

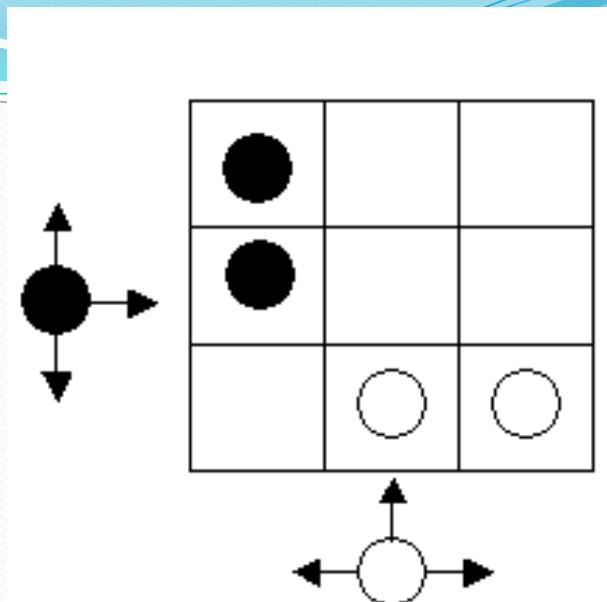
Tìm kiếm có đối thủ

TS. Nguyễn Quốc Tuấn

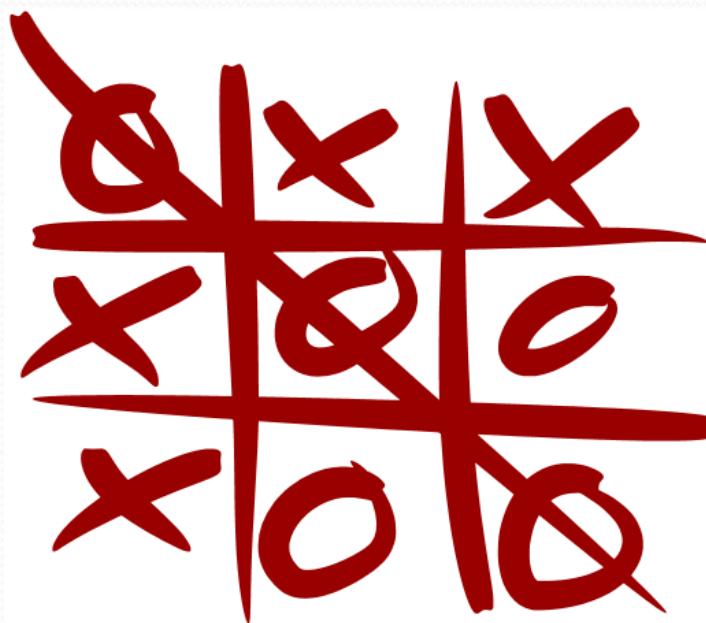
Một số ví dụ

- Ví dụ 1: Trò chơi Dodgem

- Quân đen có thể đi tới ô trống bên phải, ở trên hoặc ở dưới
- Quân trắng có thể đi tới ô trống bên trái, bên phải, ở trên
- Quân đen nếu ở cột ngoài cùng bên phải có thể đi ra ngoài bàn cờ
- Quân trắng nếu ở hàng trên cùng có thể đi ra khỏi bàn cờ
- Ai đưa cả hai quân của mình ra khỏi bàn cờ trước sẽ thắng, hoặc tạo ra tình huống mà đối phương không đi được sẽ thua.



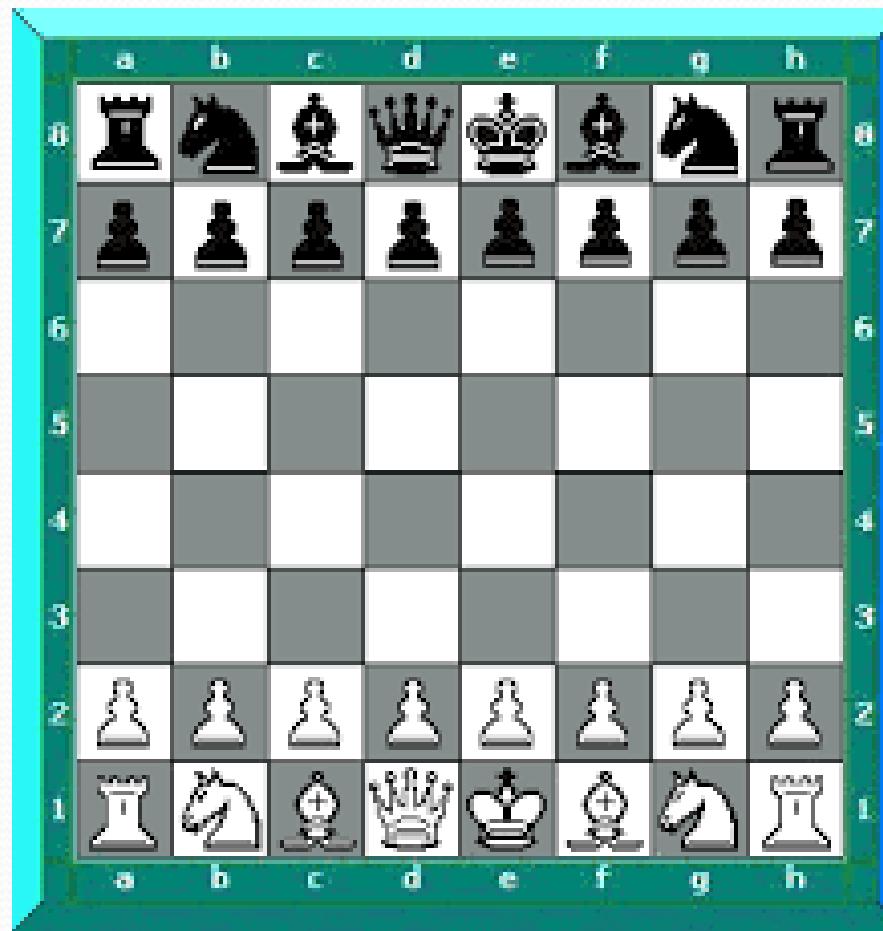
Ví dụ 2: Tic-tac-toe



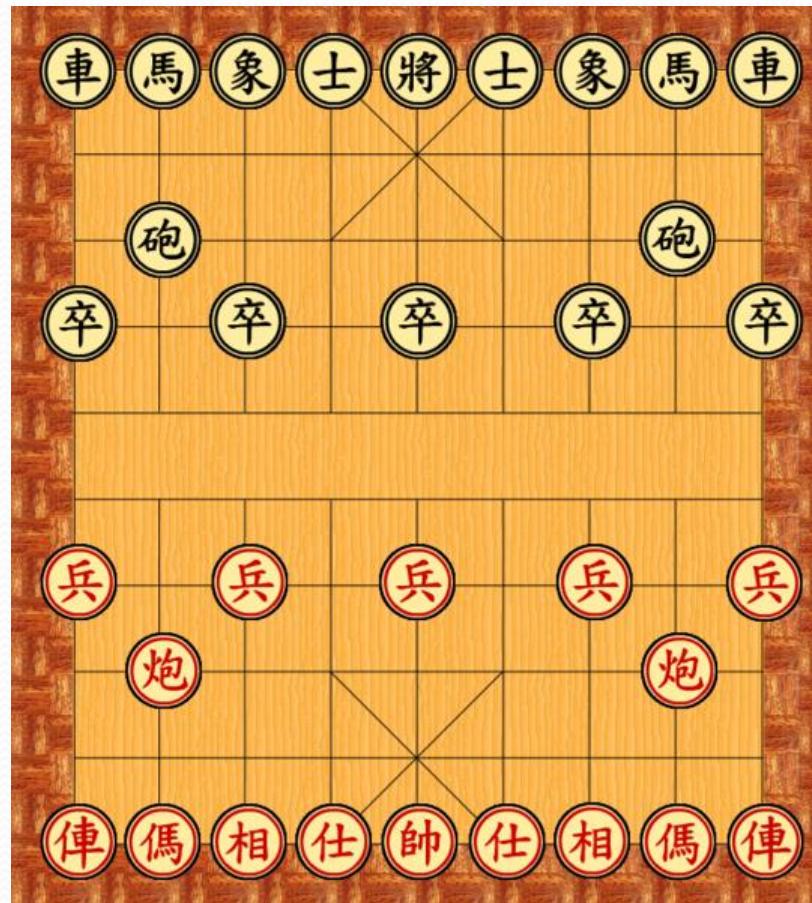
Ví dụ 3: Ô ăn quan



Ví dụ 4: Cờ vua



Ví dụ 5: Cờ tướng



Bài toán

- Giả sử một người chơi cầm quân Trắng, một người chơi cầm quân Đen
- Nghiên cứu các trò chơi:
 - Hai người chơi thay phiên nhau đưa ra các nước đi tuân theo các luật đi nào đó
 - Các luật này là như nhau cho cả hai người
 - Hai người chơi đều biết được thông tin đầy đủ về các tình thế trong trò chơi
- Bài toán: Tìm kiếm nước đi trong các nước đi có thể để thắng.

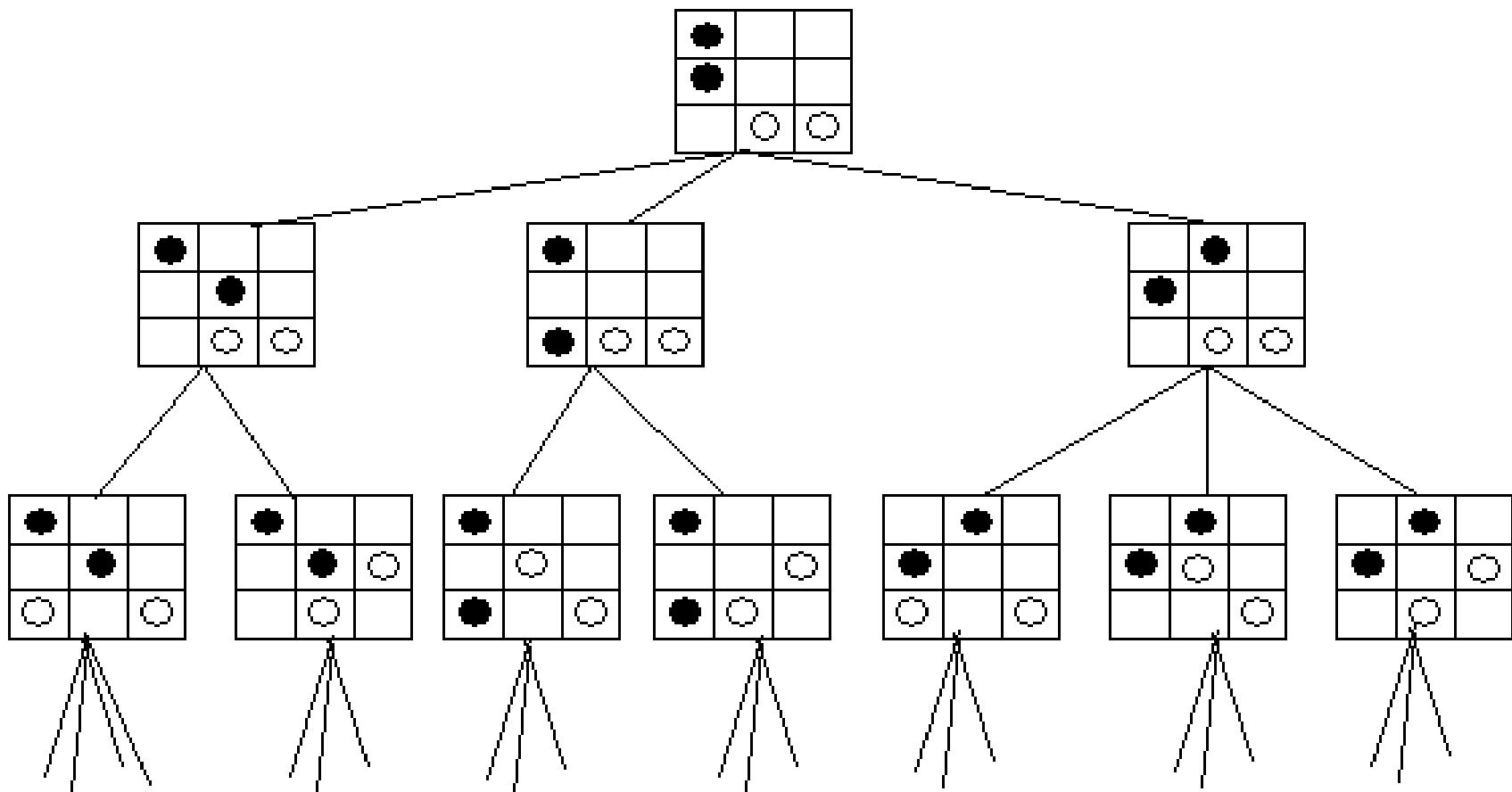
Tổng quát bài toán chơi cờ

- Vấn đề chơi cờ có thể xem như vấn đề tìm kiếm trong không gian trạng thái.
 - Mỗi trạng thái là một tình thế (sự bố trí các quân của hai bên trên bàn cờ)
 - Trạng thái ban đầu là sự sắp xếp các quân cờ của hai bên lúc bắt đầu cuộc chơi
 - Các toán tử là các nước đi hợp lệ
 - Các trạng thái kết thúc là các tình thế mà cuộc chơi dừng, thường được xác định bởi một số điều kiện dừng nào đó, các trạng thái kết thúc này thường được gắn với một giá trị gọi là giá trị của hàm kết cuộc.

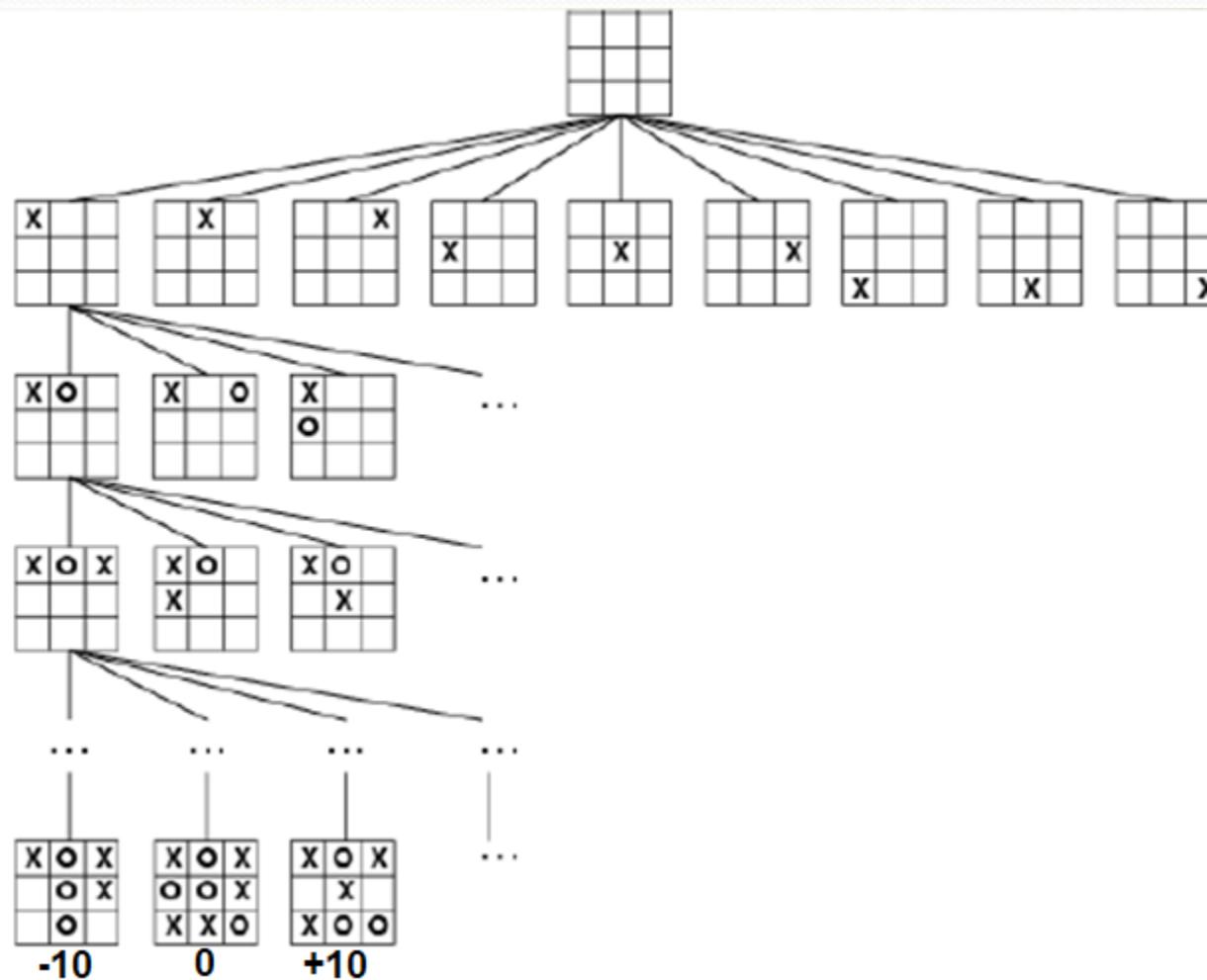
Cây trò chơi

- Cây trò chơi được xây dựng như sau:
 - Gốc của cây ứng với trạng thái ban đầu
 - Gọi đỉnh ứng với trạng thái mà Trắng (Đen) sẽ đưa ra nước đi là đỉnh Trắng (Đen)

Cây trò chơi Dodgem



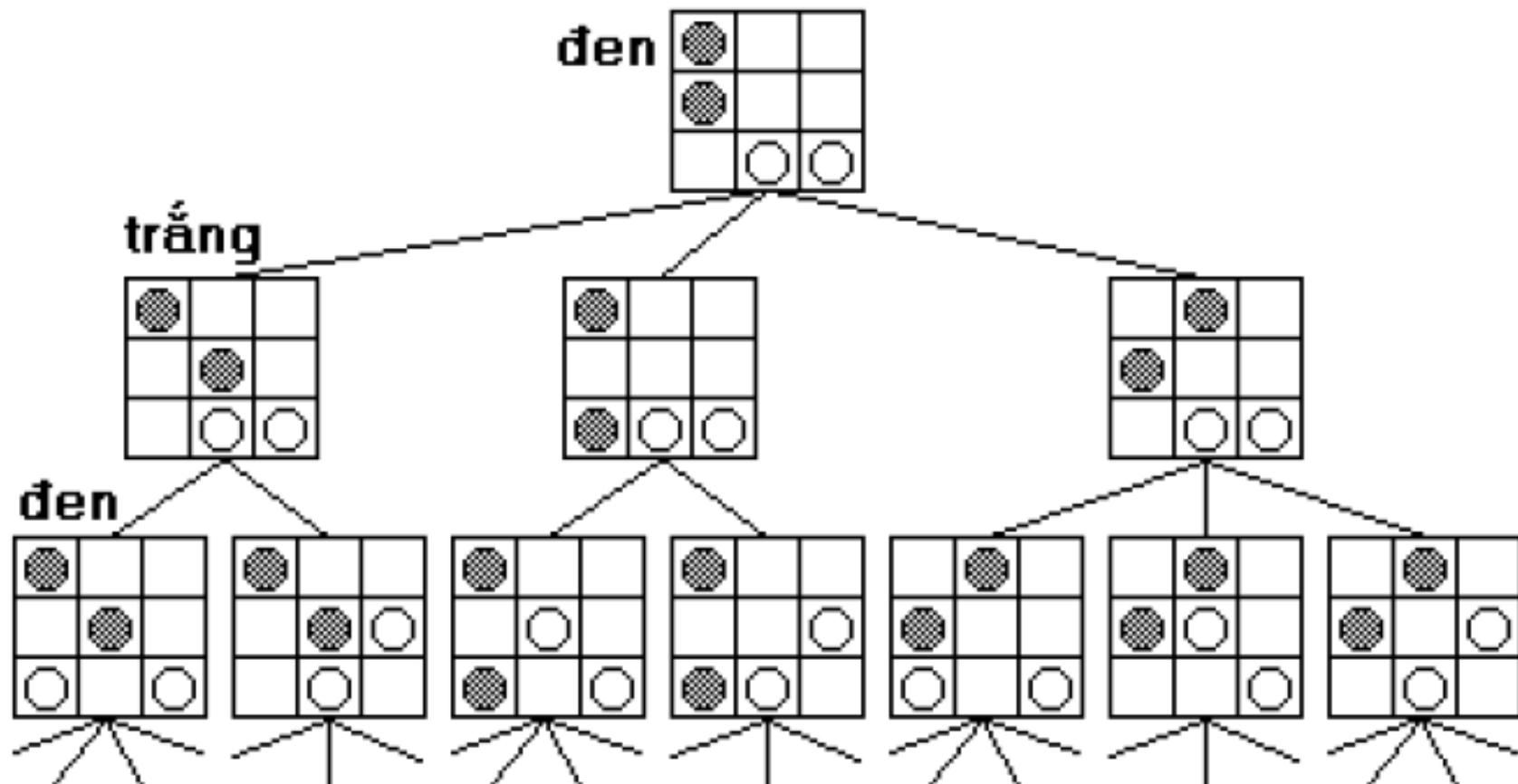
Cây trò chơi Tic-tac-toe



Chiến lược MINIMAX

- Quá trình chơi cờ là quá trình Trắng và Đen thay phiên nhau đưa ra quyết định, thực hiện một trong số các nước đi hợp lệ. Trên cây trò chơi, quá trình đó sẽ tạo ra đường đi từ gốc tới lá.
- Giả sử tới một thời điểm nào đó, đường đi đã dẫn tới đỉnh u.
 - Nếu u là đỉnh Trắng thì Trắng cần chọn đi tới một trong các đỉnh Đen v là con u.
 - Tại đỉnh Đen v mà Trắng vừa chọn. Đen sẽ phải chọn đi tới một trong các đỉnh trắng w là con của v.
- Quá trình trên sẽ dừng lại khi đạt tới một đỉnh là lá của cây.

Chiến lược MINIMAX



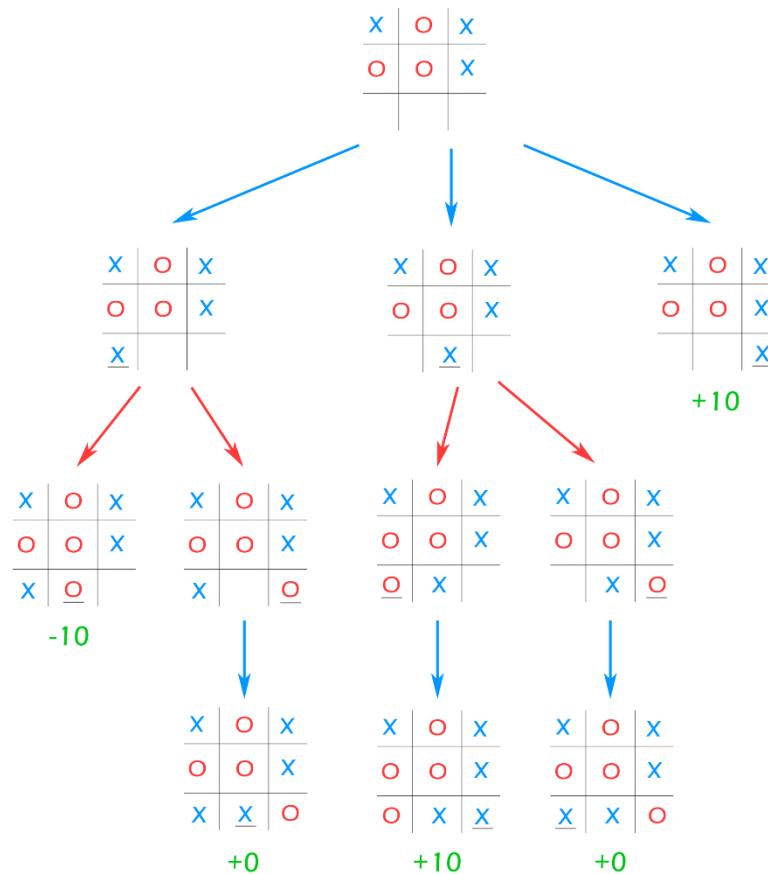
- Giả sử Trắng cần tìm nước đi tại đỉnh u.
 - Nước đi tối ưu cho Trắng là nước đi dẫn tới đỉnh con v, v là đỉnh tốt nhất (cho Trắng) trong số các đỉnh con của u
 - Đến lượt đối thủ chọn nước đi từ v, Đen cũng sẽ chọn nước đi tốt nhất cho mình.
 - Giá trị của các đỉnh lá (đỉnh ứng với các trạng thái kết thúc) là giá trị của **hàm kết cuộc**.

MAX

MIN

MAX

MIN



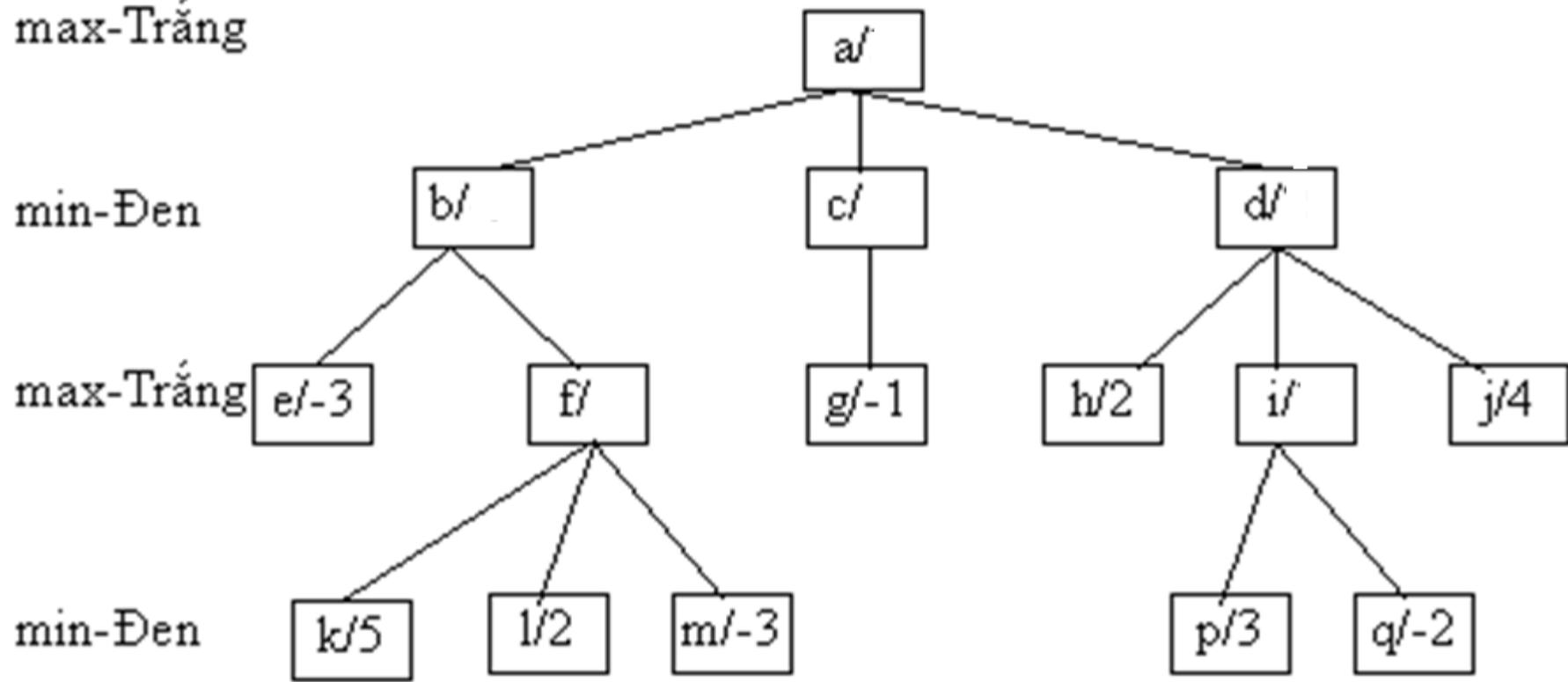
Ví dụ về xác định giá trị cho các đỉnh của cây trò chơi

max-Trắng

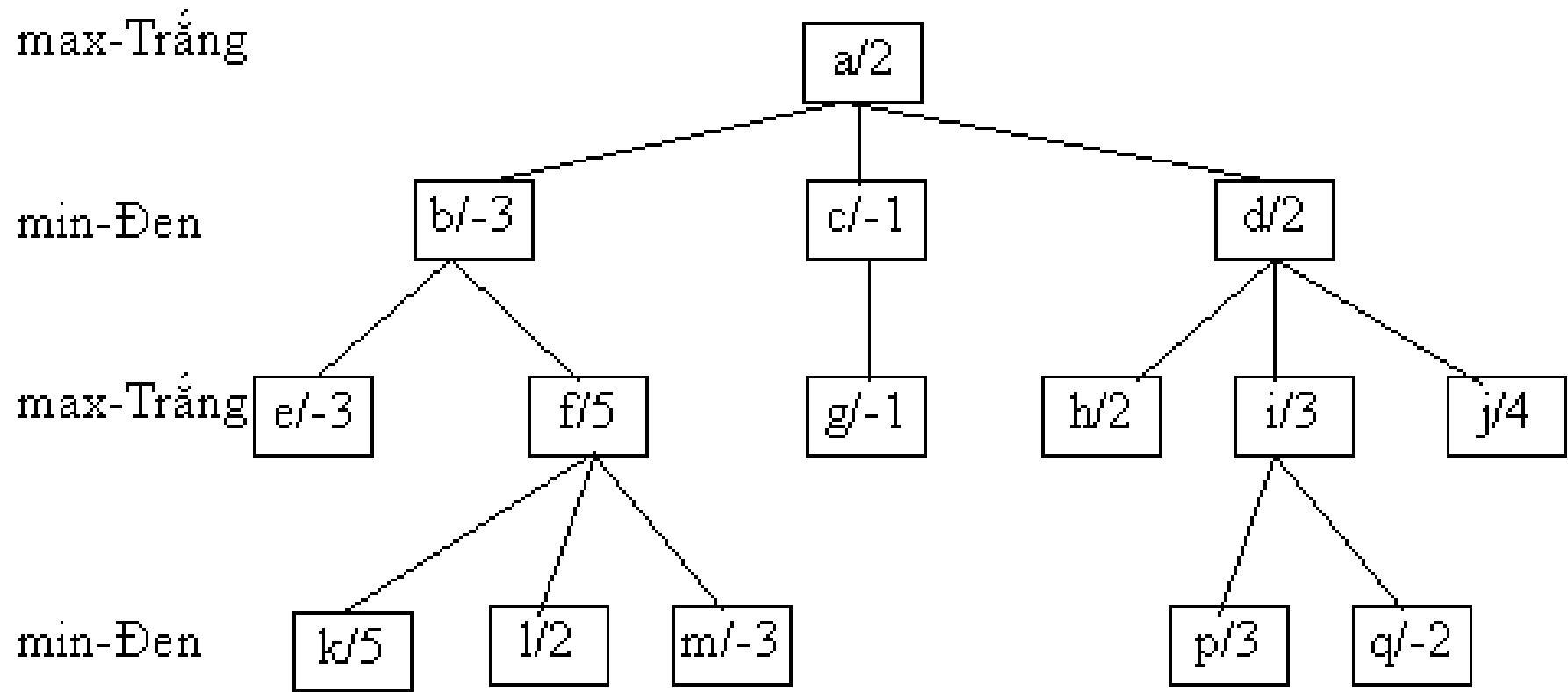
min-Đen

max-Trắng

min-Đen



Ví dụ về xác định giá trị cho các đỉnh của cây trò chơi



Các hàm đệ quy xác định giá trị cho các đỉnh

Function MaxVal(u)

Begin

if u là đỉnh kết thúc **then** MaxVal = f(u)

else MaxVal = max {MinVal(v) | v là đỉnh con u}

End;

Function MinVal(u)

Begin

if u là đỉnh kết thúc **then** MinVal = f(u)

else MinVal = min {MaxVal(v) | v là đỉnh con u}

End;

Thủ tục chọn nước đi cho Trắng

Procedure Minimax(u,v)

Begin

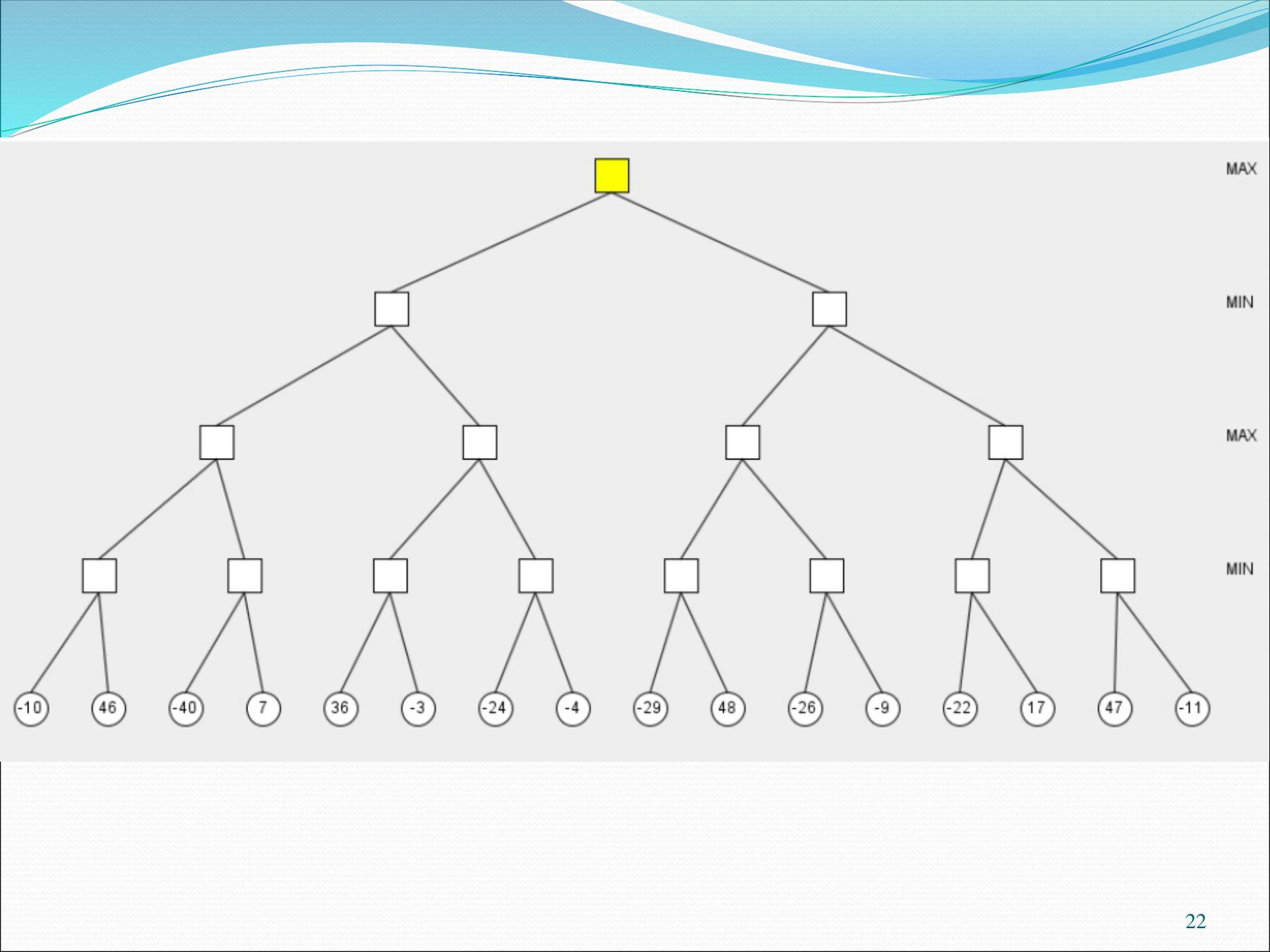
 val = $-\infty$;

for mỗi w là đỉnh con của u **do**

if val \leq MinVal(w) **then**

 {val = MinVal(w); v = w}

End;



Hạn chế

- Thuật toán Minimax là thuật toán tìm kiếm theo độ sâu.
- Về mặt lý thuyết chiến lược Minimax cho phép ta tìm ra nước đi tối ưu cho Trắng.
- Thực tế chúng ta không có đủ thời gian để xác định được nước đi tối ưu vì phải xem xét đến toàn bộ các đỉnh của cây trò chơi.

Khắc phục

- Hạn chế không gian tìm kiếm: khi cần xác định nước đi cho trắng tại u ta chỉ xem xét cây trò chơi gốc u tới độ cao h nào đó
- Vẫn sử dụng chiến lược MINIMAX để tìm kiếm nước đi
- Sử dụng ***hàm đánh giá*** cho các lá của cây trò chơi

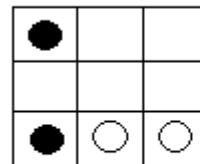
Hàm đánh giá

- Hàm đánh giá **eval** ứng với mỗi trạng thái u của trò chơi với một giá trị eval(u).
- Trạng thái u càng lợi cho Trắng nếu eval(u) là số dương càng lớn.
- Trạng thái u càng thuận lợi cho Đen nếu eval(u) là số âm càng nhỏ
- Nếu eval(u) ≈ 0 thì không có lợi thế cho ai cả
- Nếu eval(u) = +∞ thì Trắng thắng và eval(u) = -∞ thì Đen thắng
- Chất lượng của chương trình chơi cơ phụ thuộc rất nhiều vào hàm đánh giá

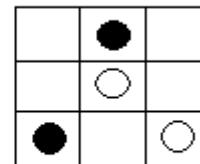
Ví dụ về xây dựng hàm đánh giá cho trò chơi Dodgem

2 trạng thái sau có lợi cho trắng hay đen, giá trị đánh giá các trạng thái này ?

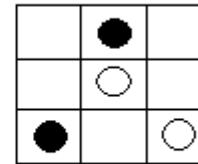
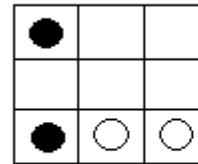
u



v



Đánh giá



1. Cho điểm các vị trí

30	35	40
15	20	25
0	5	10

Giá trị quân trắng

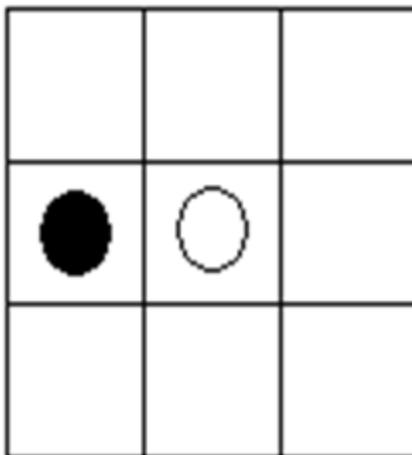
-10	-25	-40
-5	-20	-35
0	-15	-30

Giá trị quân đen

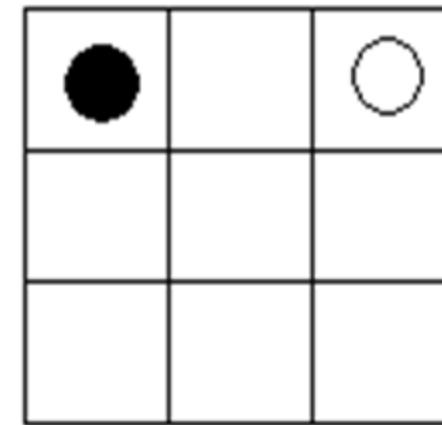
Đánh giá

2. Cho điểm cản

- Nếu quân trắng cản trực tiếp một quân đen nó sẽ được thêm 40 điểm, nếu cản gián tiếp được thêm 30 điểm
- Tương tự nếu quân đen cản trực tiếp quân trắng nó được thêm -40 điểm còn cản gián tiếp nó được thêm -30 điểm



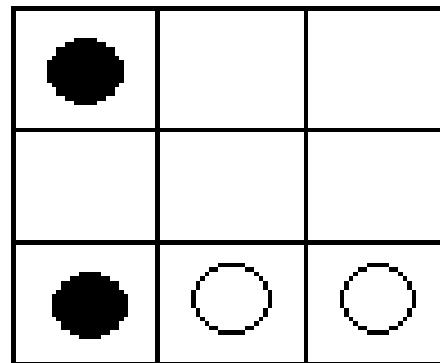
Trắng cản trực tiếp đen được 40 điểm



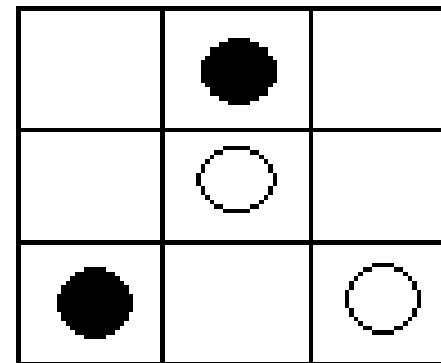
Trắng cản gián tiếp đen được 30 điểm

Áp dụng qui tắc trên cho 2 trạng thái :

u



v



30	35	40
15	20	25
0	5	10

Giá trị quân trắng

-10	-25	-40
-5	-20	-35
0	-15	-30

Giá trị quân đen

Các hàm xác định giá trị các đỉnh - cải tiến

Function MaxVal(u,h)

Begin

if h=0 hoặc u là đỉnh kết thúc **then** MaxVal = eval(u)

else MaxVal = max {MinVal(v,h-1)| v là đỉnh con u}

End;

Function MinVal(u,h)

Begin

if h=0 hoặc u là đỉnh kết thúc **then** MinVal = eval(u)

else MinVal = min {MaxVal(v,h-1)| v là đỉnh con u}

End;

Thủ tục chọn nước đi cho Trắng -cải tiến

Procedure Minimax(u,v,h);

Begin

 val = $-\infty$;

for mỗi w là đỉnh con của u **do**

if val \leq MinVal(w,h-1) **then**

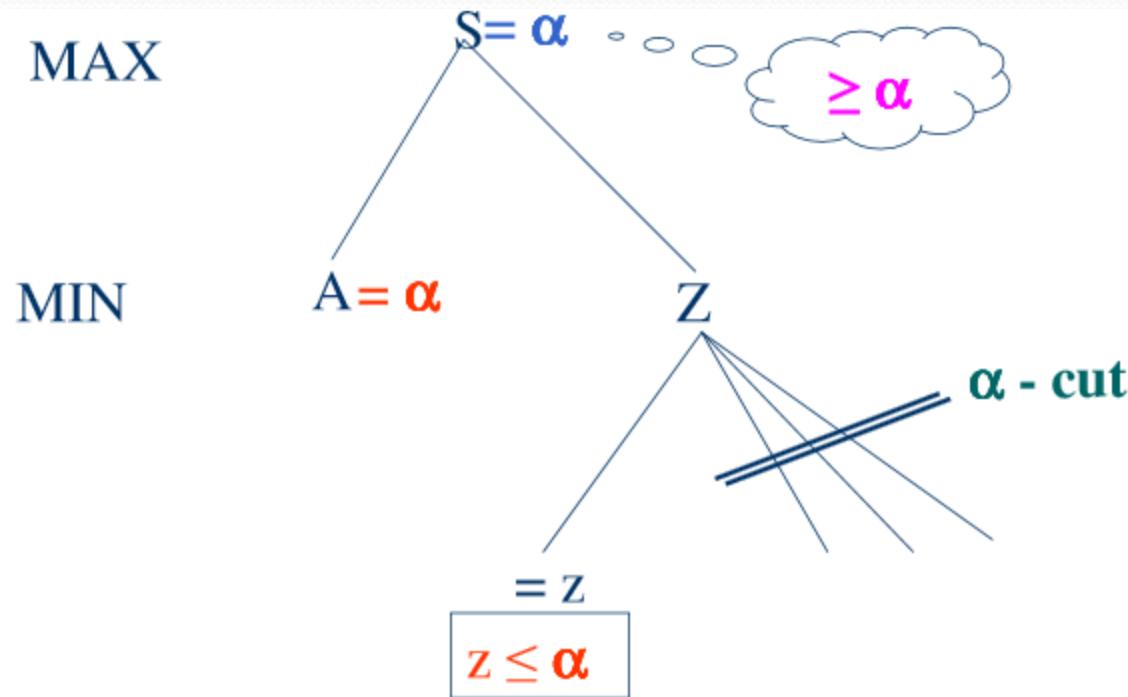
 {val = MinVal(w,h-1); v = w}

End;

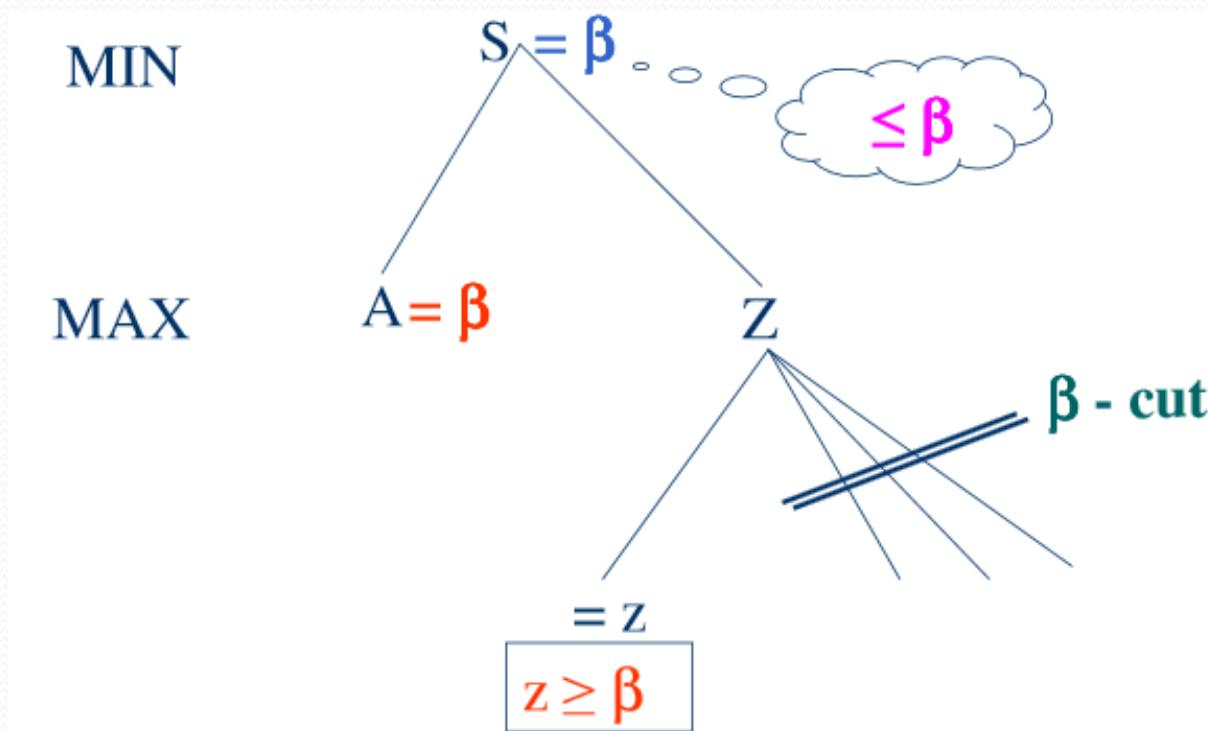
Phương pháp cắt tỉa ALPHA-BETA

- Để tìm nước đi cho Trắng tại u, chỉ xét các đỉnh của cây trò chơi gốc u với độ sâu hạn chế h
- Có thể giảm bớt số đỉnh cần đánh giá của cây trò chơi gốc u (độ sâu h) mà không ảnh hưởng gì tới sự đánh giá u
- Phương pháp cắt cùt alpha - beta cho phép ta cắt bỏ các nhánh không cần thiết cho sự đánh giá đỉnh u

Cắt tỉa α



Cắt tỉa β



Kỹ thuật cài đặt

- Dùng tham số α để ghi lại giá trị lớn nhất trong các giá trị của các đỉnh con đã đánh giá của một đỉnh trắng
- Tham số β ghi lại giá trị nhỏ nhất trong các đỉnh con đã đánh giá của một đỉnh đen
- Giá trị α, β sẽ được cập nhật trong quá trình tìm kiếm
- $\text{MaxVal}(u, \alpha, \beta)$ - hàm xác định giá trị của đỉnh trắng u
- $\text{MinVal}(u, \alpha, \beta)$ - hàm xác định giá trị của đỉnh đen u

Function MaxVal(u, α , β)

Begin

if u là đỉnh kết thúc **then return** eval(u)

else{ val= - ∞ ;

for mỗi đỉnh v là con của u **do**

 {

 val = max (val, MinVal(v, α , β));

if val $\geq \beta$ **then return** val } // cắt bỏ cây con từ các đỉnh v còn lại

α = max(α ,val);

 }

return val;

}

End:

Function MinVal(u, α , β)

Begin

if u là đỉnh kêt thúc **then** return eval(u)

else{ val=+ ∞ ;

for mỗi đỉnh v là con của u **do**{

 val=min(val, MaxVal(v, α , β));

if val $\leq \alpha$ **then** return val }// cắt bỏ cây con từ các đỉnh v còn lại

$\beta = \min (\beta, \text{val})$;

 }

 return val;

}

End;

Procedure Alpha_beta(u,v)

Begin

$\alpha = -\infty$; $\beta = +\infty$;

 for mỗi đỉnh w là con của u do

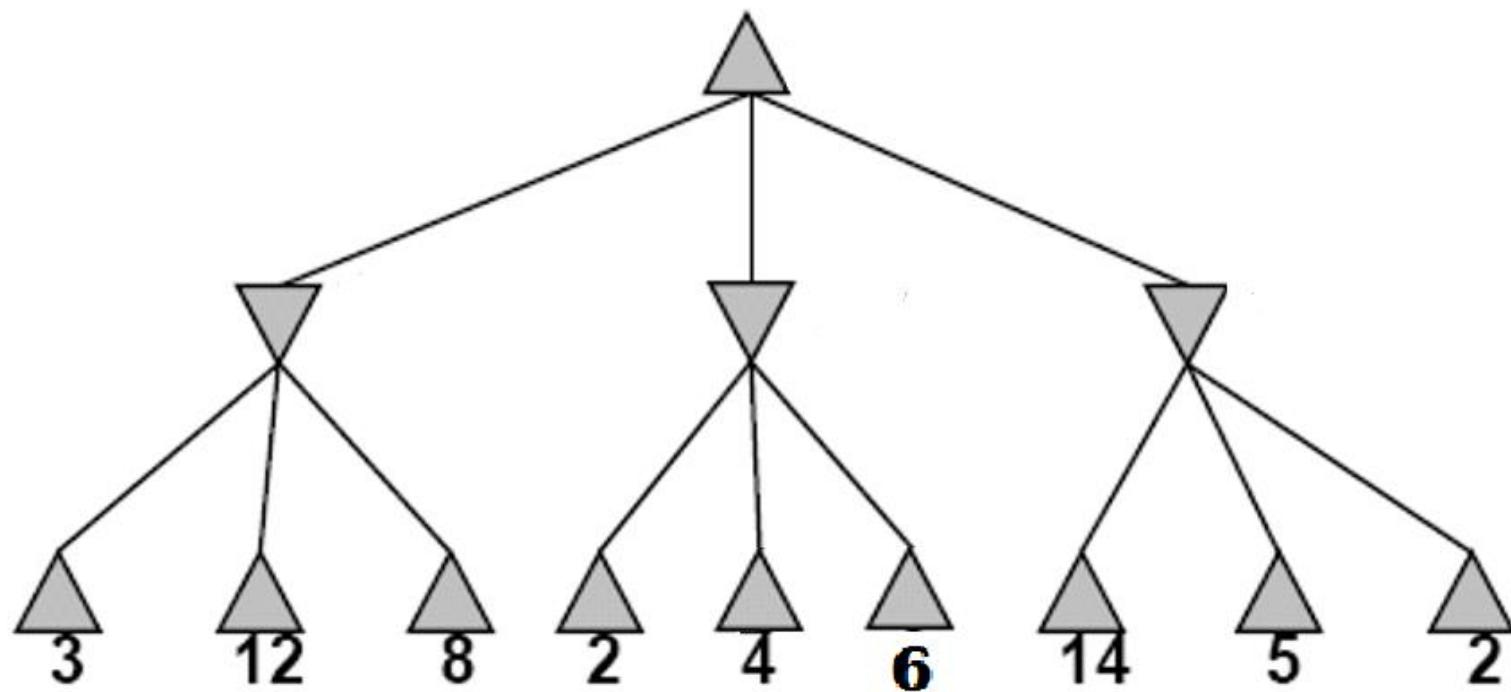
 if $\alpha \leq \text{MinVal}(w, \alpha, \beta)$ then

$\{\alpha = \text{MinVal}(w, \alpha, \beta); v = w;\}$

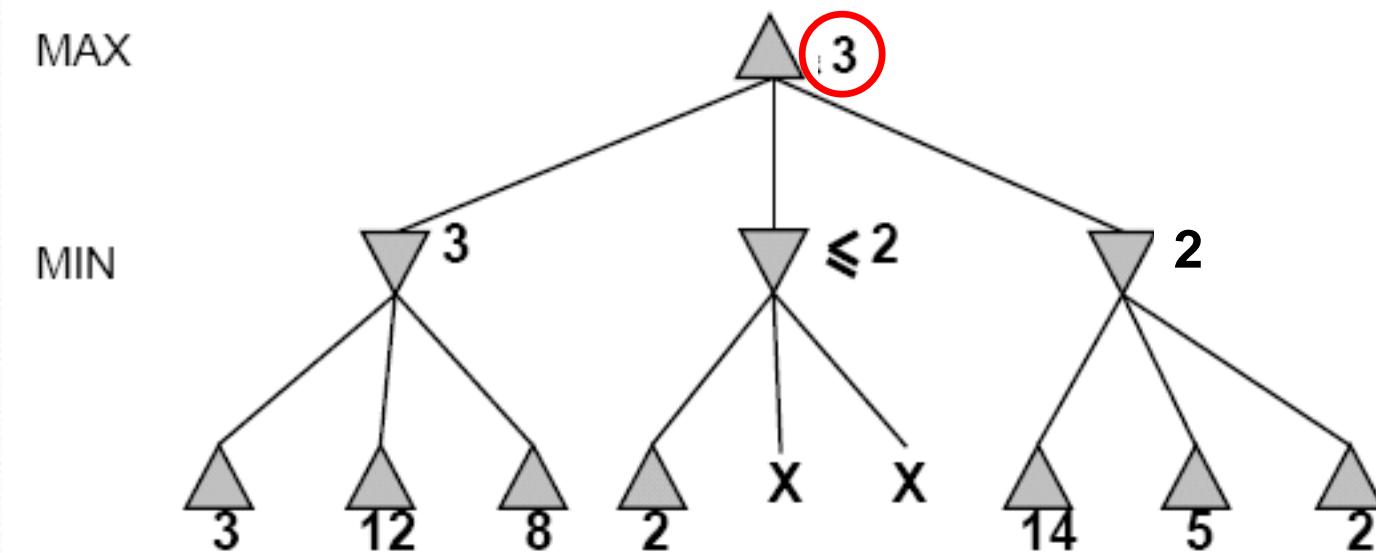
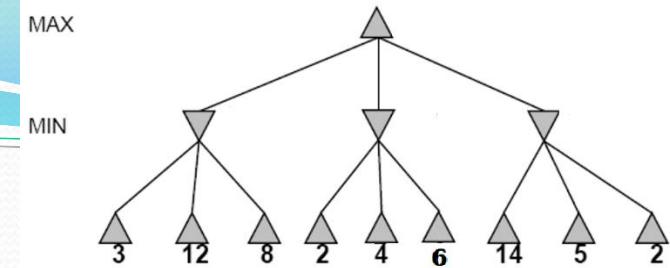
End;

MAX

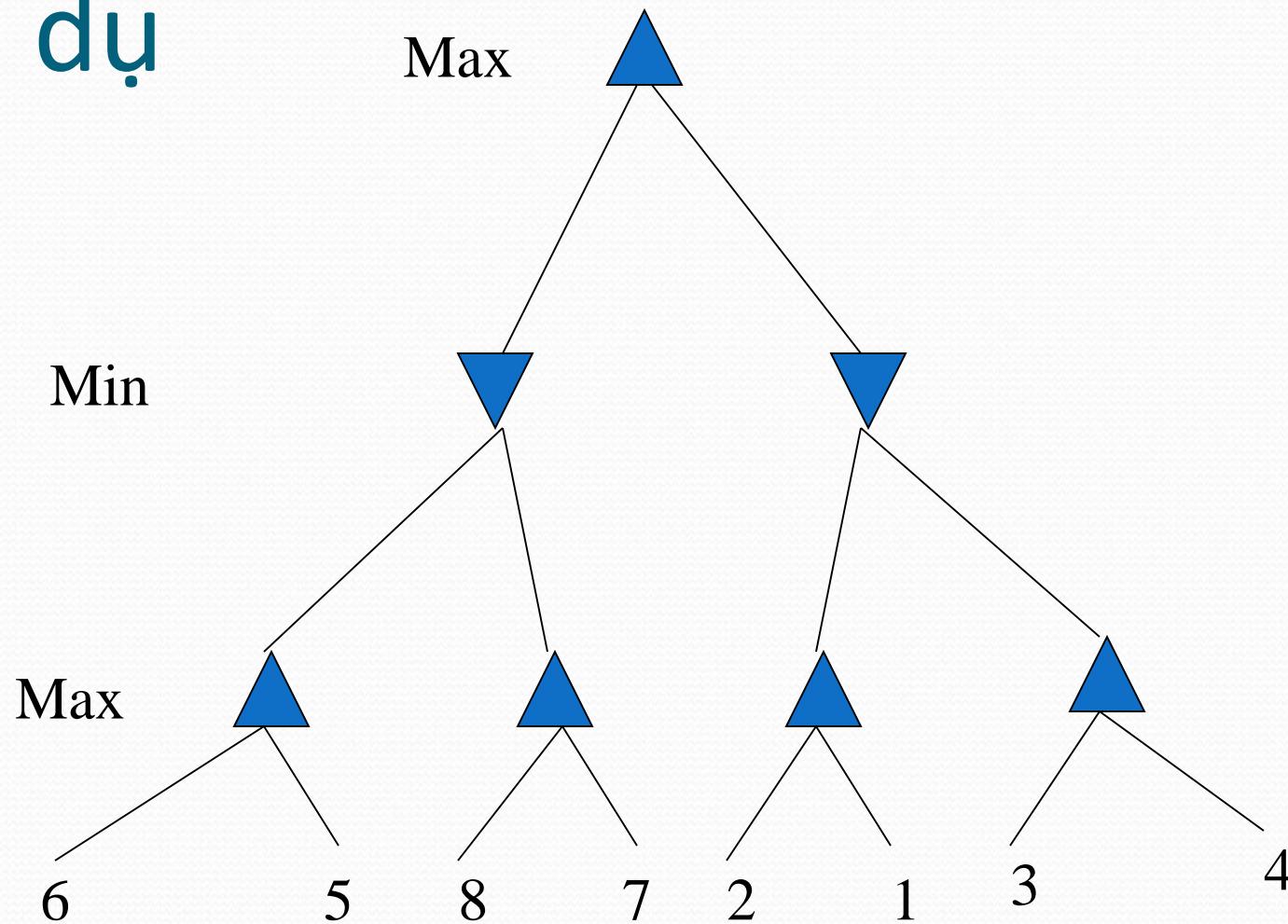
MIN



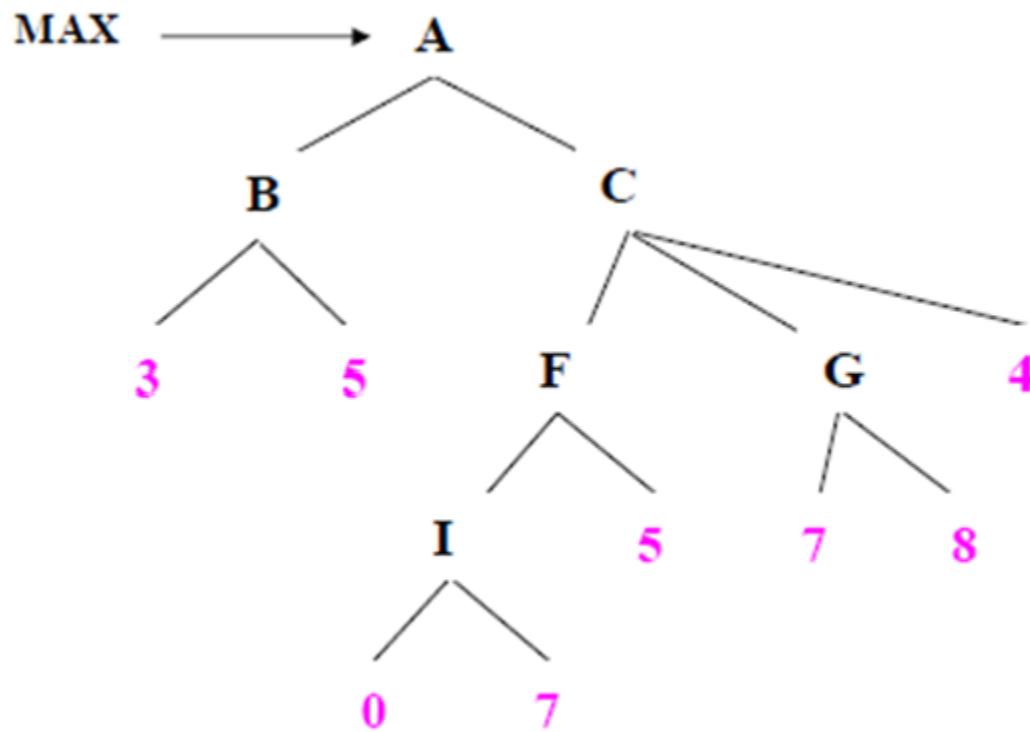
Ví dụ Alpha-Beta (tiếp)



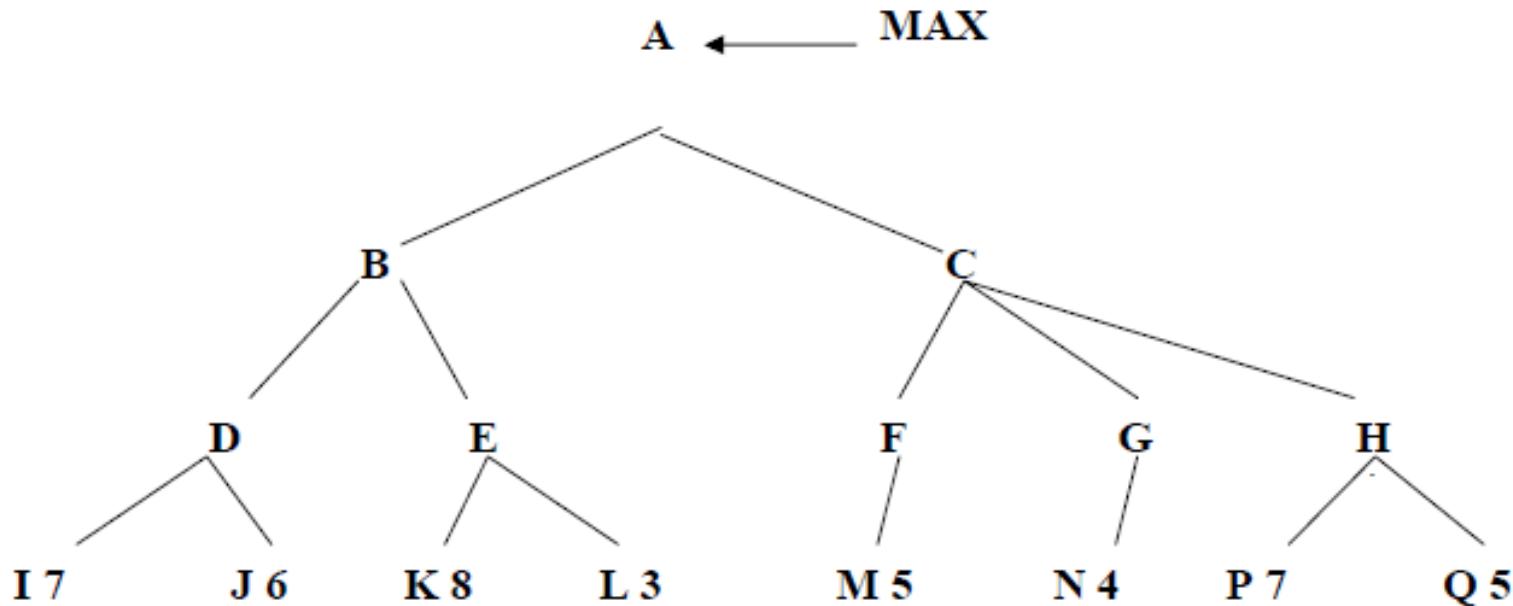
Ví dụ



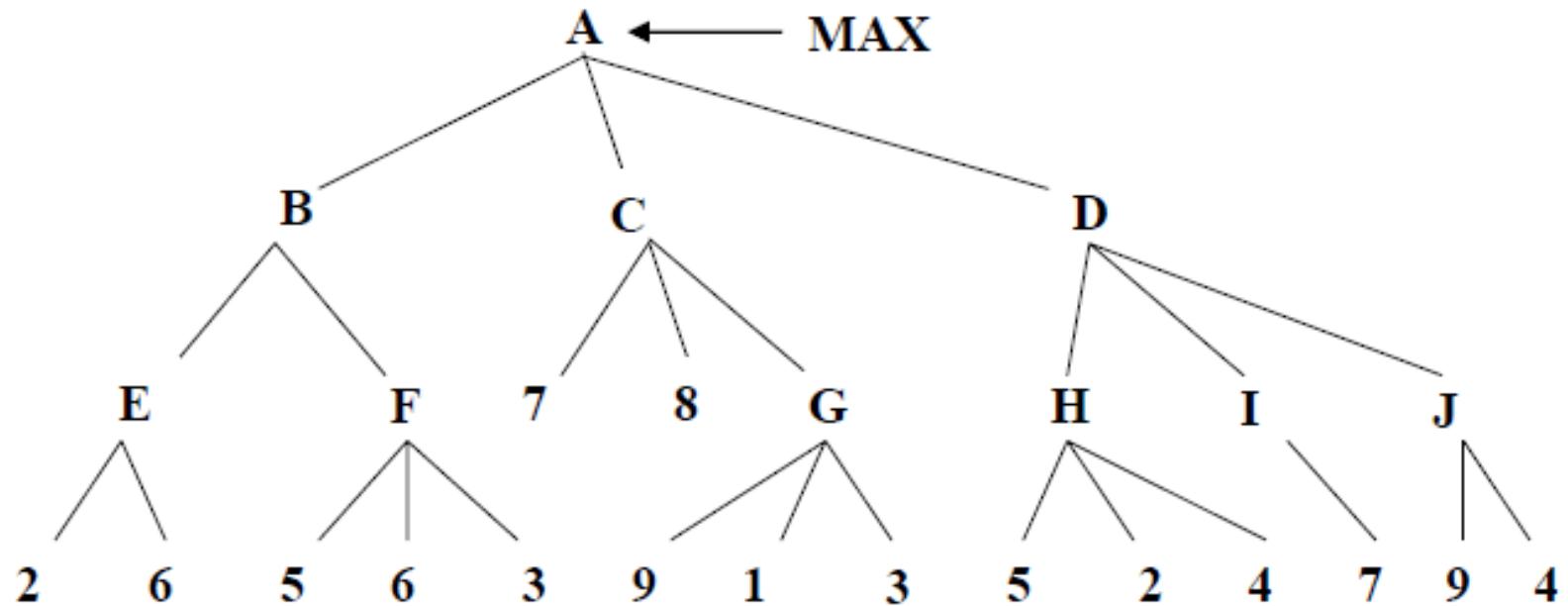
Minimax



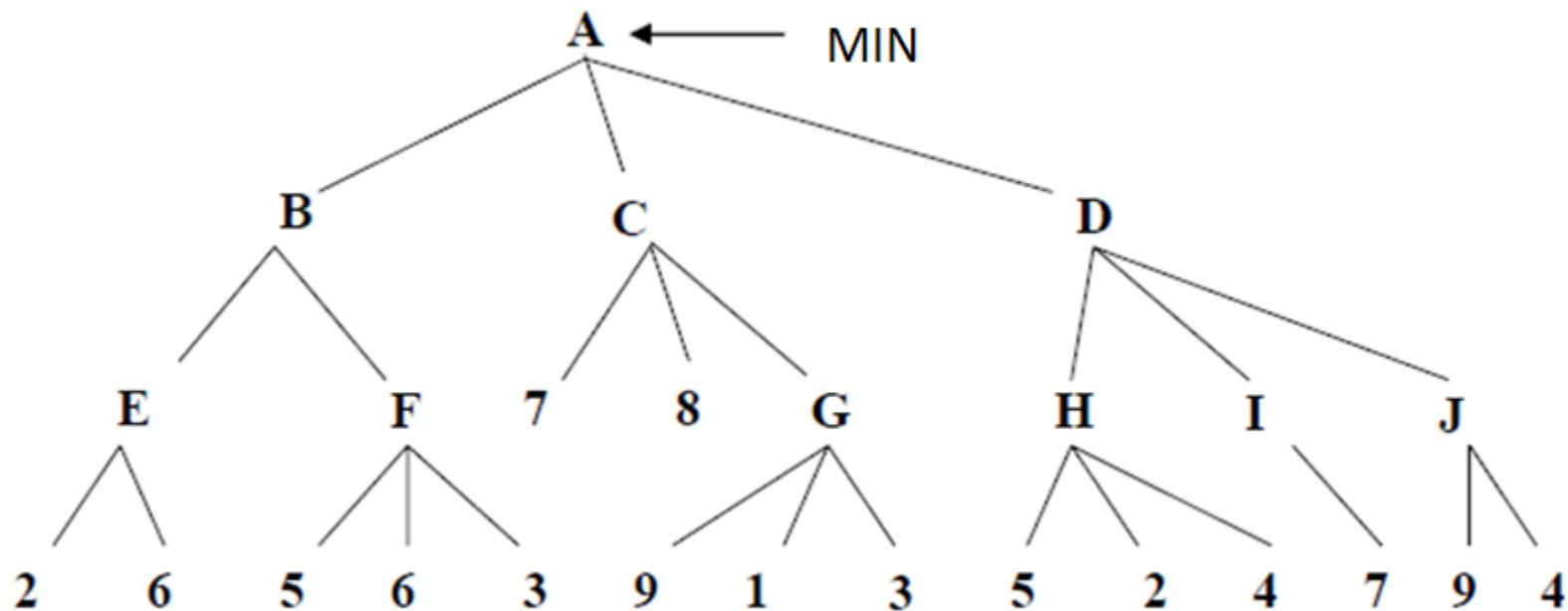
Alpha-Beta



Alpha-Beta



Alpha-Beta



max

a

min

b

c

d

max

e

min

4

8

8

3

7

9

4

1

0

5

7

8

1

5

f

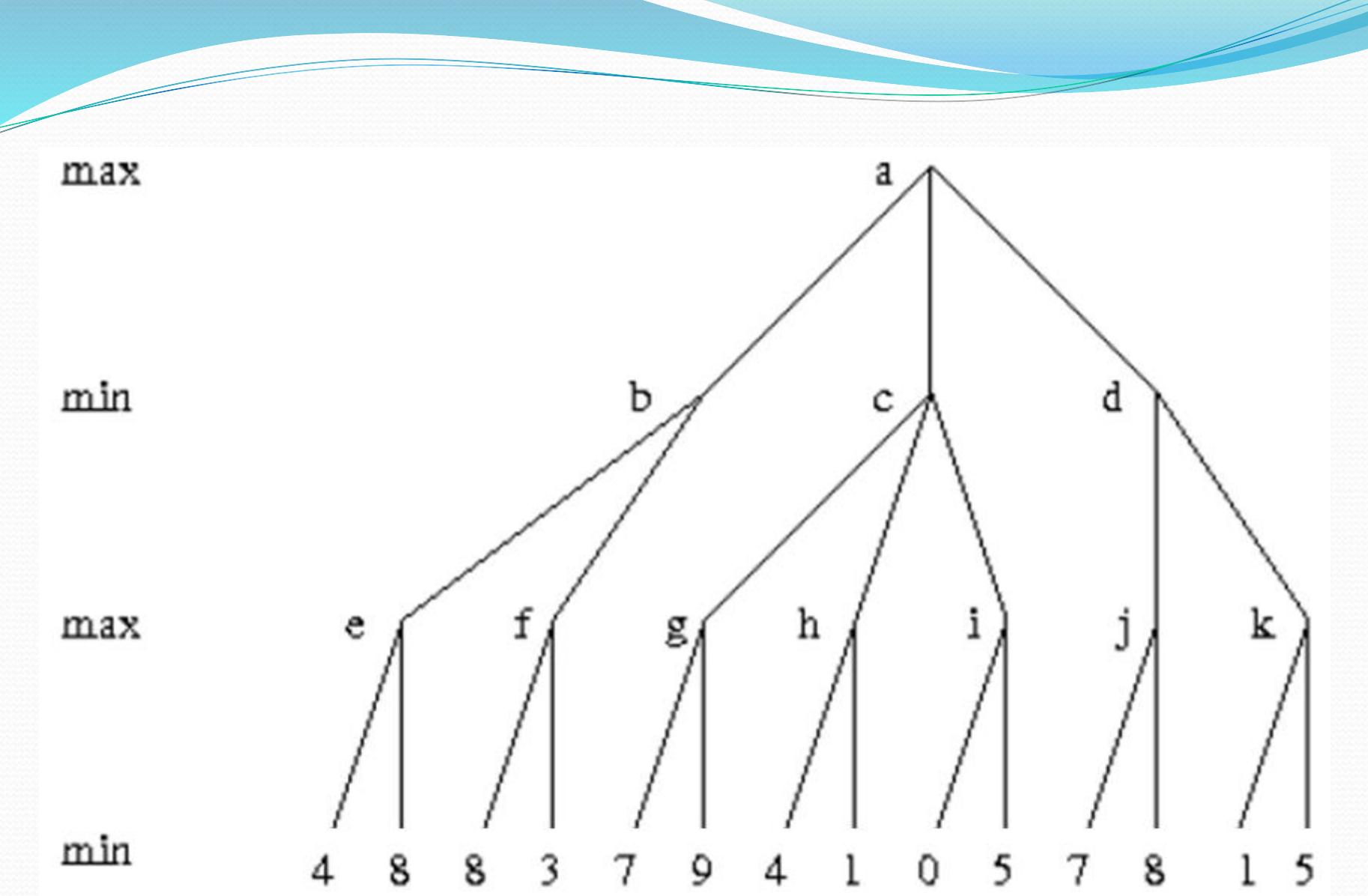
g

h

i

j

k

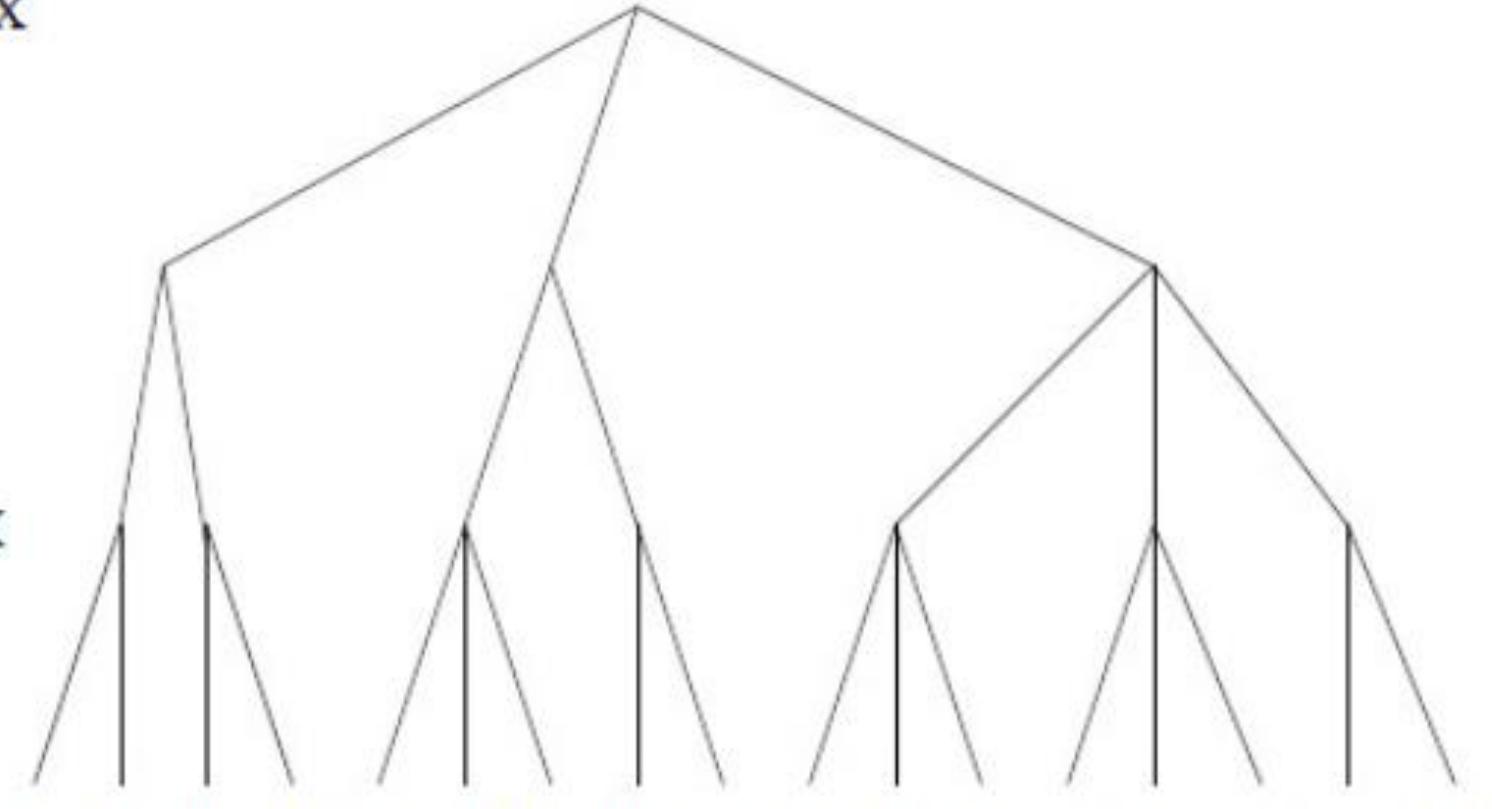




MAX

MIN

MAX



3 7 10 10 5 15 11 13 15 3 7 15 9 3 7 9 15