



CANTHO UNIVERSITY

CHƯƠNG 3

KỸ THUẬT THIẾT KẾ THUẬT TOÁN

KỸ THUẬT CẮT TỈA ALPHA-BETA TRÊN CÂY TRÒ CHƠI

Võ Huỳnh Trâm



Một số kỹ thuật thiết kế thuật toán

- Kỹ thuật **Chia để trị** (Divide and Conquer)
- Kỹ thuật **Tham ăn** /*Háu ăn/Tham lam* (Greedy)
- Kỹ thuật **Nhánh cận** (Branch and Bound)
- Kỹ thuật **Quy hoạch động** (Dynamic Programming)
- Kỹ thuật **Quay lui** (Backtracking)
- Kỹ thuật **Cắt tỉa alpha-beta** (Alpha-Beta Pruning)
trên Cây trò chơi
- Kỹ thuật *Tìm kiếm địa phương* (Local Search)



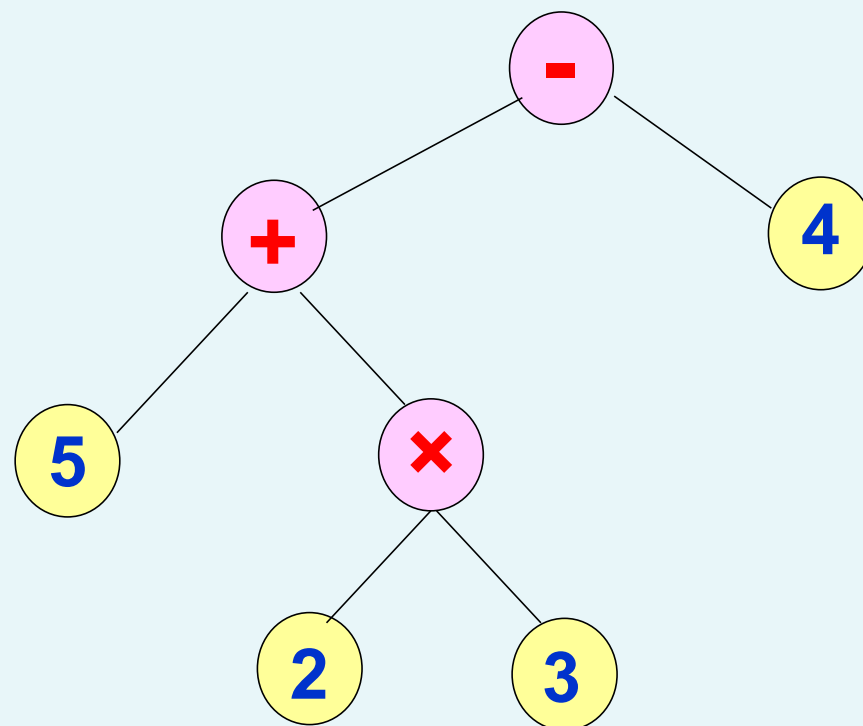
Kỹ thuật quay lui

- Là quá trình phân tích **đi xuống** và **quay lui lại** theo đường đã đi qua.
 - Tại mỗi bước: vấn đề chưa được giải quyết (do còn thiếu cứ liệu) nên phải phân tích tới các **điểm dừng** - xác định được *lời giải*/ xác định được *không thể* (/không nên) tiếp tục đi.
 - Từ các **điểm dừng**: quay lui theo đường đã qua để giải quyết các vấn đề tồn đọng \Rightarrow giải quyết vấn đề ban đầu.
- 3 kỹ thuật quay lui:
 - “**Vét cạn**” : đi tới tất cả các điểm dừng rồi mới quay lui.
 - “**Cắt tỉa Alpha-Beta**” và “**Nhánh-Cạn**”: không cần thiết phải đi tới tất cả điểm dừng, chỉ đến một số điểm và suy luận để quay lui sớm.



Định trị cây biểu thức số học: Mô tả

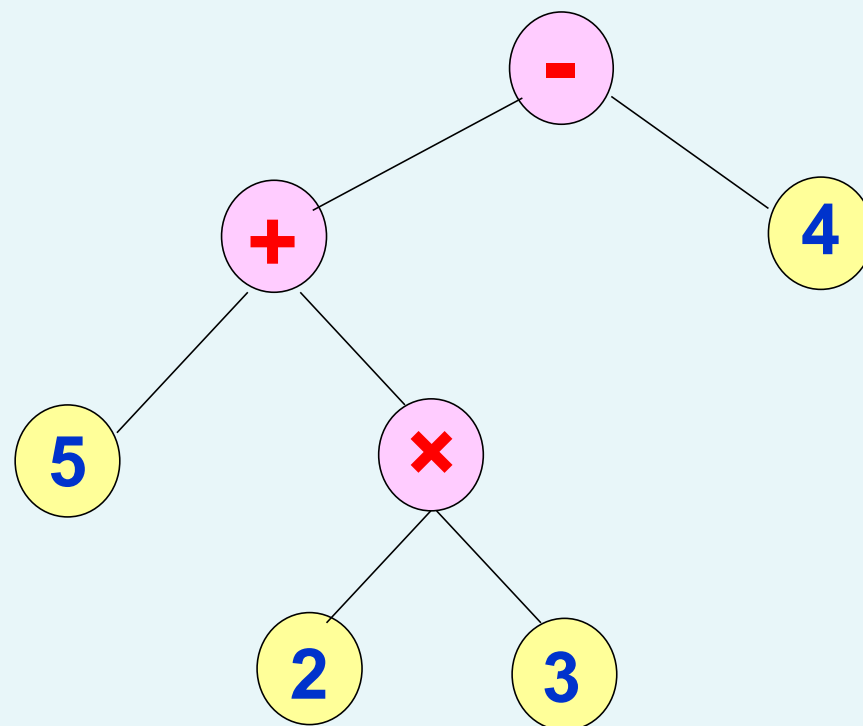
- Cây biểu thức số học là **cây nhị phân**:
 - *Nút lá* biểu diễn **toán hạng**
 - *Nút trong* biểu diễn **toán tử**.
- VD: Biểu thức $5 + 2 \times 3 - 4$ biểu diễn bởi cây trong hình





Định trị cây biểu thức số học: Ví dụ

- Để định trị nút $-$: định trị nút $+$ và nút 4 .
Nút 4 là nút lá nên giá trị là 4 .
- Để định trị nút $+$: định trị nút 5 và nút $*$.
Nút 5 là nút lá nên giá trị là 5 .
- Để định trị nút $*$: định trị nút 2 và 3 . Cả 2 nút đều là lá nên giá trị là 2 và 3 .
- Quay lui lại nút $*$, lấy toán tử $*$ áp dụng cho 2 con của nó là 2 và $3 \rightarrow$ trị của nút $*$ là 6 .
- Quay lui về nút $+$, áp dụng toán tử $+$ vào 2 con của nó là 5 và $6 \rightarrow$ trị của nút $+$ là 11 .
- Cuối cùng quay lui về nút $-$, áp dụng toán tử $-$ vào 2 con của nó là 11 và $4 \rightarrow$ trị của nút $-$ (nút gốc) là 7 . Đó chính là trị của biểu thức.





Cài đặt kỹ thuật quay lui (vét cạn): Định trị cho cây BTSH

```
float Eval (node n) {  
    if (n là nút lá)  
        return (trị của toán hạng trong n);  
    return (Toán tử trong n (Eval (Con trái của n),  
                                Eval (Con phải của n)));  
}
```

- Muốn định trị cho cây biểu thức T: gọi **Eval**(root (T))



Cây trò chơi: Mô tả

- Xét trò chơi có 2 người luân phiên đi nước cờ của mình: *cờ vua, cờ tướng, cờ carô, ...*
- Trò chơi có một *trạng thái bắt đầu* và mỗi nước đi sẽ biến đổi *trạng thái hiện hành* thành *trạng thái mới*.
- Trò chơi kết thúc khi : có một người *thắng cuộc* hoặc cả hai đấu thủ không thể phát triển được nước đi của mình (*hòa cờ*).
- **Vấn đề:** Tìm cách phân tích xem từ một trạng thái nào đó sẽ dẫn đến đấu thủ nào thắng với điều kiện cả hai đấu thủ đều có trình độ như nhau → **Cây trò chơi**



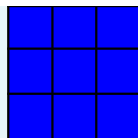
Biểu diễn trò chơi bằng cây trò chơi

- Trò chơi có thể được biểu diễn bởi *cây trò chơi*.
- Mỗi nút của cây biểu diễn cho một trạng thái.
 - **Nút gốc**: biểu diễn cho *trạng thái bắt đầu* cuộc chơi.
 - Mỗi **nút lá**: biểu diễn cho một *trạng thái kết thúc* của trò chơi (trạng thái thắng, thua hoặc hòa).
- Nếu trạng thái x được biểu diễn bởi nút n thì các con của n biểu diễn cho tất cả các trạng thái kết quả của các nước đi có thể xuất phát từ trạng thái x .



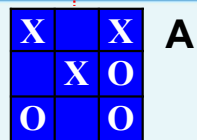
CANTHO UNIVERSITY

X-đi



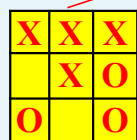
VD: *Cây trò chơi carô 9 ô*

X-đi

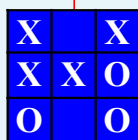


A

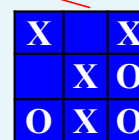
O-đi



B

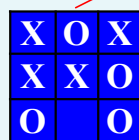


C

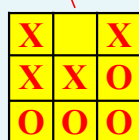


D

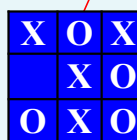
X-đi



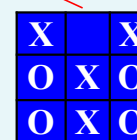
E



F

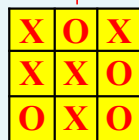


G

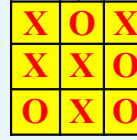


H

O-đi



I



J



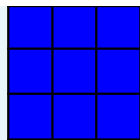
K

w.ctu.edu.vn



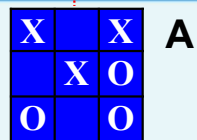
CANTHO UNIVERSITY

X-đi



VD: Gán trị cho các nút lá

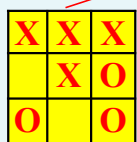
X-đi
Max



A

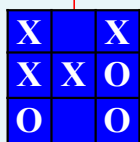
1 : nếu X thắng,
-1 : nếu X thua
0 : nếu hai đầu thủ hòa

O-đi
Min

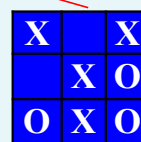


B

1

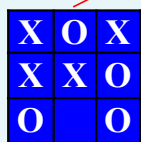


C

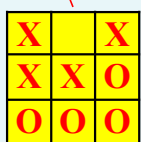


D

X-đi
Max

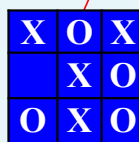


E

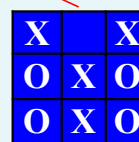


F

-1

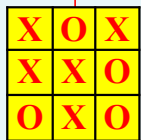


G



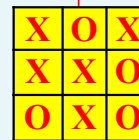
H

O-đi
Min



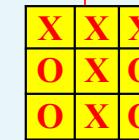
I

0



J

0



K

1



Quy tắc định trị cây trò chơi

- **Lượt đi của X:** X chọn nước đi dẫn đến trạng thái có giá trị **lớn nhất (1)** hay X chọn nước đi MAX \rightarrow nút **MAX**.
- **Lượt đi của O:** O chọn nước đi dẫn đến trạng thái có giá trị **nhỏ nhất (-1)** (X thua, O thắng) hay O chọn nước đi MIN \rightarrow nút **MIN**.

Do 2 đấu thủ đi luân phiên nhau nên các mức trên cây cũng luân phiên là MAX và MIN \rightarrow *Cây trò chơi* còn gọi là **cây MIN-MAX**.



Quy tắc định trị cây trò chơi

- Quy tắc định trị:

- **Nút lá** thì *trị của nó* = *giá trị được gán cho nút đó*.

Ngược lại:

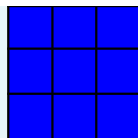
- Nếu nút là **nút MAX** thì *trị của nó* = *giá trị lớn nhất (Max)* của tất cả các trị của các con của nó.

- Nếu nút là **nút MIN** thì *trị của nó* = *giá trị nhỏ nhất (Min)* của tất cả các trị của các con của nó.



CANTHO UNIVERSITY

X-đi



Thuật toán Quay lui vét cạn

A

X-đi
Max

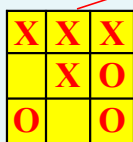
$$\begin{aligned} &-\infty \\ &\text{Max}(-\infty, 1) = 1 \\ &\text{Max}(1, -1) = 1 \\ &\text{Max}(1, 0) = 1 \end{aligned}$$



→ Trị nút A = 1

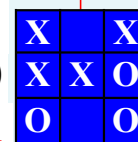
O-đi
Min

B



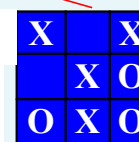
1

C



$$\begin{aligned} &\infty \\ &\text{Min}(\infty, 0) = 0 \\ &\text{Min}(0, -1) = -1 \end{aligned}$$

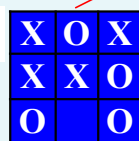
D



$$\begin{aligned} &\infty \\ &\text{Min}(\infty, 0) = 0 \\ &\text{Min}(0, 1) = 0 \end{aligned}$$

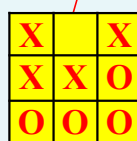
X-đi
Max

E



$$\begin{aligned} &-\infty \\ &\text{Max}(-\infty, 0) = 0 \end{aligned}$$

F



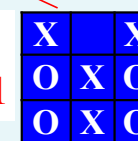
-1

G



$$\begin{aligned} &-\infty \\ &\text{Max}(-\infty, 0) = 0 \end{aligned}$$

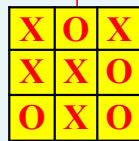
H



$$\begin{aligned} &-\infty \\ &\text{Max}(-\infty, 1) = 1 \end{aligned}$$

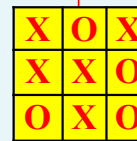
O-đi
Min

I



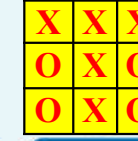
0

J



0

K



1



Kỹ thuật vét cạn định trị cây trò chơi: Giả thiết

- Hàm **Payoff** nhận vào một nút lá và trả về *giá trị* nút lá đó.
- Các hằng ∞ và $-\infty$ tương ứng các trị Payoff *lớn nhất* và *nhỏ nhất*.
- Khai báo kiểu **ModeType** = (MIN, MAX) để xác định định trị cho nút là MIN hay MAX.
- Kiểu **NodeType** được khai báo thích hợp để biểu diễn cho một nút trên cây phản ánh một trạng thái của cuộc chơi.
- Hàm **is_leaf** để xác định một nút có phải là nút lá hay không?
- Hàm **max** và **min** lấy giá trị *lớn nhất* và giá trị *nhỏ nhất* của hai giá trị.
- Hàm **Search** nhận vào một nút n và kiểu *mode* của nút đó (MIN hay MAX), trả về *giá trị* của nút.



Kỹ thuật vét cạn định trị cây trò chơi: Cài đặt bằng C/C++

```
float Search(NodeType n, ModeType  
mode) {  
    NodeType C;    /*C là nút con của nút n*/  
    float value; /*Lúc đầu cho value một giá trị  
    tạm,sau khi đã xét hết tất cả các con của nút n  
    thì value là giá trị của nút n*/  
    if (is_leaf(n))  
        return Payoff(n);  
    /*Khởi tạo giá trị tạm cho n*/  
    if (mode == MAX) value = -∞;  
    else value = ∞;
```

*/*Xét tất cả các con của n, mỗi lần xác định được
giá trị của một nút con,ta phải đặt lại giá trị
tạm value. Khi đã xét hết tất cả các con thì
value là giá trị của n*/*

```
    for (với mỗi con C của n)  
        if (mode == MAX)  
            value = max(value, Search(C,MIN));  
        else  
            value = min(value, Search(C, MAX));  
        return value;  
    }
```



Kỹ thuật quay lui vết cạn: Nhận xét

- Để định trị một nút, phải định trị tất cả các nút con cháu của nó
→ Muốn định trị cho nút gốc, phải định trị tất cả nút trên cây.
- Số lượng các nút tuy *hữu hạn* nhưng *không phải là ít*.
 - Cây trò chơi ca rô 9 ô → **9!** nút. Bàn cờ n ô → **n!** nút trên cây.
 - Đối với các loại cờ khác như cờ vua thì số lượng các nút còn lớn hơn nhiều ⇒ **bùng nổ tổ hợp** các nút.

⇒ Tìm cách sao cho khi định trị một nút thì ***không nhất thiết*** phải định trị cho tất cả các nút con cháu của nó.



Kỹ thuật quay lui vết cạn: Nhận xét

- Cờ vua
 - 1997, DeepBlue đánh bại Gary Kasparov trong trận đấu 6 ván
 - Bí quyết
 - Tìm kiếm vết cạn với độ sâu cao nhất có thể
 - Tính được 2×10^8 nước đi trong 1 giây so với 2 nước của Kasparov
 - 99.99% nước đi được xem là “ngu ngốc”
 - Hàm ước lượng giá (heuristic) rất phức tạp



Máy tính **Deep Blue**
của IBM

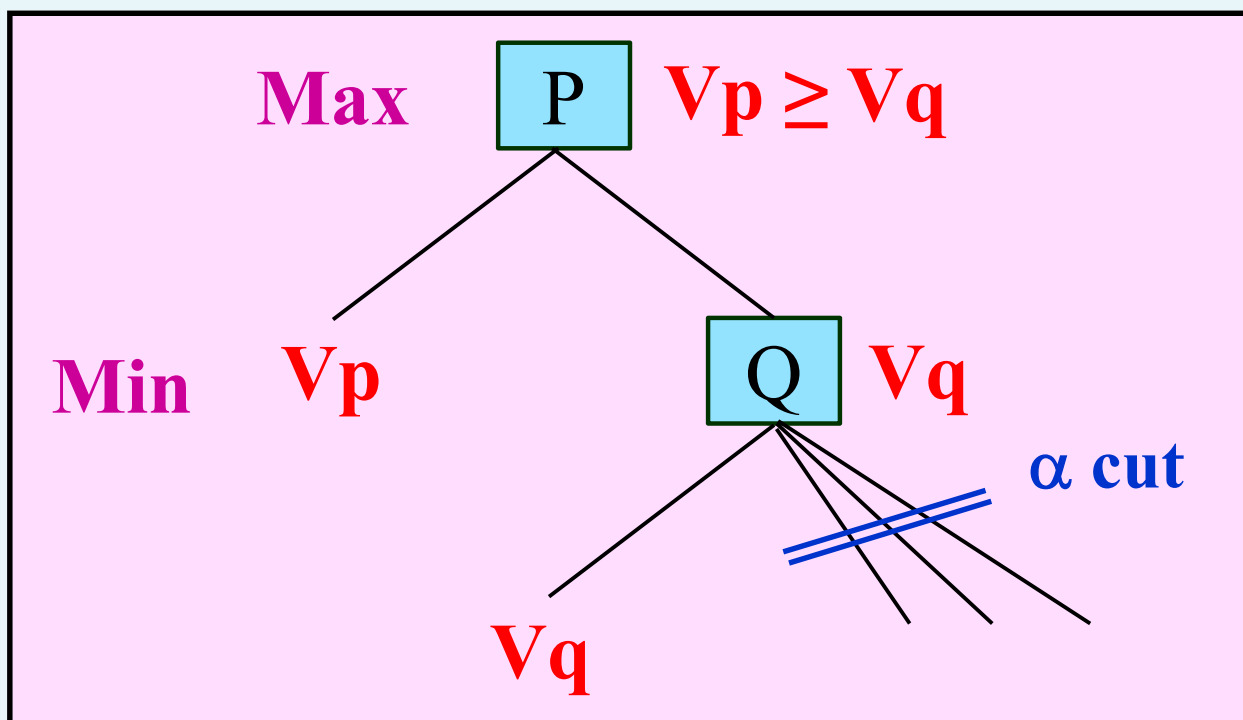


Nhà vô địch thế giới
Garry Kasparov



Quy tắc cắt tỉa Alpha: Nút MAX

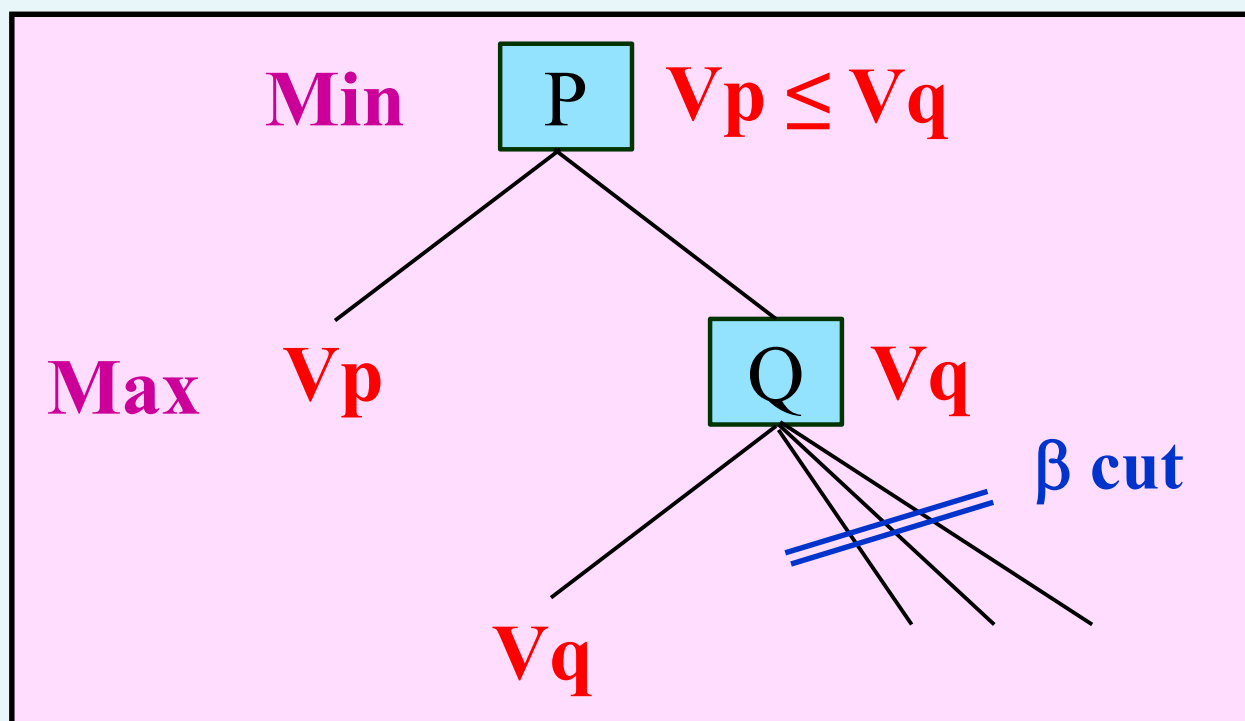
- P là nút **MAX** và đang xét nút con Q của nó (Q là nút MIN):
Nếu $V_p \geq V_q \rightarrow$ cắt các con chưa xét của Q.





Quy tắc cắt tỉa Beta: Nút MIN

- P là nút **MIN** và đang xét nút con Q của nó (Q là nút **MAX**):
Nếu $V_p \leq V_q \rightarrow$ cắt các con chưa xét của Q.





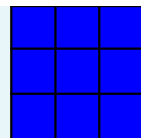
Kỹ thuật cắt tỉa Alpha-Beta (Alpha-Beta Pruning)

- Quy tắc định trị cho một nút *không phải là nút lá* :
 - Khởi đầu nút MAX có giá trị tạm $-\infty$; nút MIN là ∞ .
 - Nếu tất cả các nút con của một nút *đã được xét* hoặc *bị cắt tỉa* thì giá trị tạm của nút đó trở thành giá trị của nó.
 - Nếu một nút MAX n có giá trị tạm là V_1 và một nút con của nó có giá trị là V_2 thì đặt giá trị tạm mới của n là **$\max(V_1, V_2)$** . Nếu n là nút MIN thì đặt giá trị tạm mới của n là **$\min(V_1, V_2)$** .
- ⇒ Vận dụng quy tắc cắt tỉa Alpha-Beta để hạn chế số lượng nút phải xét.



CANTHO UNIVERSITY

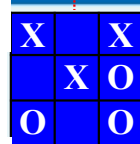
X-đi



Thuật toán *Cắt tỉa Alpha-Beta*

X-đi
Max

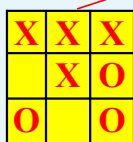
$$\begin{aligned} &-\infty \\ &\text{Max}(-\infty, 1) = 1 \\ &\text{Max}(1, 0) = 1 \\ &\text{Max}(1, 0) = \mathbf{1} \end{aligned}$$



→ Trị nút A = **1**

O-đi
Min

B

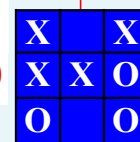


1

∞

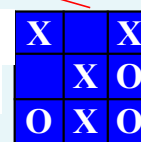
$$\text{Min}(\infty, 0) = \mathbf{0}$$

C



$$\text{Min}(\infty, 0) = \mathbf{0}$$

D

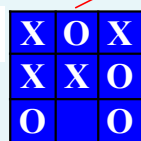


Vì $V_A = 1 > V_D = 0$

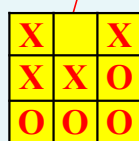
X-đi
Max

$$\begin{aligned} &-\infty \\ &\text{Max}(-\infty, 0) = \mathbf{0} \end{aligned}$$

E



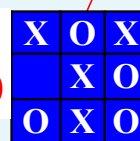
F



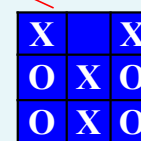
$-\infty$

$$\text{Max}(-\infty, 0) = \mathbf{0}$$

G

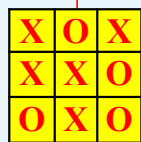


H



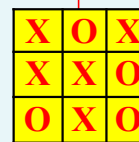
O-đi
Min

I



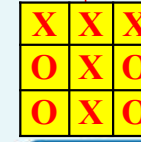
0

J



0

K





Thuật toán cắt tỉa Alpha – Beta: Cài đặt

```
float Cat_tia(NodeType Q, ModeType mode,
float Vp) {
    NodeType C; /*C là nút con của Q*/
    float Vq;
    /*Vq là giá trị tạm của Q, sau khi tất cả các con của
    nút Q đã xét hoặc bị cắt tỉa thì Vq là giá trị của nút
    Q*/
    if (is_leaf(Q)) return Payoff(Q);
    /* Khởi tạo giá trị tạm cho Q */
    if (mode == MAX) Vq = -∞;
    else Vq = ∞;
    /*Xét các con của Q, mỗi lần xác định được giá trị của
    một nút con của Q, ta phải đặt lại giá trị tạm Vq và so
    sánh với Vp để có thể cắt tỉa hay không*/
```

```
Xét C là con trái nhất của Q;
while (C là con của Q) {
    if (mode == MAX) {
        Vq = max(Vq, Cat_tia(C, MIN, Vq));
        if (Vp ≤ Vq) return Vq;
        /*cắt tỉa tất cả các con còn lại của Q*/
    } else
    {
        Vq = min(Vq, Cat_tia(C, MAX, Vq));
        if (Vp ≥ Vq) return Vq;
    }
    return Vq;
}
```



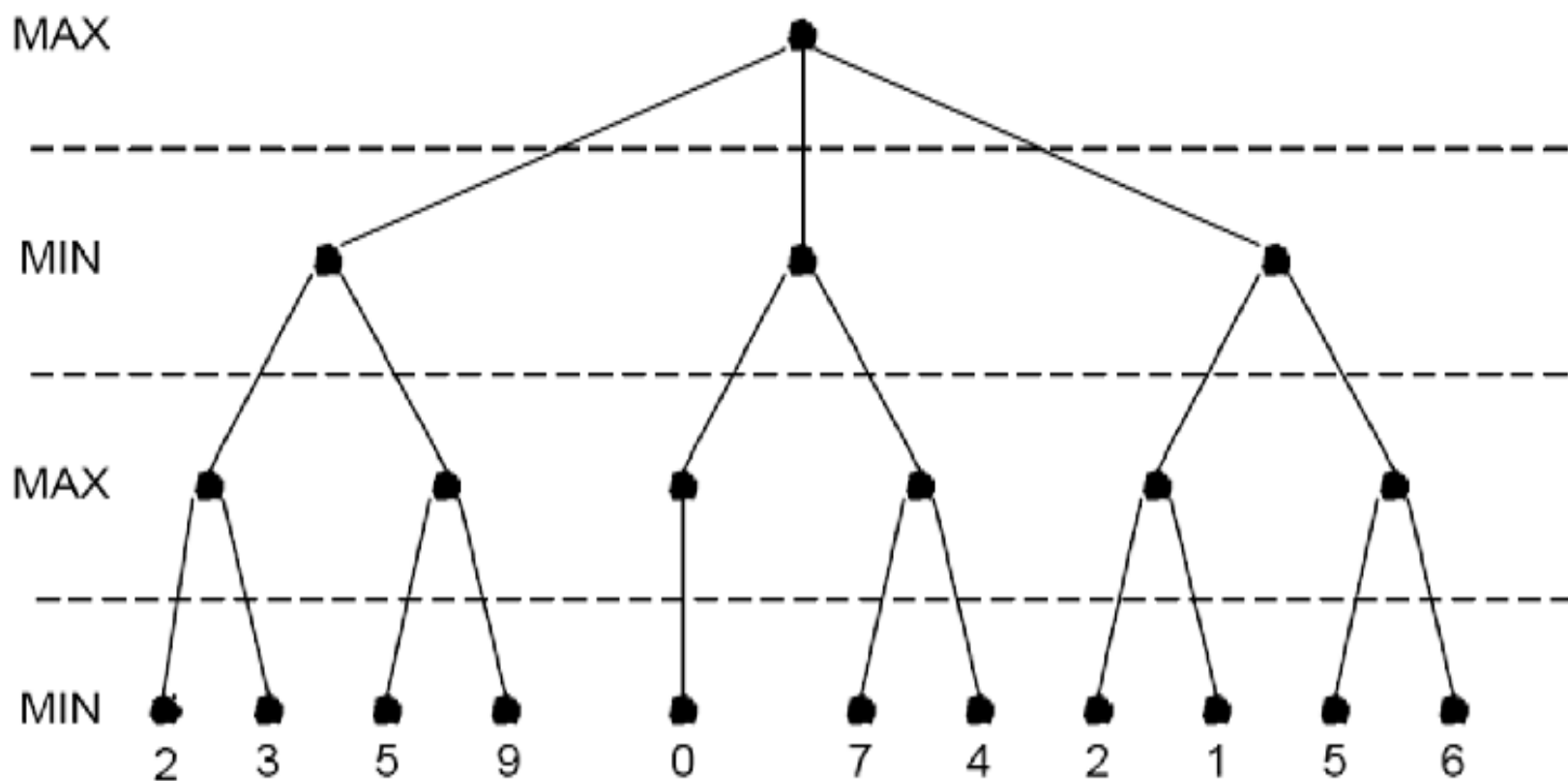

Kết luận

- Mỗi kỹ thuật chỉ phù hợp với một hoặc một số loại bài toán
- Mỗi kỹ thuật đều có ưu và khuyết điểm, không có kỹ thuật nào là dùng được trong mọi trường hợp (“*trị bá bệnh*”)
 - Kỹ thuật **Chia để trị** là kỹ thuật *cơ bản* nhất. Hãy chia nhỏ bài toán để giải quyết!
 - Kỹ thuật **Vét cạn** có độ phức tạp *quá lớn* (lũy thừa)
 - Kỹ thuật “**Tham ăn**” giúp nhanh chóng xây dựng được một phương án, dù chưa hẳn tối ưu nhưng *chấp nhận được*.
 - Kỹ thuật **Nhánh cận** cho phép tìm được phương án tối ưu, tuy nhiên cần phải có cách *ước lượng cận tốt* mới cắt được nhiều nhánh.
 - Kỹ thuật **Quy hoạch động** có thể giải được rất nhiều bài toán, tuy nhiên cần phải xây dựng được *công thức truy hồi* khi số lượng bài toán con phải giải là đa thức.



Bài tập 1

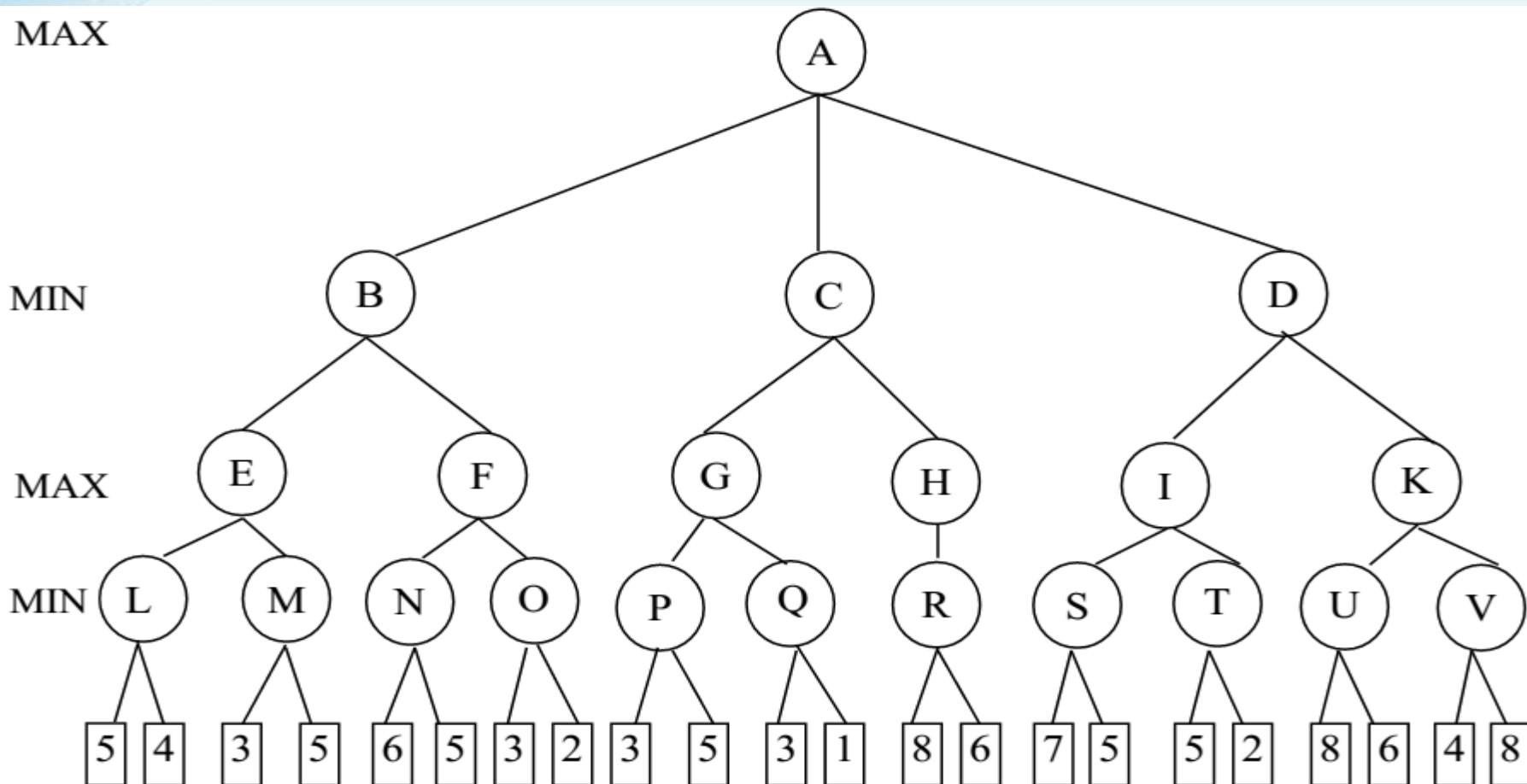
- Bài tập:** Định trị nút gốc cây trò chơi theo giải thuật cắt tỉa Alpha-Beta





Bài tập 2

- Bài tập 5-GT:** Định trị nút gốc cây trò chơi theo giải thuật cắt tỉa Alpha-Beta





Bài tập 3

- Bài tập:** Định trị nút gốc cây trò chơi theo giải thuật cắt tỉa Alpha-Beta

