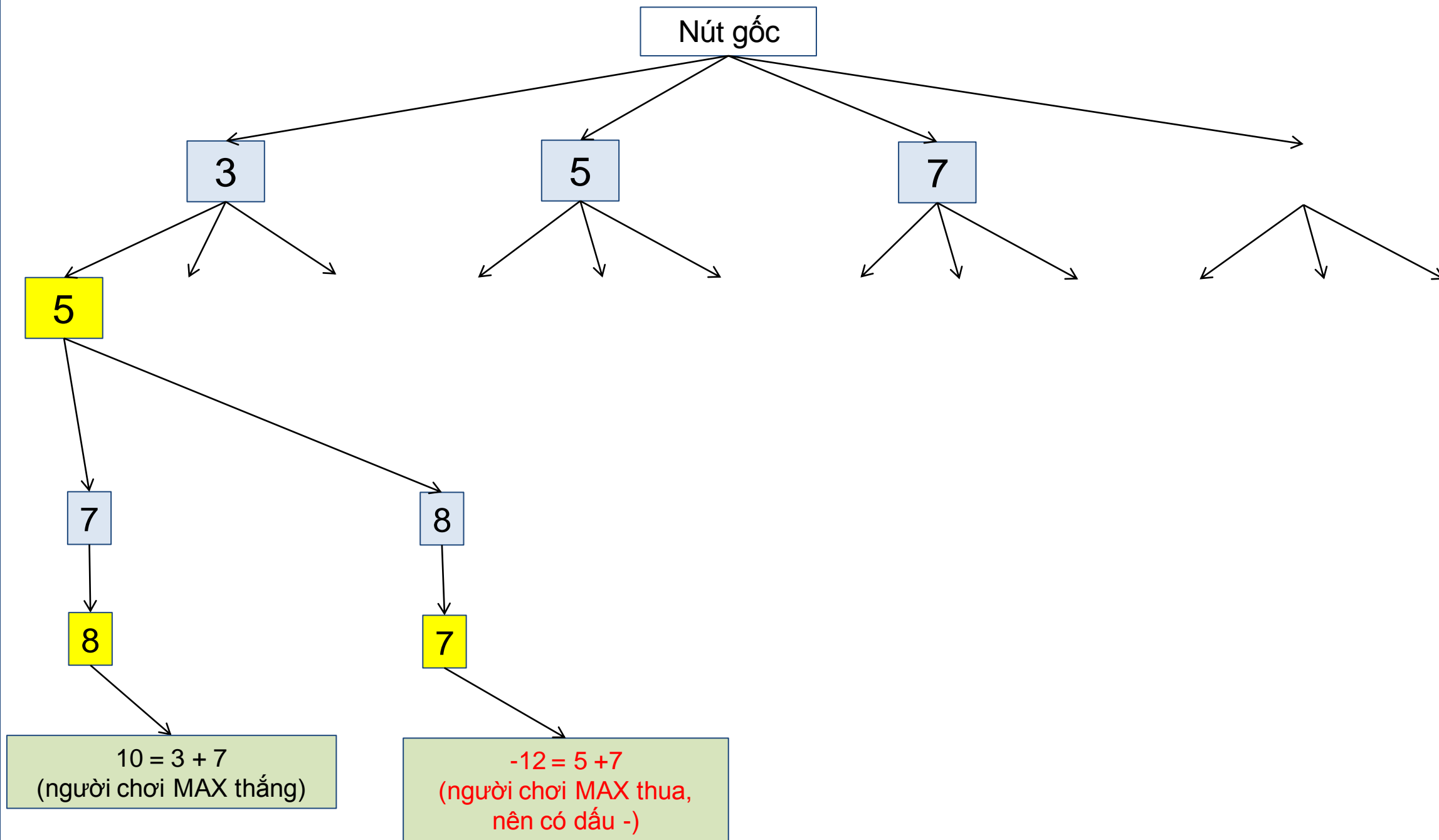
# Trò chơi bắt lá bài



- Để có thể thắng trong trò chơi trên
  - Vẽ toàn bộ không gian trạng thái
  - Gắn điểm số vào mỗi trạng thái kết thúc
  - Sử dụng giải thuật Minimax để cập nhật điểm số từ nút lá đến nút gốc
  - Chọn đường đi có điểm số lớn nhất

# Trò chơi bắt lá bài





Nút gốc

3

5

7

8

5

7

8

3

7

8

3

5

8

3

5

7

7

8

5

8

5

7

7

8

3

8

3

7

5

8

3

8

3

5

5

7

3

7

3

5

8

7

8

5

7

5

8

7

8

3

7

3

8

5

3

5

3

7

5

7

3

5

3

10

-12

8

-12

8

10

12

-10

8

-10

8

12

12

-8

10

-8

10

12

-10

-8

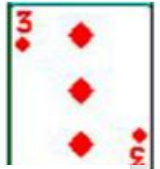
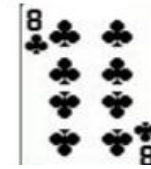
-12

-8

-12

-10

# Bài toán bắt lá bài



Player 1 Max

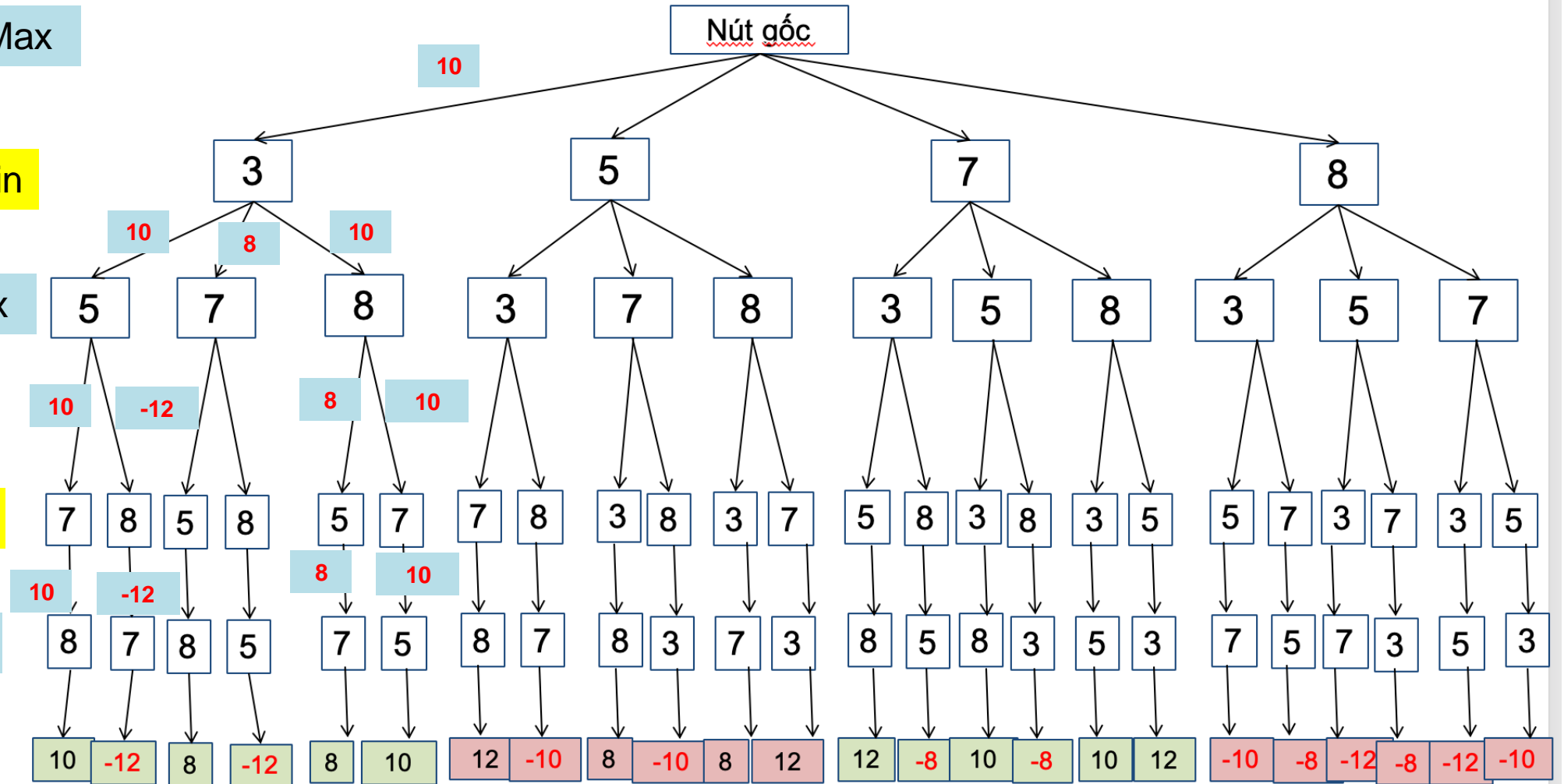
Player 2 Min

Player 1 Max

Player 2 Min

Player 1 Max

Điểm cho  
người chơi  
Player 1



Ta có thể làm gì để hạn chế số lượng trạng thái phải kiểm tra ngoài việc hạn chế số mức đi vì số trạng thái vẫn còn quá lớn?



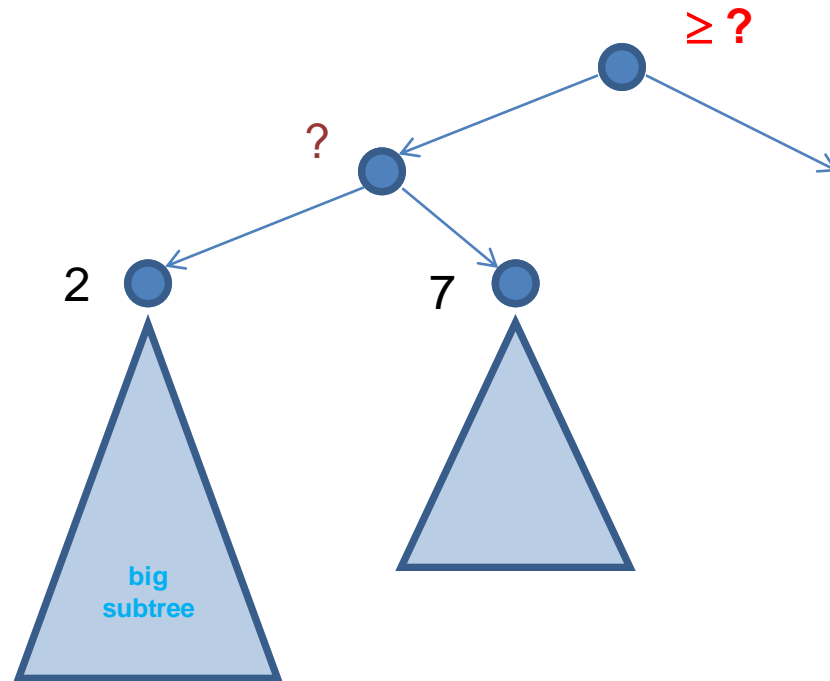
# Cắt tỉa các nhánh

- Ta có thể làm gì để hạn chế số lượng TT phải kiểm tra ngoài việc hạn chế số mức d đi vì số trạng thái vẫn còn quá lớn?
- Cờ vua: nhân tố nhánh  $b=35$ ;  $d=3$  có  $35*35*35=42.785$  trạng thái
- Giảm bớt các trạng thái cần khảo sát mà vẫn không ảnh hưởng gì đến việc giải quyết bài toán
- **Cắt tỉa các nhánh không cần khảo sát.**
- Nếu biết là trường hợp xấu thì không cần phải xét thêm.

# Thủ tục cắt tỉa $\alpha - \beta$

- Tìm kiếm alpha – beta thực hiện theo kiểu tìm kiếm sâu.
- Hai giá trị  $\alpha$ ,  $\beta$  được tạo ra trong quá trình tìm kiếm.
  - $\alpha$  gắn với các nút MAX, không giảm.
  - $\beta$  gắn với các nút MIN, không tăng.
- Luật cắt tỉa  $\alpha$ - $\beta$ : Quá trình tìm kiếm có thể kết thúc bên dưới:
  - Nút MIN có  $\beta \leq \alpha$  của nút cha MAX bất kỳ ( $\alpha$ -cut)
  - Nút MAX có  $\alpha \geq \beta$  của nút cha MIN bất kỳ ( $\beta$ -cut)
- Cắt tỉa  $\alpha$ - $\beta$  thể hiện mối quan hệ giữa các nút ở mức  $n$  và  $n+2$ , mà tại đó toàn bộ cây có gốc tại mức  $n+1$  có thể cắt bỏ.

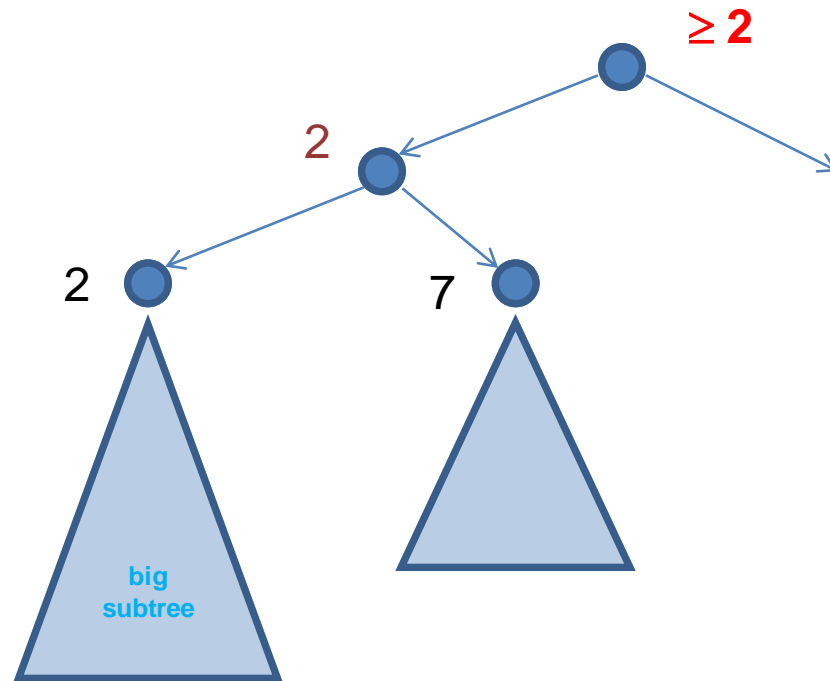
# Cắt tỉa $\alpha$ (1)



Us (max)

Them (min)

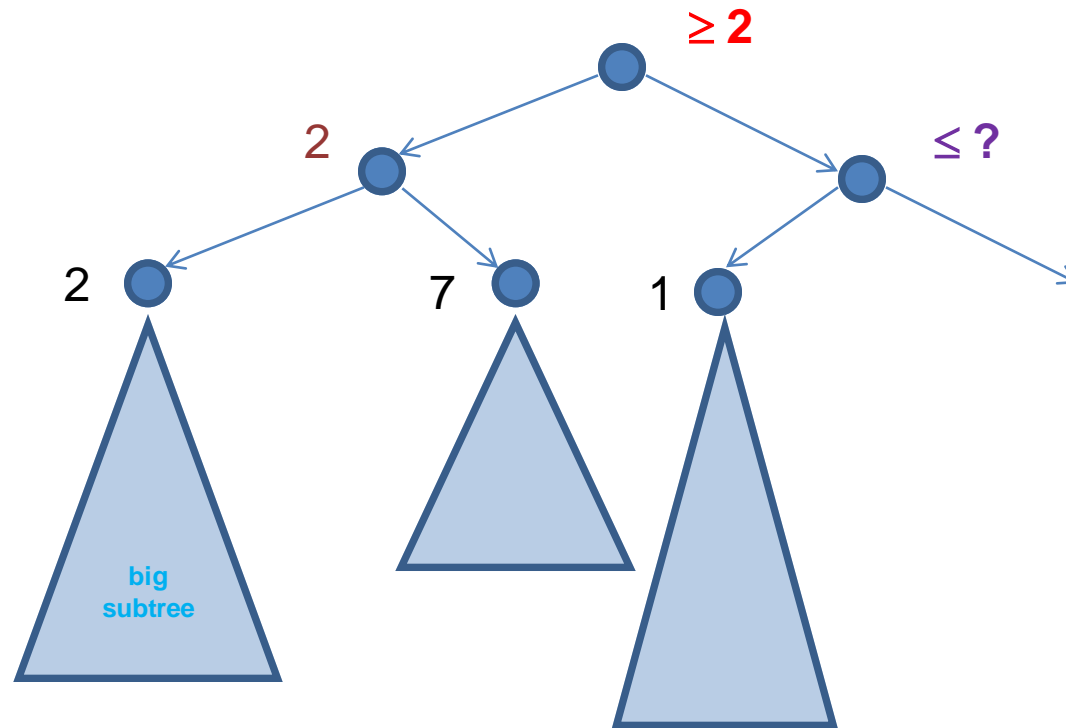
# Cắt tỉa $\alpha$ (2)



Us (max)

Them (min)

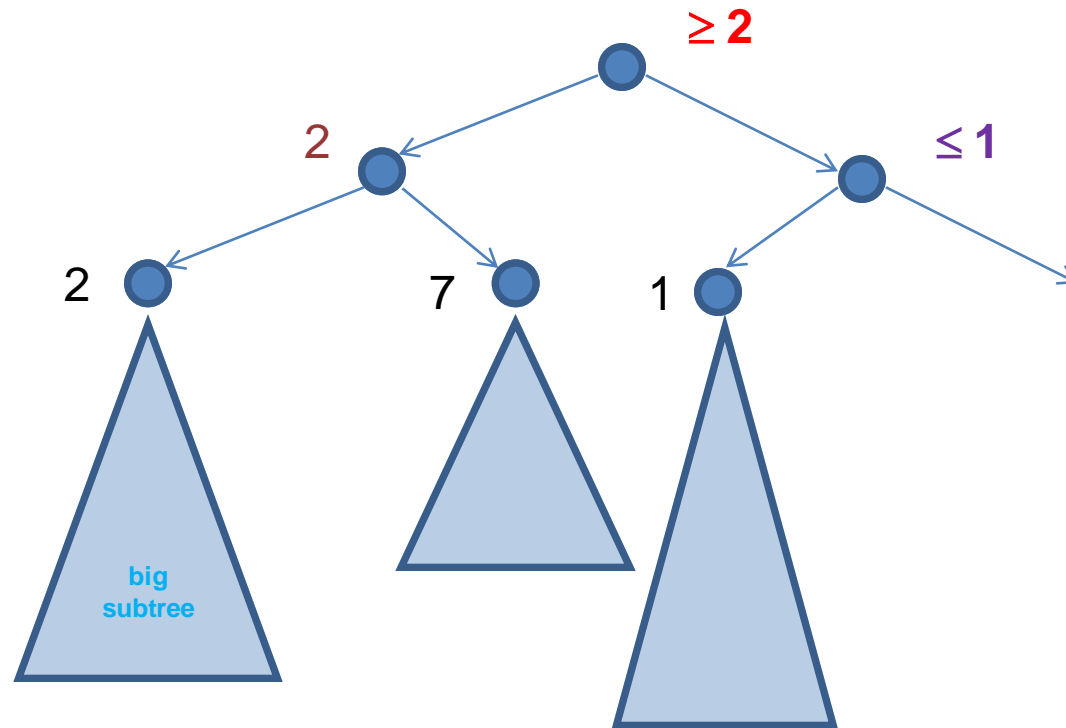
# Cắt tỉa $\alpha$ (3)



Us (max)

Them (min)

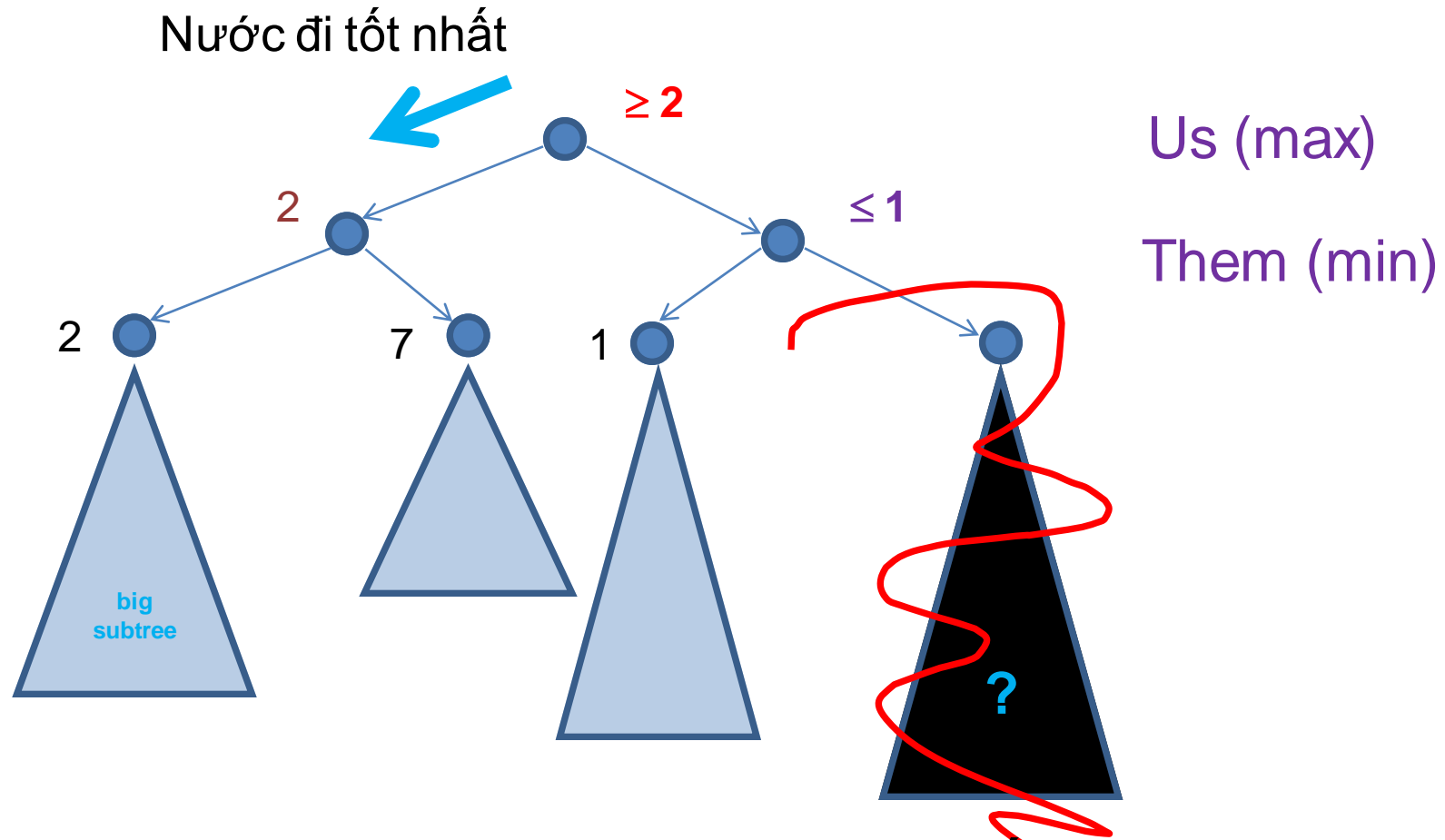
# Cắt tỉa $\alpha$ (4)



Us (max)

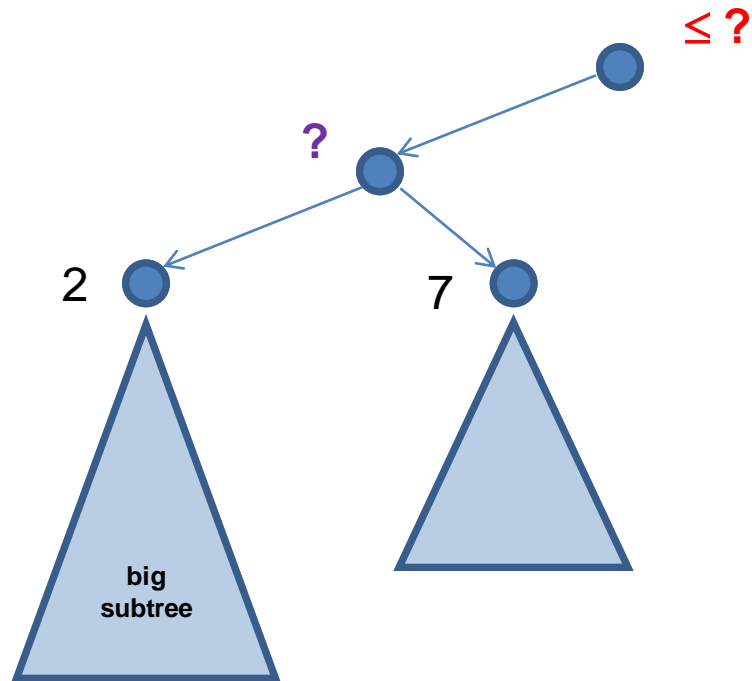
Them (min)

# Cắt tỉa $\alpha$ (5)



Nút MIN có  $\beta \leq \alpha$  của nút cha MAX bất kỳ ( $\alpha$ -cut)  
Nút MAX có  $\alpha \geq \beta$  của nút cha MIN bất kỳ ( $\beta$ -cut)

# Cắt tỉa $\beta$ (1)

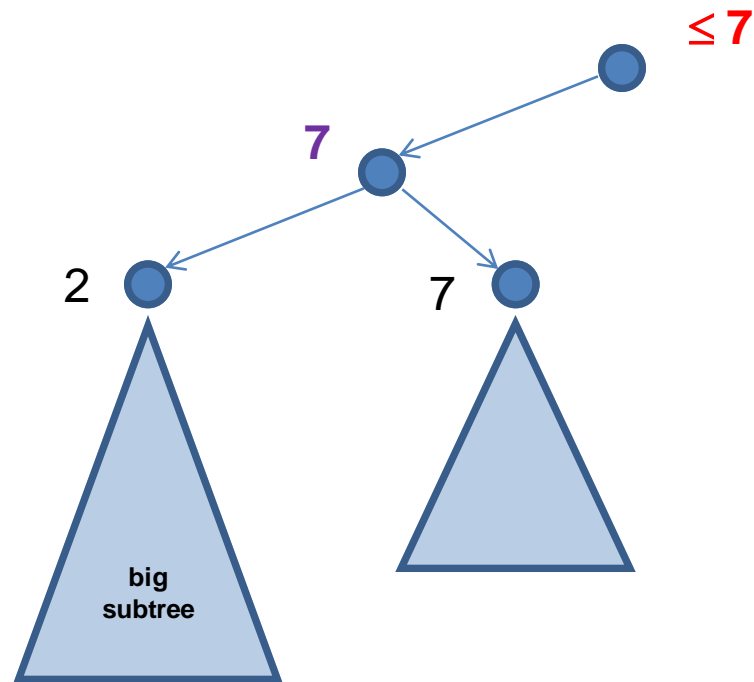


Them (min)

Us (max)



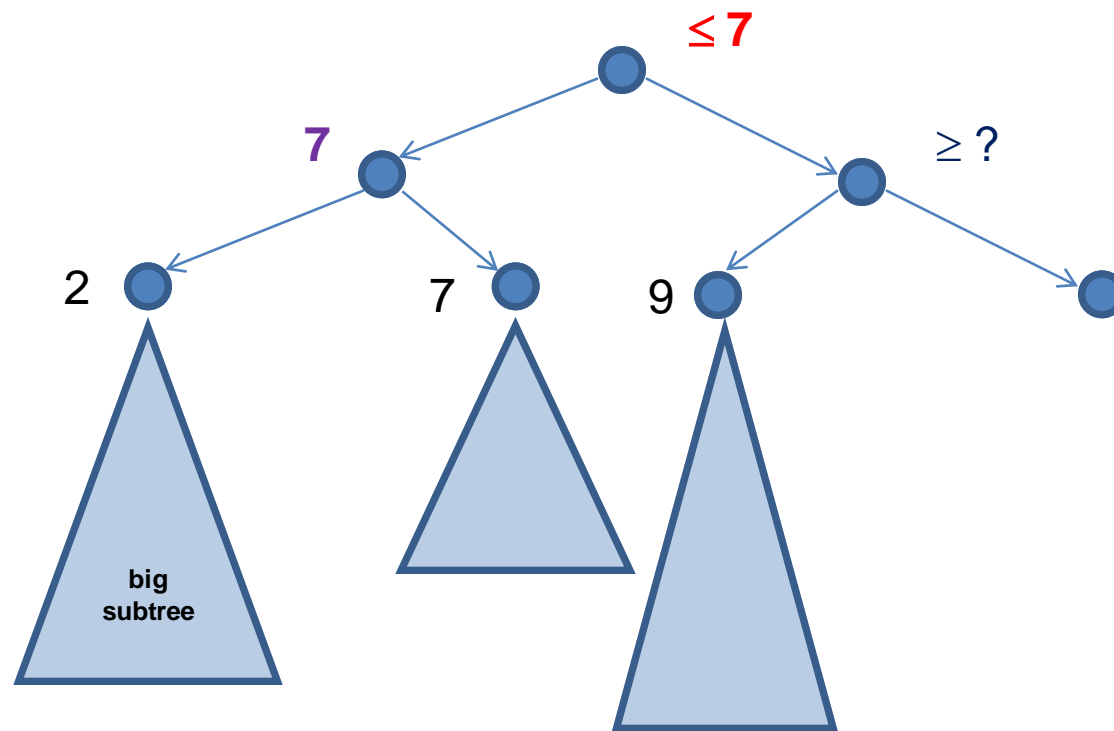
# Cắt tỉa $\beta$ (2)



Them (min)

Us (max)

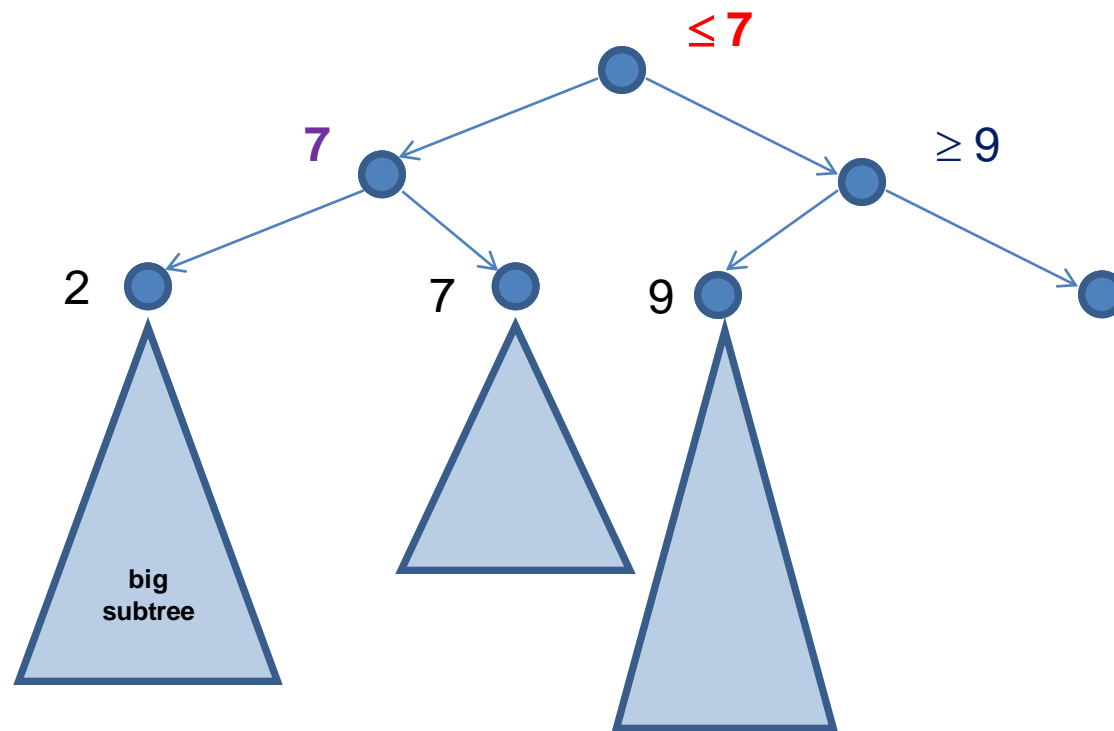
# Cắt tỉa $\beta$ (3)



Them (min)

Us (max)

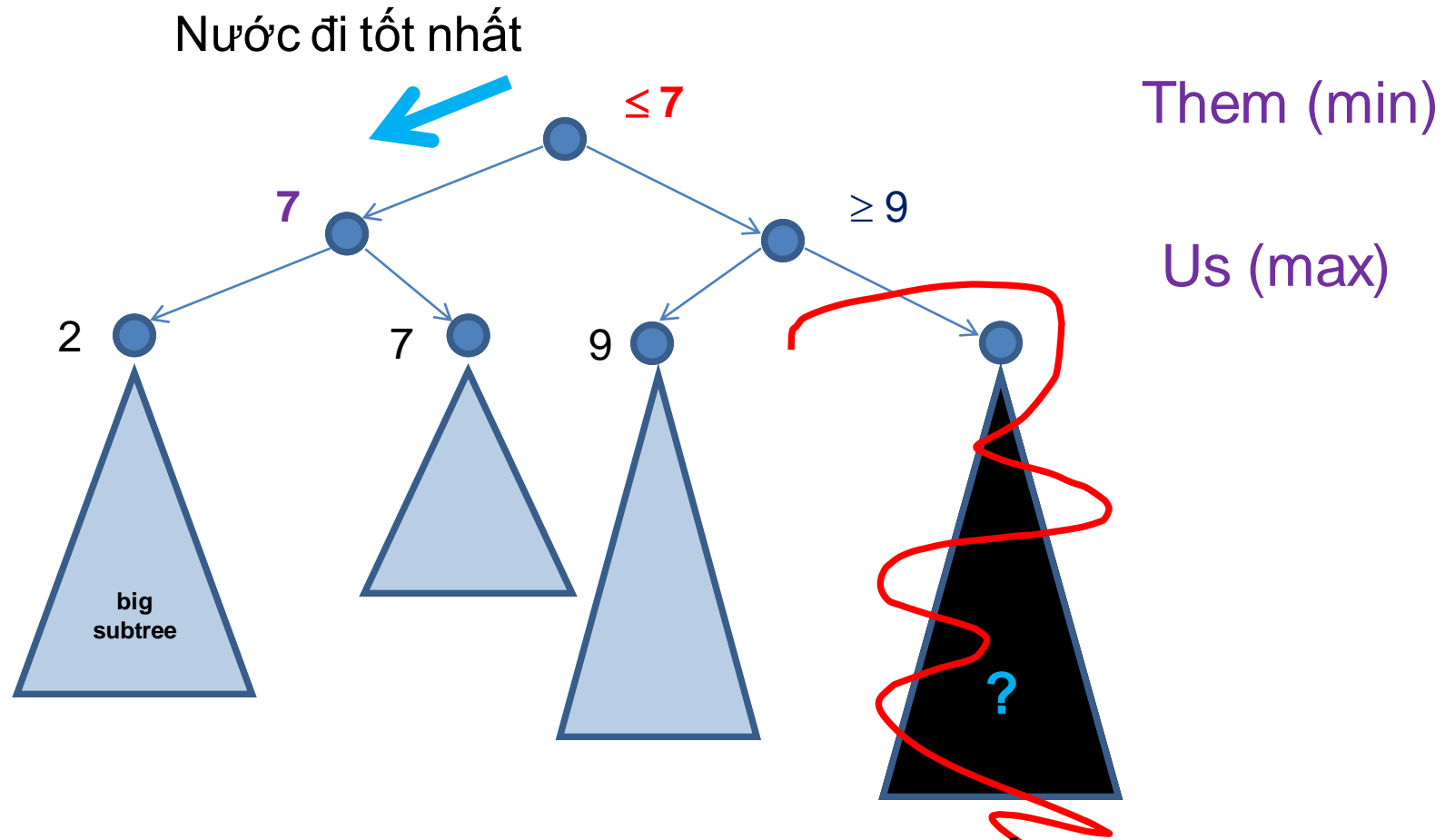
# Cắt tỉa $\beta$ (4)



Them (min)

Us (max)

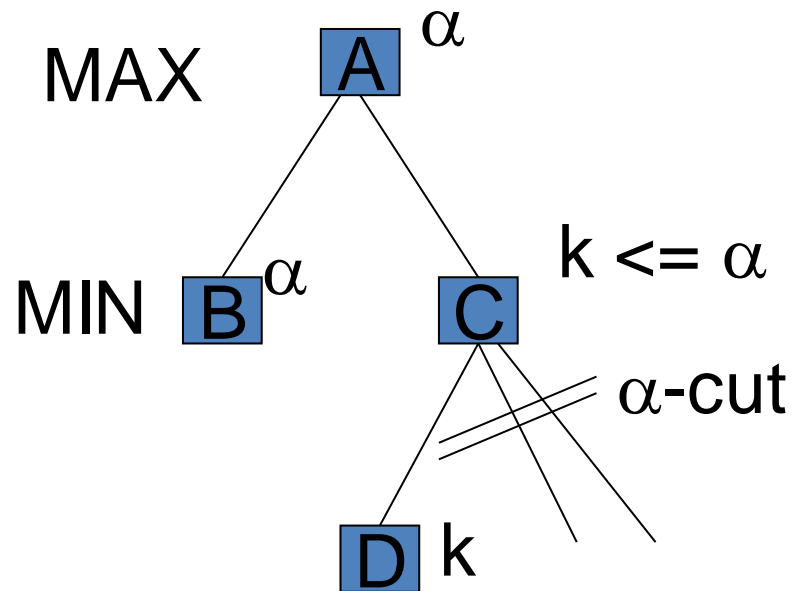
# Cắt tỉa $\beta$ (5)



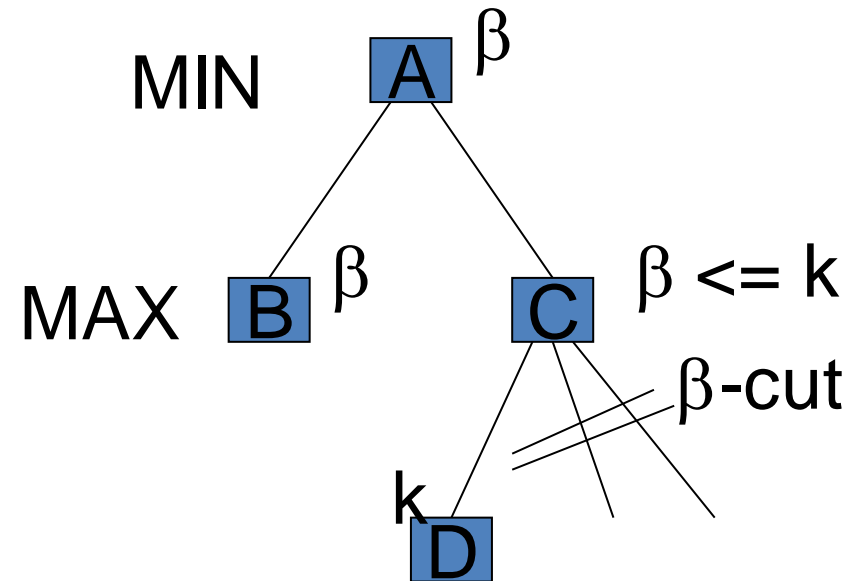
Nút MIN có  $\beta \leq \alpha$  của nút cha MAX bất kỳ ( $\alpha$ -cut)  
Nút MAX có  $\alpha \geq \beta$  của nút cha MIN bất kỳ ( $\beta$ -cut)

# Thủ tục cắt tỉa $\alpha - \beta$

## ■ $\alpha$ -cut

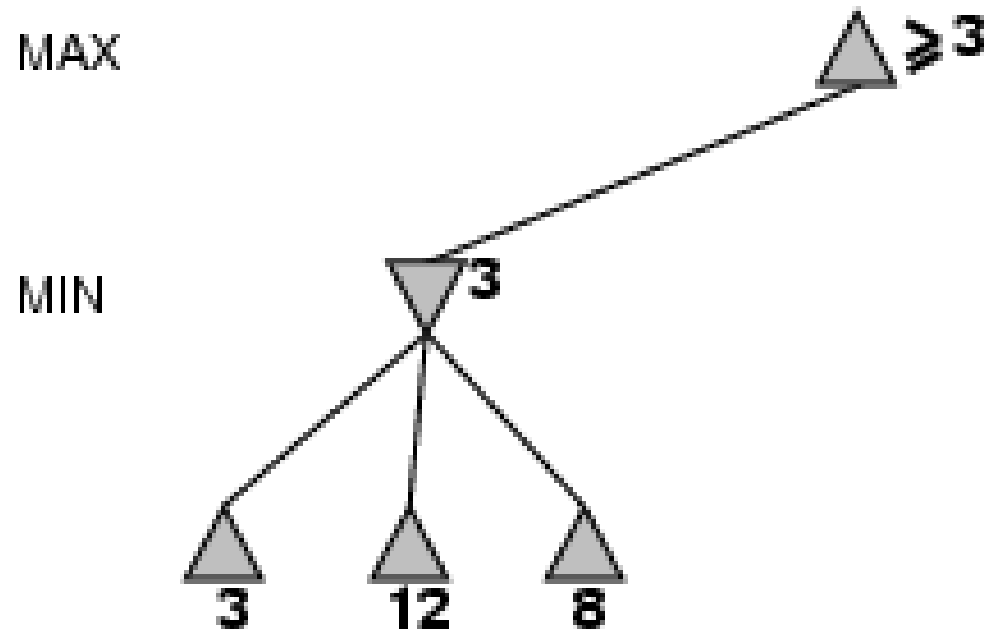


## $\beta$ -cut

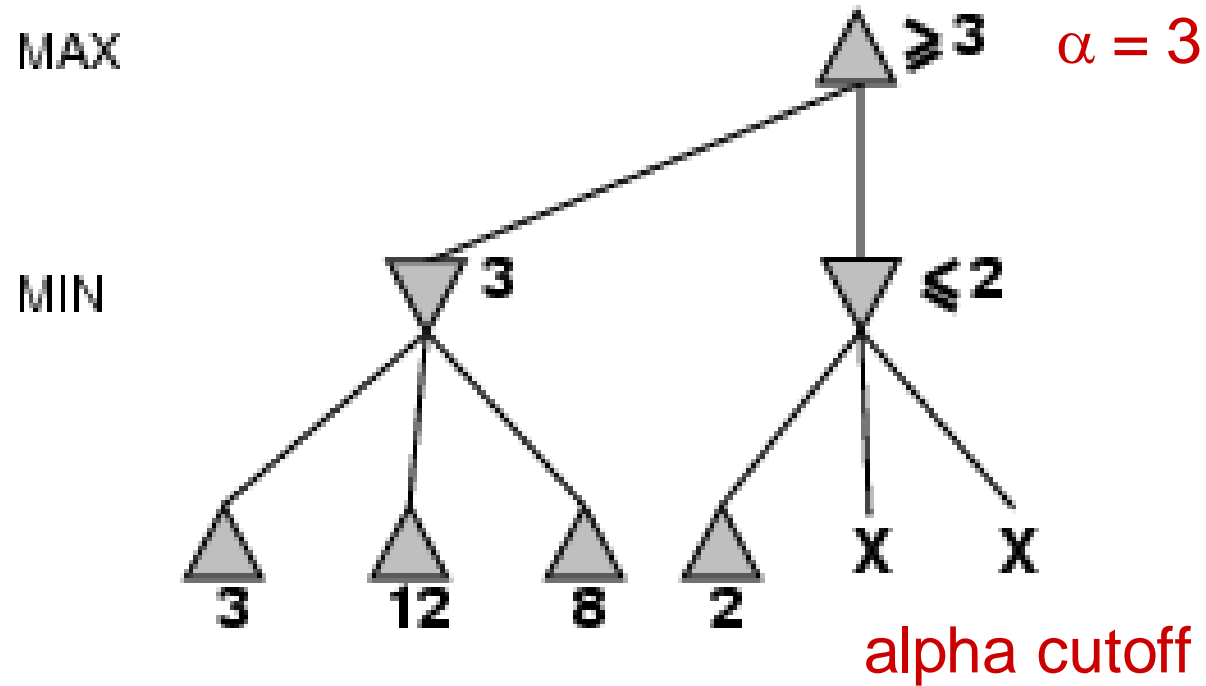


Nút MIN có  $\beta \leq \alpha$  của nút cha MAX bất kỳ ( $\alpha$ -cut)  
Nút MAX có  $\alpha \geq \beta$  của nút cha MIN bất kỳ ( $\beta$ -cut)

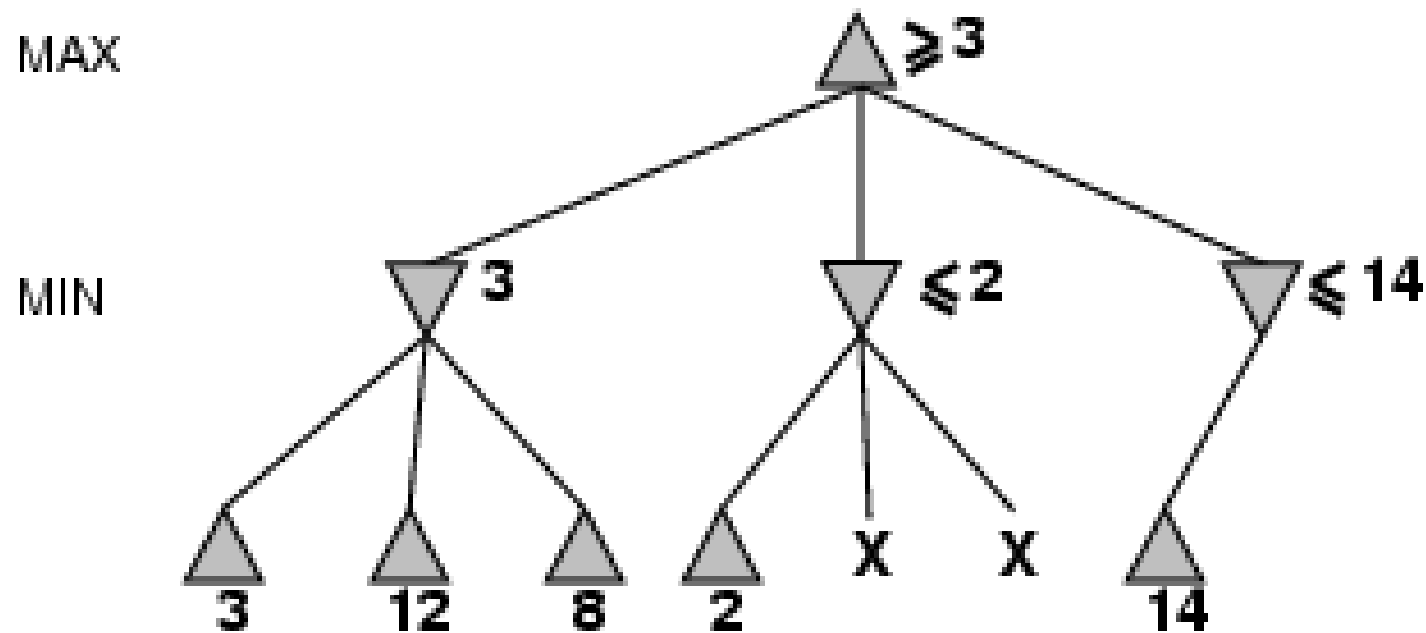
# Ví dụ 1 - cắt tỉa $\alpha - \beta$ (1)



# Ví dụ 1 - cắt tỉa $\alpha - \beta$ (2)

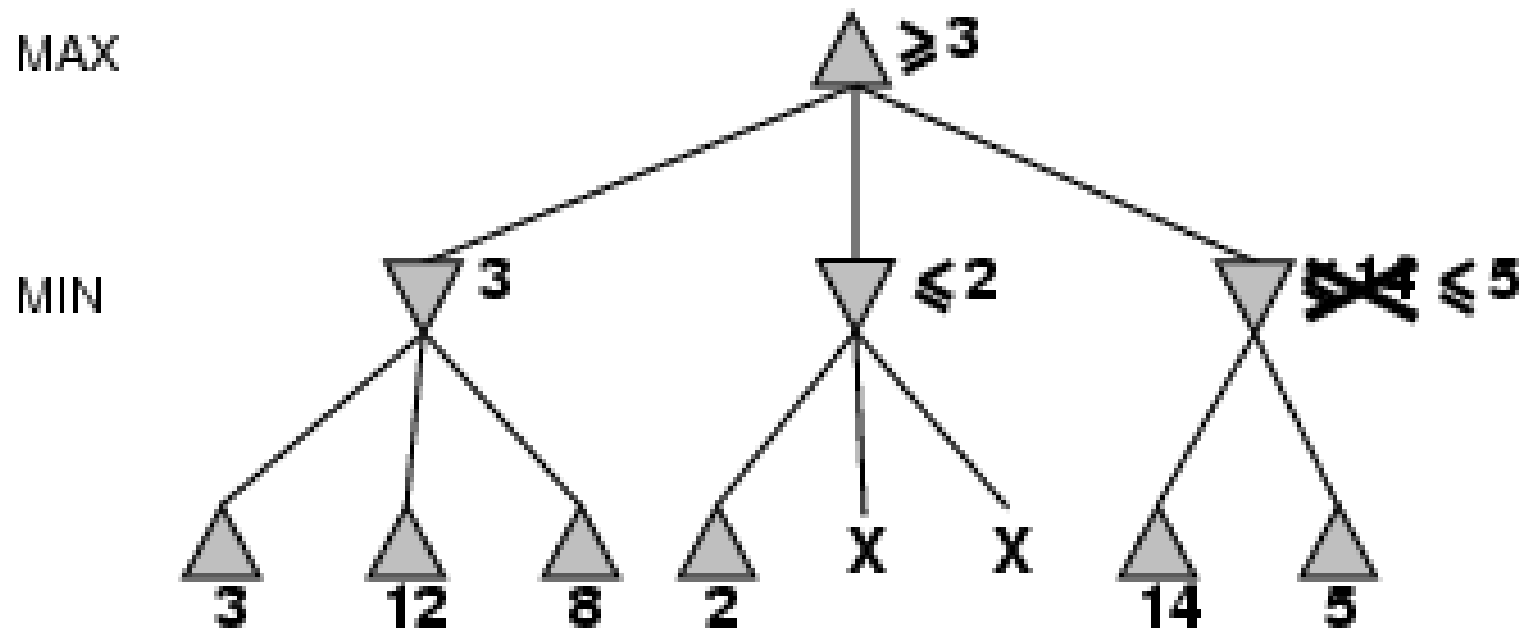


# Ví dụ 1 - cắt tỉa $\alpha - \beta$ (3)

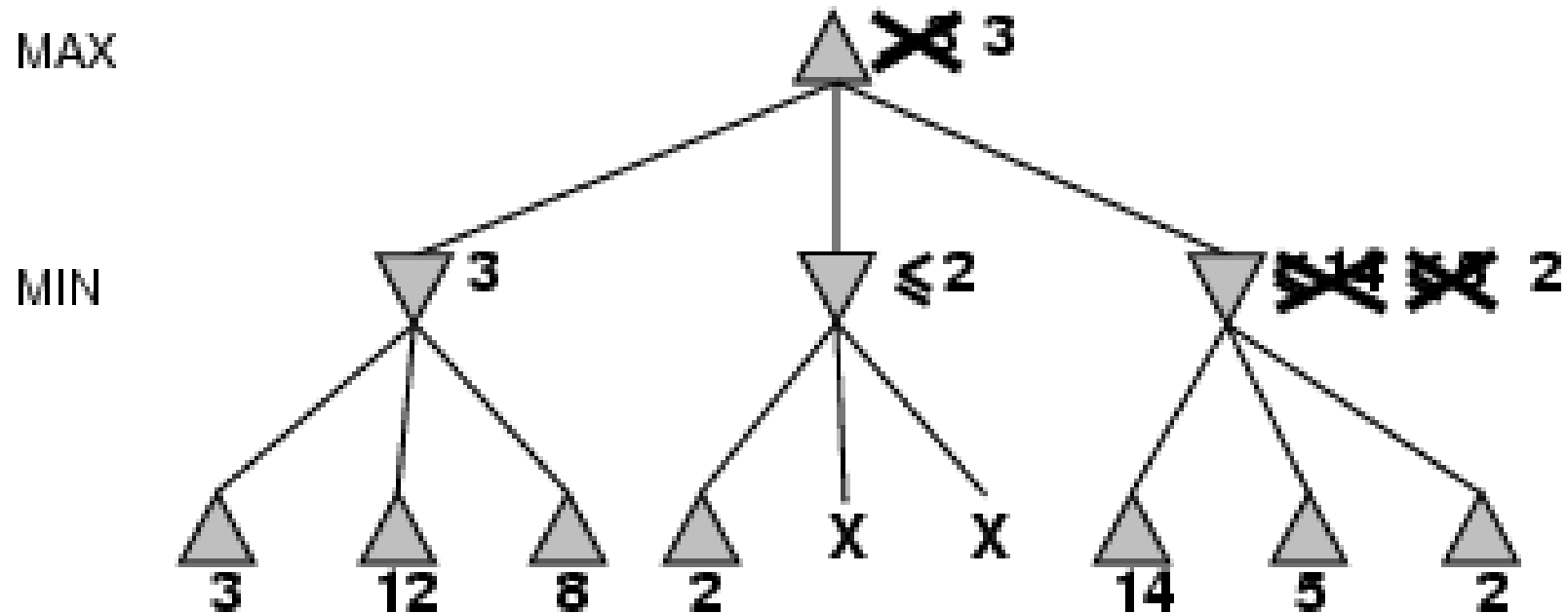




# Ví dụ 1 - cắt tỉa $\alpha - \beta$ (4)



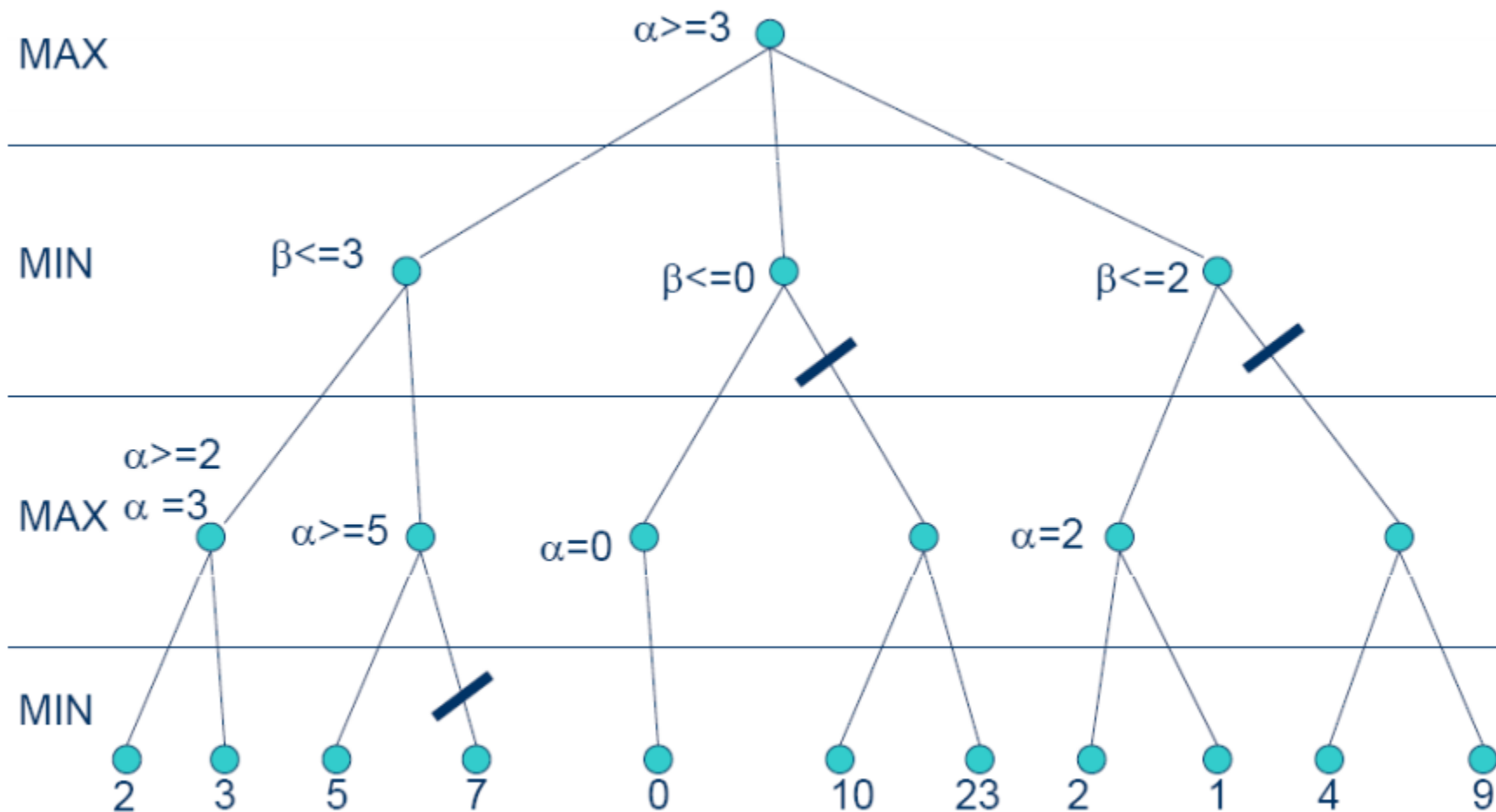
# Ví dụ 1 - cắt tỉa $\alpha - \beta$ (5)



Lưu ý:

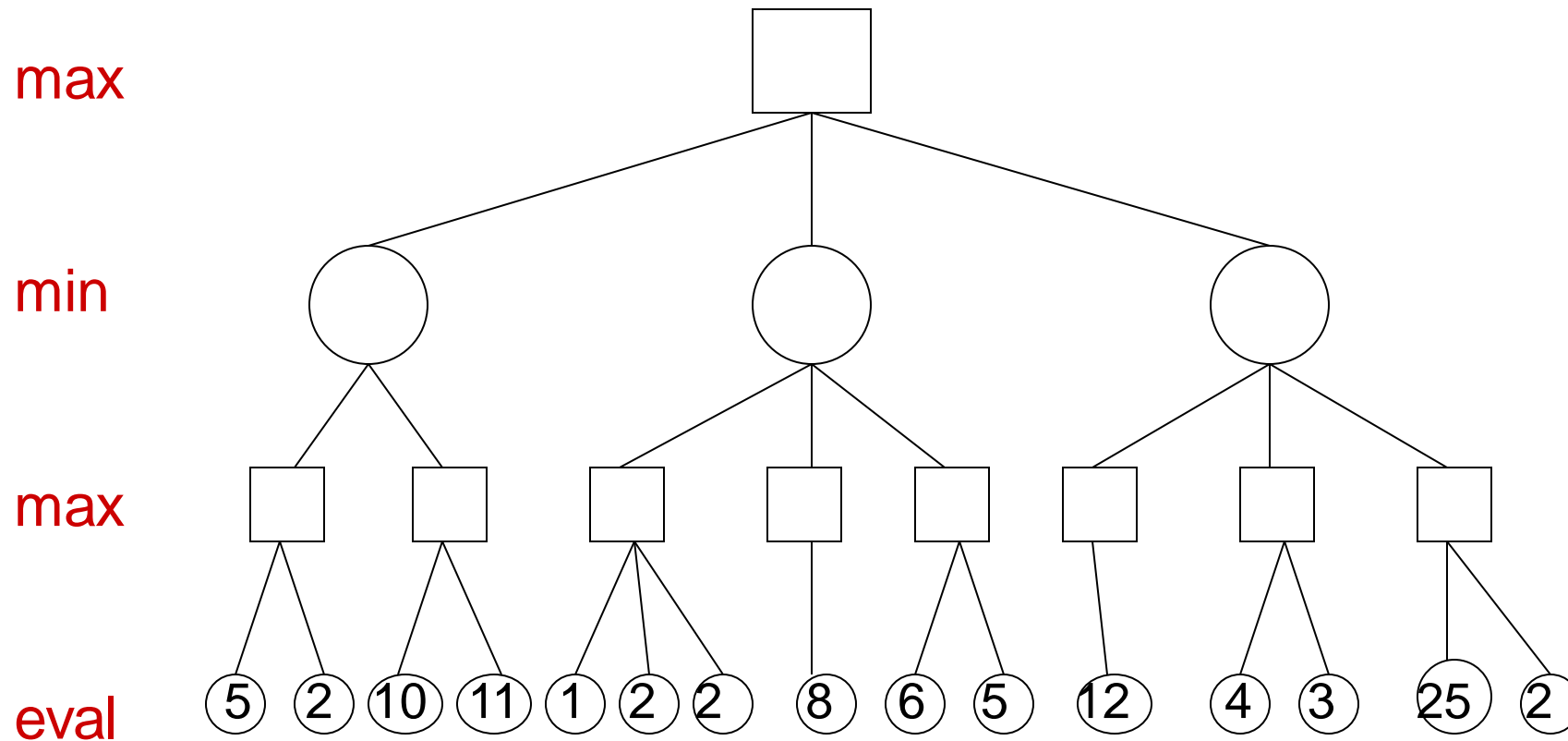
Chúng ta duyệt từ trái sang phải và phải lần lượt từng nhánh một, sau đó sang nhánh tiếp theo cùng gốc.

## Ví dụ 2 - cắt tỉa $\alpha - \beta$



$\alpha$ - $\beta$  pruning

# Bài tập: cắt tỉa $\alpha - \beta$



Nút MIN có  $\beta \leq \alpha$  của nút cha MAX bất kỳ ( $\alpha$ -cut)  
Nút MAX có  $\alpha \geq \beta$  của nút cha MIN bất kỳ ( $\beta$ -cut)

# Kết luận

- Xây dựng cây trò chơi
- Tính giá trị các nút. Giải thuật minimax.
- Lựa chọn nước đi tốt nhất.
- Cắt tỉa  $\alpha$ - $\beta$

# Tài liệu tham khảo

- Slide bài giảng về tìm kiếm đối kháng trong các trò chơi của TS Nguyễn Hữu Hòa (<http://staff.agu.edu.vn/nvhoa/AI/lecture4.pdf>)
- Thuật toán Minimax (AI trong Game) (<https://topdev.vn/blog/thuat-toan-minimax-ai-trong-game/>)
- Clip minh họa giải thuật minimax <https://youtu.be/zDskcx8FStA>