

## Nội dung

- Biểu diễn bài toán trong KGTT
- Tìm kiếm mù (uninformed search)
- Tìm kiếm heuristic (informed search)
- **Cây trò chơi, cắt tỉa alpha -beta**

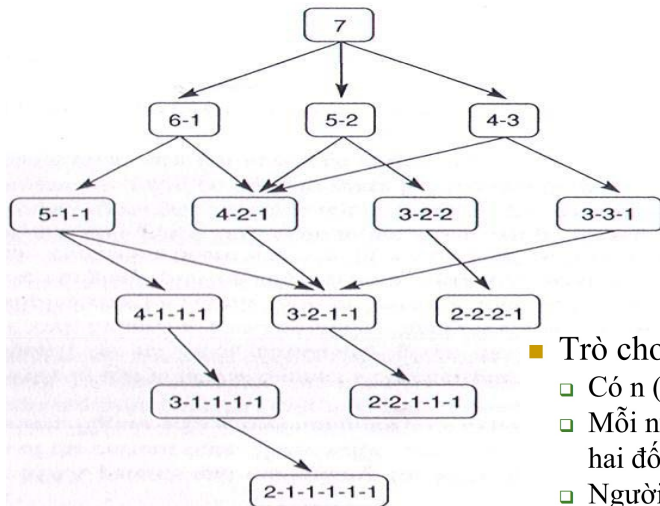
113

## Cây trò chơi

- Trò chơi một trong những đặc tính được xem là “thông minh” của con người
- Trò chơi là phiên bản “F1” của AI
- Đã đạt được những thành tựu đáng kể
- Ở đây ta chỉ xem xét các dạng trò chơi trí tuệ, đối kháng (board game)

114

## Không gian trạng thái trò chơi Nim

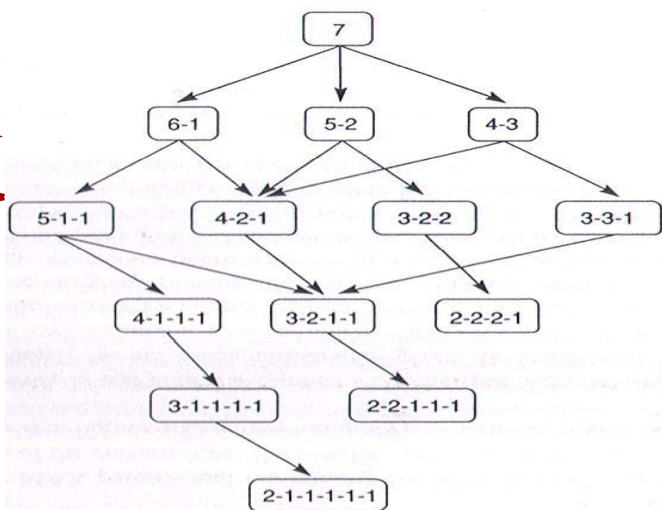
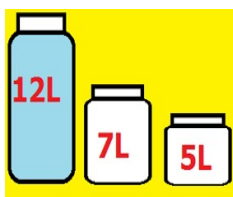


### Trò chơi Nim

- Có  $n$  ( $n > 2$ ) đồng xu
- Mỗi nước đi, người chơi chia các đồng xu này thành hai đồng nhỏ có số lượng mỗi đồng khác nhau
- Người thua sẽ là người cuối cùng không chia được theo yêu cầu của bài toán

115

## Sự khác biệt giữa các bài toán?



116

## 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*

117

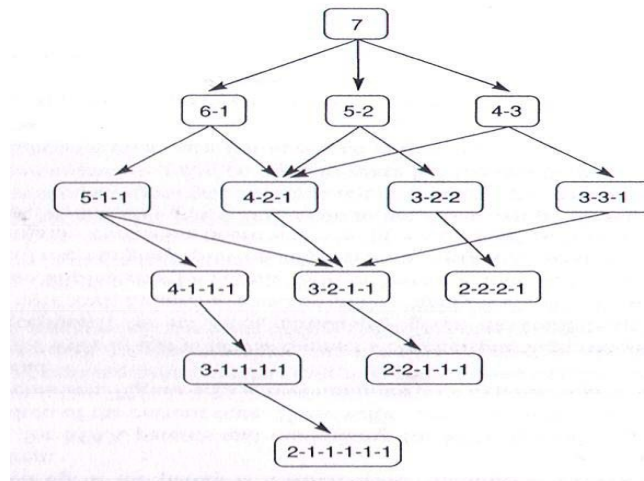
## Ứng dụng Heuristic trong các trò chơi

- 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.*

118

## Giải thuật minimax

### ■ Không gian trạng thái trò chơi Nim



119

## Giải thuật minimax

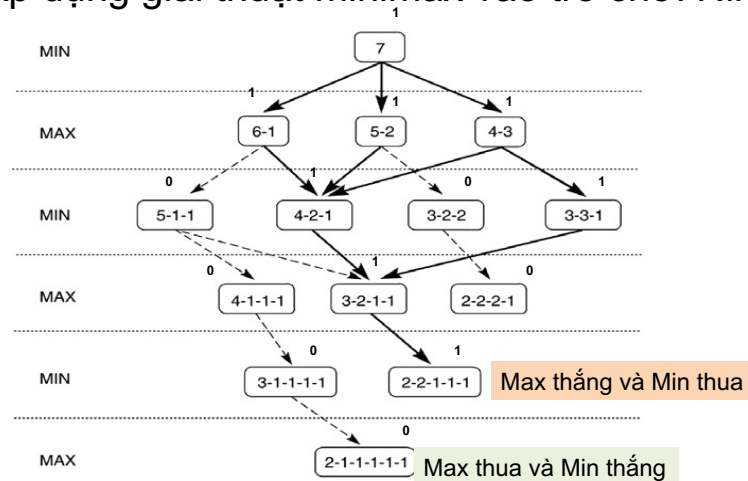
### ■ Giải thuật minimax:

- 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 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.

120

## Giải thuật minimax

### ■ Áp dụng giải thuật minimax vào trò chơi Nim



121

## Giải thuật minimax

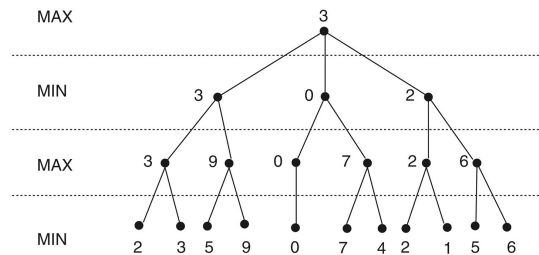
### ■ Minimax đến độ sâu lớp cố định

- 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.

122

## Minimax với độ sâu lớp cố định

- Minimax đối với một KGTT giả định.

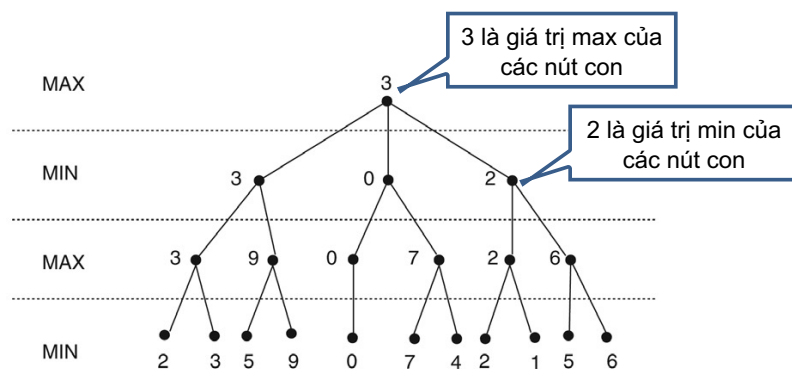


- Các nút lá được gán các giá trị **heuristic**
- Còn giá trị tại các nút trong là các giá trị nhận được dựa trên giải thuật Minimax

123

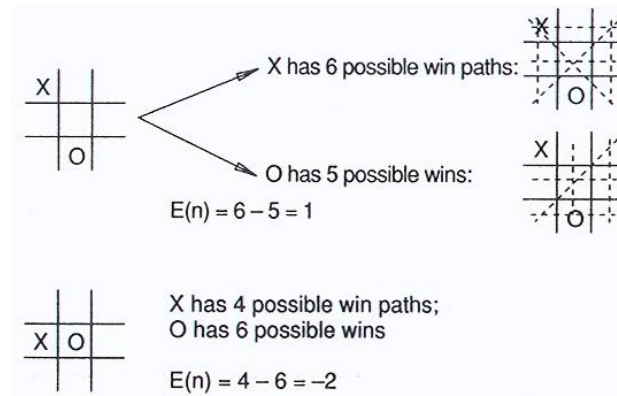
## Giải thuật minimax

- Minimax đối với một KGTT giả định:



124

## 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

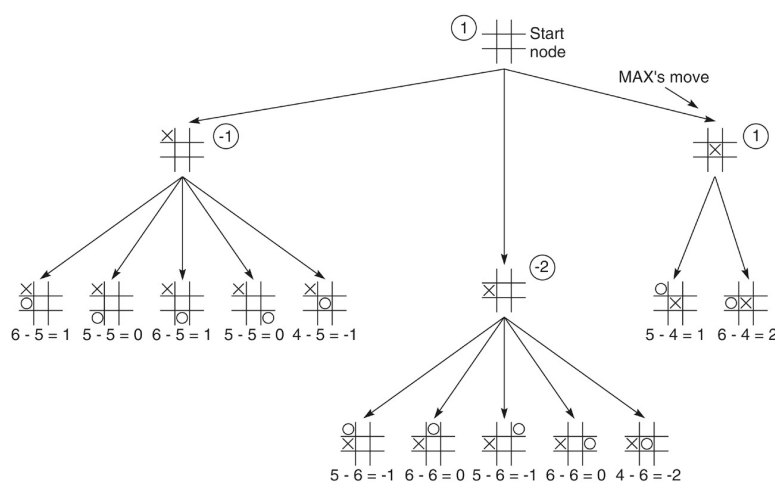
$E(n)$  là trị số đánh giá tổng cộng cho trạng thái  $n$

C 4 – Tìm kiếm Heuristic

TTNT, p.125

125

## 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

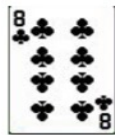
$O(n)$  là tổng số đường thắng có thể của đối thủ

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

126

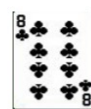
## Bài toán bắt lá bài

- 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



127

## Bài toán bắt lá bài

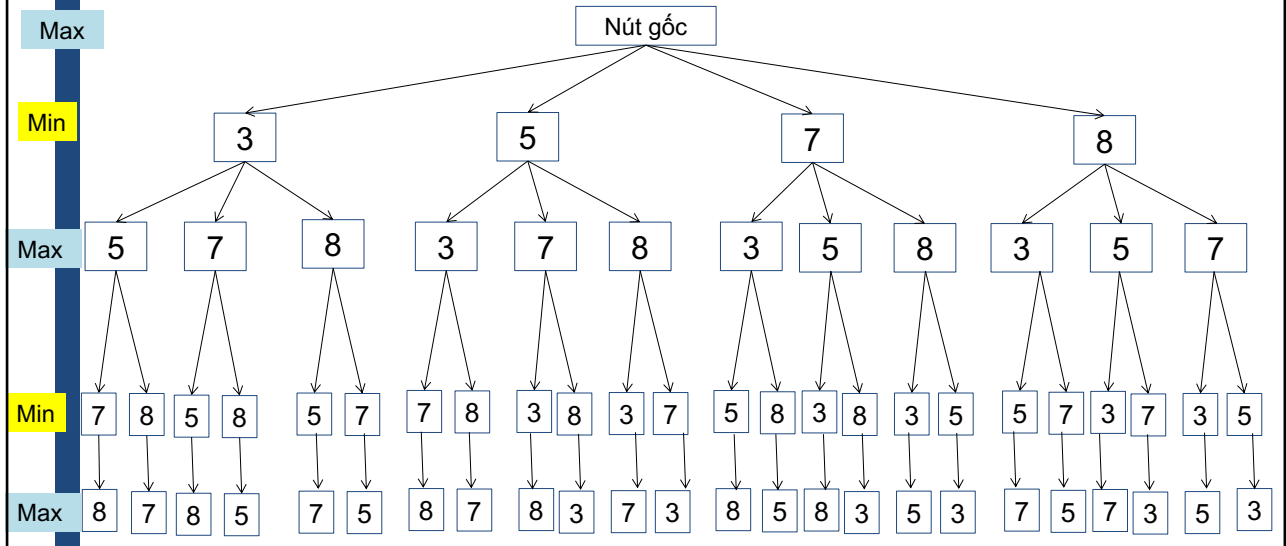


- Để có thể thắng trong trò chơi trên
  - Vẽ toàn bộ không gian trạng thái
  - Gắn điểm số vào mỗi trạng thái kết thúc
  - Sử dụng giải thuật Minimax để cập nhật điểm số từ nút lá đến nút gốc
  - Chọn đường đi có điểm số lớn nhất

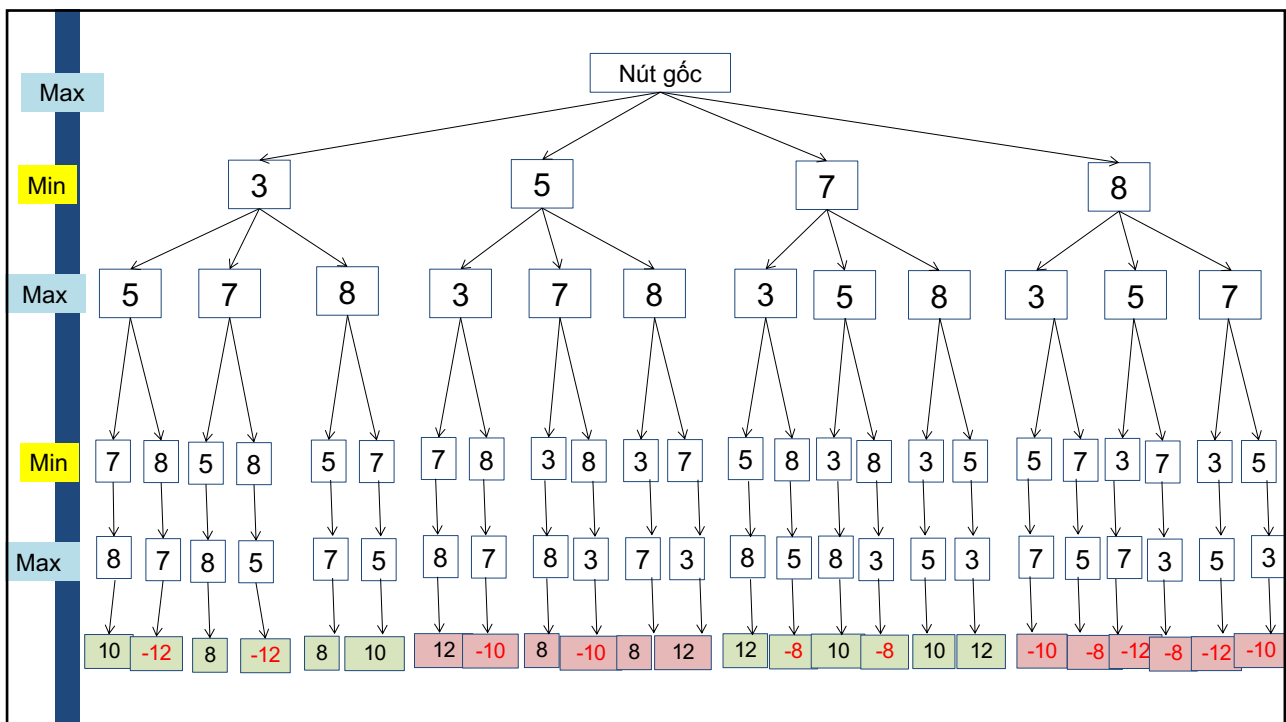
128



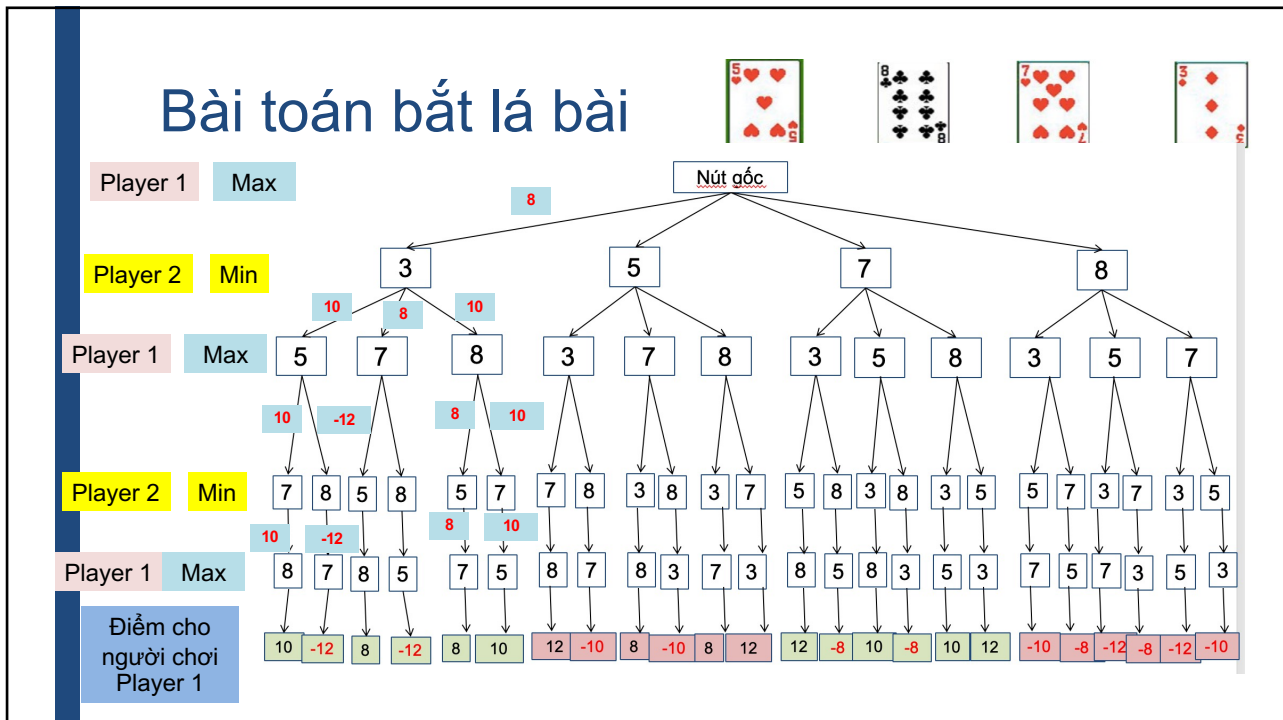
# Bài toán bắt lá bài



129



130



131

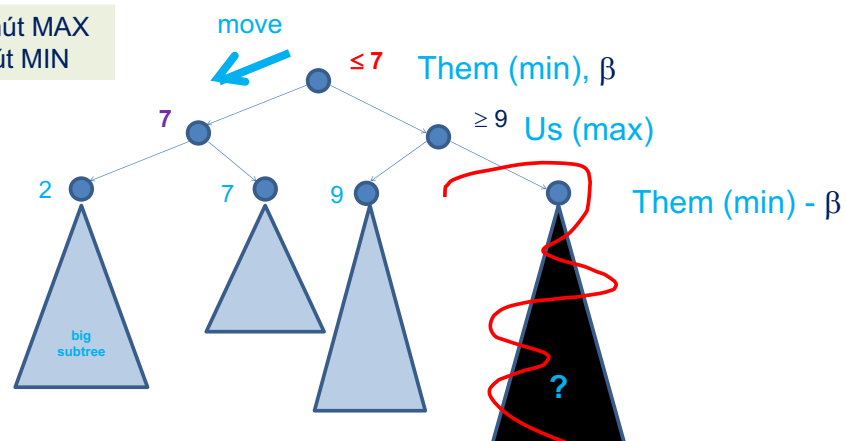
## Thủ tục cắt tỉa alpha – beta

- 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)

132

$\beta$ -cut

$\alpha$  gắn với các nút MAX  
 $\beta$  gắn với các nút MIN

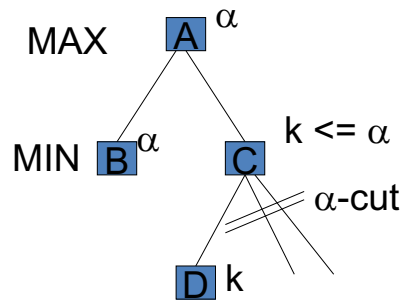


Nút MIN có  $\beta \leq \alpha$  của nút cha MAX bất kỳ ( $\alpha$ -cut)  
 Nút MAX có  $\alpha \geq \beta$  của nút cha MIN bất kỳ ( $\beta$ -cut)

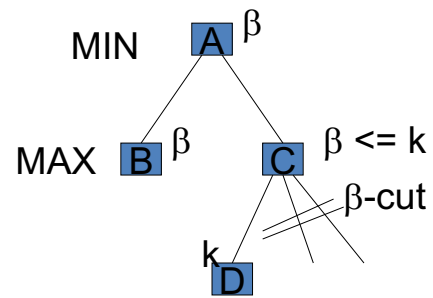
134

## Thủ tục cắt tỉa alpha – beta

### ■ $\alpha$ -cut



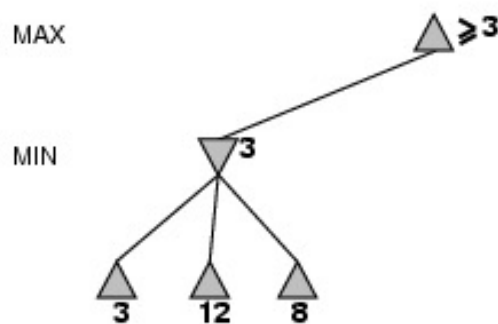
### $\beta$ -cut



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)

135

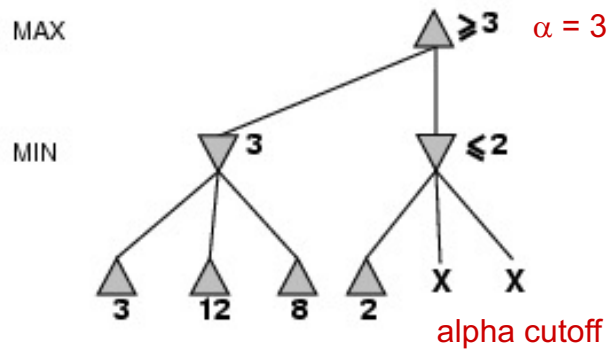
## $\alpha$ - $\beta$ pruning example



136

136

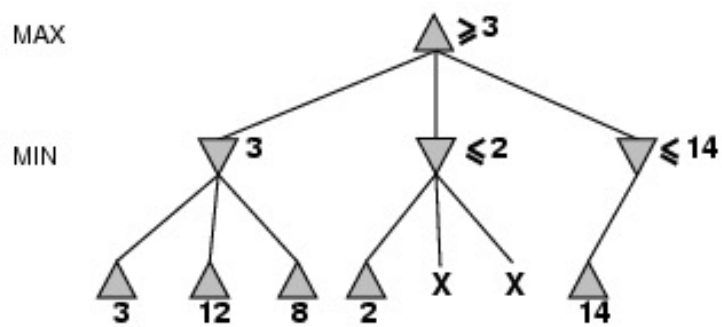
## $\alpha$ - $\beta$ pruning example



137

137

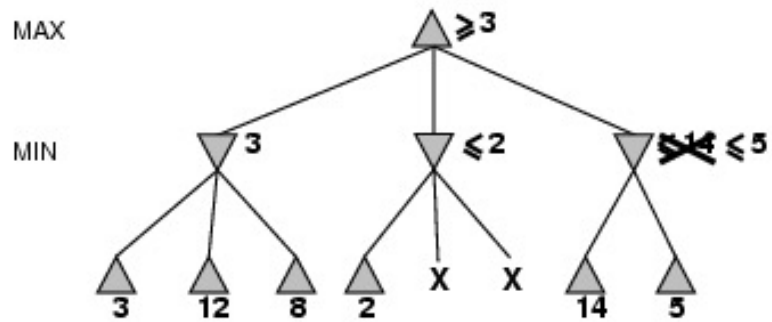
## $\alpha$ - $\beta$ pruning example



138

138

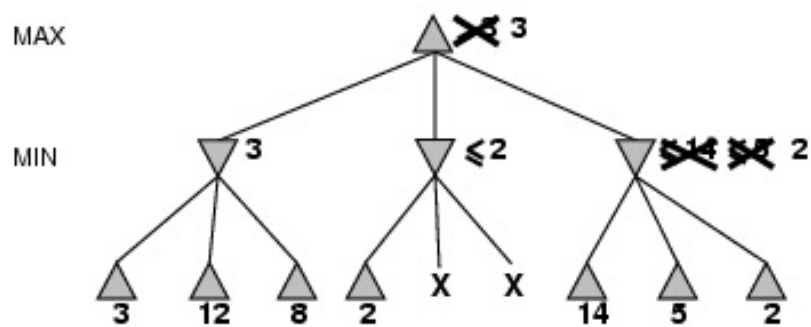
## $\alpha$ - $\beta$ pruning example



139

139

## $\alpha$ - $\beta$ pruning example

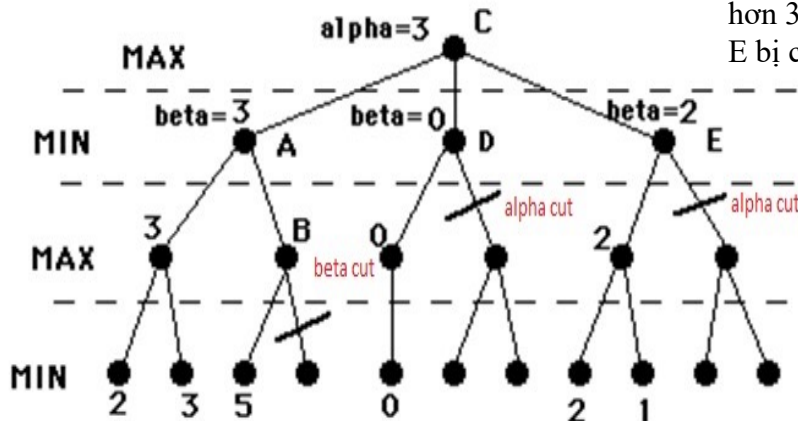


140

140

## Thủ tục cắt tỉa alpha – beta

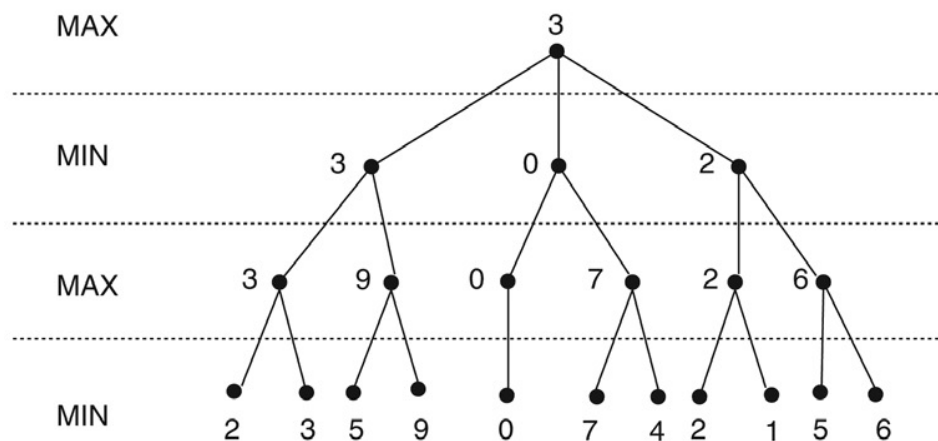
A có  $\beta = 3$  (Trị nút A sẽ không lớn hơn 3)  
 B bị cắt tỉa  $\beta$ , vì  $5 > 3$   
 C có  $\alpha = 3$  (Trị nút C sẽ không nhỏ hơn 3)  
 D bị cắt tỉa  $\alpha$ , vì  $0 < 3$   
 E bị cắt tỉa  $\alpha$ , vì  $2 < 3$  Trị nút C là 3



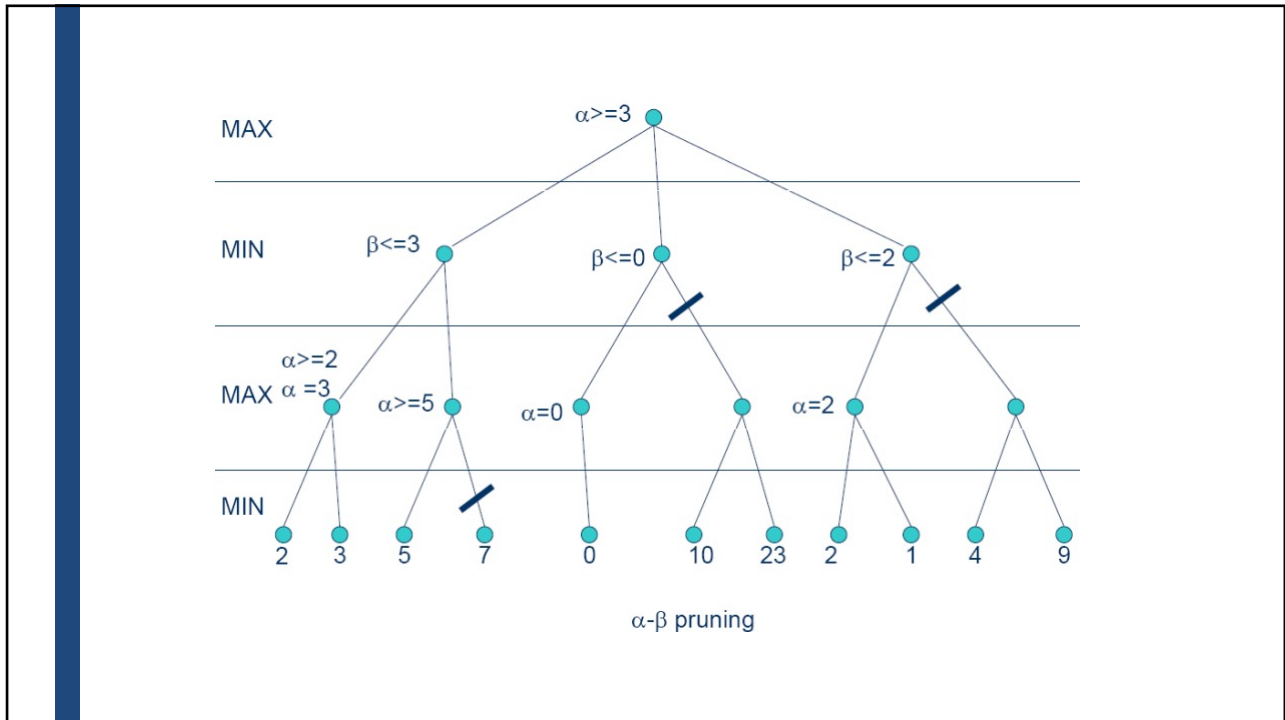
141

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)

### ■ Minimax đối với một KGTT giả định:



142



143

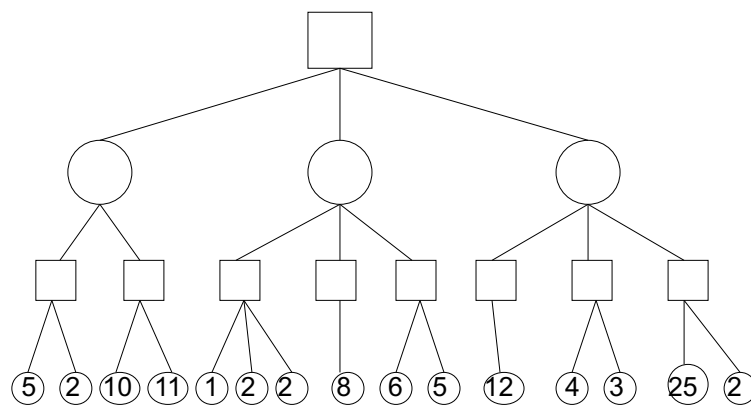
## Alpha-Beta Pruning

max

min

max

eval



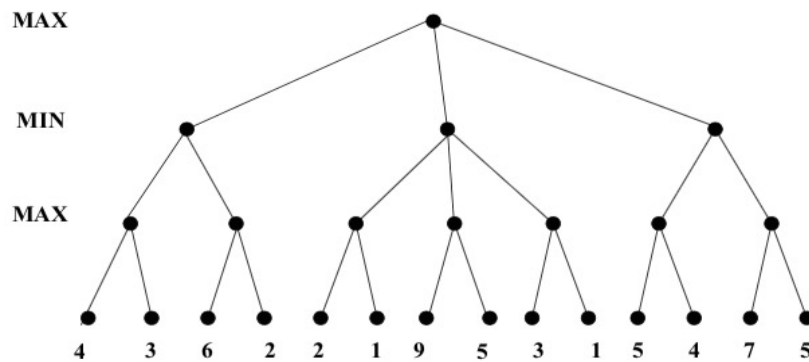
Nút MIN có  $\beta \leq \alpha$  của nút cha MAX bất kỳ ( $\alpha$ -cut)  
 Nút MAX có  $\alpha \geq \beta$  của nút cha MIN bất kỳ ( $\beta$ -cut)

144

144



## Alpha-Beta Pruning



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)

145