

Chương 1. GIỚI THIỆU



TRÍ TUỆ NHÂN TẠO

Artificial Intelligence

Đoàn Vũ Thịnh
Khoa Công nghệ Thông tin
Đại học Nha Trang
Email: thinhdv@ntu.edu.vn

Nha Trang, 06-2023

Chương 1. GIỚI THIỆU

▪ Khái niệm

Trí tuệ:

- Phản ứng một cách thích hợp những tình huống thông qua điều chỉnh hành vi hợp lý.
- Hiểu rõ mối quan hệ giữa các sự kiện của thế giới quan để đưa ra những hành vi phù hợp nhằm đạt được mục đích.

Tri thức:

- Là sự hiểu biết (thường là sâu) về một vấn đề (lĩnh vực). Ví dụ: Hiểu biết về y khoa, về máy tính, về thời tiết...
- Là thông tin chứa đựng 2 phần: Các khái niệm và các phương pháp nhận thức (quy luật, suy diễn...).

Chương 1. GIỚI THIỆU

▪ Khái niệm

Hành xử thông minh:

- Kết quả của một quá trình thu thập, xử lý và điều khiển theo những tri thức đã có hay mới phát sinh.
- Kết quả thường tốt hơn khi so với các hành xử thông thường.

Thông minh:

- Sự tổng hợp của cả 3 yếu tố: thu thập tri thức, suy luận và hành xử của đối tượng trên tri thức thu thập được.
- Không thể hánh giá riêng lẽ bất kỳ một khía cạnh nào để nói về tính thông minh.

Chương 1. GIỚI THIỆU

▪ Khái niệm

AI là gì:

- “An AI approach problem-solving is one which (George Luger):
 - *uses domain-specific knowledge*
 - *to find a good-enough solution*
 - *to a hard problem*
 - *in a reasonable amount of time.*”
- Minsky, trí tuệ nhân tạo là một ngành khoa học:
 - *Nghiên cứu, mô phỏng trên máy tính các hành vi và tư duy thông minh tương tự như con người.*
 - *Giúp máy tính có khả năng nhận thức, suy luận và phản ứng*

Chương 1. GIỚI THIỆU

- Đối tượng nghiên cứu
 - Ai là ngành nghiên cứu về các hành xử thông minh bao gồm: thu thập, lưu trữ tri thức, suy luận, hoạt động và kỹ năng.
 - Đối tượng nghiên cứu là các “hành xử thông minh” chứ không phải là “sự thông minh”.
 - ‘Không có’ sự thông minh mà chỉ có biểu hiện thông minh qua hành xử.

Chương 1. GIỚI THIỆU

- Mục tiêu nghiên cứu
 - Trí tuệ nhân tạo nhằm tạo ra “Máy người”?
 - Xây dựng lý thuyết về thông minh hể giải thích các hoạt động thông minh
 - Tìm hiểu cơ chế sự thông minh của con người
 - Cơ chế lưu trữ tri thức
 - Cơ chế khai thác tri thức
 - Xây dựng cơ chế hiện thực sự thông minh
 - Áp dụng các hiểu biết này vào các máy móc phục vụ con người

Chương 1. GIỚI THIỆU

- Đặc điểm
 - Sử dụng máy tính vào suy luận trên các ký hiệu, nhận dạng qua mẫu, học, và các suy luận khác...
 - Tập trung vào các vấn đề “khó” không thích hợp với các lời giải mang tính thuật toán.
 - Quan tâm đến các kỹ thuật giải quyết vấn đề sử dụng các thông tin không chính xác, không đầy đủ, mơ hồ...
 - Cho lời giải ‘đủ tốt’ (không phải lời giải chính xác hay tối ưu)
 - Sử dụng heuristics – “bí quyết”, “mẹo”
 - Sử dụng tri thức chuyên môn

Chương 1. GIỚI THIỆU

■ Đặc điểm

- Chương trình chưa tự sinh ra được heuristic
- Chưa có khả năng xử lý song song của con người
- Chưa có khả năng diễn giải một vấn đề theo nhiều phương pháp khác nhau như con người.
- Chưa có khả năng xử lý thông tin trong môi trường liên tục như con người.
- Chưa có khả năng học như con người.
- Chưa có khả năng tự thích nghi với môi trường.

Chương 1. GIỚI THIỆU

▪ Lịch sử phát triển

- 1943, Warren McCulloch và Walter Pitts nghiên cứu 3 cơ sở lý thuyết cơ bản: triết học cơ bản và chức năng các neuron thần kinh; phân tích các mệnh đề logic; lý thuyết dự đoán của Turing.
⇒ *Các tác giả đã đề xuất mô hình neuron nhân tạo, mỗi neuron đặc trưng bởi hai trạng thái “bật”, “tắt” và phát hiện mạng neuron có khả năng học.*
- 1950, Alan Turing đã đưa ra các “trắc nghiệm thông minh” để nhận biết máy tính có thông minh hay không.
⇒ *Máy tính có thể thất bại trong các trắc nghiệm thông minh nhưng nó vẫn có thể thông minh.*

Chương 1. GIỚI THIỆU

▪ Lịch sử phát triển

- 1956, John McCarthy đưa ra thuật ngữ Trí tuệ nhân tạo tại Hội thảo đầu tiên về chủ đề này.
⇒ *Đề xuất ngôn ngữ lập trình Lisp – một trong những ngôn ngữ lập trình hàm tiêu biểu, được sử dụng trong lĩnh vực AI.*
- 60, 70 Joel Moses viết chương trình Macsyma - chương trình toán học sử dụng cơ sở tri thức đầu tiên thành công.
- Marvin Minsky và Seymour Papert đưa ra các chứng minh đầu tiên về giới hạn của các mạng nơ-ron đơn giản.
- Alain Colmerauer phát triển Ngôn ngữ lập trình logic Prolog.

Chương 1. GIỚI THIỆU

▪ Lịch sử phát triển

- Ted Shortliffe xây dựng thành công một số hệ chuyên gia đầu tiên trợ giúp chẩn đoán trong y học, các hệ thống này sử dụng ngôn ngữ luật để biểu diễn tri thức và suy diễn.
- 1980, các hệ chuyên gia (expert systems) mô phỏng tri thức và các kỹ năng phân tích của chuyên gia con người.
- 90-2000, AI đã đạt được những thành tựu to lớn nhất, AI được áp dụng trong logic, khai phá dữ liệu, y học và công nghiệp.
- Những năm gần đây AI có bước đột phá mạnh mẽ: Robot, Hội họa, chat box...

Chương 1. GIỚI THIỆU

- Lĩnh vực nghiên cứu
 - Lập luận (reasoning), suy diễn (reference) tự động:
 - Lập luận là suy diễn logic, dùng để chỉ một tiến trình rút ra kết luận (tri thức mới) từ những giả thiết đã cho (được biểu diễn dưới dạng cơ sở tri thức).
 - ⇒ Để thực hiện lập luận người ta cần có các phương pháp lưu trữ cơ sở tri thức và các thủ tục lập luận trên cơ sở tri thức đó.
 - ⇒ Muốn máy tính có thể lưu trữ và xử lý tri thức thì cần có các phương pháp biểu diễn tri thức.

Chương 1. GIỚI THIỆU

- Lĩnh vực nghiên cứu
 - Biểu diễn tri thức: ngôn ngữ biểu diễn và kỹ thuật xử lý TT
 - Một ngôn ngữ biểu diễn tri thức được đánh giá là “tốt” nếu nó có tính biểu đạt cao (khả năng biểu diễn một phạm vi rộng lớn các thông tin trong một miền ứng dụng) và các tính hiệu quả của thuật toán lập luận (chi phí về thời gian và không gian dành cho việc lập luận) trên ngôn ngữ đó.
 - Tuy nhiên, hai yếu tố này dường như đối nghịch nhau.
 - ⇒ *Thách thức lớn trong lĩnh vực AI là xây dựng các ngôn ngữ biểu diễn tri thức mà có thể cân bằng hai yếu tố này, tức là ngôn ngữ có tính biểu đạt đủ tốt (tùy theo từng ứng dụng) và có thể lập luận hiệu quả.*

Chương 1. GIỚI THIỆU

- Lĩnh vực nghiên cứu
- Lập kế hoạch:
 - Khả năng suy ra các mục đích cần đạt được đối với các nhiệm vụ đặt ra, và xác định dãy các hành động cần thực hiện để đạt được mục đích đó.
- Học máy:
 - Một lĩnh vực nghiên cứu của AI đang được phát triển mạnh mẽ và có nhiều ứng dụng trong các lĩnh vực khác nhau như khai phá dữ liệu, khám phá tri thức,...

Chương 1. GIỚI THIỆU

- Lĩnh vực nghiên cứu
 - Xử lý ngôn ngữ tự nhiên:
 - một nhánh của AI, tập trung vào các ứng dụng trên ngôn ngữ của con người. Các ứng dụng trong nhận dạng tiếng nói, nhận dạng chữ viết, dịch tự động, tìm kiếm thông tin,...
 - Hệ chuyên gia:
 - Có khả năng xử lý lượng thông tin lớn và cung cấp các kết luận dựa trên những thông tin đó.
 - Robotics

HẾT CHƯƠNG 1

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI



TRÍ TUỆ NHÂN TẠO

Artificial Intelligence

Đoàn Vũ Thịnh
Khoa Công nghệ Thông tin
Đại học Nha Trang
Email: thinhdv@ntu.edu.vn

Nha Trang, 06-2023

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

Để thiết kế giải thuật chung giải các bài toán này, chúng ta nên phát biểu bài toán theo dạng 5 thành phần:

- Trạng thái bài toán,
- Trạng thái đầu,
- Trạng thái đích,
- Các phép chuyển trạng thái,
- Lược đồ chi phí các phép chuyển trạng thái (chi phí)

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Trạng thái bài toán: Gọi số nước có trong 3 can lần lượt là a , b , c ($a \leq 3$, $b \leq 5$, $c \leq 9$) là trạng thái của bài toán
- Trạng thái đầu: $(0, 0, 0)$
- Trạng thái đích: $(0, 0, 7)$
- Các phép chuyển trạng thái: từ trạng thái (a,b,c) có thể chuyển sang trạng thái (x,y,z) thông qua các thao tác như làm rỗng 1 can, chuyển từ can này sang can kia đến khi hết nước ở can nguồn hoặc can đích bị đầy.
- Lược đồ chi phí các phép chuyển trạng thái (chi phí): 1

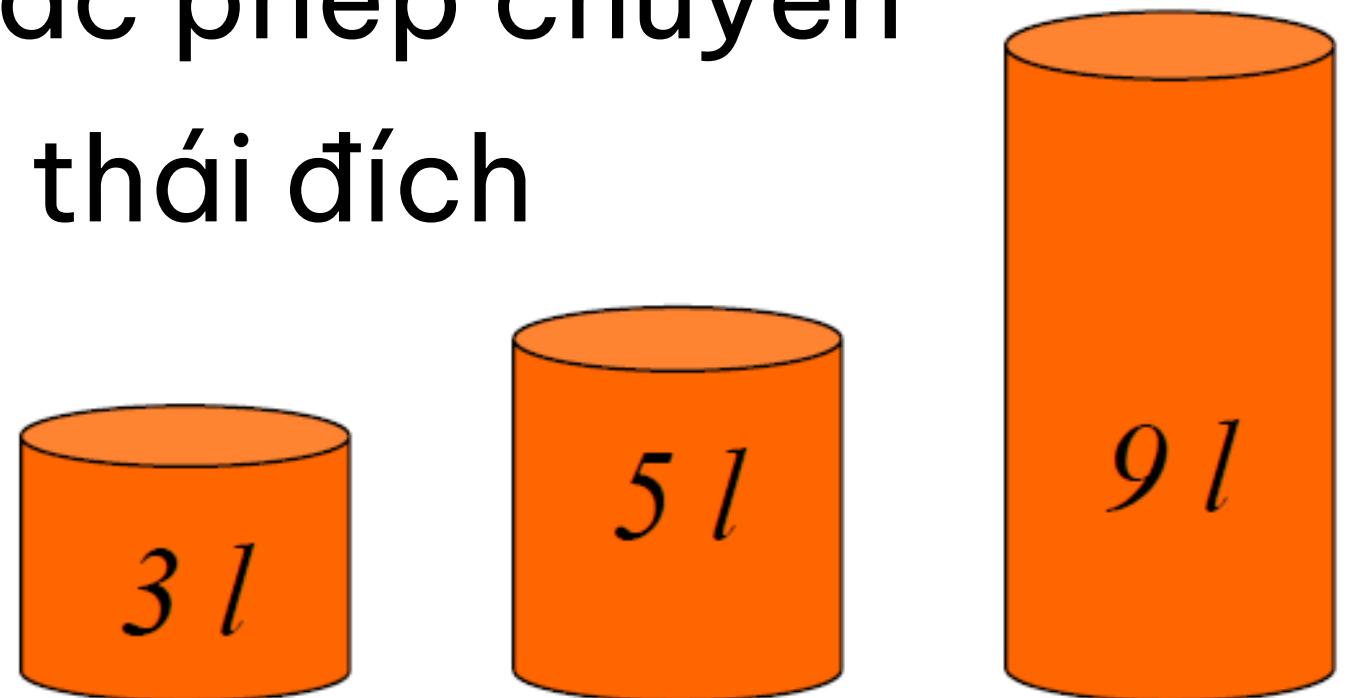


Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Một lời giải của bài toán là một dãy các phép chuyển trạng thái từ trạng thái đầu đến trạng thái đích

Bước	a	b	c
0	0	0	0
1	3	0	0
2	0	0	3
3	3	0	3
4	0	0	6
5	3	0	6
6	0	3	6
7	3	3	6
8	1	5	6
9	0	5	7

Chi phí = 9

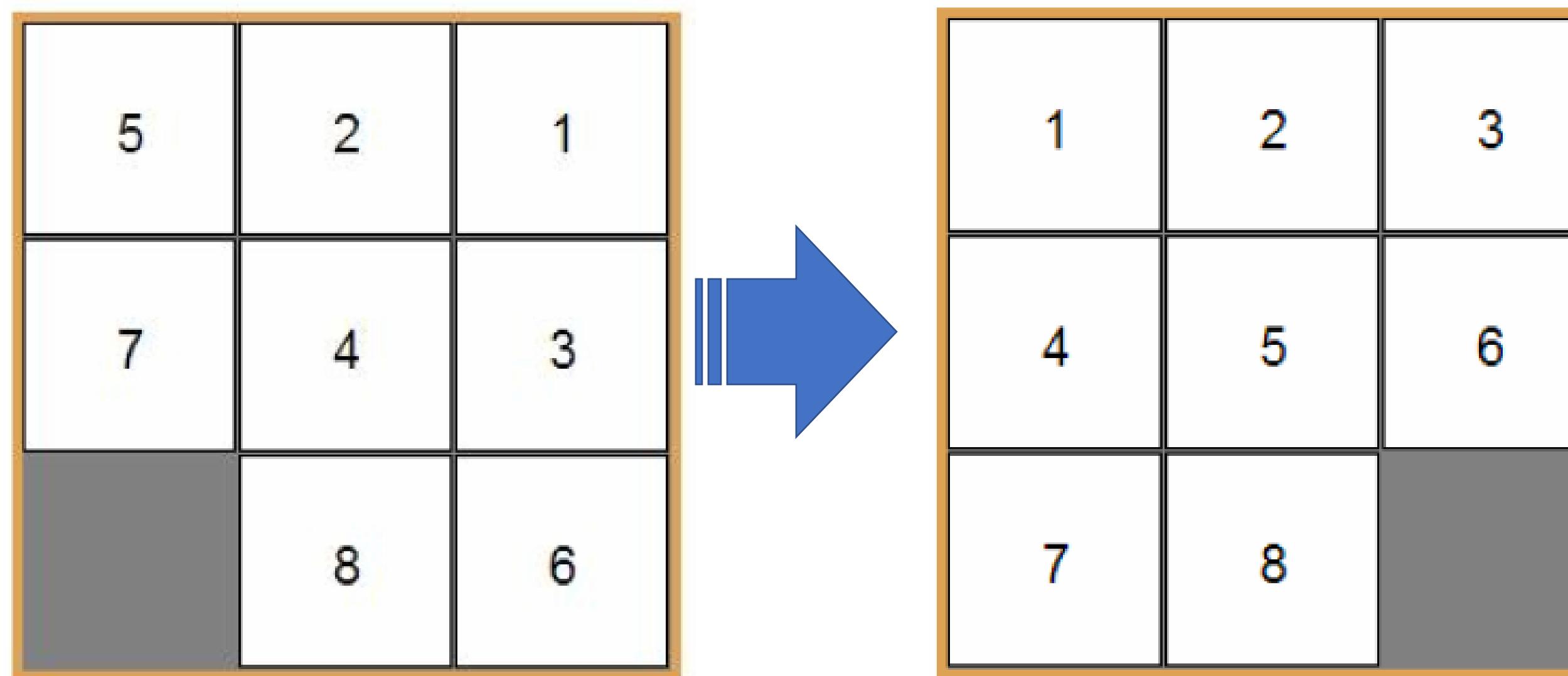


Chi phí = 5

Bước	a	b	c
0	0	0	0
1	0	5	0
2	3	2	0
3	3	0	2
4	3	5	2
5	3	0	7

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Bàn cờ kích thước 3×3 , trên bàn cờ có 8 quân cờ đánh số từ 1 đến 8 và có một ô trống. Có thể chuyển một quân cờ có chung cạnh với ô trống sang ô trống. Tìm dãy các phép chuyển để từ trạng thái ban đầu về trạng thái đích.



Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Bài toán di chuyển 8 số trên bàn cờ có thể phát biểu dưới dạng 5 thành phần:
- Biểu diễn trạng thái: mảng 2 chiều kích thước 3×3 , phần tử của mảng lưu số hiệu quân cờ (từ 0 đến 9, 0 là vị trí trống).
- Trạng thái đầu (hình)
- Trạng thái đích (hình)
- Phép chuyển trạng thái: đổi chỗ ô có số hiệu 0 với một trong các ô có cùng cạnh.
- Chi phí: mỗi phép chuyển có chi phí 1.

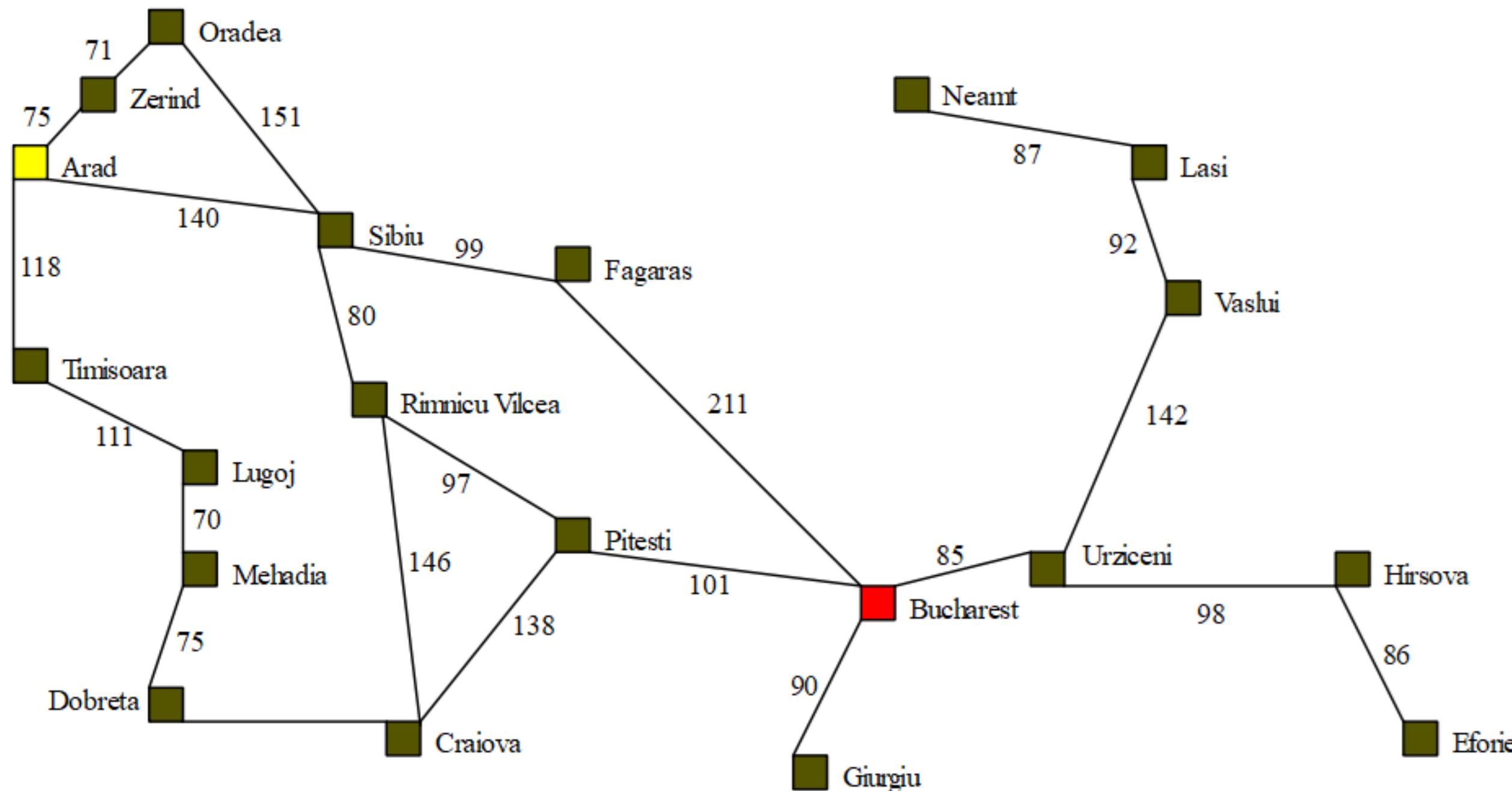
Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

5	2	1	5	2	1	0	2	1	2	0	1	2	4	1	2	4	1
7	4	3	0	4	3	5	4	3	5	4	3	5	0	3	0	5	3
0	8	6	7	8	6	7	8	6	7	8	6	7	8	6	7	8	6

4	1	3	4	1	3	4	1	3	4	1	0	4	0	1	0	4	1
0	2	5	2	0	5	2	5	0	2	5	3	2	5	3	2	5	3
7	8	6	7	8	6	7	8	6	7	8	6	7	8	6	7	8	6

0	1	3	1	0	3	1	2	3	1	2	3	1	2	3	1	2	3
4	2	5	4	2	5	4	0	5	4	5	0	4	5	6	7	8	0
7	8	6	7	8	6	7	8	6	7	8	6	7	8	0	Chi phí = 16		

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI



Một ôtô robot tìm đường từ Arad đến Bucharest. robot này không có bản đồ đầy đủ, nhưng khi nó đến một thành phố mới, nó có bộ cảm biến đọc được biển chỉ đường đến các thành lân cận.

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Bài toán tìm đường có thể phát biểu theo 5 thành phần:
- Trạng thái: vị trí của ôtô robot (tên thành phố)
- Trạng thái đầu: Thành phố Arad
- Trạng thái đích: Thành phố Bucharest
- Phép chuyển trạng thái: từ thành phố sang thành phố lân cận
- Chi phí: khoảng cách giữa 2 thành phố
- Một ví dụ của lời giải: **Arad \Rightarrow Sibiu \Rightarrow Fagaras \Rightarrow Bucharest.**

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

▪ Giải thuật tổng quát tìm kiếm lời giải

▪ Không gian trạng thái của bài toán

- Mỗi bài toán với 5 thành phần như mô tả ở trên, chúng ta có thể xây dựng được một cấu trúc đồ thị (không gian trạng thái) với **các nút là các trạng thái của bài toán**, các **cung là phép chuyển trạng thái**. Không gian trạng thái có thể là vô hạn hoặc hữu hạn. Ví dụ, với bài toán di chuyển 8 số trên bàn cờ, không gian trạng thái có số lượng là 8!
- Lời giải của bài toán là một đường đi trong không gian trạng thái có **điểm đầu là trạng thái đầu và điểm cuối là trạng thái đích**. Nếu không gian trạng thái của bài toán là nhỏ, có thể liệt kê và lưu vừa trong bộ nhớ của máy tính thì việc tìm đường đi trong không gian trạng thái có thể áp dụng các thuật toán tìm đường đi trong lý thuyết đồ thị.

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

▪ Giải thuật tổng quát tìm kiếm lời giải

Function General_Search(*problem, strategy*) returns a *solution*, or failure

```
cây-tìm-kiếm ← trạng-thái-đầu;  
while (1)  
{  
    if (cây-tìm-kiếm không thể mở rộng được nữa) then return failure  
    nút-lá ← Chọn-1-nút-lá(cây-tìm-kiếm, strategy)  
    if (node-lá là trạng-thái-đích) then return Đường-đi(trạng-thái-đầu, nút-lá)  
    else mở-rộng(cây-tìm-kiếm, các-trạng-thái-kề(nút-lá))  
}
```

- Tìm kiếm theo chiều rộng (nút lá nào xuất hiện trong cây sớm hơn thì được chọn trước để phát triển cây).
- Tìm kiếm theo chiều sâu (ngược lại)

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

▪ Giải thuật tổng quát tìm kiếm lời giải

▪ Đánh giá giải thuật: b^d

- Tính đầy đủ: có tìm được lời giải của bài toán không?
- Độ phức tạp thời gian: thời gian của giải thuật thực hiện?
- Độ phức tạp không gian: Kích cỡ của bộ nhớ cần cho giải thuật? cấu trúc dữ liệu lưu các trạng thái lá của cây tìm kiếm
- Tính tối ưu: Giải thuật có tìm ra lời giải có chi phí tối ưu (nhỏ nhất hoặc lớn nhất tùy theo ngữ cảnh của bài toán)?

b: số nhánh tối đa của một nút, hay là số phép chuyển trạng thái tối đa của một trạng thái tổng quát

d: độ sâu của lời giải có chi phí nhỏ nhất

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

▪ Tìm kiếm theo chiều rộng (Breath First Search)

Begin

Open={s};

//s: đỉnh xuất phát

Close=∅;

//Close: tập các đỉnh đã xét

While(Open != ∅){

//Open: tập các đỉnh có thể xét ở bước kế tiếp

n = Retrieve(Open); //n: đỉnh đang xét

If(n==g) Return TRUE; //g: đỉnh kết thúc

Open = Open \cup $\Gamma(n)$; // $\Gamma(n)$: các đỉnh có thể đi trực tiếp từ n

Close = Close \cup {n}

}

End;

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều rộng (Breath First Search)

Begin

 Open={s};

 Close=∅;

 While(Open != ∅){

 n = Retrieve(Open);

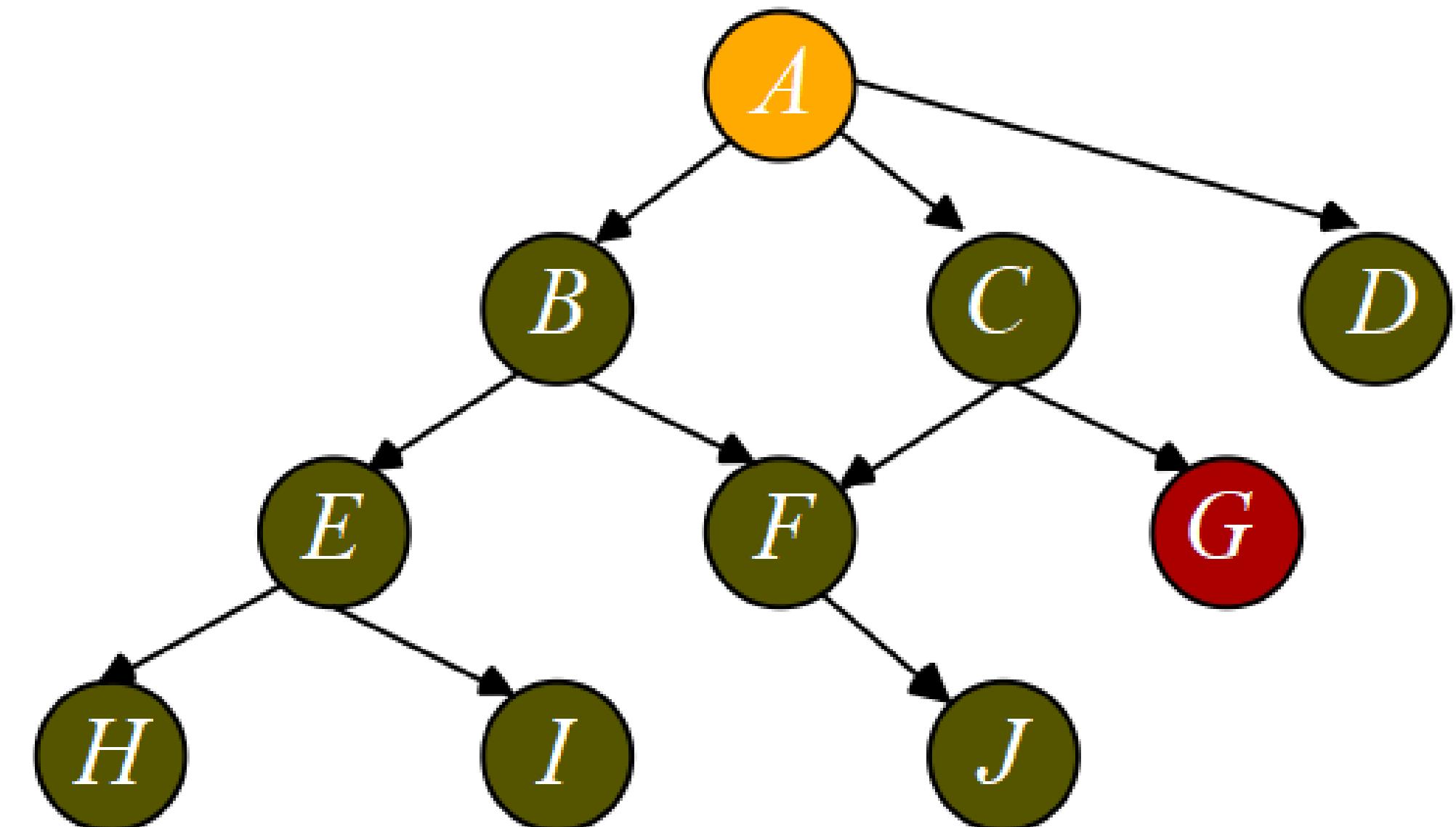
 If(n==g) Return TRUE;

 Open = Open ∪ Γ(n);

 Close = Close ∪ {n}

}

End;



s = A là đỉnh bắt đầu

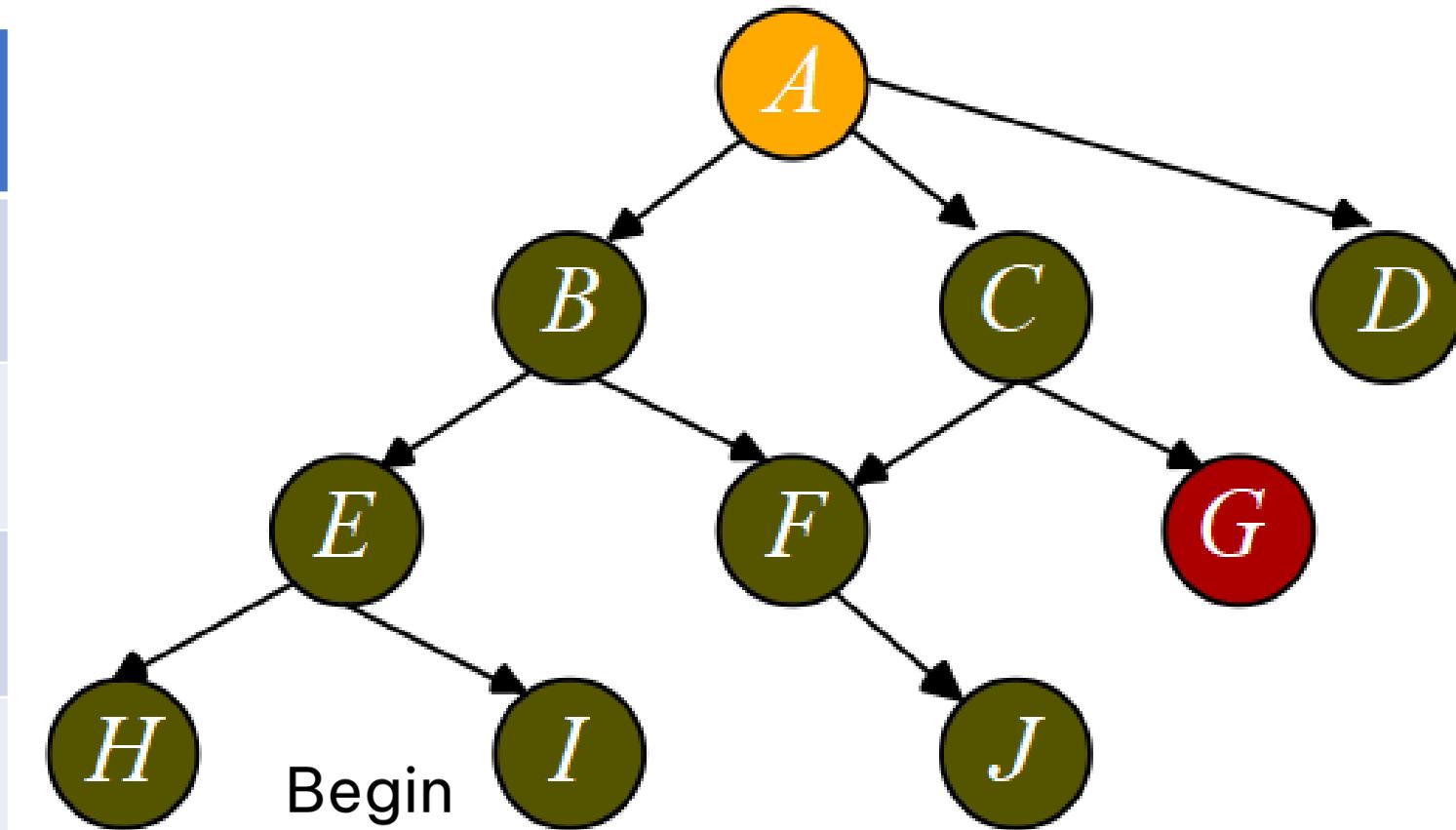
g=G là đỉnh kết thúc

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều rộng (Breath First Search)

Bước	n	$\Gamma(n)$	Open	Close
0			{A}	\emptyset
1	A	{B,C,D}	{B,C,D}	{A}
2	B	{E,F}	{C,D,E,F}	{A,B}
3	C	{F,G}	{D,E,F,G}	{A,B,C}
4	D	\emptyset	{E,F,G}	{A,B,C,D}
5	E	{H,I}	{F,G,H,I}	{A,B,C,D,E}
6	F	{J}	{G,H,I,J}	{A,B,C,D,E,F}
7	G	TRUE		

■ $A \Rightarrow C \Rightarrow G$



```

Begin
  Open={s};
  Close=∅;
  While(Open != ∅){
    n = Retrieve(Open);
    If(n==g) Return TRUE;
    Open = Open ∪  $\Gamma(n)$ ;
    Close = Close ∪ {n}
  }
End;
  
```

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều rộng (Breath First Search)

Begin

Open={s};

Close=∅;

While(Open != ∅){

n = Retrieve(Open);

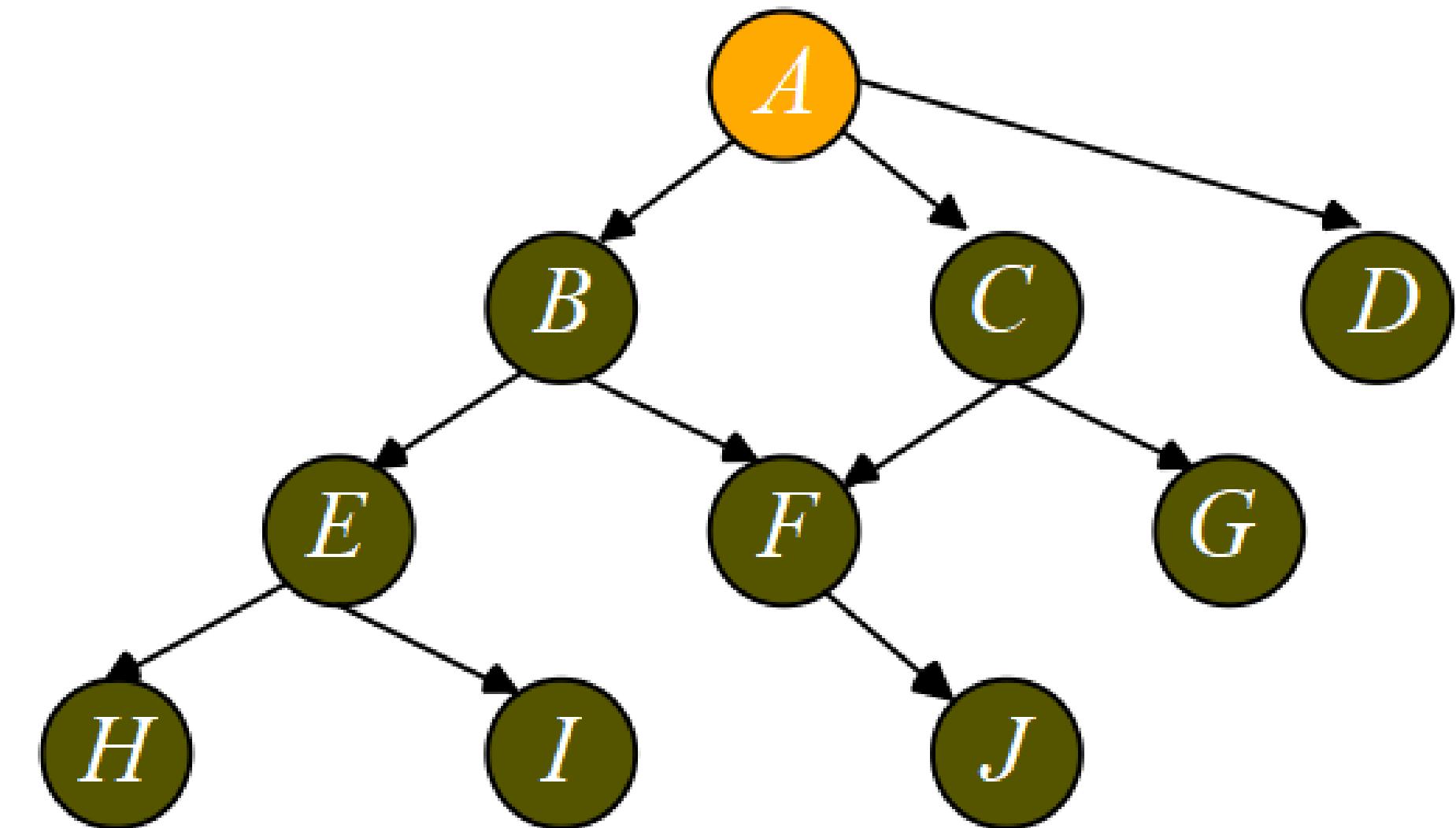
If(n==g) Return TRUE;

Open = Open \cup $\Gamma(n)$;

Close = Close \cup {n}

}

End;



Ví dụ 2

s = A là đỉnh bắt đầu

g=U là đỉnh kết thúc

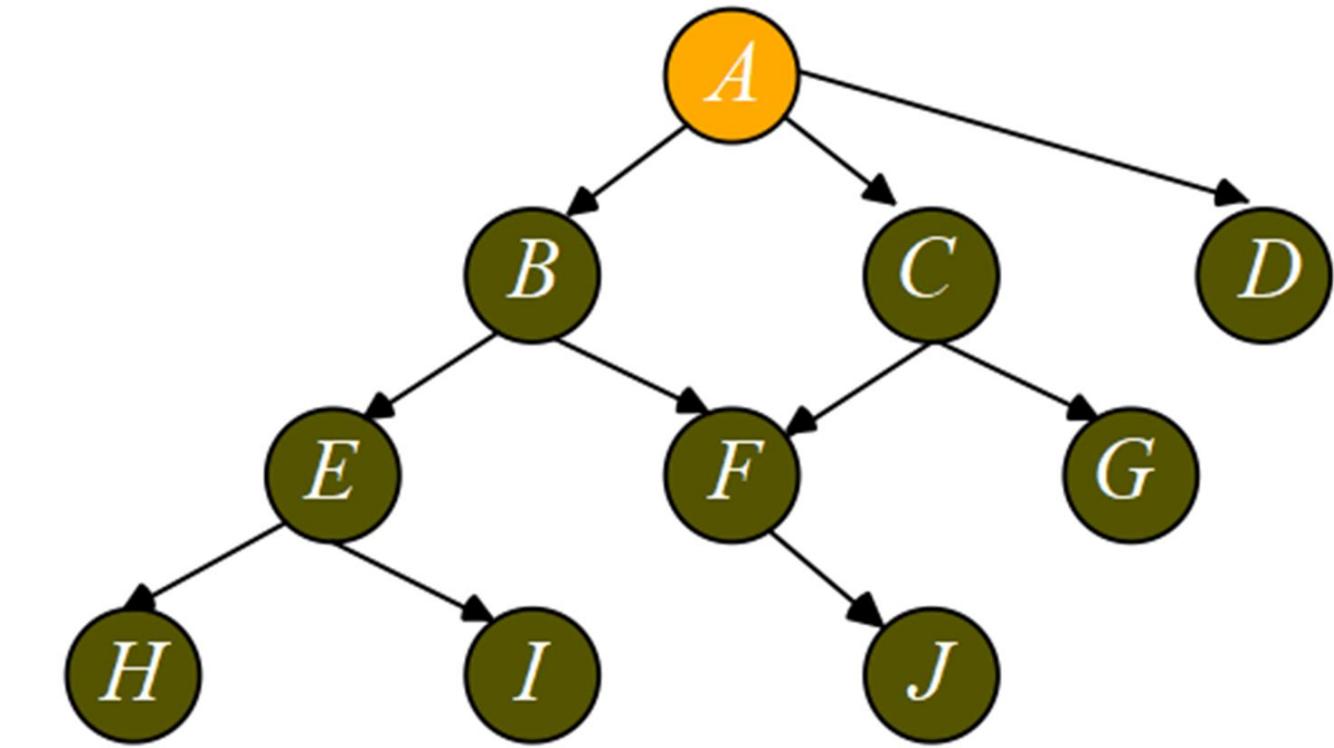
Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều rộng (Breath First Search)

Bước	n	$\Gamma(n)$	Open	Close
0			{A}	\emptyset
1	A	{B,C,D}	{B,C,D}	{A}
2	B	{E,F}	{C,D,E,F}	{A,B}
3	C	{F,G}	{D,E,F,G}	{A,B,C}
4	D	\emptyset	{E,F,G}	{A,B,C,D}
5	E	{H,I}	{F,G,H,I}	{A,B,C,D,E}

$s = A$ là đỉnh bắt đầu

$g=U$ là đỉnh kết thúc



Begin

```

Open={s};
Close= $\emptyset$ ;
While(Open !=  $\emptyset$ ){
    n = Retrieve(Open);
    If(n==g) Return TRUE;
    Open = Open  $\cup$   $\Gamma(n)$ ;
    Close = Close  $\cup$  {n}
}
End;
  
```

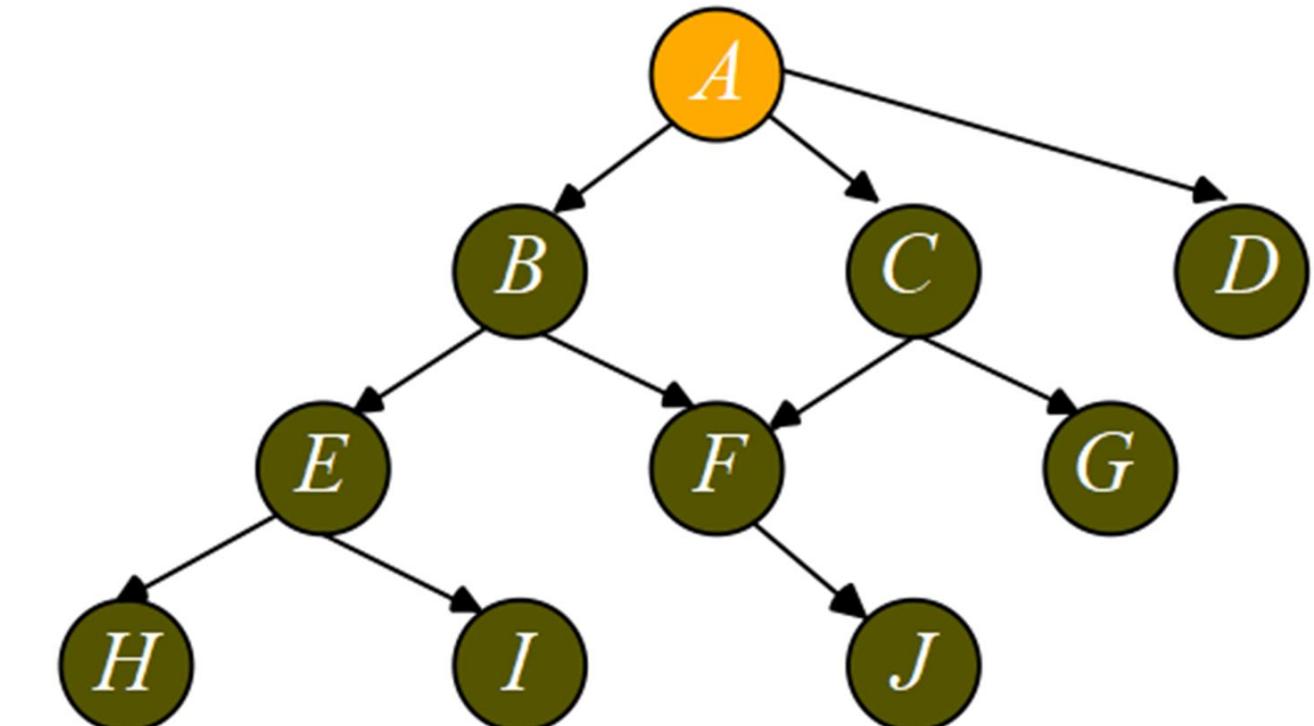
Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều rộng (Breath First Search)

Bước	n	$\Gamma(n)$	Open	Close
6	F	{J}	{G,H,I,J}	{A,B,C,D,E,F}
7	G	\emptyset	{H,I,J}	{A,B,C,D,E,F,G}
8	H	\emptyset	{I,J}	{A,B,C,D,E,F,G,H}
9	I	\emptyset	{J}	{A,B,C,D,E,F,G,H,I}
10	J	\emptyset	\emptyset	{A,B,C,D,E,F,G,H,I,J}
11		FALSE		

s = A là đỉnh bắt đầu

g=U là đỉnh kết thúc



Begin

```

Open={s};
Close= $\emptyset$ ;
While(Open !=  $\emptyset$ ){
    n = Retrieve(Open);
    If(n==g) Return TRUE;
    Open = Open  $\cup$   $\Gamma(n)$ ;
    Close = Close  $\cup$  {n}
}
End;
  
```

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều rộng (Breath First Search)

Begin

 Open={s};

 Close=∅;

 While(Open != ∅){

 n = Retrieve(Open);

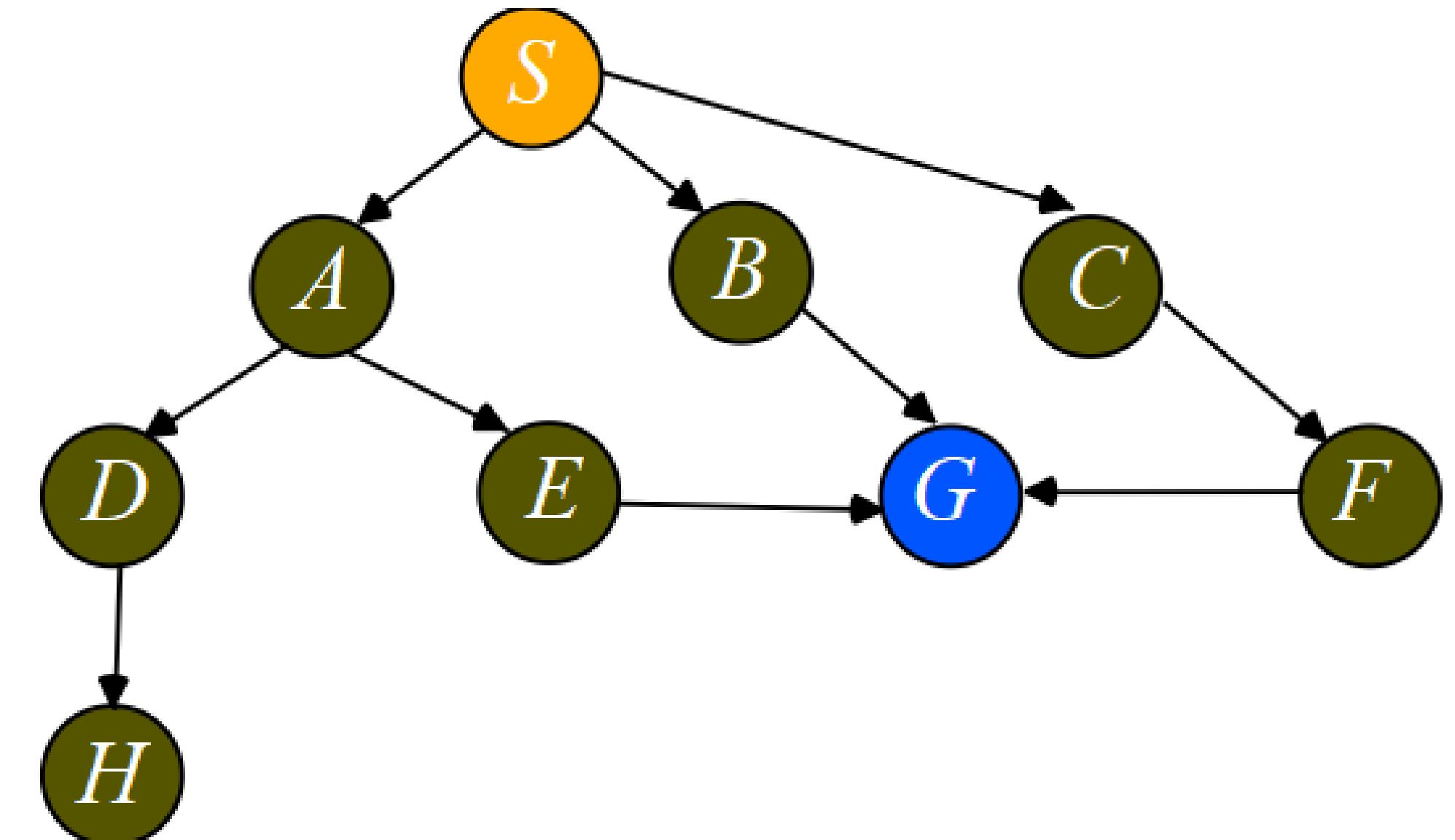
 If(n==g) Return TRUE;

 Open = Open ∪ Γ(n);

 Close = Close ∪ {n}

}

End;



Ví dụ 3

S là đỉnh bắt đầu G là đỉnh kết thúc

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

▪ Tìm kiếm theo chiều rộng (Breath First Search)

Begin

 Open={s};

 Close=∅;

 While(Open != ∅){

 n = Retrieve(Open);

 If(n==g) Return TRUE;

 Open = Open \cup $\Gamma(n)$;

 Close = Close \cup {n}

}

End;

Nguyên tắc của BFS:

- *Tìm 1 nút biên và các nút kề*
- *Lấy 1 nút đầu của OPEN ra khỏi Queue*
- *Đưa 1 nút vào cuối OPEN*
- *Không đưa nút đã duyệt/hoặc đã có vào Queue*
- *Khi thêm dựa theo thứ tự alphaB*

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

▪ Tìm kiếm theo chiều sâu (Depth First Search)

Begin

 Open={s};

 Close=∅;

 While(Open != ∅){

 n = Retrieve(Open);

 If(n==g) Return TRUE;

 Open = Open \cup $\Gamma(n)$;

 Close = Close \cup {n}

}

End;

Nguyên tắc của DFS:

- *Tìm 1 nút biên và các nút kề*
- *Lấy 1 nút đầu của OPEN ra khỏi Queue*
- *Chèn 1 nút vào đầu OPEN*
- *Không đưa nút đã duyệt/hoặc đã có vào Queue*
- *Khi thêm dựa theo thứ tự alphaB*

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều sâu (Depth First Search)

Begin

 Open={s};

 Close=∅;

 While(Open != ∅){

 n = Retrieve(Open);

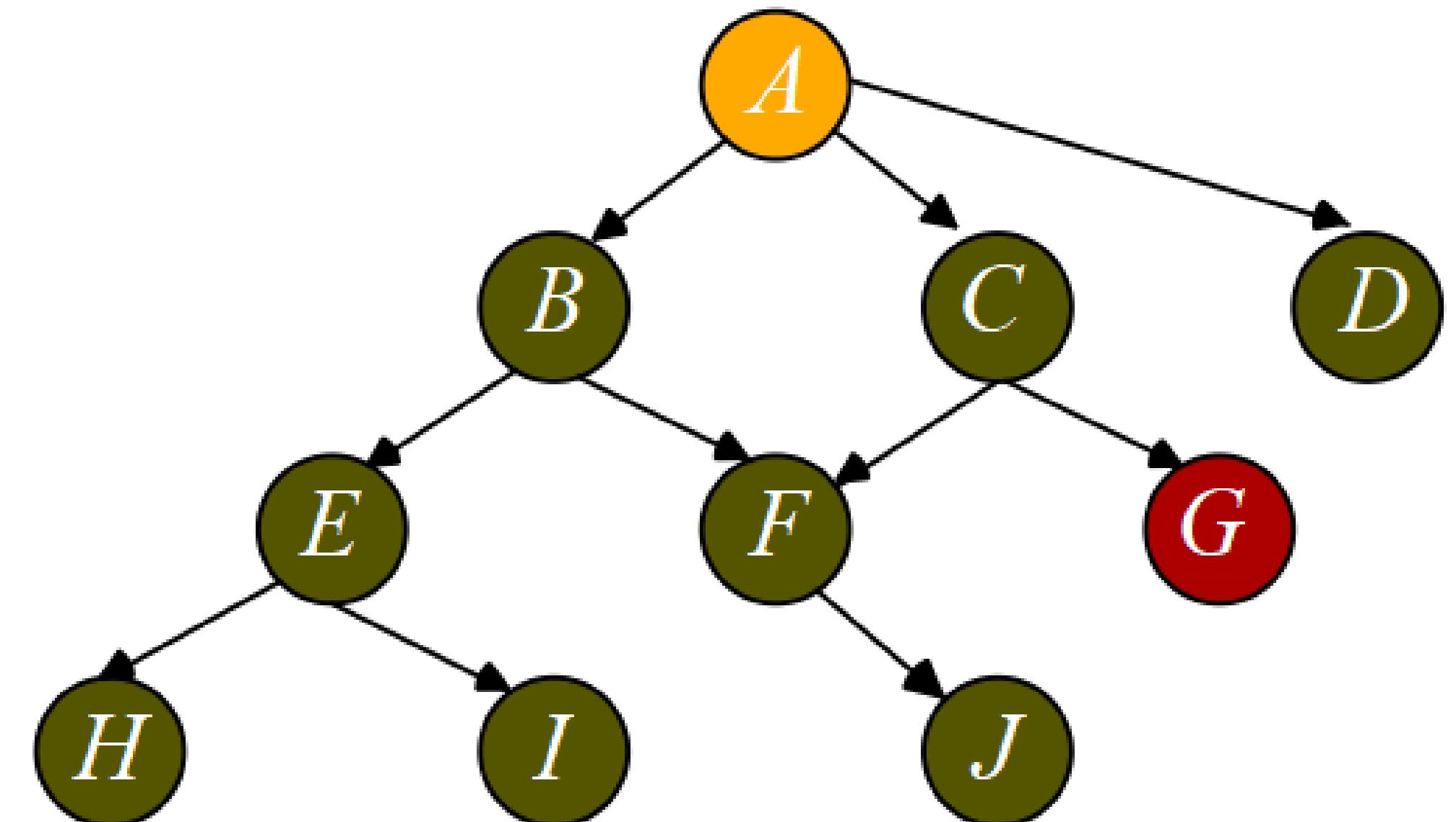
 If(n==g) Return TRUE;

 Open = Open ∪ Γ(n);

 Close = Close ∪ {n}

}

End;



Ví dụ 1

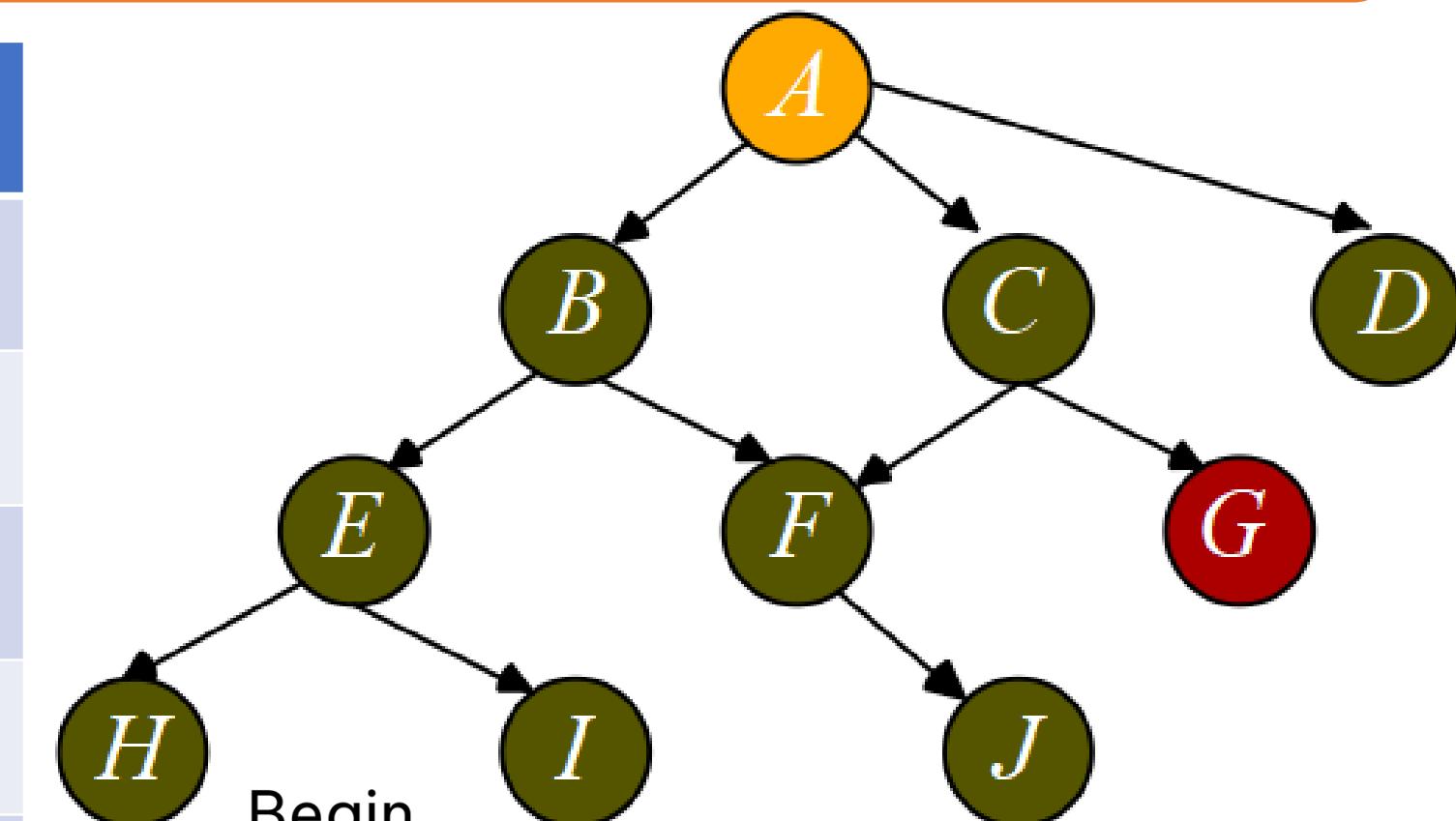
s = A là đỉnh bắt đầu

g=G là đỉnh kết thúc

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

▪ Tìm kiếm theo chiều sâu (Depth First Search)

Bước	n	$\Gamma(n)$	Open	Close
0			{A}	\emptyset
1	A	{B,C,D}	{B,C,D}	{A}
2	B	{E,F}	{E,F,C,D}	{A,B}
3	E	{H,I}	{H,I,F,C,D}	{A,B,E}
4	H	\emptyset	{I,F,C,D}	{A,B,E,H}
5	I	\emptyset	{F,C,D}	{A,B,E,H,I}
6	F	{J}	{J,C,D}	{A,B,E,H,I,F}
7	J	\emptyset	{C,D}	{A,B,E,H,I,F,J}
8	C	{F,G}	{G,D}	{A,B,E,H,I,F,J,C}
9	G	TRUE		■ A \Rightarrow C \Rightarrow G



```

Begin
  Open={s};
  Close= $\emptyset$ ;
  While(Open !=  $\emptyset$ ){
    n = Retrieve(Open);
    If(n==g) Return TRUE;
    Open = Open  $\cup$   $\Gamma(n)$ ;
    Close = Close  $\cup$  {n}
  }
End;
  
```

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều sâu (Depth First Search)

Begin

 Open={s};

 Close=∅;

 While(Open != ∅){

 n = Retrieve(Open);

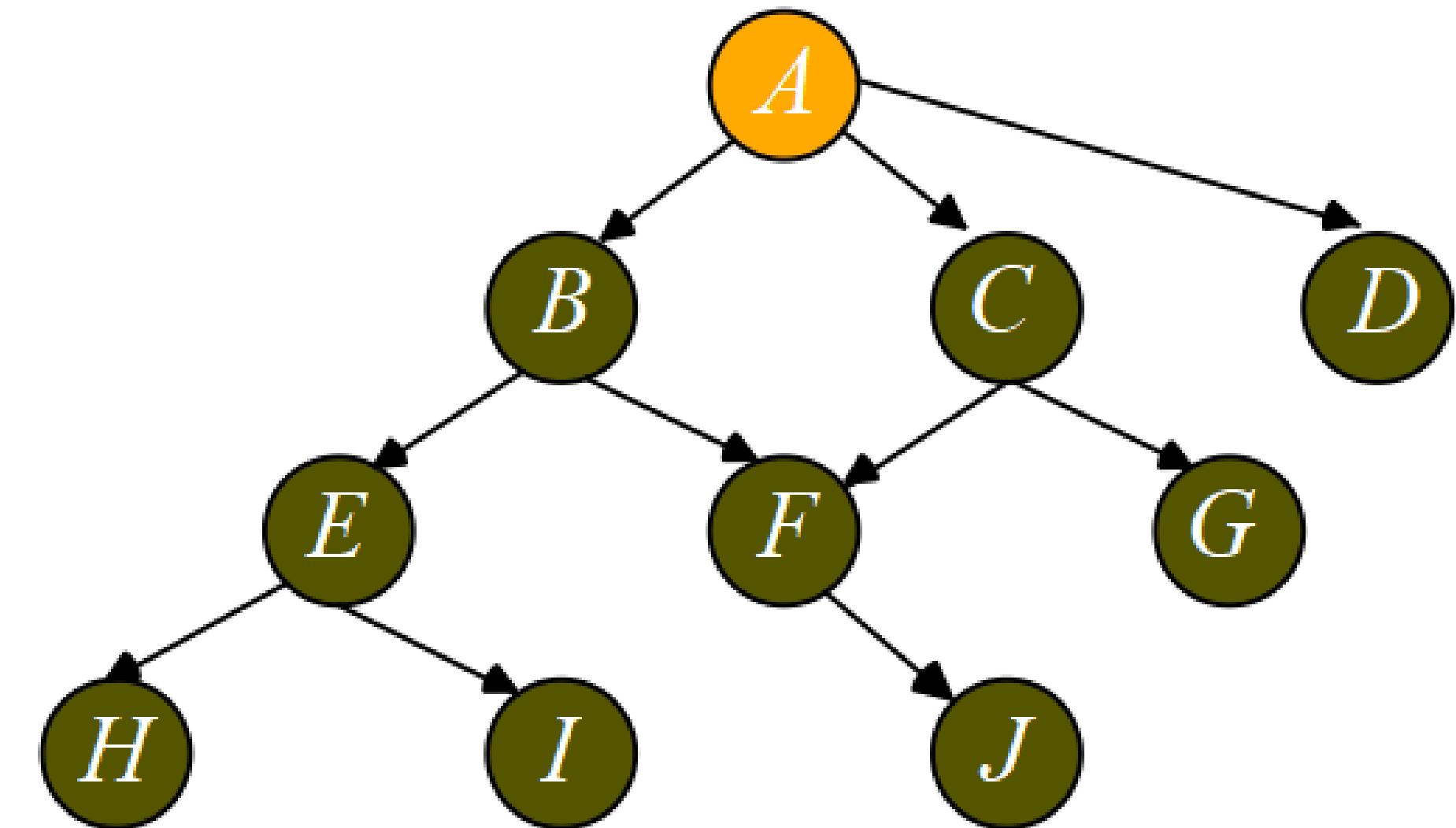
 If(n==g) Return TRUE;

 Open = Open ∪ Γ(n);

 Close = Close ∪ {n}

 }

End;



Ví dụ 2

s = A là đỉnh bắt đầu

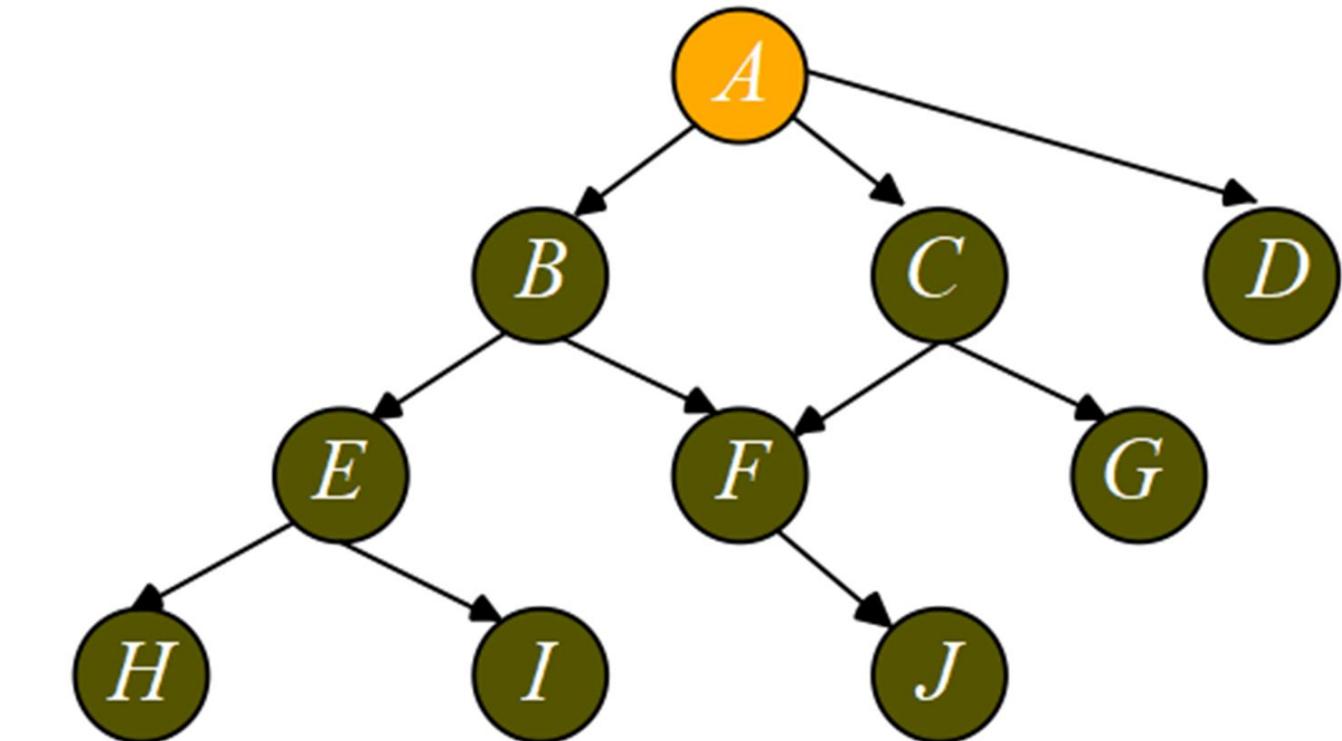
g=U là đỉnh kết thúc

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều sâu (Depth First Search)

Bước	n	$\Gamma(n)$	Open	Close
0			{A}	\emptyset
1	A	{B,C,D}	{B,C,D}	{A}
2	B	{E,F}	{E,F,C,D}	{A,B}
3	E	{H,I}	{H,I,F,C,D}	{A,B,E}
4	H	\emptyset	{I,F,C,D}	{A,B,E,H}
5	I	\emptyset	{F,C,D}	{A,B,E,H,I}

s = A là đỉnh bắt đầu g=U là đỉnh kết thúc



```

Begin
  Open={s};
  Close= $\emptyset$ ;
  While(Open !=  $\emptyset$ ){
    n = Retrieve(Open);
    If(n==g) Return TRUE;
    Open = Open  $\cup$   $\Gamma(n)$ ;
    Close = Close  $\cup$  {n}
  }
End;
  
```

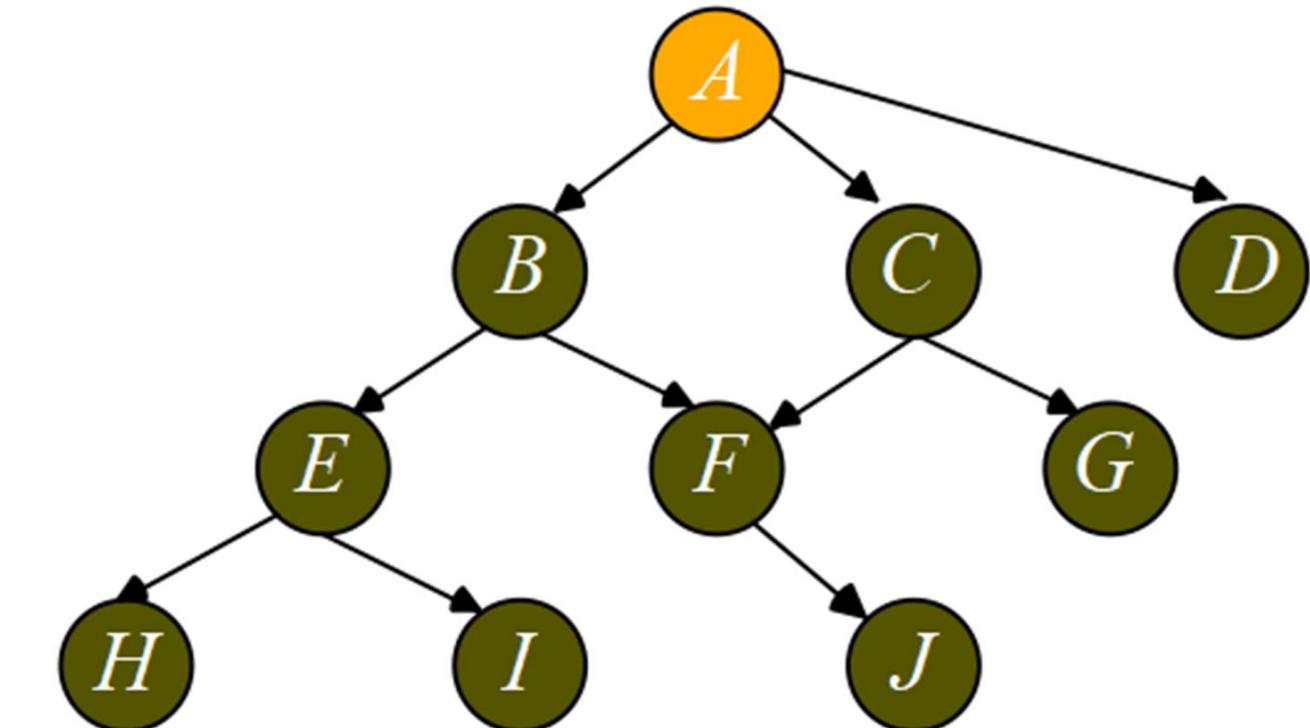
Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều sâu (Depth First Search)

Bước	n	$\Gamma(n)$	Open	Close
6	F	{J}	{J,C,D}	{A,B,E,H,I,F}
7	J	\emptyset	{C,D}	{A,B,E,H,I,F,J}
8	C	{F,G}	{G,D}	{A,B,E,H,I,F,J,C}
9	G	\emptyset	{D}	{A,B,E,H,I,F,J,C,G}
10	D	\emptyset	\emptyset	{A,B,E,H,I,F,J,C,G,D}
11	\emptyset	FALSE		

s = A là đỉnh bắt đầu

g=U là đỉnh kết thúc



Begin

Open={s};

Close= \emptyset ;

While(Open != \emptyset){

 n = Retrieve(Open);

 If(n==g) Return TRUE;

 Open = Open \cup $\Gamma(n)$;

 Close = Close \cup {n}

}

End;

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều sâu (Depth First Search)

Begin

 Open={s};

 Close=∅;

 While(Open != ∅){

 n = Retrieve(Open);

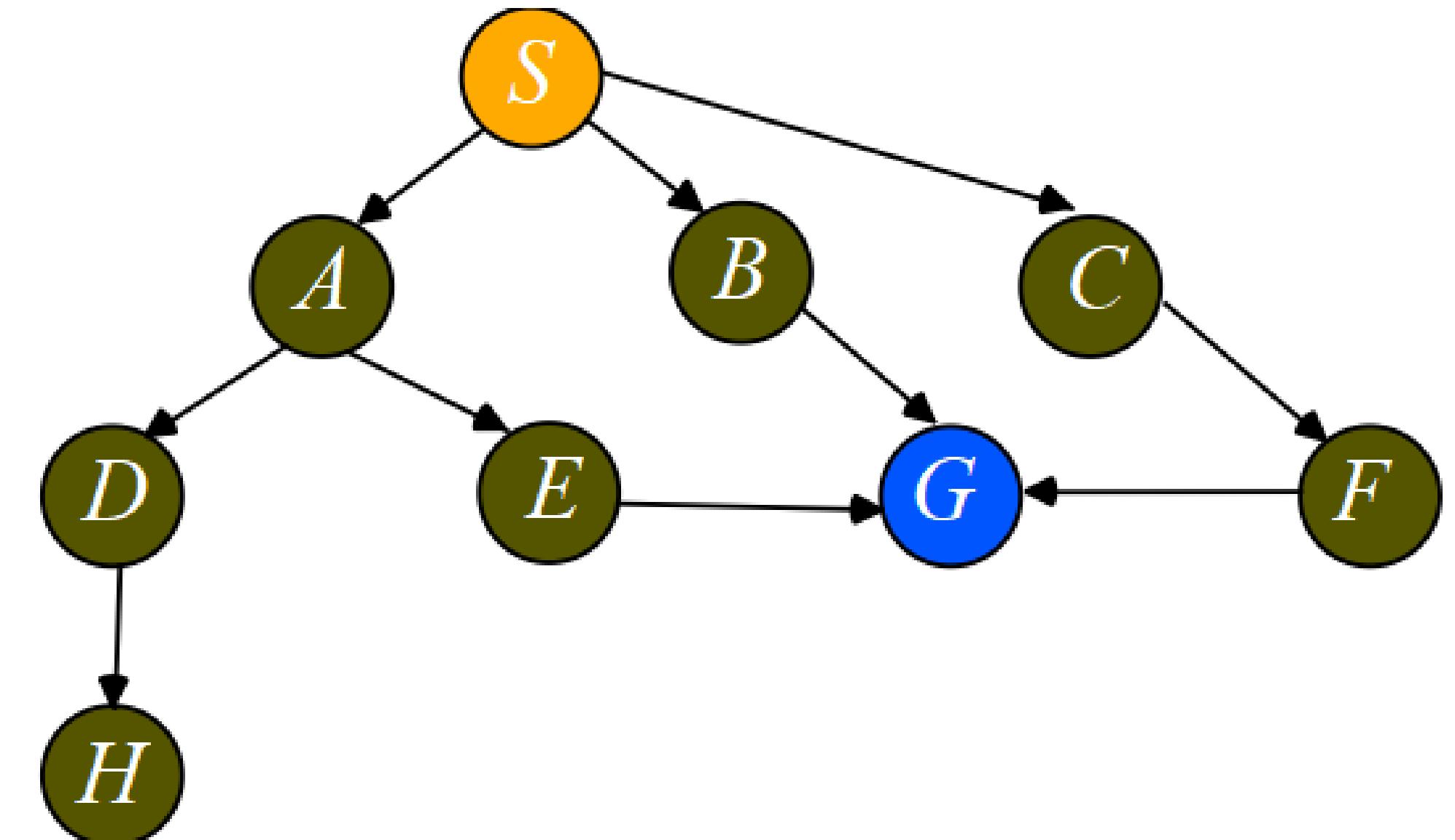
 If(n==g) Return TRUE;

 Open = Open ∪ Γ(n);

 Close = Close ∪ {n}

}

End;



Ví dụ 3

S là đỉnh bắt đầu G là đỉnh kết thúc

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

■ BFS

Open = [A]; closed = []

Open = [B,C,D];

closed = [A]

Open = [C,D,E,F];

closed = [B,A]

Open = [D,E,F,G,H]; closed = [C,B,A]

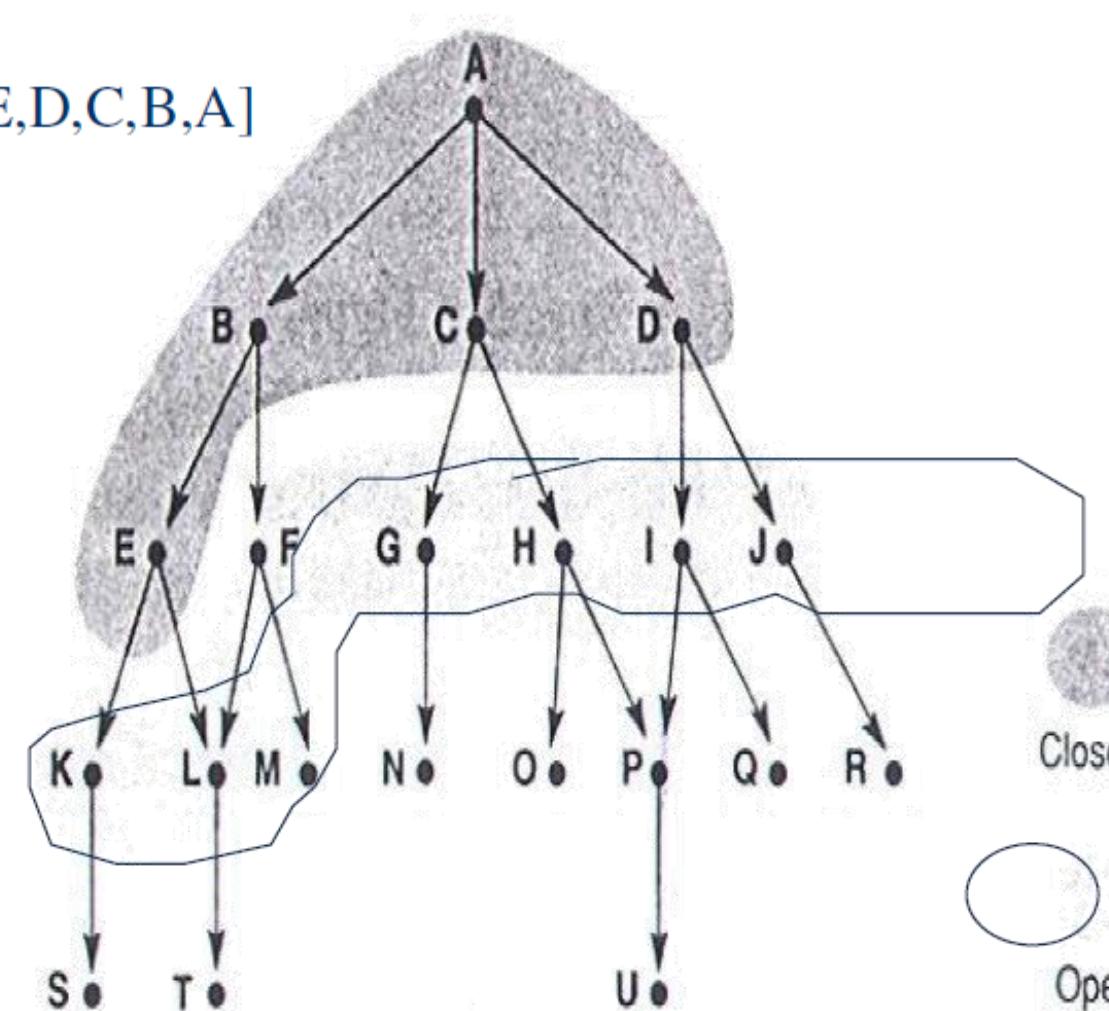
Open = [E,F,G,H,I,J]; closed = [D,C,B,A]

Open = [F,G,H,I,J,K,L];closed = [E,D,C,B,A]

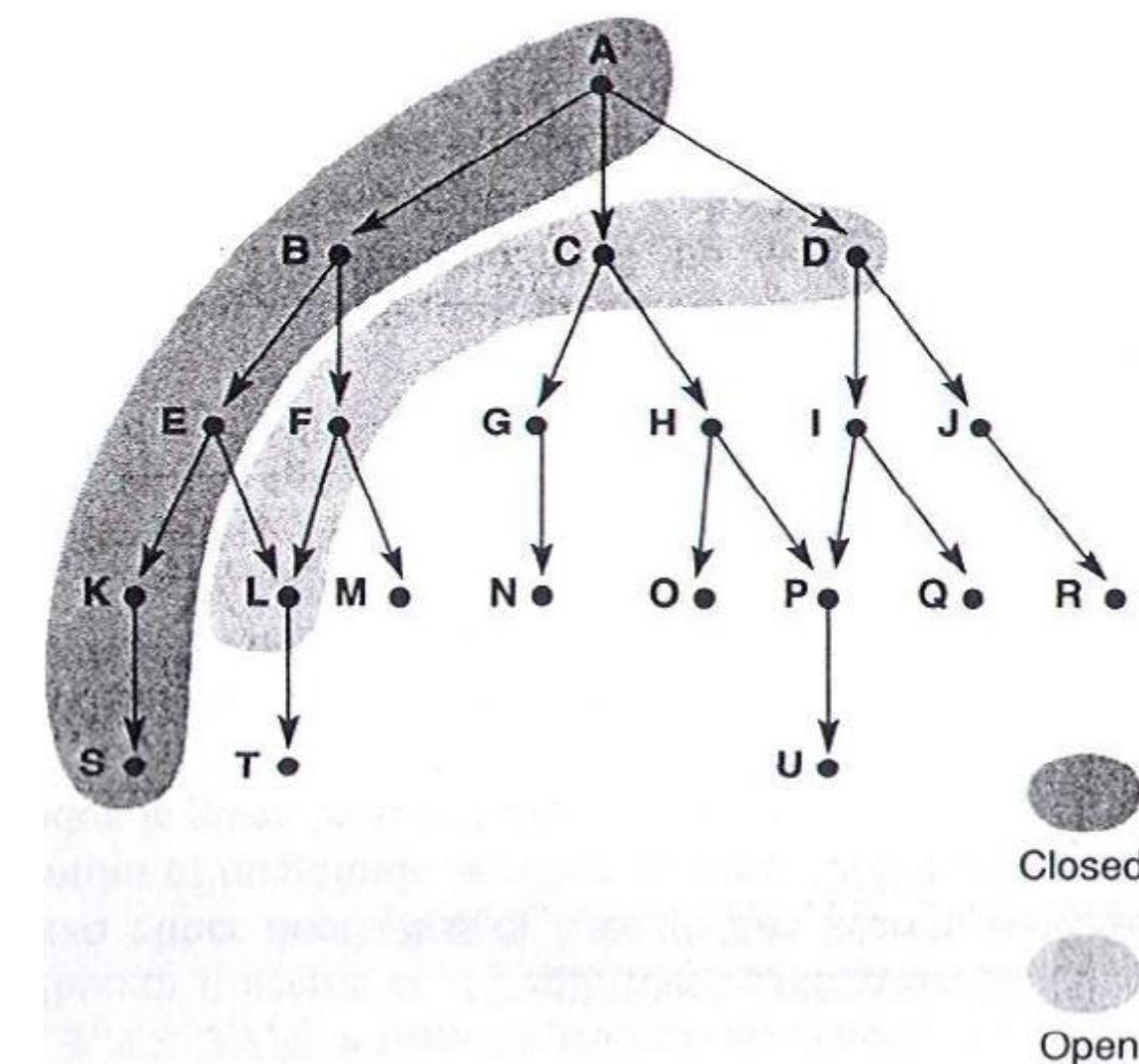
Open = [G,H,I,J,K,L,M];(vì L đã có trong open);

closed = [F,E,D,C,B,A]

...



■ DFS



Open = [A]; closed = []

Open = [B,C,D]; closed = [A]

Open = [E,F,C,D];closed = [B,A]

Open = [K,L,F,C,D];
closed = [E,B,A]

Open = [S,L,F,C,D];
closed = [K,E,B,A]

Open = [L,F,C,D];
closed = [S,K,E,B,A]

Open = [T,F,C,D];
closed = [L,S,K,E,B,A]

Open = [F,C,D];
closed = [T,L,S,K,E,B,A]

...

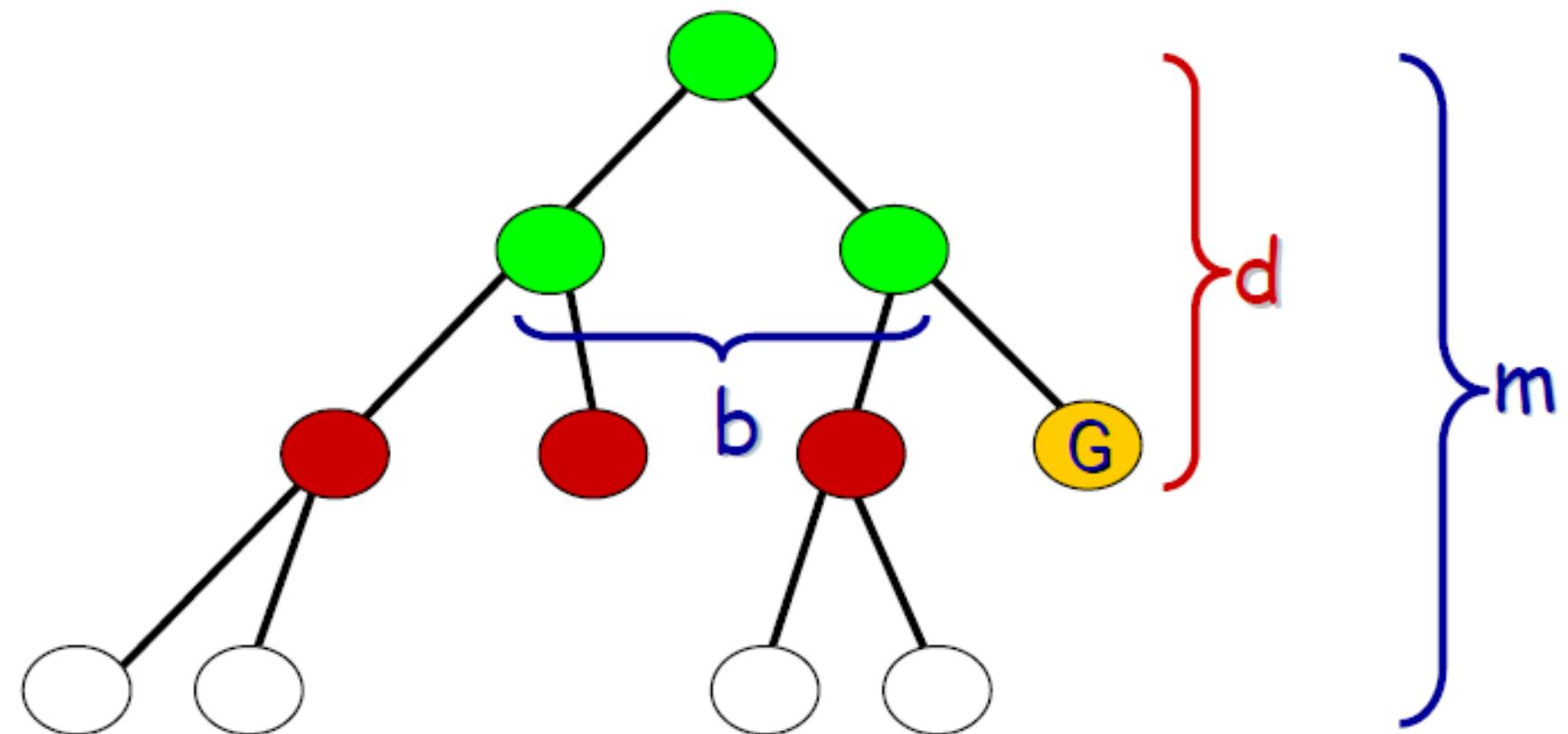
Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

▪ BFS

- OPEN được tổ chức dạng LIFO
- Nghiêm với số cung bé nhất
- Độ phức tạp thời gian: $O(b^d)$
- Độ phức tạp không gian: $O(b^d)$

▪ DFS

- OPEN được tổ chức dạng FIFO
- Thường cho kết quả nhanh hơn
- Độ phức tạp thời gian: $O(b^m)$
- Độ phức tạp không gian: $O(b \cdot m)$



Hàng đợi trong giải thuật BFS chỉ chứa các nút lá, vì vậy có kích thước là b^d

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

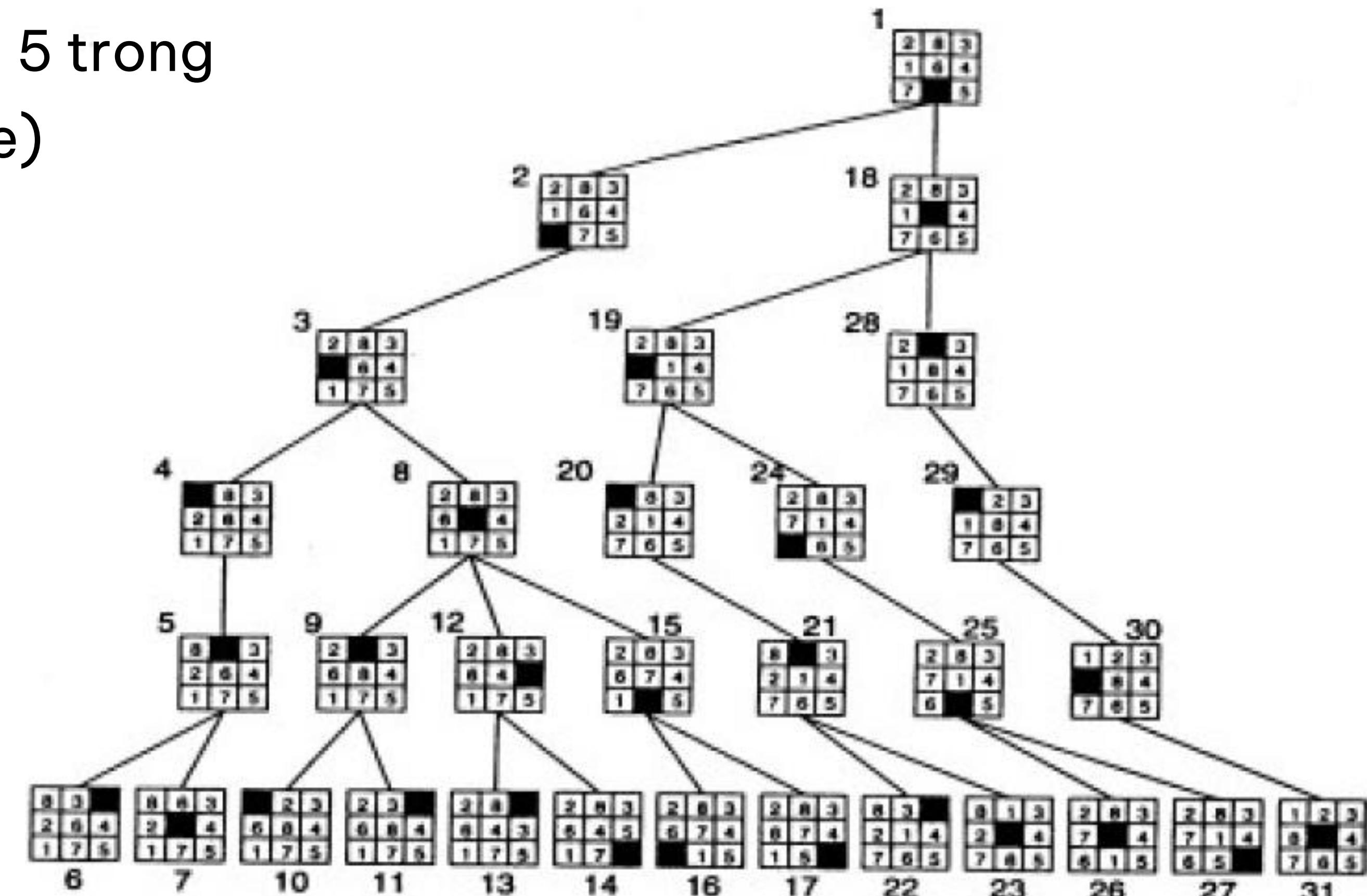
- Tìm kiếm theo chiều sâu có giới hạn (Depth Limited Search)
 - Giải thuật tìm kiếm theo chiều sâu ở trên có ưu điểm là nó có thể sinh ra lời giải nhanh chóng mà không tốn kém bộ nhớ.
 - Tuy nhiên nếu không gian trạng thái của bài toán là vô hạn thì rất có thể nó không tìm được lời giải của bài toán khi hướng tìm kiếm không chứa trạng thái đích.
 - ⇒ Để khắc phục nhược điểm này, chúng ta có thể đặt giới hạn độ sâu trong giải thuật: nếu độ sâu của trạng thái đang xét vượt quá ngưỡng nào đó thì chúng ta không bổ sung các nút kề với trạng thái này nữa mà chuyển sang hướng tìm kiếm khác.

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều sâu có giới hạn (Depth Limited Search)
 - Chiến lược giới hạn
 - Cố định một độ sâu MAX, như các kỳ thủ chơi cờ tính trước một số nước nhất định.
 - Tùy theo cấu hình phần cứng của máy tính.
 - Sử dụng Meta Knowledge trong giới hạn độ sâu
 - Giới hạn độ sâu cũng đồng nghĩa với việc co hẹp không gian trạng thái và nguy cơ dẫn đến mất nghiệm.

Chương 2. BÀI TOÁN VÀ PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

- Tìm kiếm theo chiều sâu có giới hạn (Depth Limited Search)
 - DFS có giới hạn = 5 trong trò chơi 8 - puzzle)



HẾT CHƯƠNG 2

Chương 3. HEURISTIC



TRÍ TUỆ NHÂN TẠO

Artificial Intelligence

Đoàn Vũ Thịnh
Khoa Công nghệ Thông tin
Đại học Nha Trang
Email: thinhdv@ntu.edu.vn

Nha Trang, 06-2023

Chương 3. HEURISTIC

- Heuristic là gì?
 - Heuristic là những tri thức được rút tỉa từ những kinh nghiệm, “trực giác” của con người (mẹo).
 - Heuristic có thể là những tri thức “đúng” hay “sai”.
 - Heuristic là những meta knowledge và “thường đúng”.
- Heuristic dùng để làm gì?
 - Khi bài toán có thể không có nghiệm chính xác do các mệnh đề không phát biểu chặt chẽ hay thiếu dữ liệu để khẳng định KQ.
 - Có nghiệm chính xác nhưng chi phí tính toán để tìm ra nghiệm là quá lớn (hệ quả của bùng nổ tổ hợp).
 - Heuristic giúp tìm kiếm đạt kết quả với chi phí thấp hơn.

Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
- Dạng tổng quát $\min\{f(x): x \in D\}$, D là tập các điểm rời rạc $\in \mathbb{R}^n$
- Ví dụ, máy ATM có $m=4$ loại tiền (100, 50, 20 và 10). Nếu đưa số tiền cần rút là n . Tìm phương án sao cho số tờ tiền là ít nhất.
- Gọi $x = (x_1, x_2, x_3, x_4)$ là các phương án rút tiền, với $x_1 = 100, x_2 = 50, x_3 = 20$ và $x_4 = 10$.
- $\min\{f(x_1 + x_2 + x_3 + x_4)\}$
- Điều kiện: $100 * x_1 + 50 * x_2 + 20 * x_3 + 10 * x_4 = n$
- Thông thường người ta sẽ vét cận $\min(f(x): x \in D)$

Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
 - Phương pháp tham lam là đưa ra các quyết định dựa vào thông tin đang có và trong tương lai sẽ không xem xét lại các tác động của các quyết định trong quá khứ.
 - Hàm Solution (S): nhận biết tính chấp nhận được của lời giải $S (S = \emptyset)$
 - Hàm Select (C): chọn từ tập C ứng viên triển vọng nhất để bổ sung vào lời giải hiện có.
 - Hàm Feasibe ($S+x$): Kiểm tra tính chấp nhận được của lời giải bộ phận $S+x$.
 - Ý tưởng: Xuất phát từ lời giải \emptyset , xây dựng lời giải của bài toán theo từng bước, mỗi bước sẽ chọn 1 p.tử để bổ sung vào lời giải hiện có

Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)

- Procedure Greedy;

- Begin

- $S = \emptyset;$

- While($C \neq \emptyset \text{ && } \text{not}(\text{Solution}(S))$):{

- $x \leftarrow \text{Select}(C);$

- $C = C \setminus x;$

- if($\text{Feasible}(S \cup C)$): $S = S \cup x;$

- }

- If($\text{Solution}(C)$): return $S;$

- End;

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

▪ Ví dụ bài toán cái túi: V_{max} và $w \leq W_{max}$

▪ Tham lam 1

▪ Sắp xếp đồ vật theo giá trị giảm dần

▪ Chọn đồ từ đầu – cuối

▪ Tham lam 2

▪ Sắp xếp đồ vật theo thứ tự tăng dần

▪ Chọn đồ từ đầu – cuối

▪ Tham lam 3

▪ Sắp xếp đồ vật theo tỷ trọng v/w

$$\frac{V_1}{W_1} > \frac{V_2}{W_2} > \frac{V_3}{W_3} > \dots$$

Đồ vật	1	2	3	Đồ vật	1	2	3	Đồ vật	1	2	3
Gía trị	20	16	8	Gía trị	10	16	28	Gía trị	60	100	120
T. lượng	14	6	10	T. lượng	5	6	10	T. lượng	10	20	30
Wmax = 19			Wmax = 19			Wmax = 50			Greedy 1: i=1, Value = 20		
Greedy 2: i={1,2}, Value = 24			Greedy 3: i={1,2}, Value = 160			Tối ưu: i={2,3}, Value = 24			Tối ưu: i={2,3}, Value = 220		

Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
- Cho các tờ tiền có mệnh giá lần lượt là: 500, 200, 100, 100, 50, 50, 20, 20, 20, 20, 10.
- Cho số tiền $m = 390 \Rightarrow$ *Cách đổi với số tờ ít nhất.*
- B1. Sắp xếp tờ tiền theo mệnh giá giảm dần: 500, 200, 200...10
- B2. Duyệt từ đầu danh sách \Rightarrow cuối danh sách ở B1
 - $\text{if}(m >= A[i]): m = m - A[i]; //$ số tiền giảm dần
- B3. Kiểm tra $m < 0$: Không đổi được
- B4. Muốn in ra các tờ tiền đã đổi: dùng mảng VET[] để lưu trữ
 - Mỗi lần m đổi được: gán $VET[i] = 1$
 - Duyệt từ đầu \Rightarrow cuối danh sách VET[]:
 - $\text{if}(VET[i] == 1): \text{in } A[i];$

Chương 3. HEURISTIC

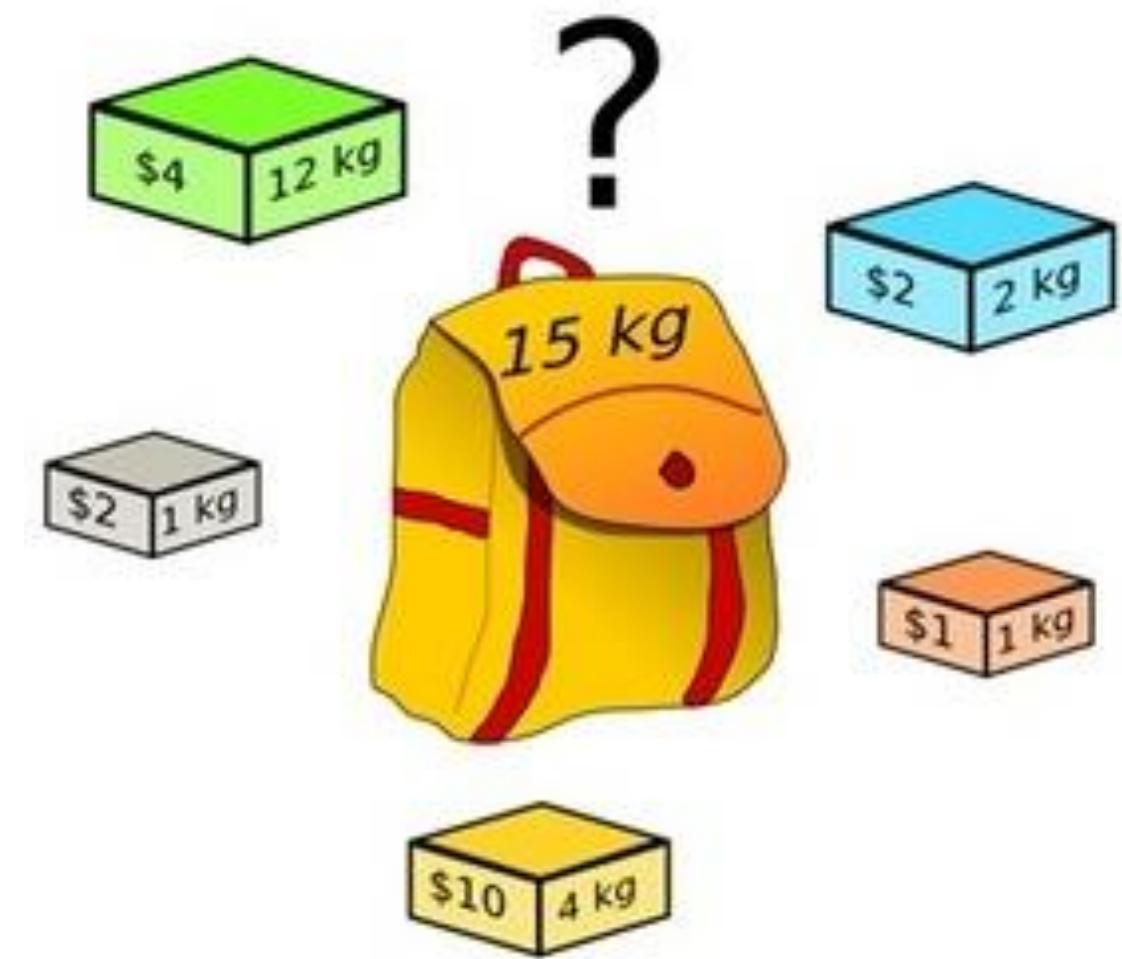
▪ Heuristic Greedy (Giải thuật tham lam)

Test	Input	Output
1	390 10 200,100,100,50,50,50,50,20,20,10	5 200,100,50,20,20
2	100 11 50,20,20,20,20,20,2,2,2,2,2	8 50,20,20,2,2,2,2,2
3	100 6 50,20,20,20,20,20	-1

- Test 1: Nghiệm tối ưu
- Test2: Nghiệm không tối ưu (**đổi thành 5 tờ 20 thì sẽ tối ưu hơn**)
- Test3: Có nghiệm nhưng trả lời vô nghiệm

Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
- Một chiếc ba lô có thể tích là Pmax. Có n vật, vật thứ i có thể tích là $P[i]$ và giá trị là $W[i]$.
- Hãy tìm cách sắp các vật vào ba lô sao cho tổng giá trị là lớn nhất. Với trọng lượng tối đa cho trước là P_{max} .

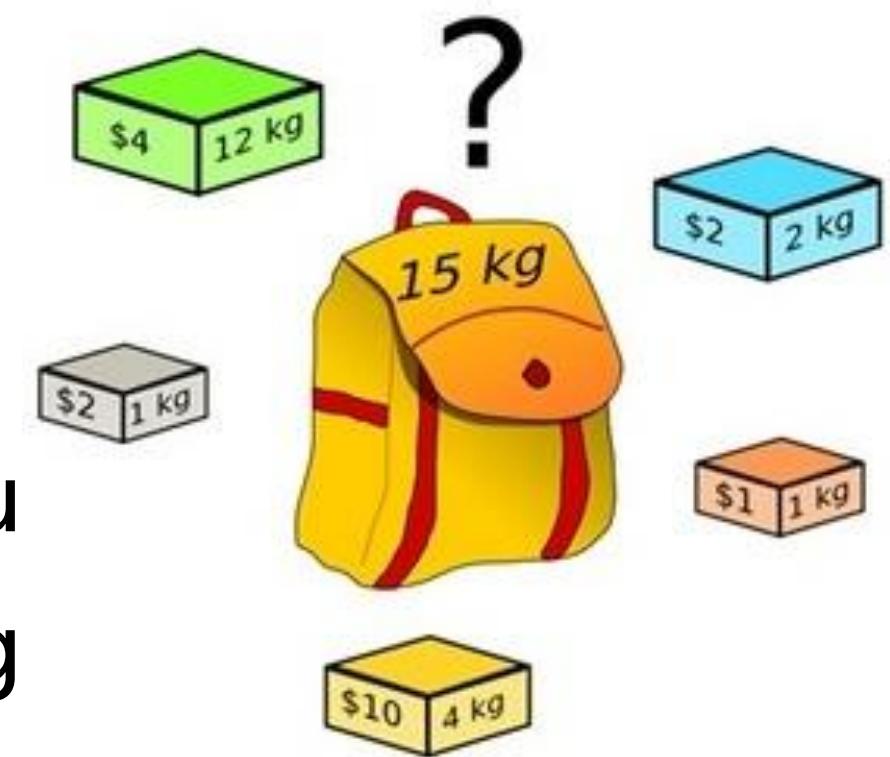


- Tham lam: Vật nào có kích thước bé nhưng giá trị lớn hơn cho vào túi trước.
- Trọng số ưu tiên: $\frac{W_i}{P_i}$ được sắp theo thứ tự giảm dần.

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

- Tổ chức dữ liệu:
- Mảng 2 chiều P[], W[]: thể tích và giá trị
- Mảng 1 chiều cs[]: thứ tự giảm dần theo trọng số ưu tiên
- Mảng cs chỉ chứa các chỉ số của phần tử có trọng số ưu tiên giảm dần, thứ tự các vật theo đề bài không đổi trong suốt quá trình xử lý.



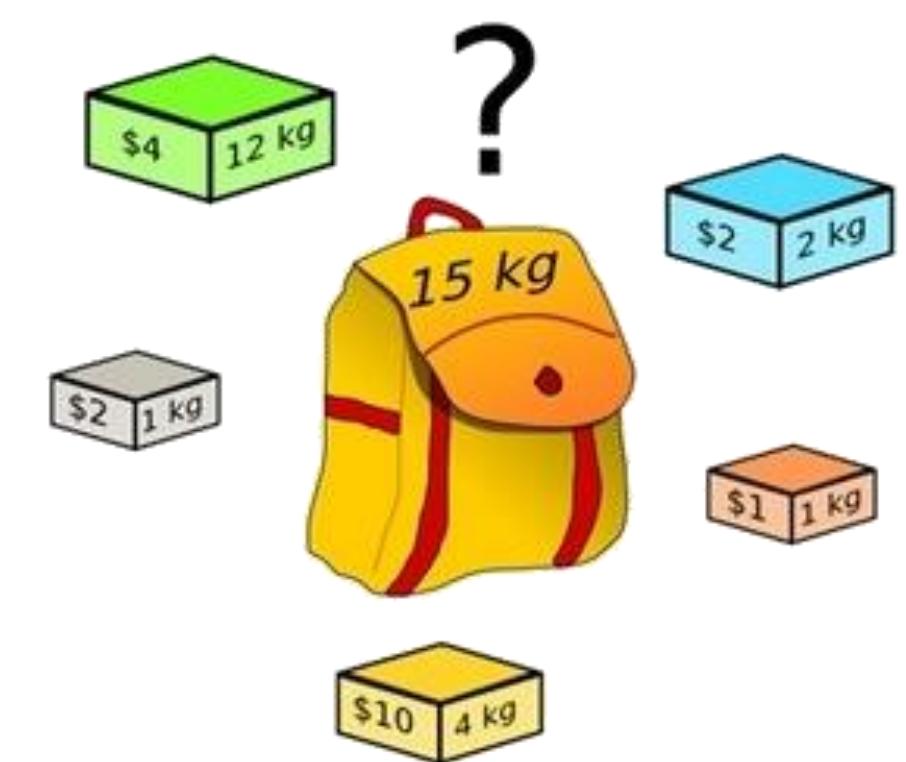
- B1. Sắp xếp các vật theo thứ tự giảm dần của trọng số ưu tiên
- B2. Xếp các vật vào ba lô.

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

▪ B1. Sắp xếp các vật theo thứ tự giảm dần của trọng số ưu tiên

```
for(i = 1; i < n; i + +){  
    for(j = i + 1; j ≤ n; j + +){  
        if(W[cs[i]]/P[cs[i]] < W[cs[j]]/P[cs[j]]]):  
            tam = cs[j]; cs[j] = cs[i]; cs[i] = tam;
```



▪ B2. Xếp các vật vào ba lô.

- $S = 0; i = 1;$
- $while((S < Pmax) \&\& (i \leq n)):$
 - $if(S + P[cs[i]] \leq Pmax):$
 - $S += P[cs[i]];$
 - $printf(\$5d, cs[i]);$
 - $i ++;$

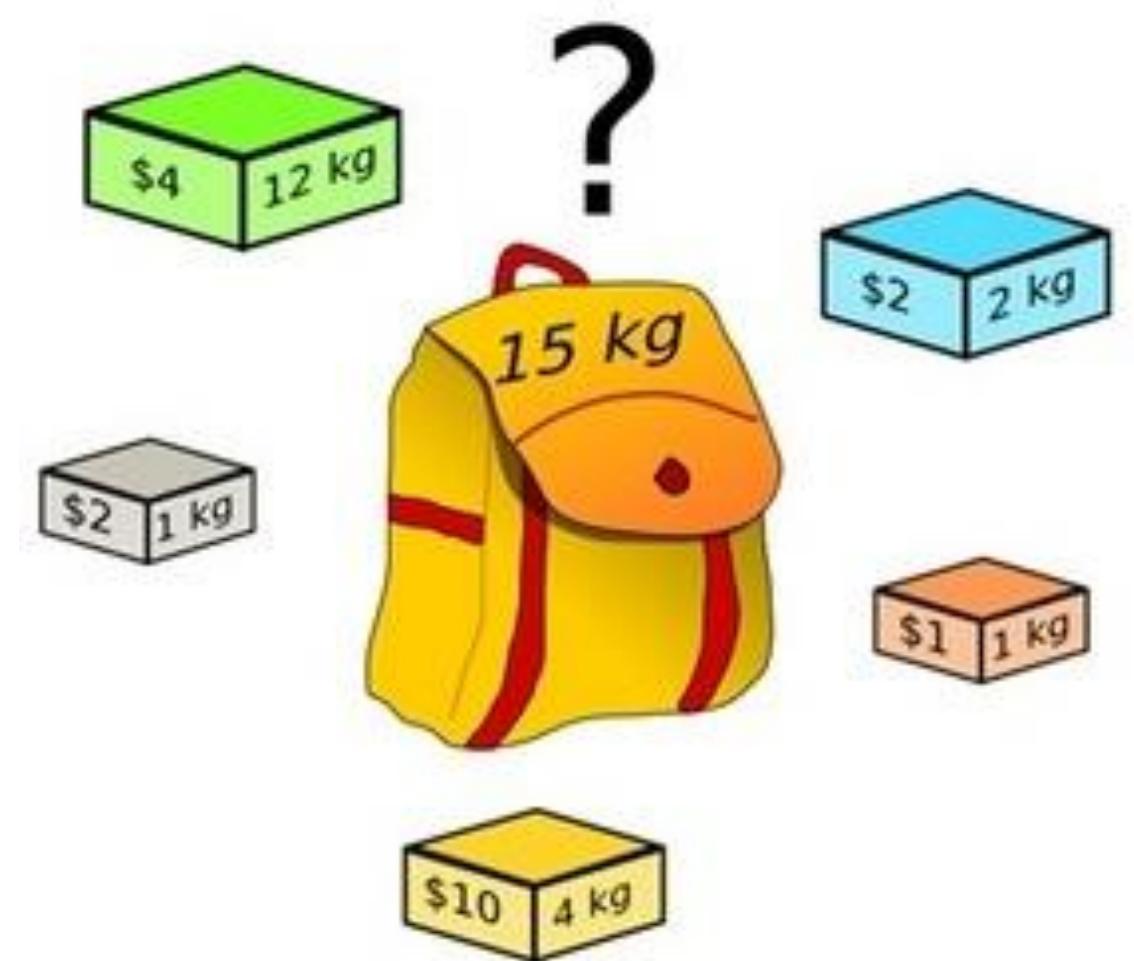
Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

Đồ vật	1	2	3	4	5
Gía trị	10\$	2\$	2\$	1\$	4\$
T. lượng	4kg	1kg	2kg	1kg	12kg
Wmax = 15kg					
Greedy 3: i={1,2,3,4}, Value = 15\$, Weight = 8kg					

Đồ vật	1	5	3	2	4
Gía trị	10\$	4\$	2\$	2\$	1\$
T. lượng	4kg	12kg	2kg	1kg	1kg
Wmax = 15kg					
Greedy 2: i={1,2,3,4}, Value = 15\$, Weight = 8kg					

Đồ vật	4	2	3	5	1
Gía trị	1\$	2\$	2\$	4\$	10\$
T. lượng	1kg	1kg	2kg	12kg	4kg
Wmax = 15kg					
Greedy 1: i={1,2,3,4}, Value = 15\$, Weight = 8kg					



Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
 - Cho n thành phố (1,2,...n) và khoảng cách giữa chúng ($c_{i,j}$). Hãy tìm hành trình của một người đưa thư, đi qua tất cả các thành phố rồi quay về thành phố xuất phát, sao cho tổng chiều dài đường đi là ngắn nhất.

	A	B	C	D	E	F	G	H
A	0	730	640	840	800	430	380	1010
B	730	0	710	1040	500	300	540	470
C	640	710	0	1420	1050	600	920	1160
D	840	1040	1420	0	740	950	570	900
E	800	500	1050	740	0	520	460	200
F	430	300	600	950	520	0	390	690
G	380	540	920	570	460	390	0	660
H	1010	470	1160	900	200	690	660	0

- Vét cạn: $(n-1)!$. Với n lớn?
- Greedy: Mỗi bước chọn $i \rightarrow j$ sao cho j gần i nhất trong những thành phố chưa đến nối với i.

Chương 3. HEURISTIC

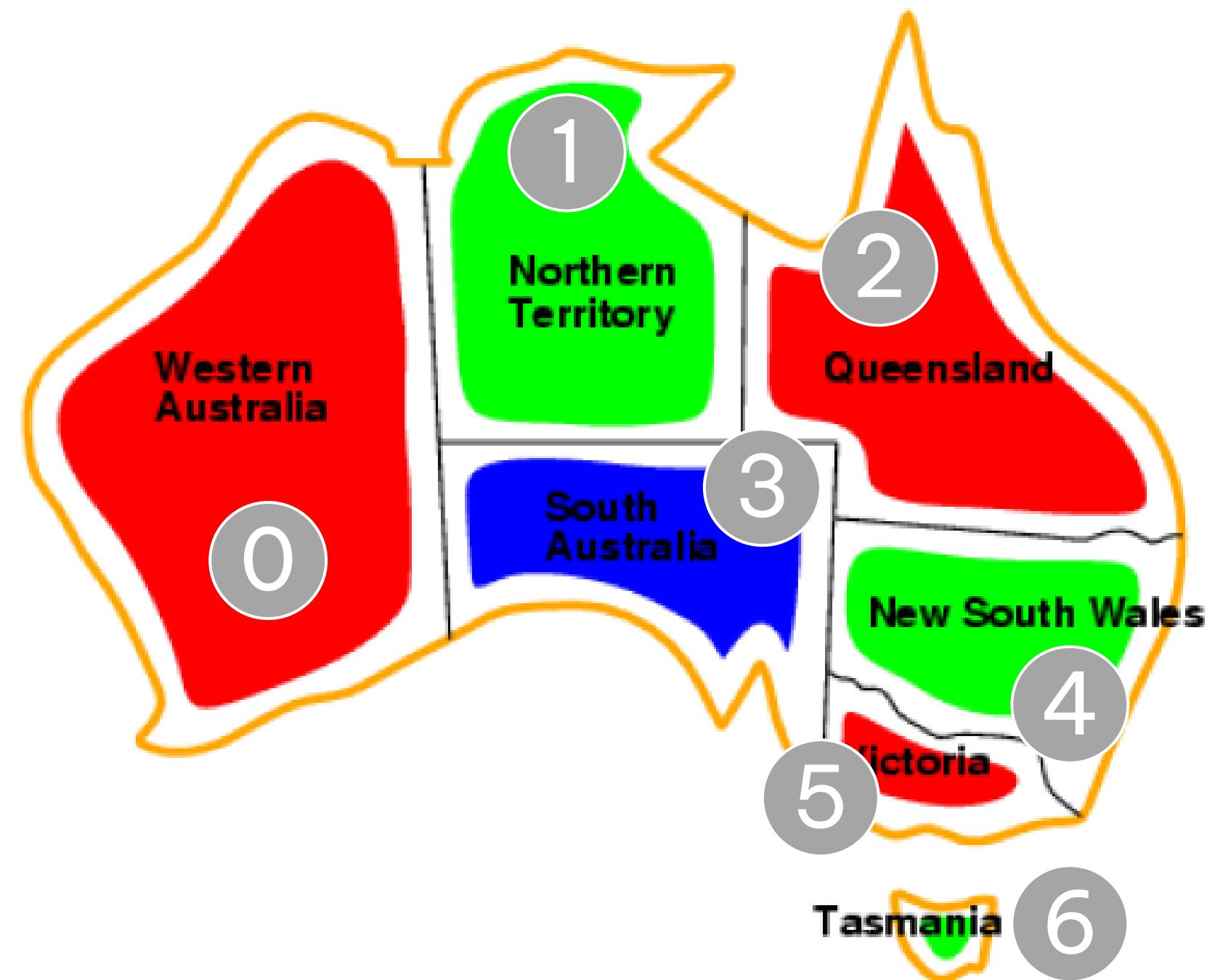
- Heuristic Greedy (Giải thuật tham lam)
 - Cho n thành phố (1,2,...n) và khoảng cách giữa chúng ($c_{i,j}$). Hãy tìm hành trình của một người đưa thư, đi qua tất cả các thành phố rồi quay về thành phố xuất phát, sao cho tổng chiều dài đường đi là ngắn nhất.

	A	B	C	D	E	F	G	H
A	0	730	640	840	800	430	380	1010
B	730	0	710	1040	500	300	540	470
C	640	710	0	1420	1050	600	920	1160
D	840	1040	1420	0	740	950	570	900
E	800	500	1050	740	0	520	460	200
F	430	300	600	950	520	0	390	690
G	380	540	920	570	460	390	0	660
H	1010	470	1160	900	200	690	660	0

- Greedy 1: A → G → F → B → H → E
→ D → C → A

Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
 - Sử dụng ba màu để tô bản đồ các tỉnh của một nước sao cho các tỉnh kế nhau thì có màu khác nhau.
 - Ví dụ, nước Australia có 7 bang như hình vẽ, chỉ sử dụng ba màu: đỏ, xanh lơ và xanh da trời để tô màu 7 bang của nước Australia sao cho không có hai bang nào kề nhau lại có màu giống nhau.
 - Tập ràng buộc: $WA \neq NT$, $WA \neq SA$, $NT \neq SA$, $NT \neq Q$, $SA \neq Q$, $SA \neq NSW$, $SA \neq V$, $Q \neq NSW$, $NSW \neq V$



Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
 - **Bước 1:** Lập danh sách $V' := [v_1, v_2, \dots, v_n]$ là các đỉnh của đồ thị được sắp xếp theo thứ tự bậc giảm dần: $d(v_1) > d(v_2) > \dots > d(v_n)$. Ban đầu tất cả các đỉnh trong V' đều chưa được tô màu. Gán $i := 1$;
 - **Bước 2:** Tô màu i cho đỉnh đầu tiên trong danh sách V' . Duyệt lần lượt các đỉnh khác trong V' (nếu có) và chỉ tô màu i cho các đỉnh không kề đỉnh đã có màu i .
 - **Bước 3:** Kiểm tra nếu tất cả các đỉnh trong V đã được tô màu thì thuật toán kết thúc, đồ thị đã sử dụng i màu để tô. Ngược lại, nếu vẫn còn đỉnh chưa được tô thì chuyển sang bước 4.
 - **Bước 4:** Loại khỏi danh sách V' các đỉnh đã tô màu. Sắp xếp lại các đỉnh trong V' theo thứ tự bậc giảm dần. Gán $i := i + 1$ và quay lại bước 2.

Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
 - Cho n công việc, công việc i hoàn thành trong thời gian t_i , các công việc được thực hiện trên M máy công suất như nhau, mỗi máy đều có thể thực hiện bất kỳ trong n công việc, mỗi công việc làm liên tục trên 1 máy cho đến khi hoàn thành. Hãy tổ chức máy thực hiện đủ n công việc sao cho thời gian thực hiện càng nhỏ càng tốt.
 - Xét 6 công việc và thời gian hoàn thành các công việc tương ứng:

CV1	CV2	CV3	CV4	CV5	CV6
2	5	8	1	5	1

Chương 3. HEURISTIC

- Heuristic Greedy (Giải thuật tham lam)
 - Sắp xếp các công việc giảm dần theo thời gian hoàn thành
 - Phân công M công việc đầu tiên cho M máy. Thời gian lớn nhất để hoàn thành của M công việc này: Tmax
 - Loop1:
 - Loop2:
 - Chọn máy có thời gian đã làm $< Tmax$, thêm công việc mới cho máy này (theo thứ tự công việc đã sắp xếp).
 - Lặp lại cho đến khi thêm công việc mới thì không có máy nào có tổng thời gian thực hiện $< Tmax$
 - Tìm máy có thời gian thực hiện bé nhất, phân công công việc cho máy này và thay đổi giá trị của Tmax
 - Lặp lại cho đến khi hết việc

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

CV1	CV2	CV3	CV4	CV5	CV6	Số máy
2	5	8	1	5	1	M=3

▪ Sắp xếp công việc giảm dần theo thời gian hoàn thành

CV3	CV2	CV5	CV1	CV4	CV6
8	5	5	2	1	1

- Phân công lần 1:
 - Tmax = 8
 - Min_t = M2 (5)
- M1: CV3 (8)
- M2: CV2 (5)
- M3: CV5 (5)
- Tổng thời gian hoàn thành: 8 đơn vị thời gian

- Phân công lần 2:
 - M2 = M2 + CV1 (2) + CV4 (1) = 8 >= Tmax
 - M3 = M3 + CV6 (1) = 6

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

CV1	CV2	CV3	CV4	CV5	Số máy
3	3	2	2	2	M=2

▪ Sắp xếp công việc giảm dần theo thời gian hoàn thành

CV1	CV2	CV3	CV4	CV5
3	3	2	2	2

- Phân công lần 1:
 - M1: CV1 (3)
 - M2: CV2 (3)
- Tmax = 3
- Min_t = M1 (3)

▪ Phân công lần 2:

$$\text{▪ } M_1 = M_1 + CV_2 (2) = 5$$

$$>=T_{\max} \Rightarrow T_{\max} = 5$$

$$\text{▪ } M_2 = M_2 + CV_4 (2) = 5$$

▪ Tổng thời gian hoàn thành: 7 đơn vị thời gian

- Tmax = 5
- Min_t = M1 (5)

- Phân công lần 3:
 - M1 = M1 + CV3 (2) = 7
- Tmax = 7
- Min_t = M1 (7)
- M2 = M2 + Ø

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

CV1	CV2	CV3	CV4	CV5	CV6	CV7	Số máy
1	8	1	5	4	3	4	M=3

▪ Sắp xếp công việc giảm dần theo thời gian hoàn thành

CV2	CV4	CV5	CV7	CV6	CV1	CV3
8	5	4	4	3	1	1

- Phân công lần 1:
 - M1: CV2(8)
 - M2: CV4 (5)
 - M3: CV5 (4)

- $T_{max} = 8$
- $Min_t = M3 (4)$

- Phân công lần 2:
 - $M3 = M3 + CV7 (4) = 8$
 $>= T_{max} \Rightarrow T_{max} = 8$
 - $M2 = M2 + CV4 (4) = 8$

Chương 3. HEURISTIC

▪ Heuristic Greedy (Giải thuật tham lam)

CV1	CV2	CV3	CV4	CV5	CV6	CV7	Số máy
1	8	1	5	4	3	4	M=3

▪ Sắp xếp công việc giảm dần theo thời gian hoàn thành

CV2	CV4	CV5	CV7	CV6	CV1	CV3
8	5	4	4	3	1	1

- Phân công lần 2:
 - $M_3 = M_3 + CV_7 (4) = 8 \geq T_{max} \Rightarrow T_{max} = 8$
 - $M_2 = M_2 + CV_4 (4) = 8$
 - Tổng thời gian hoàn thành: 9 đơn vị thời gian
- Phân công lần 3:
 - $M_1 = M_1 + CV_1 (1) = 9 \geq T_{max} \Rightarrow T_{max} = 9$
 - $M_2 = M_2 + CV_3 (1) = 9$

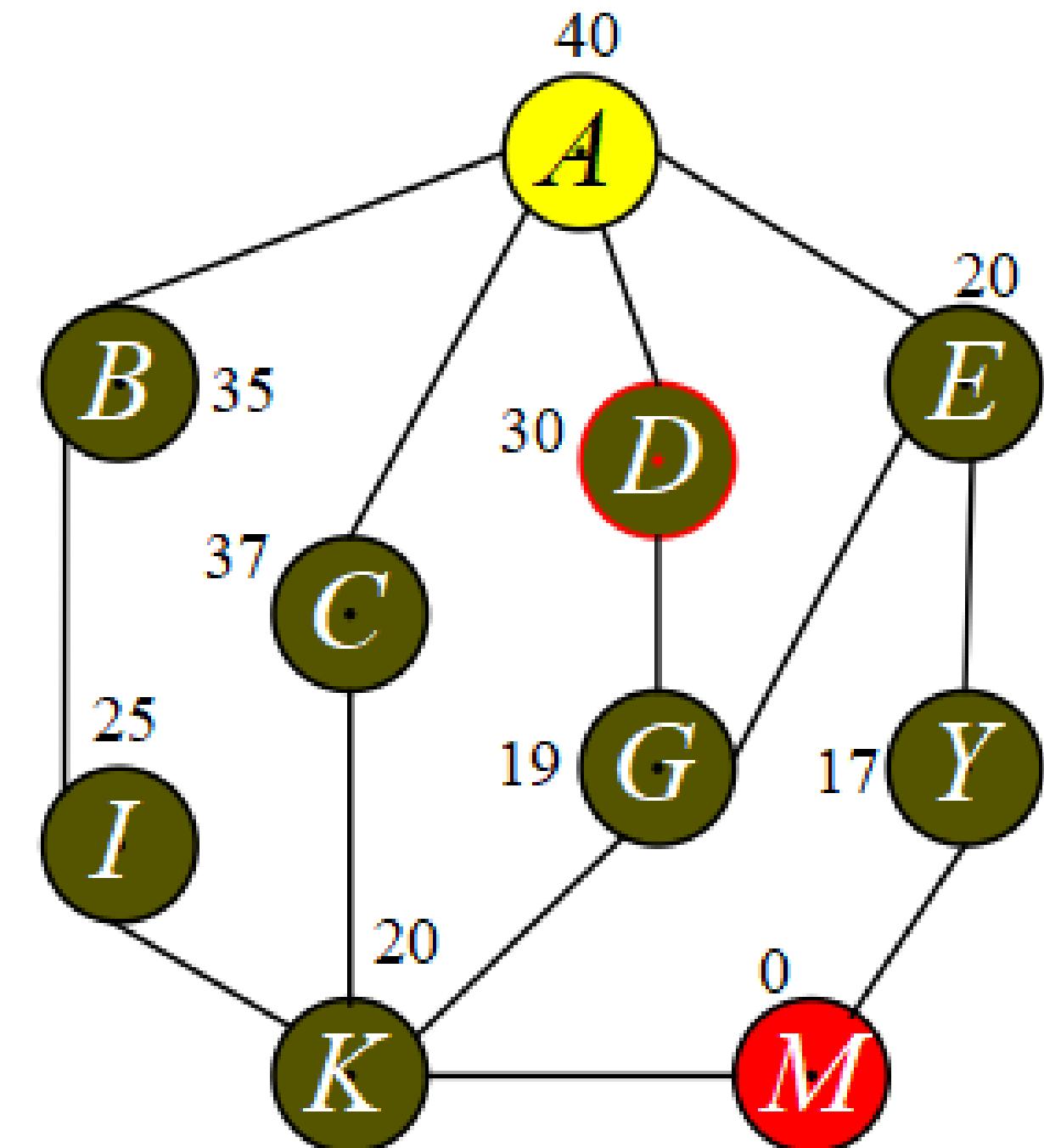
▪ $T_{max} = 8$
▪ $Min_t = M_1(8)$

Chương 3. HEURISTIC

- Tìm kiếm leo đồi (Hill Climbing Search)
- Là giải thuật tối ưu của tìm kiếm cục bộ (Stefanie Gunther, 1983) nhằm tìm kiếm trạng thái tốt hơn có sử dụng hàm lượng giá h' .
- Bước 1. Khởi tạo ngăn xếp OPEN = start
- Bước 2. Loop:
 - if ($\text{OPEN} = \emptyset$): thất bại
 - $n = \text{phần tử đầu danh sách OPEN}$
 - if ($n = \text{goal}$): thành công (END)
 - for ($m: \text{mỗi đỉnh kề } \notin \{\text{OPEN} \cup \text{CLOSE}\}$) $\rightarrow \Gamma(n) // n$ chưa xét đến
 - Sắp xếp danh sách $\Gamma(n)$ theo thứ tự tăng dần của hàm đánh giá h'
 - Chèn $\Gamma(n)$ vào đầu danh sách OPEN

Chương 3. HEURISTIC

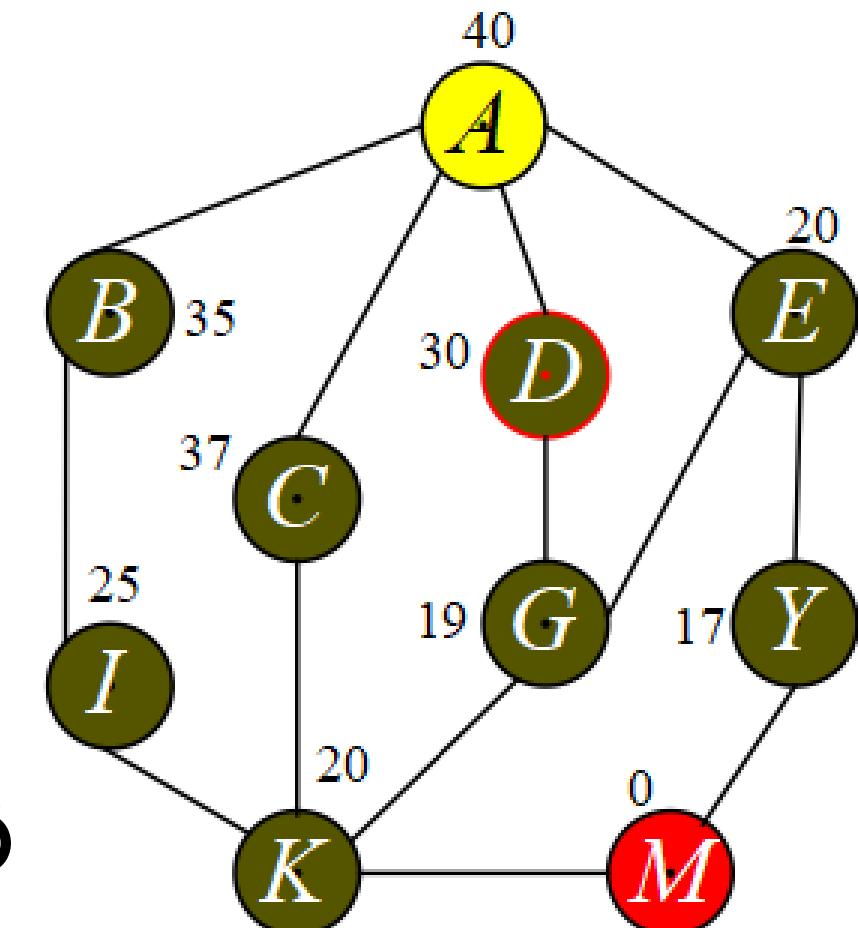
- Tìm kiếm leo đồi (Hill Climbing Search)
- Trình bày thuật toán leo đồi dốc đứng. Áp dụng thuật toán để tìm đường đi ngắn nhất từ đỉnh A đến M trên đồ thị, với các ước lượng heuristic của các trạng thái so với trạng thái đích được cho trong hình.



Chương 3. HEURISTIC

▪ Tìm kiếm leo đồi (Hill Climbing Search)

Bước	n	$\Gamma(n)$	OPEN	CLOSE
0			\emptyset	\emptyset
1	A	B(35), C(37), D(30), E(20)	E(20), D(30), B(35), C(37)	A
2	E	G(19), Y(17)	Y(17), G(19), D(30), B(35), C(37)	E
3	Y	M(0)	M(0), G(19), D(30), B(35), C(37)	Y
4	M	TRUE		

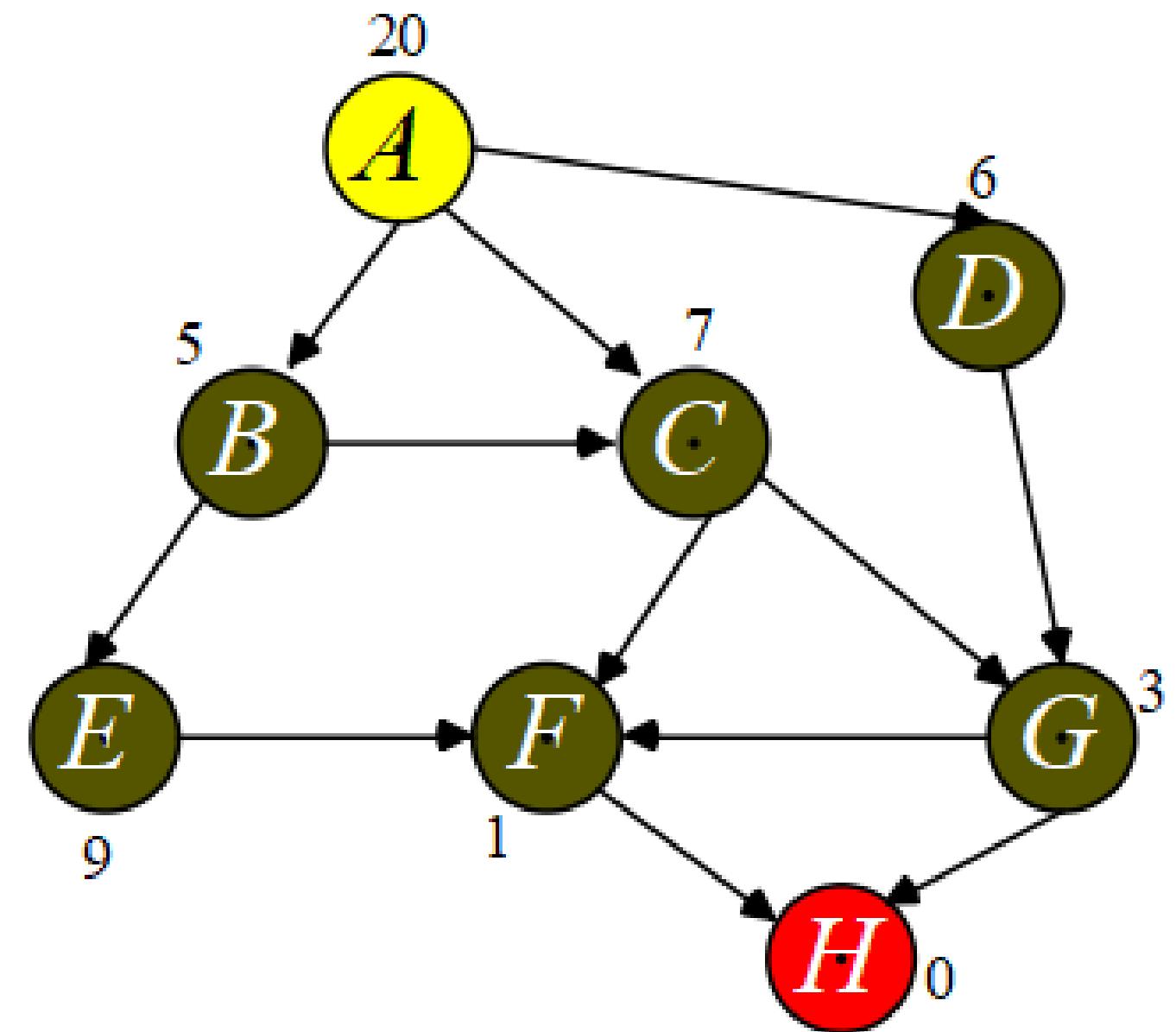


▪ A \Rightarrow E \Rightarrow Y \Rightarrow M

- Giải thuật có khuynh hướng sa lầy ở cực đại cục bộ
- Lời giải tìm được không tối ưu
- Không tìm được lời giải dù rằng có lời giải
- Có thể rơi vào vòng lặp vô hạn vì không lưu giữ trạng thái của các đỉnh đã xét.

Chương 3. HEURISTIC

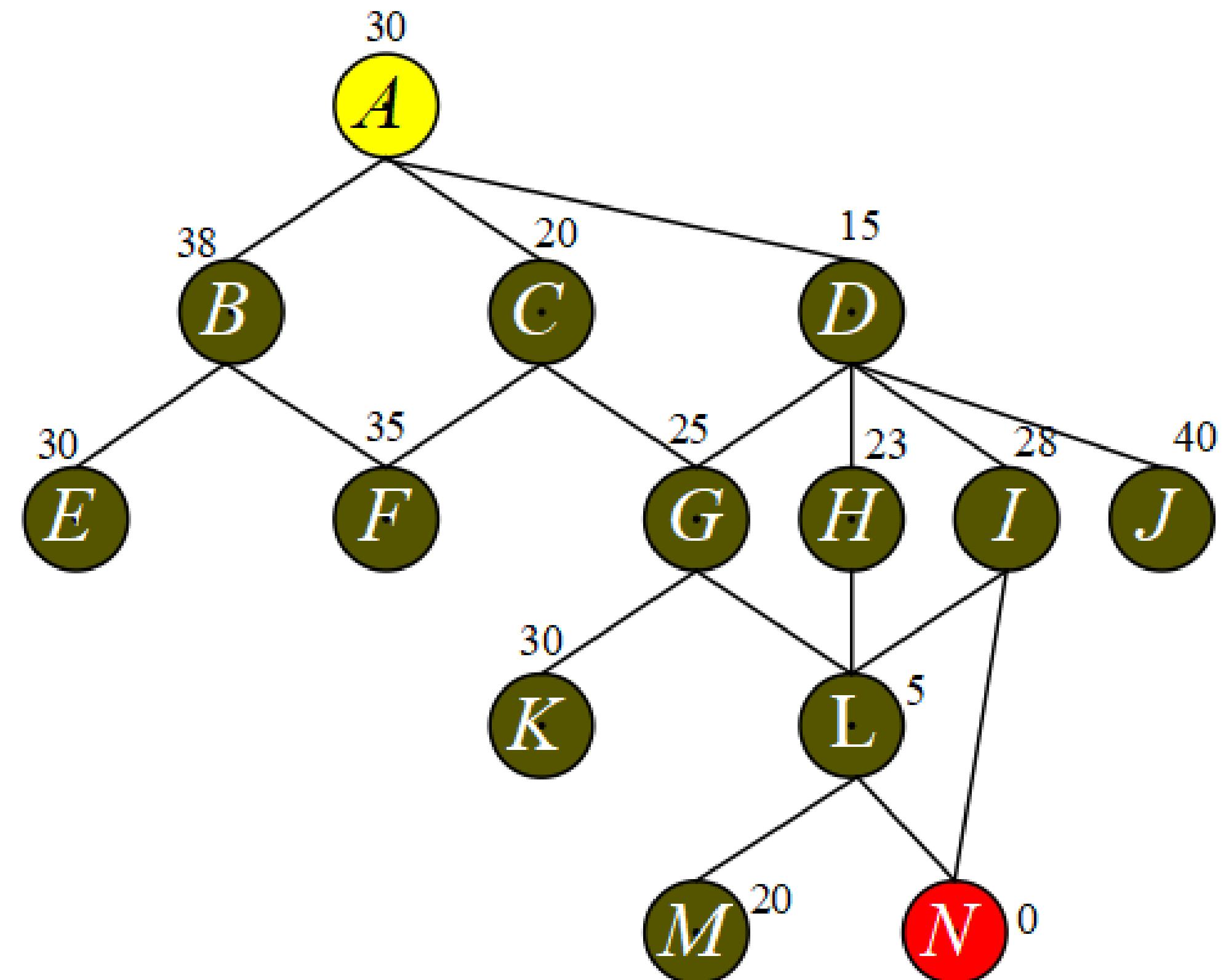
- Tìm kiếm leo đồi (Hill Climbing Search)
- Trình bày thuật toán leo đồi dốc đứng. Áp dụng thuật toán để tìm đường đi ngắn nhất từ đỉnh A đến M trên đồ thị, với các ước lượng heuristic của các trạng thái so với trạng thái đích được cho trong hình.



Chương 3. HEURISTIC

- Tìm kiếm leo đồi (Hill Climbing Search)

- Trình bày thuật toán leo đồi dốc đúng. Áp dụng thuật toán để tìm đường đi ngắn nhất từ đỉnh A đến N trên đồ thị, với các ước lượng heuristic của các trạng thái so với trạng thái đích được cho trong hình.



HẾT CHƯƠNG 3

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC



TRÍ TUỆ NHÂN TẠO *Artificial Intelligence*

Đoàn Vũ Thịnh
Khoa Công nghệ Thông tin
Đại học Nha Trang
Email: thinhdv@ntu.edu.vn

Nha Trang, 06-2023

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Các giải thuật tìm kiếm không sử dụng thông tin phản hồi (hay là giải thuật tìm kiếm mù): duyệt theo chiều rộng (BFS) và duyệt theo chiều sâu (DFS) có đặc điểm:
- Xây dựng tất cả không gian lời giải tiềm năng theo cách vét cạn, không bỏ sót và không lặp lại.
- Trong rất nhiều trường hợp, các giải thuật như vậy không khả thi vì không gian trạng thái bài toán quá lớn, tốc độ xử lý và bộ nhớ của máy tính không cho phép duyệt các lời giải tiềm năng.

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Giải thuật tìm kiếm tốt nhất đầu tiên (Best First Search)
- OPEN: tập chứa các trạng thái được sinh ra nhưng chưa được xét đến (vì ta đã chọn trạng thái khác). Thực ra OPEN là một hàng đợi ưu tiên mà trong đó, phần tử có độ ưu tiên cao nhất là phần tử tốt nhất.
- CLOSE: tập chứa các trạng thái đã được xét đến. Chúng ta cần lưu trữ những trạng thái này trong bộ nhớ để đề phòng trường hợp khi một trạng thái mới được tạo ra trùng với trạng thái mà ta đã xét trước đó.

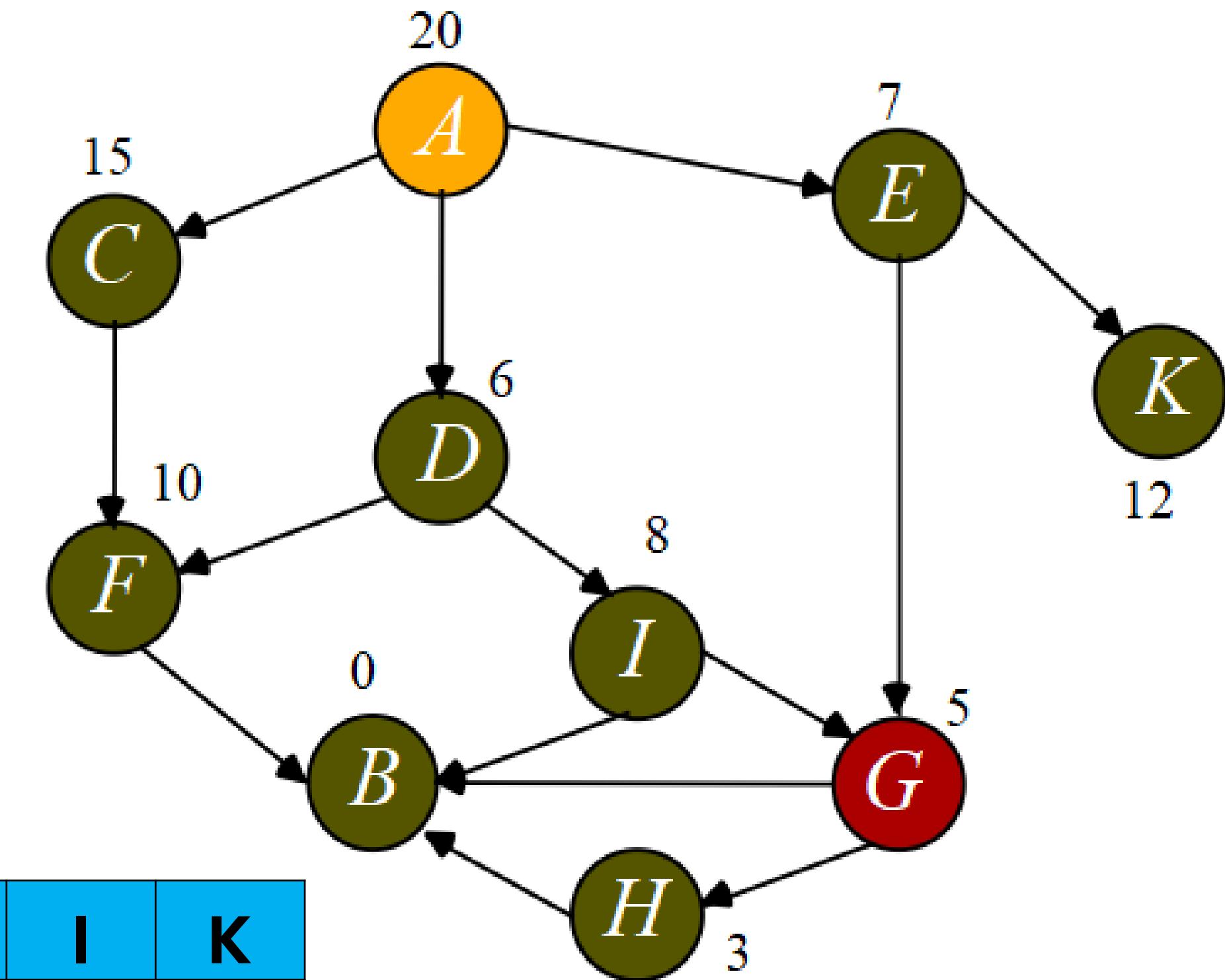
Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Giải thuật tìm kiếm tốt nhất đầu tiên (Best First Search)
 - 1. Đặt OPEN chứa trạng thái đầu
 - 2. Lặp cho đến khi tìm được trạng thái đích hoặc không còn nút nào trong OPEN, thực hiện:
 - 2.1. Chọn trạng thái tốt nhất (T_{max}) trong OPEN (Xóa T_{max} ra khỏi danh sách OPEN)
 - 2.2. Nếu T_{max} là trạng thái đích thì Kết thúc
 - 2.3. Ngược lại, tạo ra các trạng thái kế tiếp T_k có thể có từ trạng thái T_{max} . Đối với mỗi T_k thực hiện:
 - Tính $f(T_k)$; Thêm T_k vào OPEN

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Giải thuật tìm kiếm tốt nhất đầu tiên (Best First Search)
- Trình bày thuật toán Best-First Search (BFS). Áp dụng BFS để tìm đường đi ngắn nhất từ đỉnh A đến G trên đồ thị với các ước lượng heuristic của các trạng thái so với trạng thái đích được liệt kê:

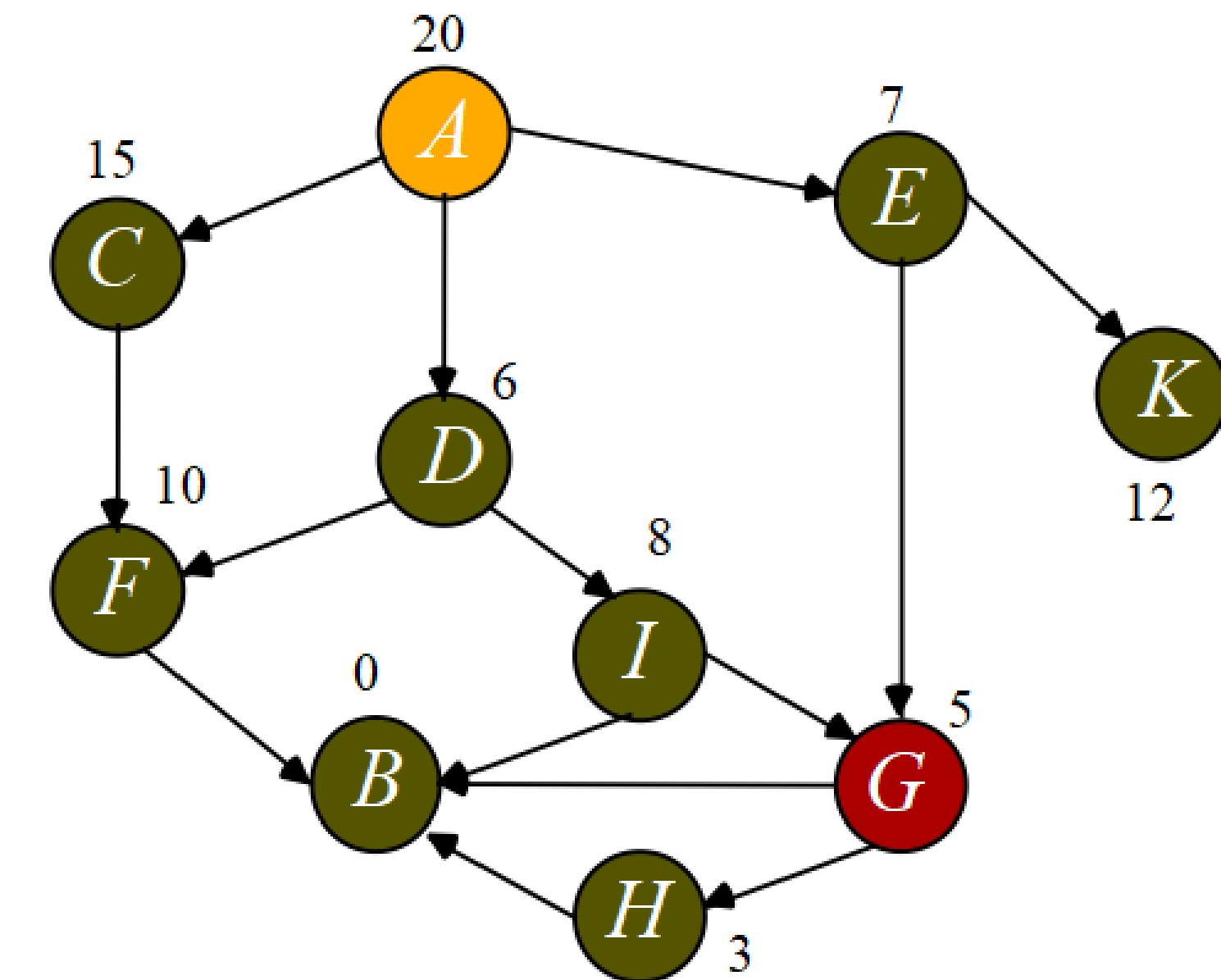
A	B	C	D	E	F	G	H	I	K
20	0	15	6	7	10	5	3	8	12



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Giải thuật tìm kiếm tốt nhất đầu tiên (Best First Search)

Bước	n	$\Gamma(n)$	Open	Close	$f_{SORT\uparrow}$
0			{A}	\emptyset	
1	A	{C,D,E}	{D,E,C}	{A}	$f(C, D, E)$
2	D	{F,I}	{E,I,F,C}	{A,D}	$f(C, E, F, I)$
3	E	{K,G}	{G,I,F,K,C}	{A,D,E}	$f(C, F, I, K, G)$
4	G	TRUE			

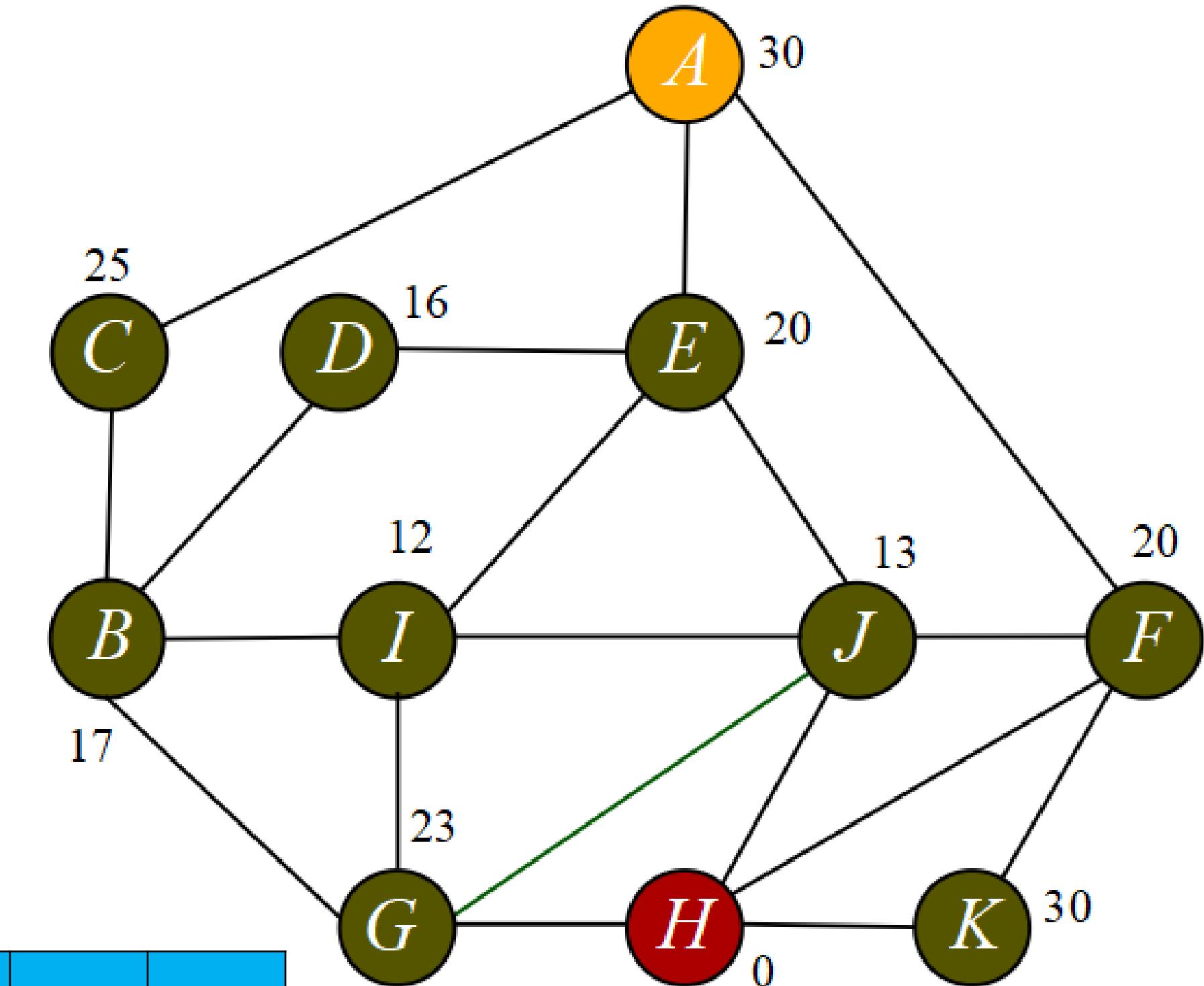


- $A \Rightarrow E \Rightarrow G$

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Giải thuật tìm kiếm tốt nhất đầu tiên (Best First Search)
- Trình bày thuật toán Best-First Search (BFS). Áp dụng BFS để tìm đường đi ngắn nhất từ đỉnh **A** đến **H** trên đồ thị với các ước lượng heuristic của các trạng thái so với trạng thái đích được liệt kê:

A	B	C	D	E	F	G	H	I	J	K
30	17	25	16	20	20	23	0	12	13	30



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Giải thuật AT (Algorithm for Tree)
 - Thuật giải BFS khá đơn giản. Tuy vậy, trên thực tế cũng như tìm kiếm theo chiều rộng và chiều sâu, hiếm khi ta dùng BFS một cách trực tiếp. Thông thường, người ta thường dùng các phiên bản của BFS là AT, AKT và A*
 - Thông thường, trong các phương án tìm kiếm BFS. Độ tốt f của một trạng thái được tính dựa vào 2 giá trị **g** và **h'**.
 - h' : ước lượng chi phí từ trạng thái hiện hành – trạng thái đích
 - g : chiều dài đoạn đường đã đi từ trạng thái ban đầu đến trạng thái hiện tại (g là chi phí thực sự chứ không phải ước lượng)

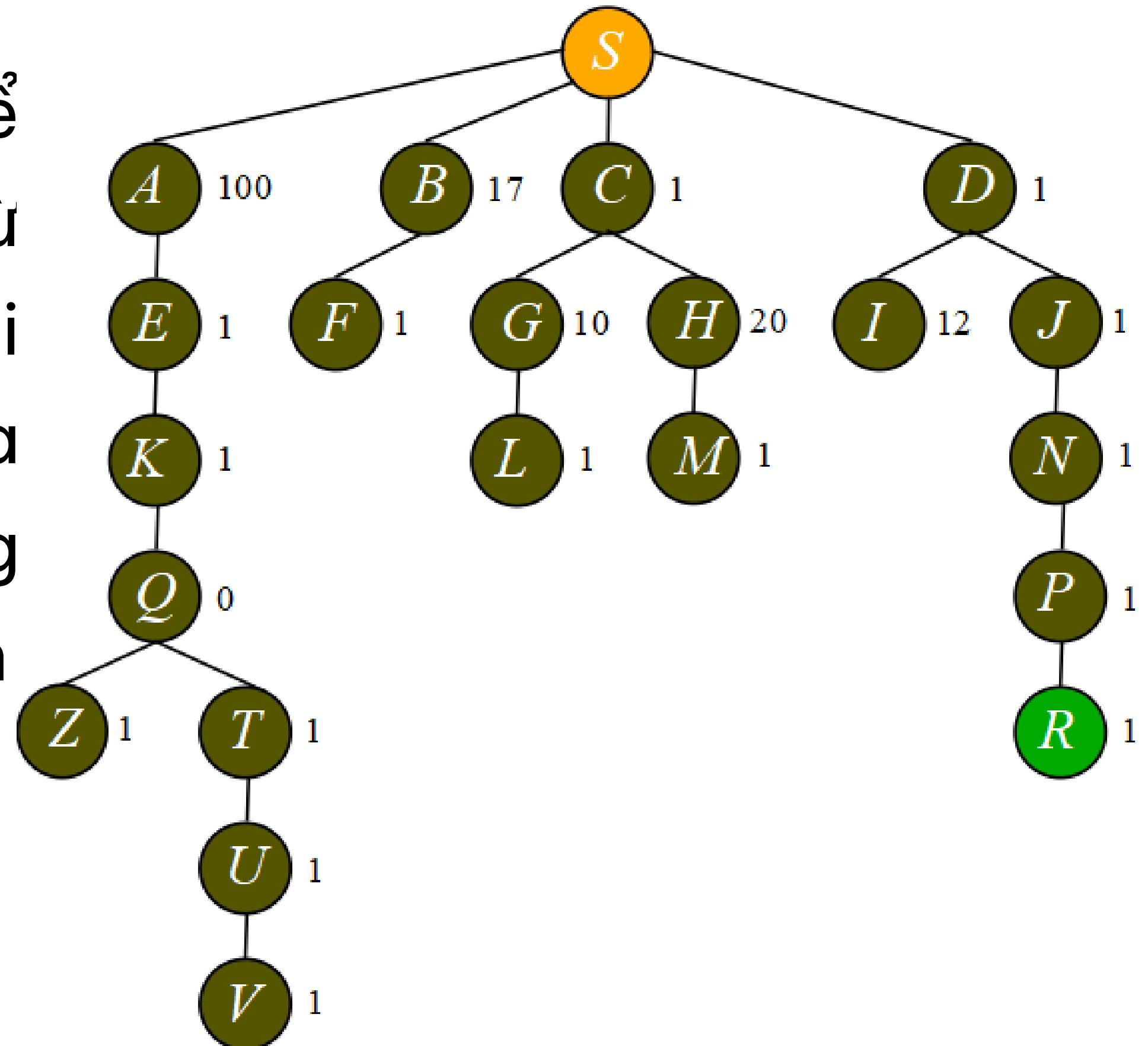
Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Giải thuật AT (Algorithm for Tree)
 - Bước 1: Chọn $s = \text{đỉnh xuất phát}; g(s)=0$
 - Bước 2: Chọn 1 đỉnh từ tập OPEN: $=n$, có $g(n) = \min$
 - 2.1. Nếu $n \equiv \text{đích}$: đường đi từ $s \Rightarrow n$ là tối ưu
 - 2.2. Nếu $\text{OPEN} = \emptyset$: không tìm thấy đường đi
 - 2.3. Nếu $\exists!$ nhiều hơn 1 đỉnh n có $g(n) = \min$:
 - Kiểm tra ($\exists! \equiv \text{đích}$): dừng
 - Ngược lại, chọn ngẫu nhiên một đỉnh: $=n$
 - Bước 3: Đưa đỉnh đã duyệt vào CLOSE, tạo $\Gamma(n)$: các đỉnh mà n trỏ tới
 - for mỗi nút con m của $\Gamma(n)$:
 - *if* ($m \notin \text{OPEN}$):
 - $g(m) := g(n) + \text{cost}(n, m); \text{OPEN} := \text{OPEN} \cup \{m\}$
 - So sánh $g(m)$ với $g_{\text{NEW}}(m)$ và cập nhật
 - Bước 4: Quay lại bước 2

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Giải thuật AT (Algorithm for Tree)

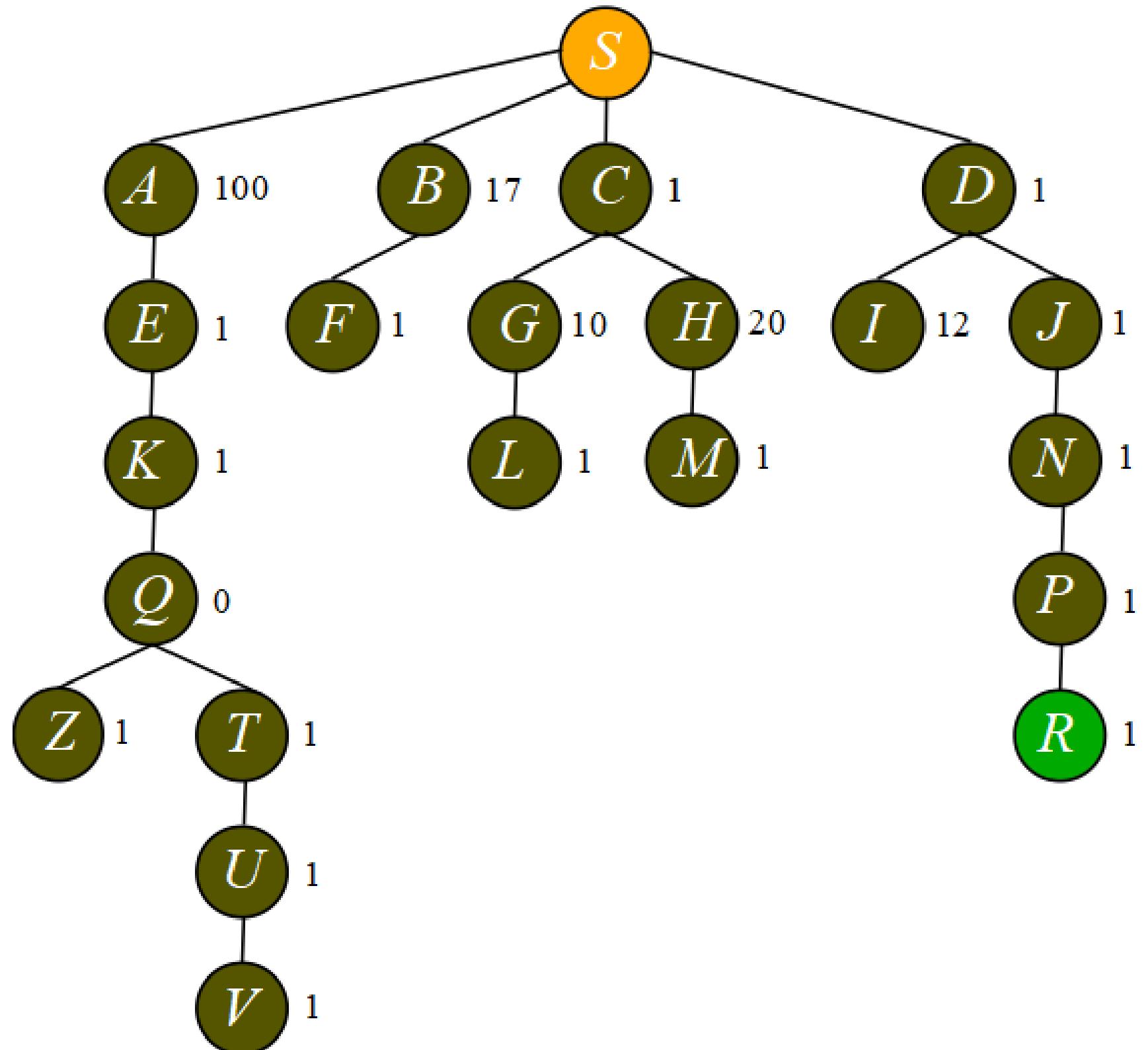
- Trình bày thuật toán AT để tìm đường đi ngắn nhất từ đỉnh **S** đến **R** trên đồ thị với các ước lượng heuristic của các trạng thái so với trạng thái đích được liệt kê như hình



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

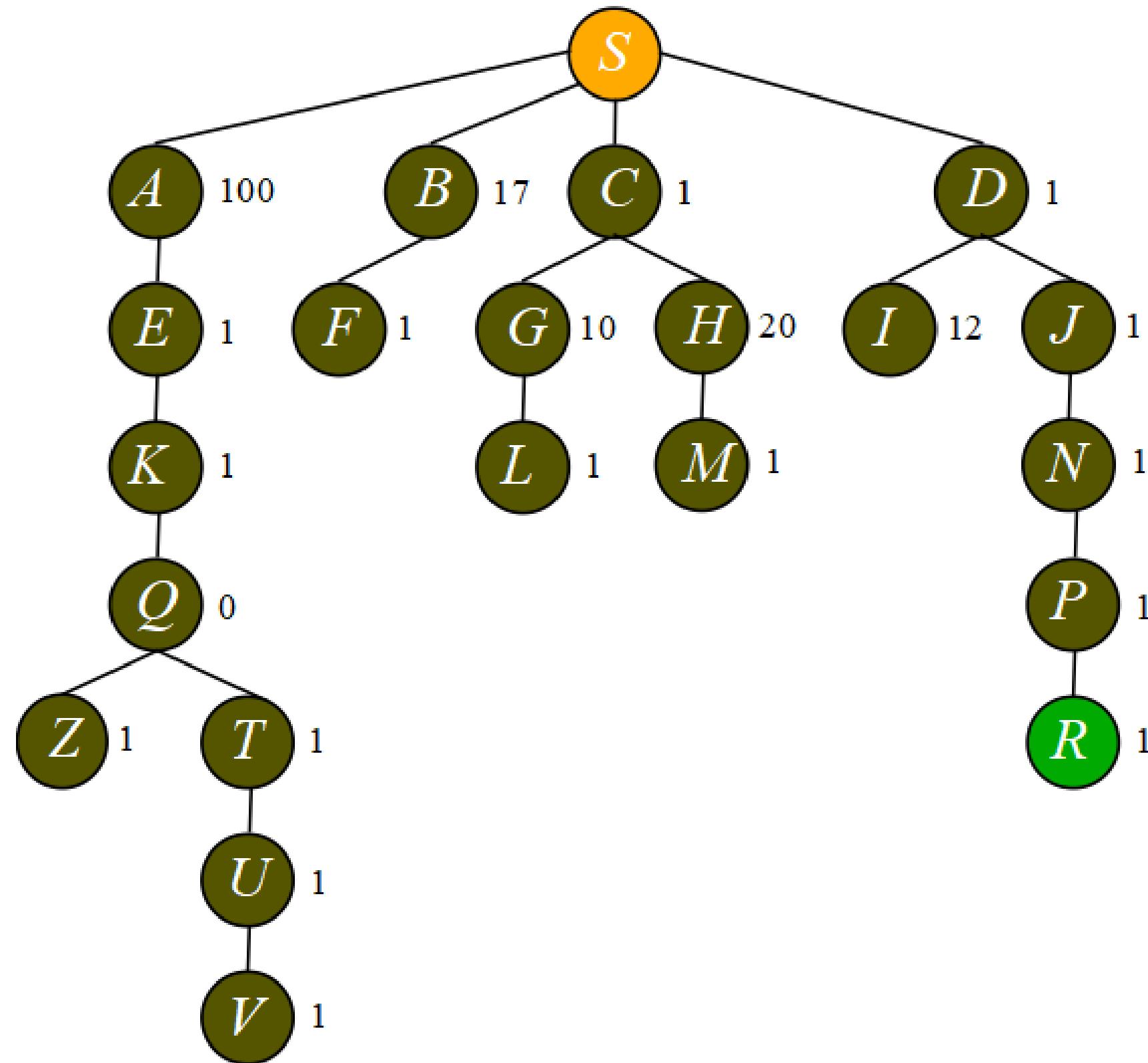
▪ Giải thuật AT (Algorithm for Tree)

- Bước 1: OPEN(S); $g(S)=0$
- Bước 2: $n=S$, $\Gamma(n)=\{A,B,C,D\}$, CLOSE={S}
 - $g(A \notin \text{OPEN})=g(S) + g(S \rightarrow A) = 0+100 = 100$
 - $g(B \notin \text{OPEN})=g(S) + g(S \rightarrow B) = 0+17 = 17$
 - $g(C \notin \text{OPEN})=g(S) + g(S \rightarrow C) = 0+1 = 1$
 - $g(D \notin \text{OPEN})=g(S) + g(S \rightarrow D) = 0+1 = 1$
 - $g(\min) = g(C)$
 - $\text{OPEN}=\{C,D,B,A\}$



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Giải thuật AT (Algorithm for Tree)



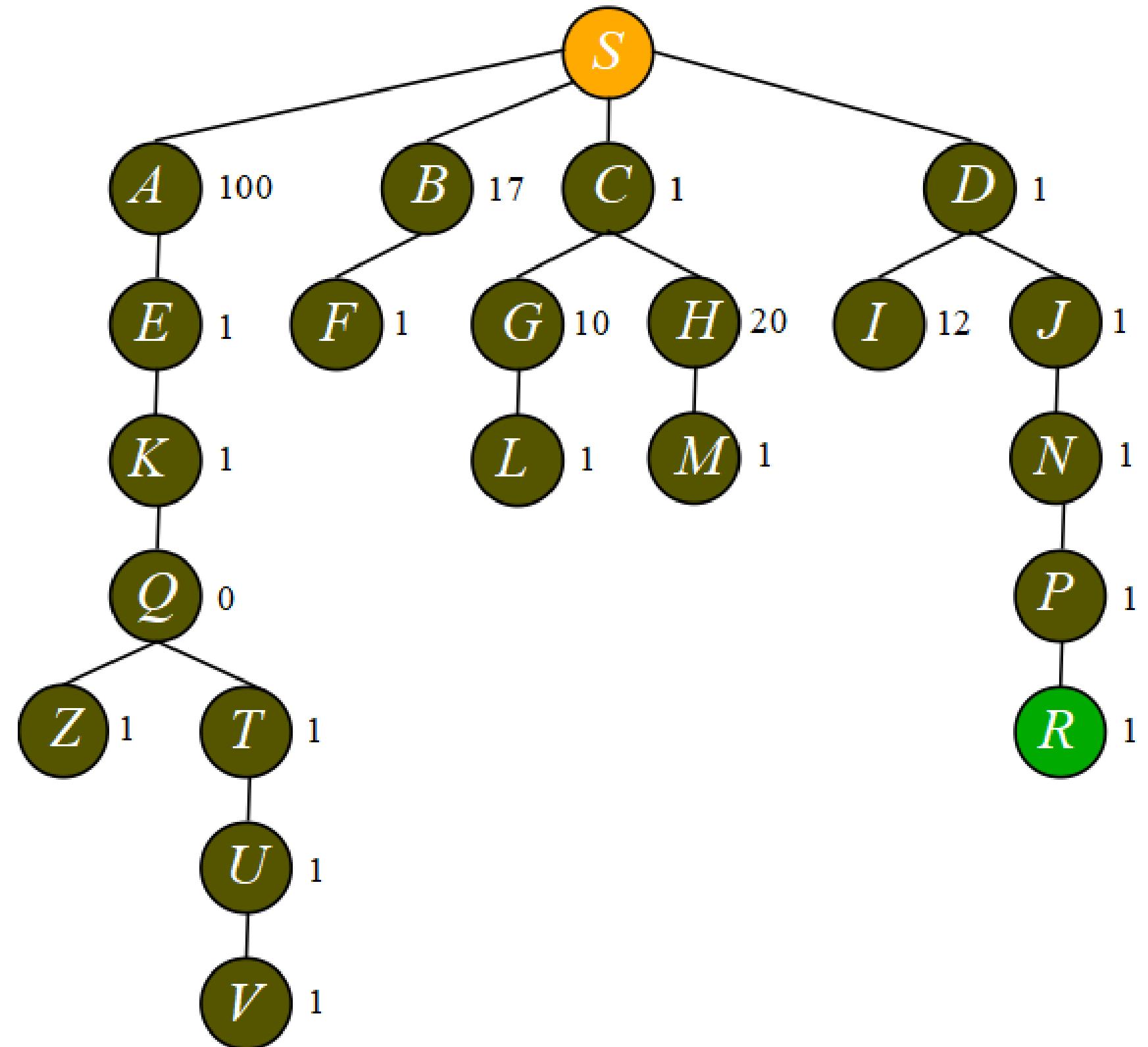
- Bước 3: $n=C$, $\Gamma(n)=\{G,H\}$, CLOSE= $\{S,C\}$
 - $g(A) = 100$
 - $g(B) = 17$
 - $g(D) = 1$
 - $g(G \notin \text{OPEN}) = g(C) + g(C \rightarrow G)$
 $= 1+10 = 11$
 - $g(H \notin \text{OPEN}) = g(C) + g(C \rightarrow H)$
 $= 1+20 = 21$
 - $g(\min) = g(D)$
 - $\text{OPEN}=\{D,G,B,C,A\}$

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Giải thuật AT (Algorithm for Tree)

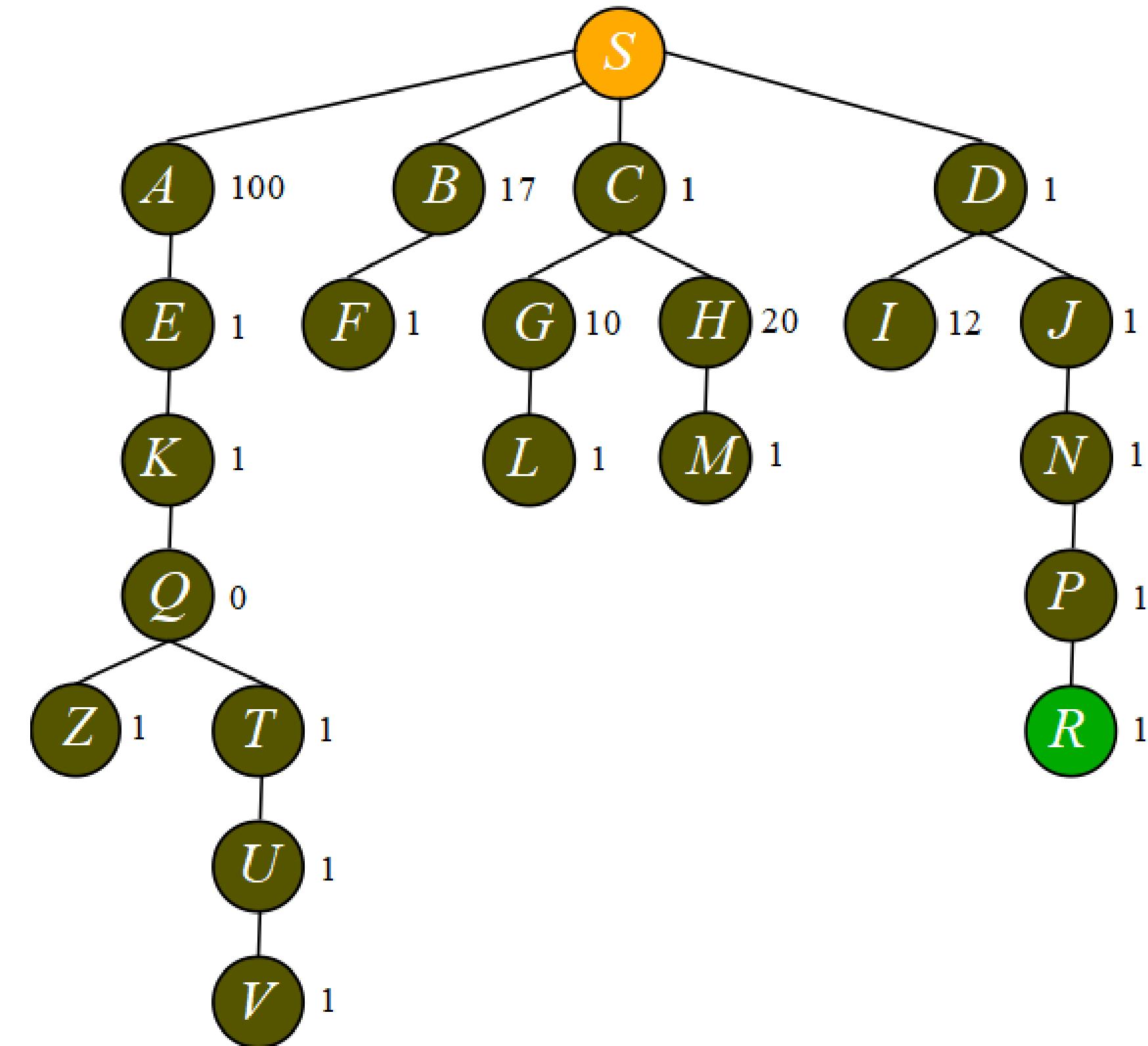
▪ Bước 4: $n=D$, $\Gamma(n)=\{I, J\}$, CLOSE= $\{S, C, D\}$

- $g(A)= 100$
- $g(B)= 17$
- $g(G)= 11$
- $g(H)= 21$
- $g(I \notin \text{OPEN})=g(D) + g(D \rightarrow I) = 1+ 12 = 13$
- $g(J \notin \text{OPEN})=g(D) + g(D \rightarrow J) = 1+ 1 = 2$
- $g(\min) = g(J)$
- $\text{OPEN}=\{J, G, I, B, H, A\}$



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Giải thuật AT (Algorithm for Tree)



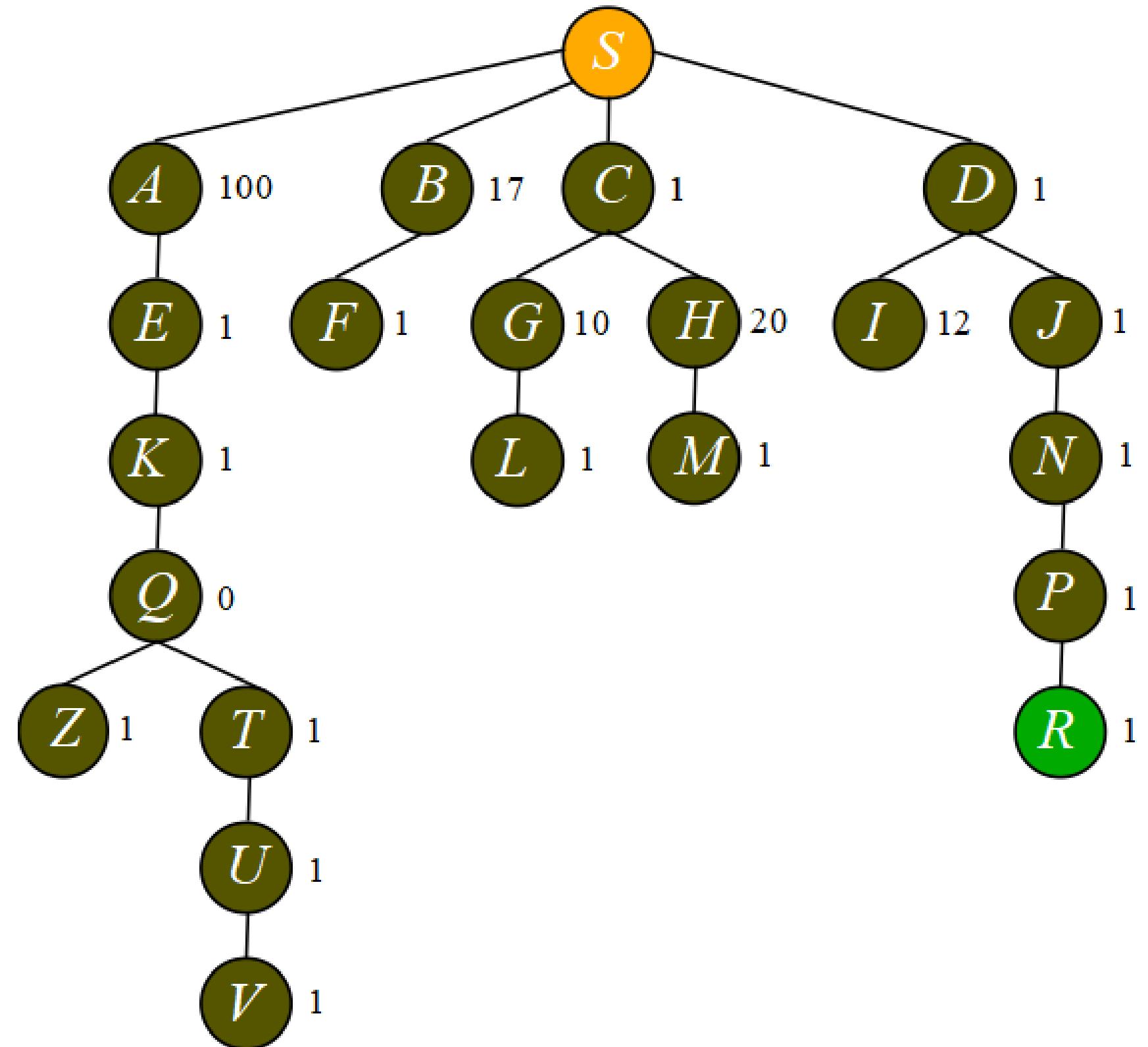
- Bước 5: $n=J$, $\Gamma(n)=\{N\}$, CLOSE= $\{S,C,D,J\}$
 - $g(A)=100$
 - $g(B)=17$
 - $g(G)=11$
 - $g(H)=21$
 - $g(I)=13$
 - $g(N \notin \text{OPEN})=g(J) + g(J \rightarrow N) = 2+1 = 3$
 - $g(\min) = g(N)$
 - $\text{OPEN}=\{N,G,I,B,H,A\}$

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Giải thuật AT (Algorithm for Tree)

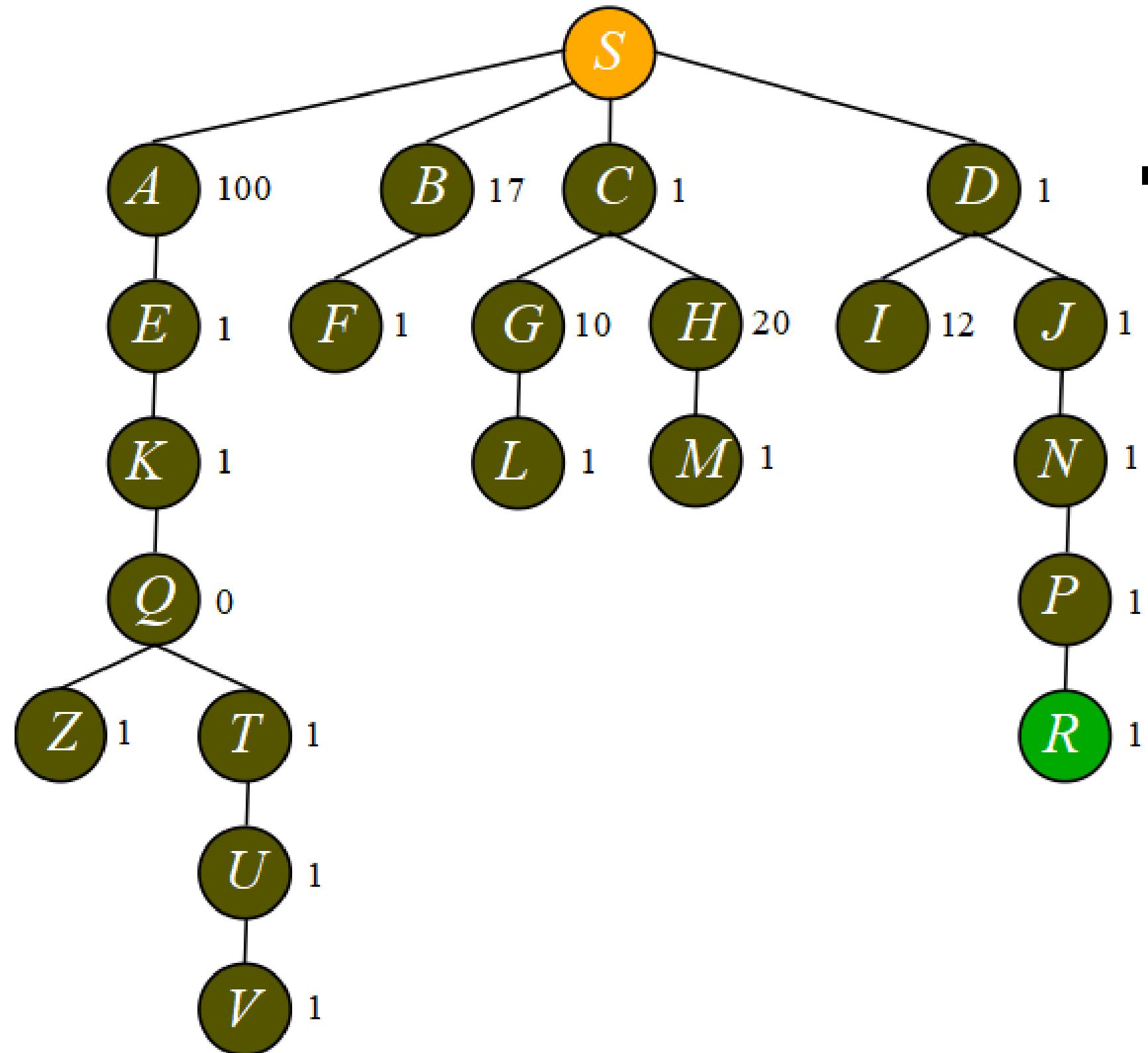
- Bước 6: $n=N$, $\Gamma(n)=\{P\}$, CLOSE= $\{S,C,D,J,N\}$

- $g(A)= 100$
- $g(B)= 17$
- $g(G)= 11$
- $g(H)= 21$
- $g(I)= 13$
- $g(P \notin OPEN) = g(N) + g(N \rightarrow P) = 3 + 1 = 4$
- $g(\min) = g(P)$
- $OPEN=\{P,G,I,B,H,A\}$



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Giải thuật AT (Algorithm for Tree)



- Bước 7: $n=P$, $\Gamma(n)=\{R\}$, $CLOSE=\{S,C,D,J,N,P\}$
 - $g(A)= 100$
 - $g(B)= 17$
 - $g(G)= 11$
 - $g(H)= 21$
 - $g(I)= 13$
 - $g(R \notin OPEN)=g(P) + g(P \rightarrow R) = 4+1 = 5$
 - $g(\min) = g(R)$
 - $OPEN=\{R,G,I,B,H,A\}$
- Bước 8: $n=R$ **TRUE**

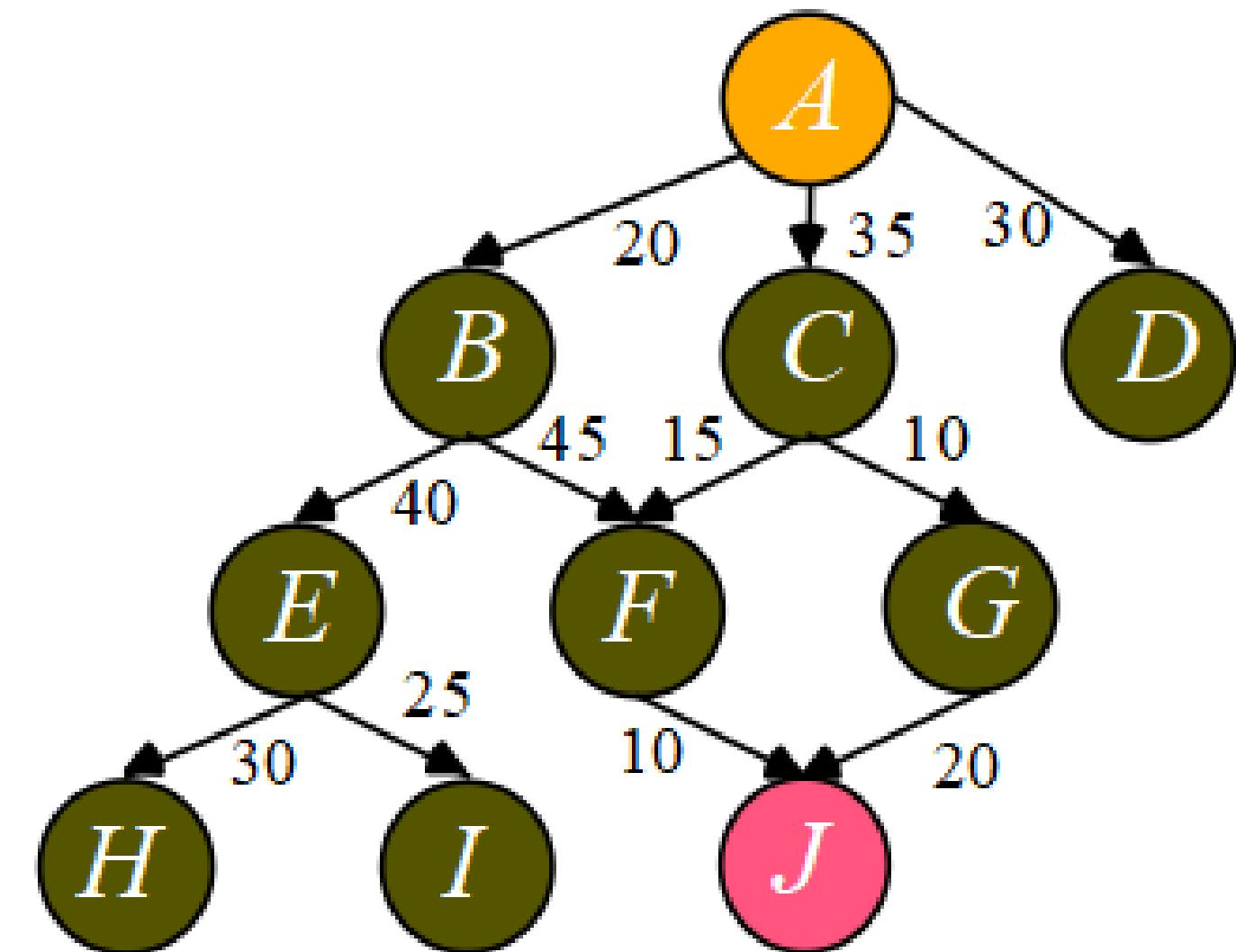
Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Giải thuật Cost Minimization Search (CMS)
 - Bước 1: Chọn $s = \text{đỉnh xuất phát}; g(s)=0$
 - Bước 2: Chọn 1 đỉnh từ tập OPEN: $=n$, có $g(n) = \min$
 - 2.1. Nếu $n \equiv \text{đích}$: đường đi từ $s \Rightarrow n$ là tối ưu
 - 2.2. Nếu $\text{OPEN} = \emptyset$: không tìm thấy đường đi
 - 2.3. Nếu $\exists!$ nhiều hơn 1 đỉnh n có $g(n) = \min$:
 - Kiểm tra ($\exists! \equiv \text{đích}$): dừng
 - Ngược lại, chọn ngẫu nhiên một đỉnh: $=n$
 - Bước 3: Đưa đỉnh đã duyệt vào CLOSE, tạo $\Gamma(n)$: các đỉnh mà n trỏ tới
 - for mỗi nút con m của $\Gamma(n)$:
 - $g(m) := g(n) + cost(n, m)$
 - $OPEN := OPEN \cup \{m\}$
 - So sánh $g(m)$ với $g_{\text{NEW}}(m)$ và cập nhật
 - Bước 4: Quay lại bước 2

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Giải thuật Cost Minimization Search (CMS)

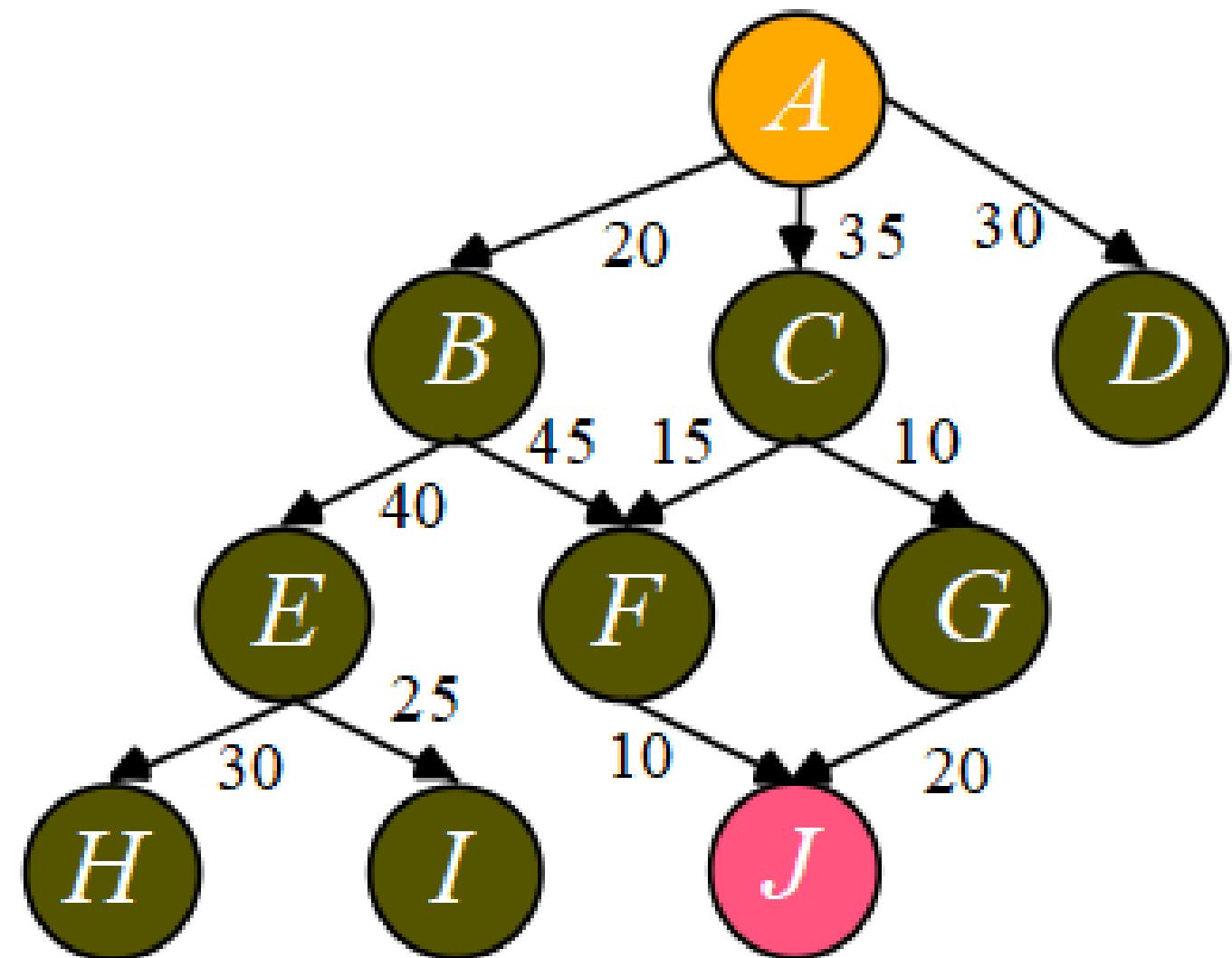
- Trình bày thuật toán CMS để tìm đường đi ngắn nhất từ đỉnh A đến J trên đồ thị với các ước lượng heuristic của các trạng thái so với trạng thái đích được liệt kê như hình



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Giải thuật Cost Minimization Search (CMS)

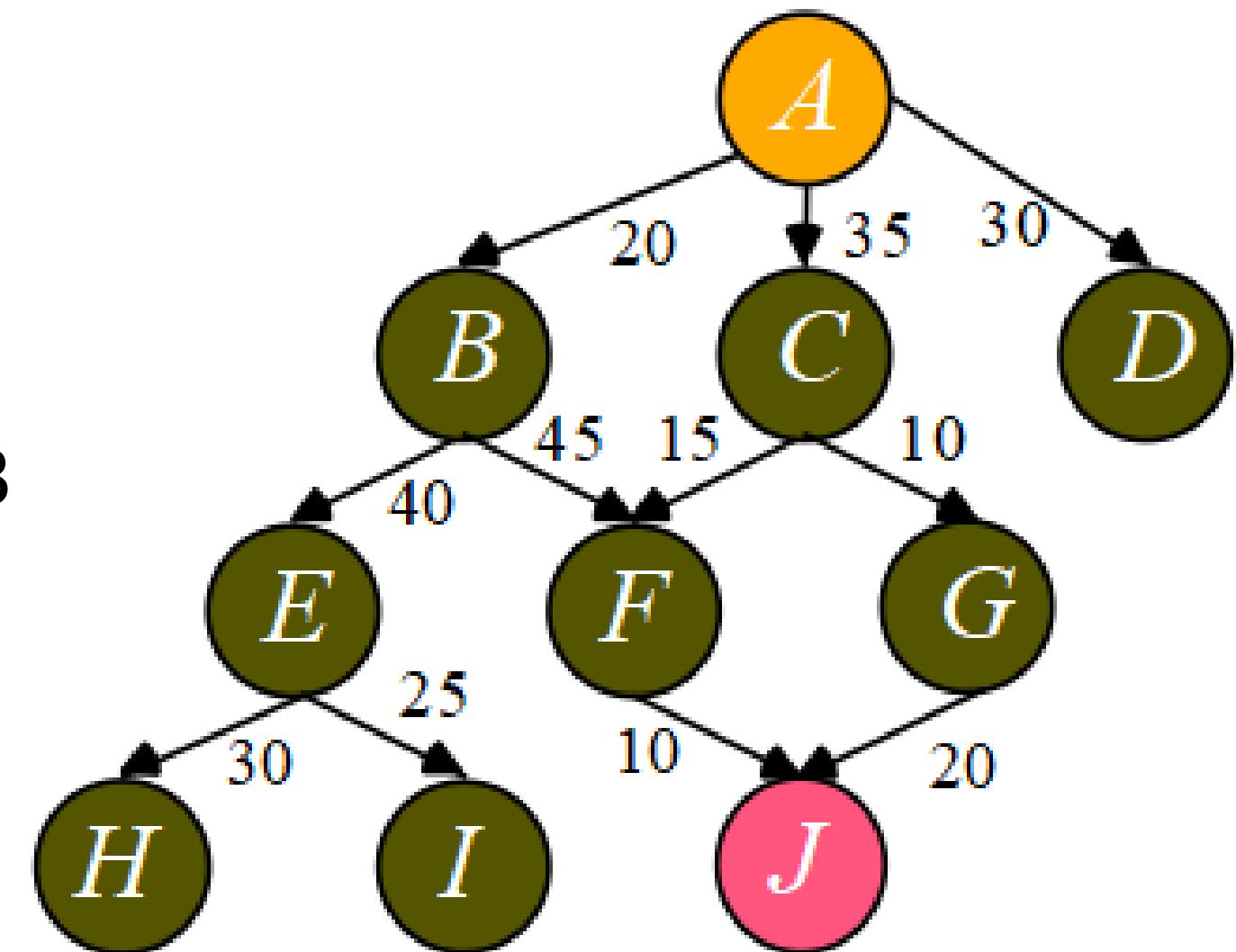
- Bước 1: OPEN(A); $g(A)=0$
- Bước 2: $n=A$, $\Gamma(n)=\{B,C,D\}$, CLOSE={A}
 - $g(B) = g(A) + g(A \rightarrow B) = 0+20 = 20$; Father(B) = A
 - $g(C) = g(A) + g(A \rightarrow C) = 0+35= 35$; Father(C) = A
 - $g(D) = g(A) + g(A \rightarrow D) = 0+30 = 30$; Father(D) = A
 - $g(\min) = g(B)$
 - OPEN={B,D,C}



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Giải thuật Cost Minimization Search (CMS)

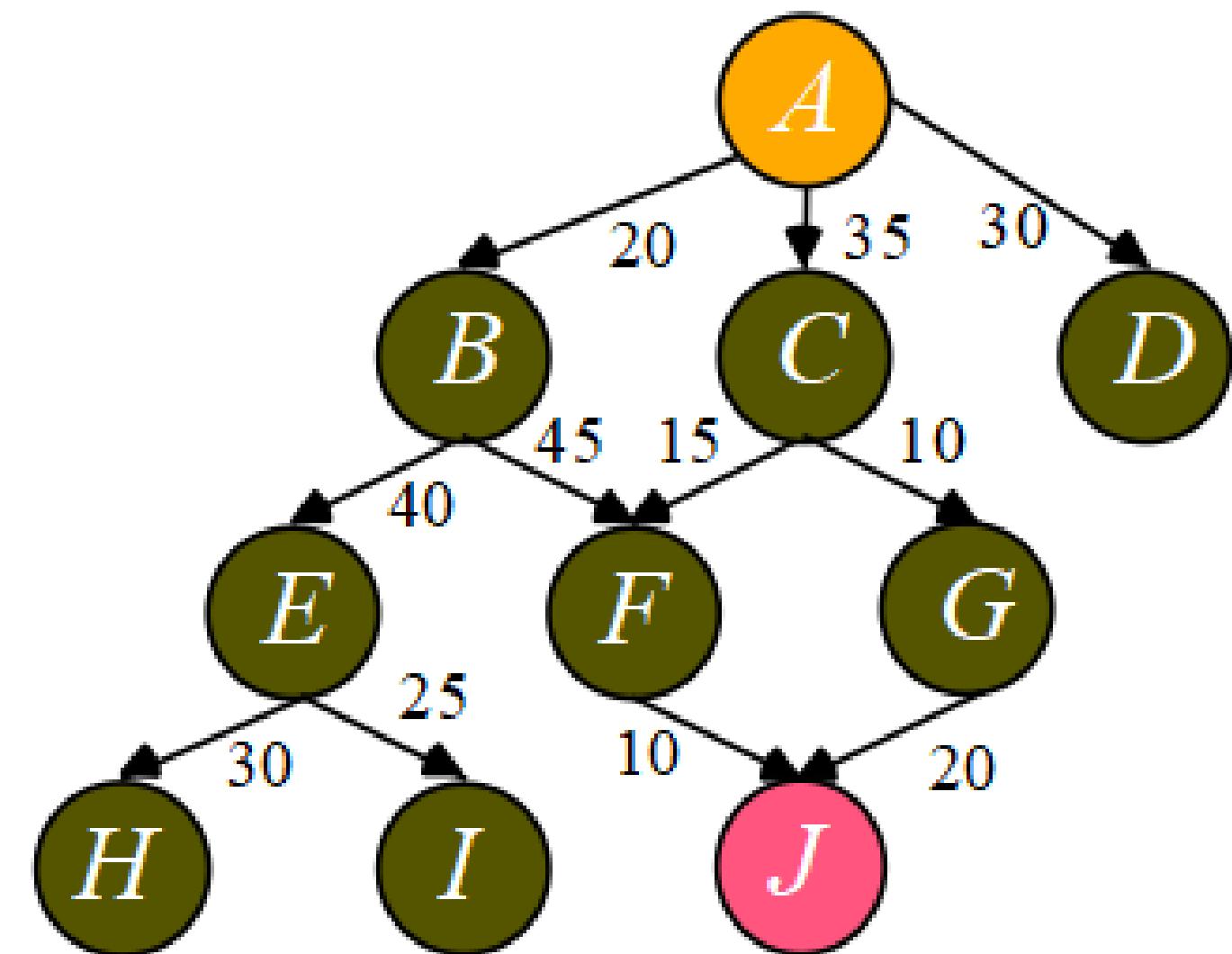
- Bước 3: $n=B$, $\Gamma(n)=\{E,F\}$, CLOSE= $\{A,B\}$
 - $g(C) = 35$
 - $g(D) = 30$
 - $g(E) = g(B) + g(B \rightarrow E) = 20 + 40 = 60$; Father(E) = B
 - $g(F) = g(B) + g(B \rightarrow F) = 20 + 45 = 65$; Father(F) = B
 - $g(\min) = g(D)$
 - OPEN= $\{D,C,E,F\}$



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Giải thuật Cost Minimization Search (CMS)

- Bước 4: $n=D$, $\Gamma(n)=\emptyset$, CLOSE={A,B,D}
 - $g(C) = 35$
 - $g(E) = 60$
 - $g(F) = 65$
 - OPEN={C,E,F}
- Bước 5: $n=C$, $\Gamma(n)=\{F,G\}$, CLOSE={A,B,D,C}
 - $g(E) = 60$
 - $g(F) = 65$
 - $g(F) = g(C) + g(C \rightarrow F) = 35 + 10 = 50$
 - $g(F) = \min(50, 65) = 50$; Father(F) = C
 - $g(G) = g(C) + g(C \rightarrow G) = 35 + 10 = 45$; Father(G) = C
 - $g(\min) = g(G)$
 - OPEN={G,F,E}



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Giải thuật Cost Minimization Search (CMS)

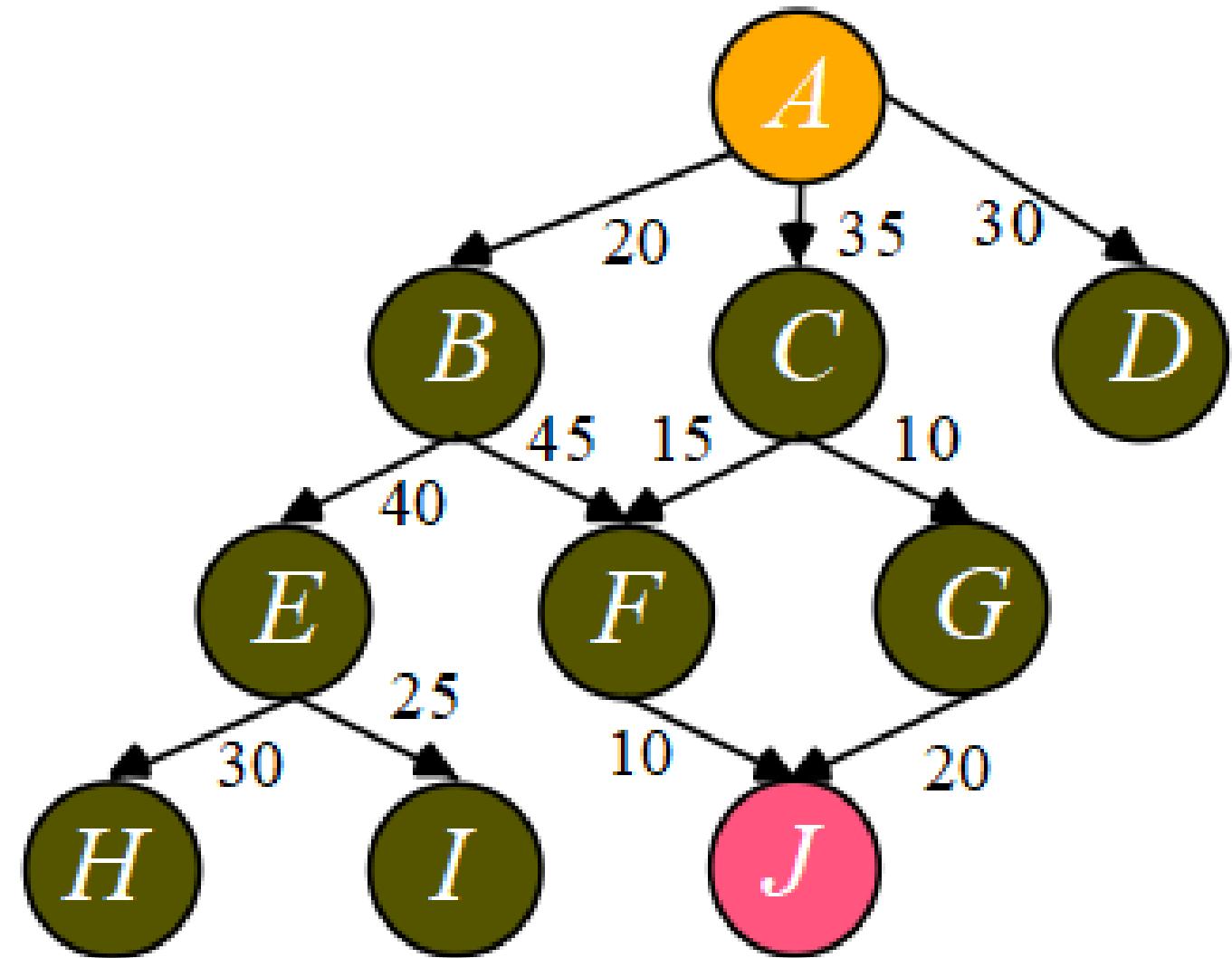
- Bước 6: $n=G$, $\Gamma(n)=\{J\}$, CLOSE={A,B,D,C,G}

- $g(E) = 60$
 - $g(F) = 65$
 - $g(J) = g(G) + g(G \rightarrow J) = 45 + 20 = 65$; Father(J) = G
 - OPEN={F,E,J}

- Bước 7: $n=F$, $\Gamma(n)=\{J\}$, CLOSE={A,B,D,C,G,F}

- $g(E) = 60$
 - $g(J) = 65$
 - $g(J) = g(G) + g(G \rightarrow J) = 45 + 20 = 65$
 - $g(J) = \min(65, 65) = 65$; Father(J) = F
 - OPEN={E,J}

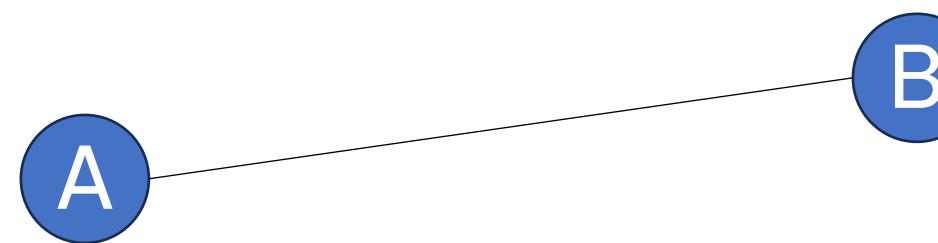
- Bước 8: $n=J$, TRUE



▪ $A \Rightarrow C \Rightarrow F \Rightarrow J$

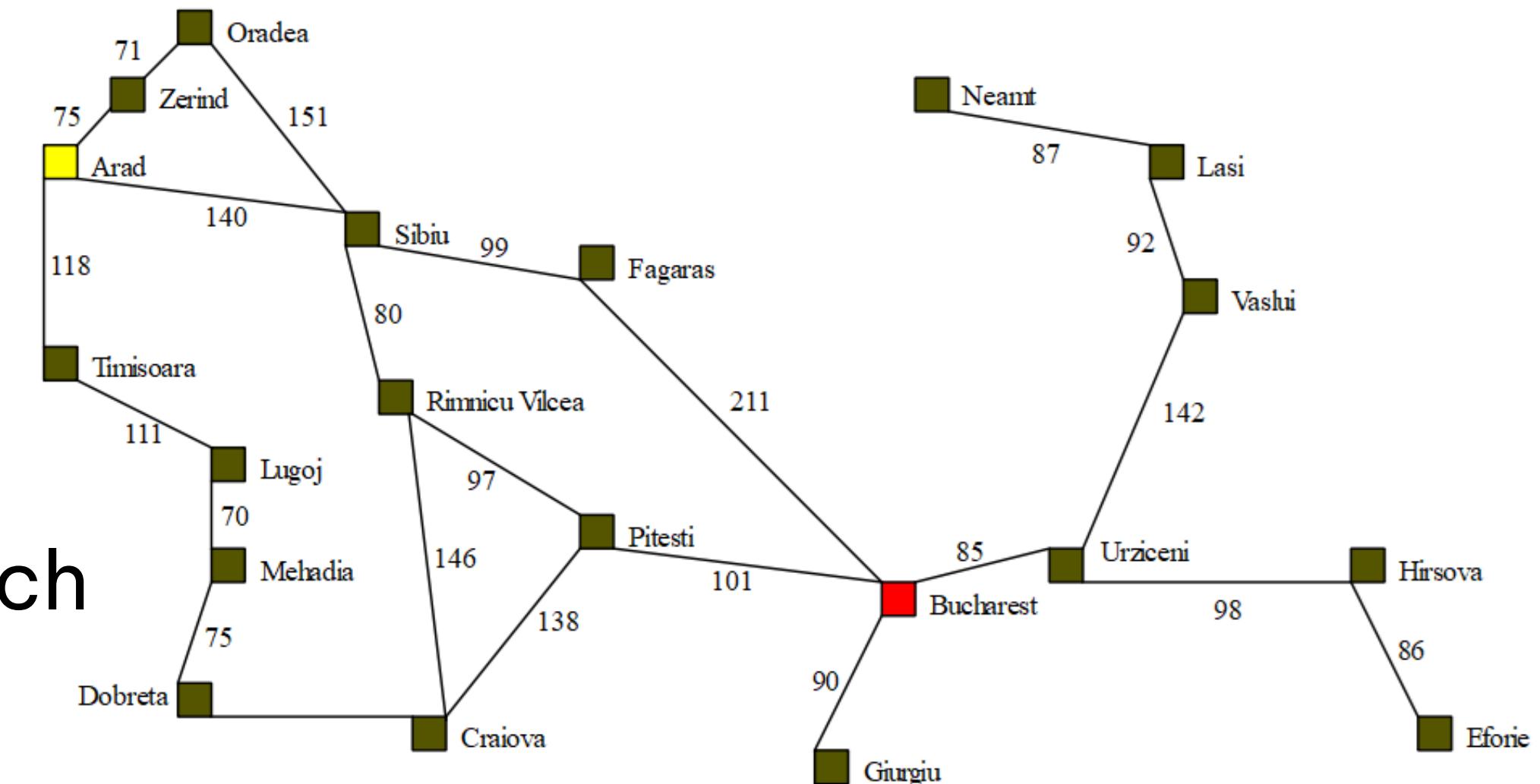
Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Algorithm for Knowledgeable Tree Search (AKT)
- Thuật giải AKT là mở rộng của giải thuật A^T bằng cách sử dụng thêm thông tin ước lượng h' (khoảng cách Manhattan).
- Độ tốt của hàm $f = g + h'$



$$h' = |X_A - X_B| + |Y_A - Y_B|$$

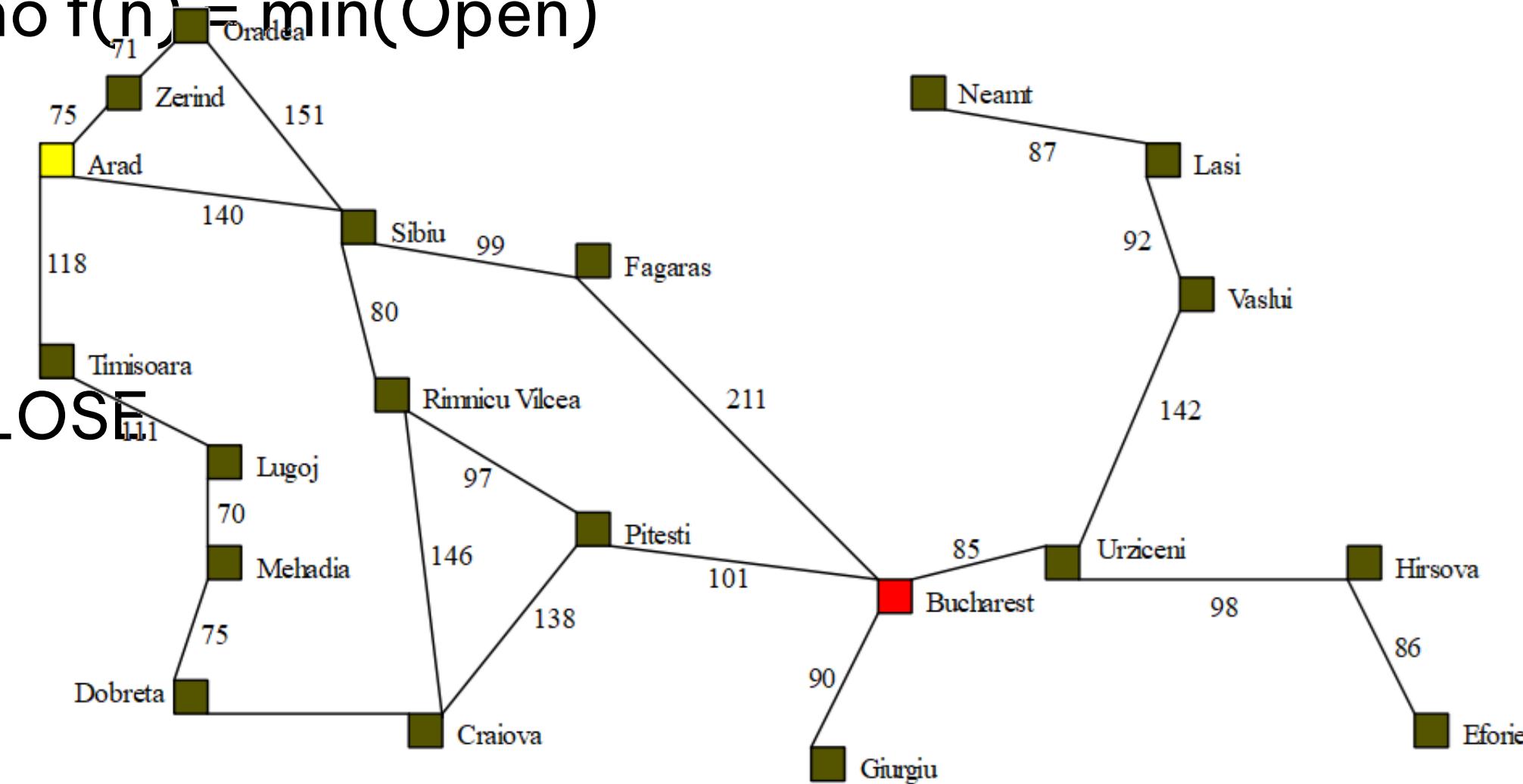
- h' còn được gọi là khoảng cách Euclid



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Algorithm for Knowledgeable Tree Search (AKT)

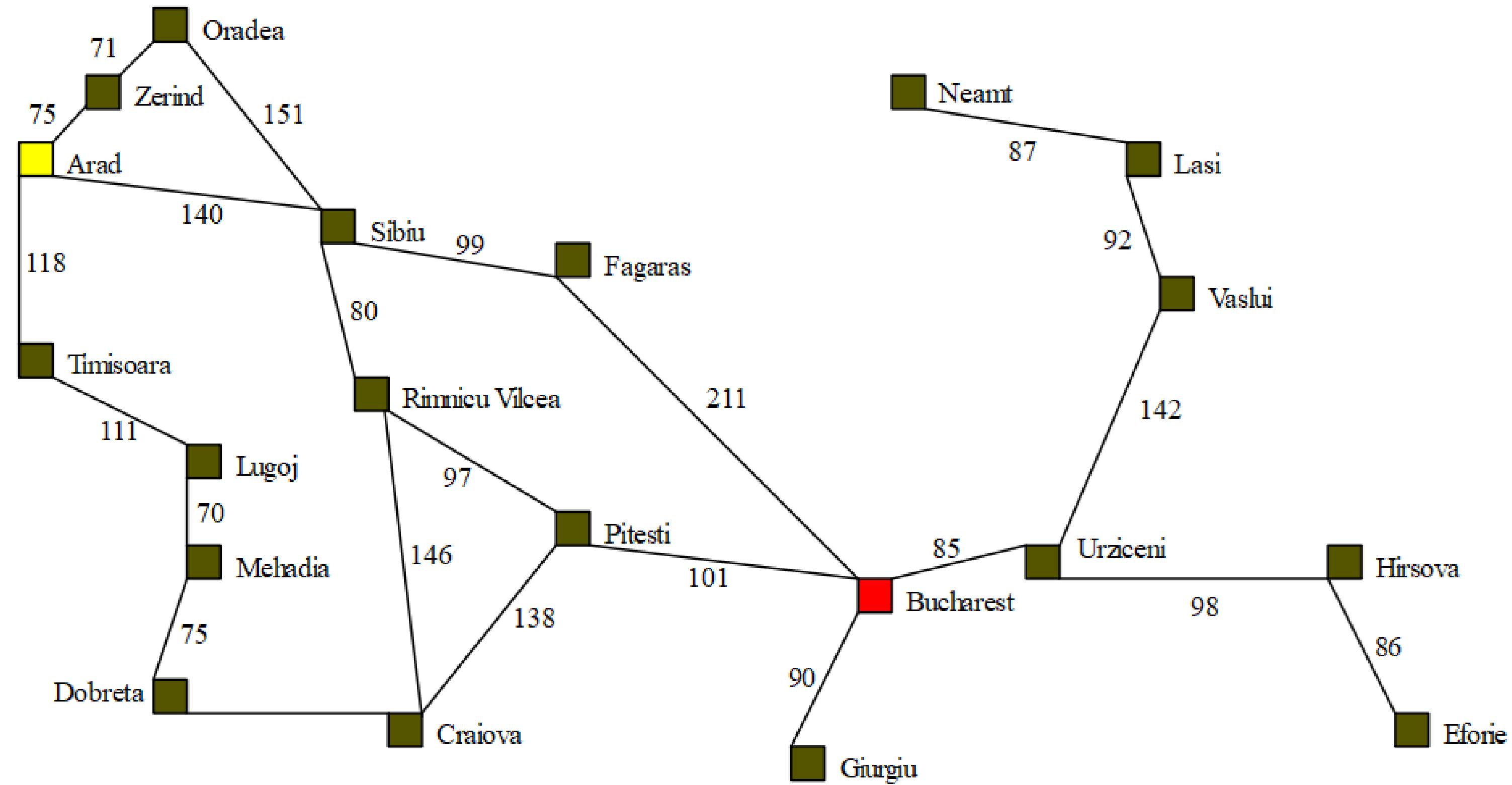
- Open:= {Start};
- While (Open $\neq \emptyset$)
 - n= Retrieve(Open); // Chọn n sao cho $f(n) = \min(\text{Open})$
 - If (n \equiv Goal): Return True
 - Else
 - Tạo $\Gamma(n)$;
 - for mỗi nút con m của $\Gamma(n) \notin \text{CLOSE}$
 - $g(m)=g(n)+\text{Cost}(n,m)$
 - $f(m)=g(m)+h'(m)$
 - OPEN = OPEN $\cup \{m\}$
 - CLOSE = CLOSE $\cup \{n\}$
 - Return False;



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Algorithm for Knowledgeable Tree Search (AKT)
- Tìm đường đi ngắn nhất từ Arad → Bucharest:

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Lasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Algorithm for Knowledgeable Tree Search (AKT)

▪ Tìm đường đi ngắn nhất từ Arad → Bucharest:

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Lasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

- Start = Arad, $\Gamma(n) = \{\text{Timisoara, Sibiu, Zerind}\}$,
- CLOSE = \emptyset
 - $g(\text{Timisoara}) = g(\text{Arad}) + \text{cost}(\text{Arad} \rightarrow \text{Timisoara}) = 0 + 118 = 118$
 - $f(\text{Timisoara}) = g(\text{Timisoara}) + h'(\text{Timisoara}) = 118 + 329 = 447$
 - $g(\text{Sibiu}) = g(\text{Arad}) + \text{cost}(\text{Arad} \rightarrow \text{Sibiu}) = 0 + 140 = 140$
 - $f(\text{Sibiu}) = g(\text{Sibiu}) + h'(\text{Sibiu}) = 140 + 253 = 393$
 - $g(\text{Zerind}) = g(\text{Arad}) + \text{cost}(\text{Arad} \rightarrow \text{Zerind}) = 0 + 75 = 75$
 - $f(\text{Zerind}) = g(\text{Zerind}) + h'(\text{Zerind}) = 75 + 374 = 449$
- OPEN= {Sibiu – 393, Timisoara – 447, Zerind-449}
 - $f_{\min} = f(\text{Sibiu})$,
 - Father(Sibiu) = Arad
 - Father(Timisoara) = Arad
 - Father(Zerind) = Arad

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Algorithm for Knowledgeable Tree Search (AKT)

▪ Tìm đường đi ngắn nhất từ Arad → Bucharest:

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Lasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

- $n = \text{Sibiu}$, $\Gamma(n) = \{\text{Arad}, \text{Fagaras}, \text{Oradea}, \text{Rimnicu Vilcea}\}$,
- CLOSE = {Arad, Sibiu}
 - $g(\text{Fagaras}) = g(\text{Sibiu}) + \text{cost}(\text{Sibiu} \rightarrow \text{Fagaras}) = 140 + 99 = 239$
 - $f(\text{Fagaras}) = g(\text{Fagaras}) + h'(\text{Fagaras}) = 239 + 178 = 417$
 - $g(\text{Oradea}) = g(\text{Sibiu}) + \text{cost}(\text{Sibiu} \rightarrow \text{Oradea}) = 140 + 151 = 291$
 - $f(\text{Oradea}) = g(\text{Oradea}) + h'(\text{Oradea}) = 291 + 380 = 671$
 - $g(\text{Rimnicu Vilcea}) = g(\text{Sibiu}) + \text{cost}(\text{Sibiu} \rightarrow \text{Rim.Vilcea}) = 140 + 80 = 220$
 - $f(\text{Rimnicu Vilcea}) = g(\text{Fagaras}) + h'(\text{Rimnicu Vilcea}) = 220 + 193 = 413$
- OPEN= {Rimnicu Vilcea - 413, Fagaras - 417, Timisoara - 447, Zerind-449}
 - $f_{\min} = f(\text{Rimnicu Vilcea})$,
 - Father(Fagaras) = Sibiu
 - Father(Oradea) = Sibiu
 - Father(Rimnicu Vilcea) = Sibiu

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Algorithm for Knowledgeable Tree Search (AKT)

▪ Tìm đường đi ngắn nhất từ Arad → Bucharest:

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Lasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

- $n = \text{Rimnicu Vilcea}$, $\Gamma(n) = \{\text{Sibiu, Craiova, Pitesti}\}$,
- CLOSE = {Arad, Sibiu, Rimnicu Vilcea}
 - $g(\text{Craiova}) = g(\text{R.Vilcea}) + \text{cost}(\text{R.Vilcea} \rightarrow \text{Craiova}) = 220 + 146 = 366$
 - $f(\text{Craiova}) = g(\text{Craiova}) + h'(\text{Craiova}) = 366 + 160 = 526$
 - $g(\text{Pitesti}) = g(\text{R.Vilcea}) + \text{cost}(\text{R.Vilcea} \rightarrow \text{Pitesti}) = 220 + 97 = 317$
 - $f(\text{Pitesti}) = g(\text{Pitesti}) + h'(\text{Pitesti}) = 317 + 98 = 415$
- OPEN= {Pitesti - 415, Fagaras - 417, Timisoara - 447, Zerind-449, Craiova - 526}
 - $f_{\min} = f(\text{Pitesti})$,
 - Father(Craiova) = Rimnicu Vilcea
 - Father(Pitesti) = Rimnicu Vilcea

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Algorithm for Knowledgeable Tree Search (AKT)

▪ Tìm đường đi ngắn nhất từ Arad → Bucharest:

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Lasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

- $n = \text{Pitesti}$, $\Gamma(n) = \{\text{Rimnicu Vilcea}, \text{Bucharest}, \text{Craiova}\}$,
- CLOSE = {Arad, Sibiu, Rimnicu Vilcea, Pitesti}
 - $g(\text{Bucharest}) = g(\text{Pitesti}) + \text{cost}(\text{Pitesti} \rightarrow \text{Bucharest}) = 317 + 101 = 418$
 - $f(\text{Bucharest}) = g(\text{Bucharest}) + h'(\text{Bucharest}) = 418 + 0 = 418$
 - $g_{\text{new}}(\text{Craiova}) = g(\text{Pitesti}) + \text{cost}(\text{Pitesti} \rightarrow \text{Craiova}) = 317 + 138 = 455$
 - $f_{\text{new}}(\text{Craiova}) = g(\text{Craiova}) + h'(\text{Craiova}) = 455 + 160 = 615$
- OPEN= {Fagaras - 417, Bucharest - 418, Timisoara - 447, Zerind-449, **Craiova - 526 < f_{new} 615**}
 - $f_{\min} = f(\text{Fagaras})$,
 - Father(Fagaras) = Sibiu (từ slide 28)
 - Father(Craiova) = Rimnicu Vilcea (vẫn giữ nguyên)

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

▪ Algorithm for Knowledgeable Tree Search (AKT)

▪ Tìm đường đi ngắn nhất từ Arad → Bucharest:

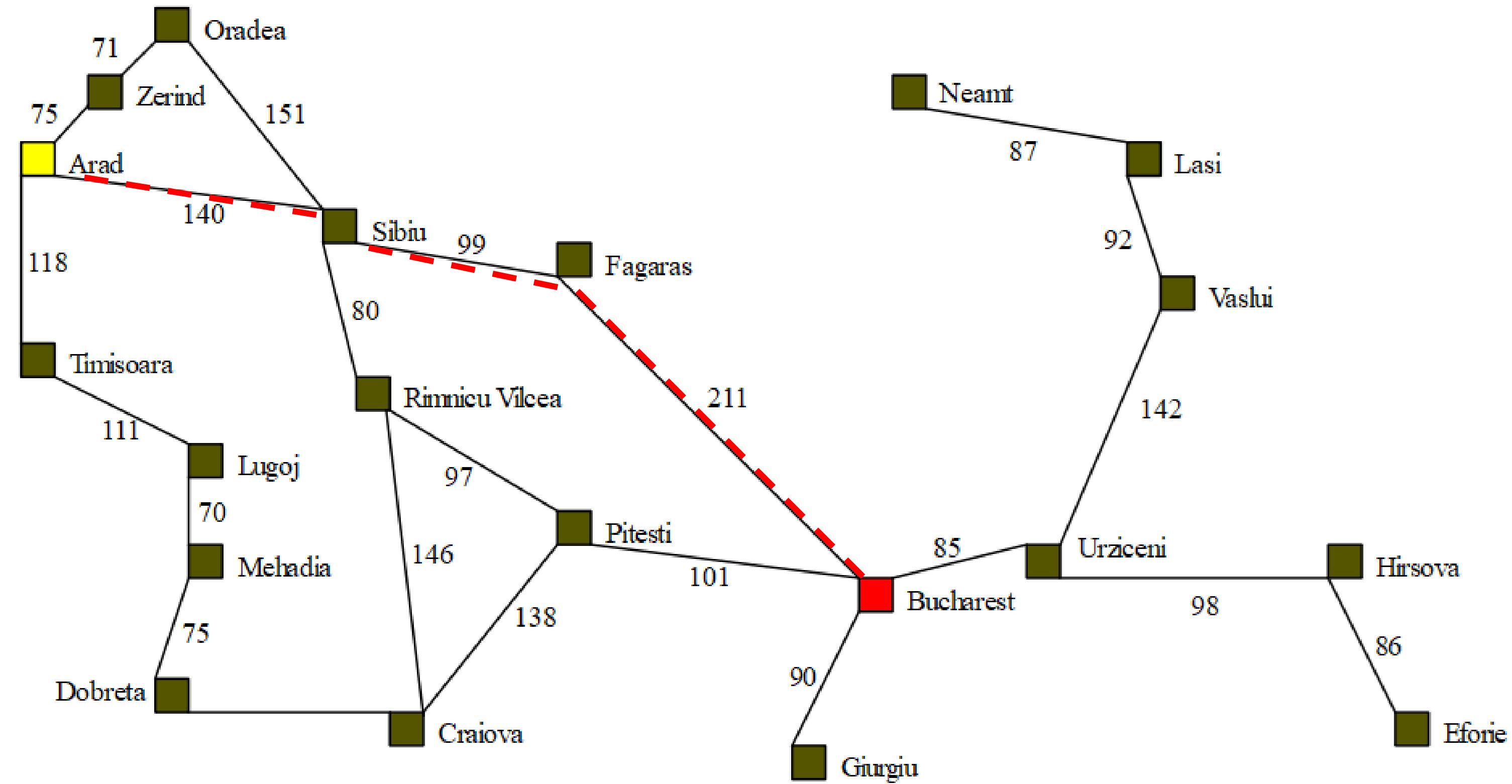
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Lasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

- $n = \text{Fagaras}$, $\Gamma(n) = \{\text{Sibiu}, \text{Bucharest}\}$,
- $\text{CLOSE} = \{\text{Arad}, \text{Sibiu}, \text{Rimnicu Vilcea}, \text{Pitesti}, \text{Fagaras}\}$
 - $g_{\text{new}}(\text{Bucharest}) = g(\text{Fagaras}) + \text{cost}(\text{Fagaras} \rightarrow \text{Bucharest}) = 417 + 0 = 417$
 - $f_{\text{new}}(\text{Bucharest}) = g_{\text{new}}(\text{Bucharest}) + h'(\text{Bucharest}) = 417 + 0 = 417$
- $\text{OPEN} = \{\text{Bucharest}_{\text{new}} - 417, \text{Timisoara} - 447, \text{Zerind}-449, \text{Craiova} - 526 < f_{\text{new}} 615\}$
 - $f_{\text{min}} = f(\text{Bucharest})$,
 - **Father(Bucharest) = Fagaras**
- $n = \text{Bucharest}$, TRUE

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Algorithm for Knowledgeable Tree Search (AKT)
- Tìm đường đi ngắn nhất từ Arad → Bucharest:

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Lasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- A star(A*)
- Thuật giải A* tránh việc xét các nhánh tìm kiếm đã xác định (cho đến thời điểm hiện tại) có chi phí cao.
- Sử dụng hàm đánh giá $f(u) = g(u) + h(u)$
 - $g(u)$: chi phí của đường đi từ nút gốc nút hiện tại
 - $h(u)$: ước lượng heuristic = khoảng cách Manhattan
 - $f(u)$: chi phí tổng thể ước lượng của đường đi qua nút hiện tại (u) → đích
- A* là phiên bản đặc biệt của A^{KT} áp dụng cho trường hợp đồ thị
- A* mở rộng A^{KT} bằng cách bổ sung cách giải quyết trường hợp khi mở một nút mà nút này đã có sẵn trong OPEN hoặc CLOSE.

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- A star(A*)
- OPEN: tập chứa các trạng thái đã được sinh ra nhưng chưa xét đến
 - Các phần tử được sắp xếp theo độ tốt (giá trị của hàm f)
 - Phần tử có độ ưu tiên cao nhất = phần tử tốt nhất
- CLOSE: Danh sách các trạng thái đã xét
- Hàm REMOVE(): Lấy phần tử ở đầu danh sách OPEN
- Hàm INSERT(): Chèn phần tử vào OPEN theo thứ tự ưu tiên
- Hàm FATHER(x): Trạng thái cha của X

```
function A_Star_Search(){  
    Input: start, goal;  
    Output: Đường đi từ start → goal hoặc thất bại  
}
```

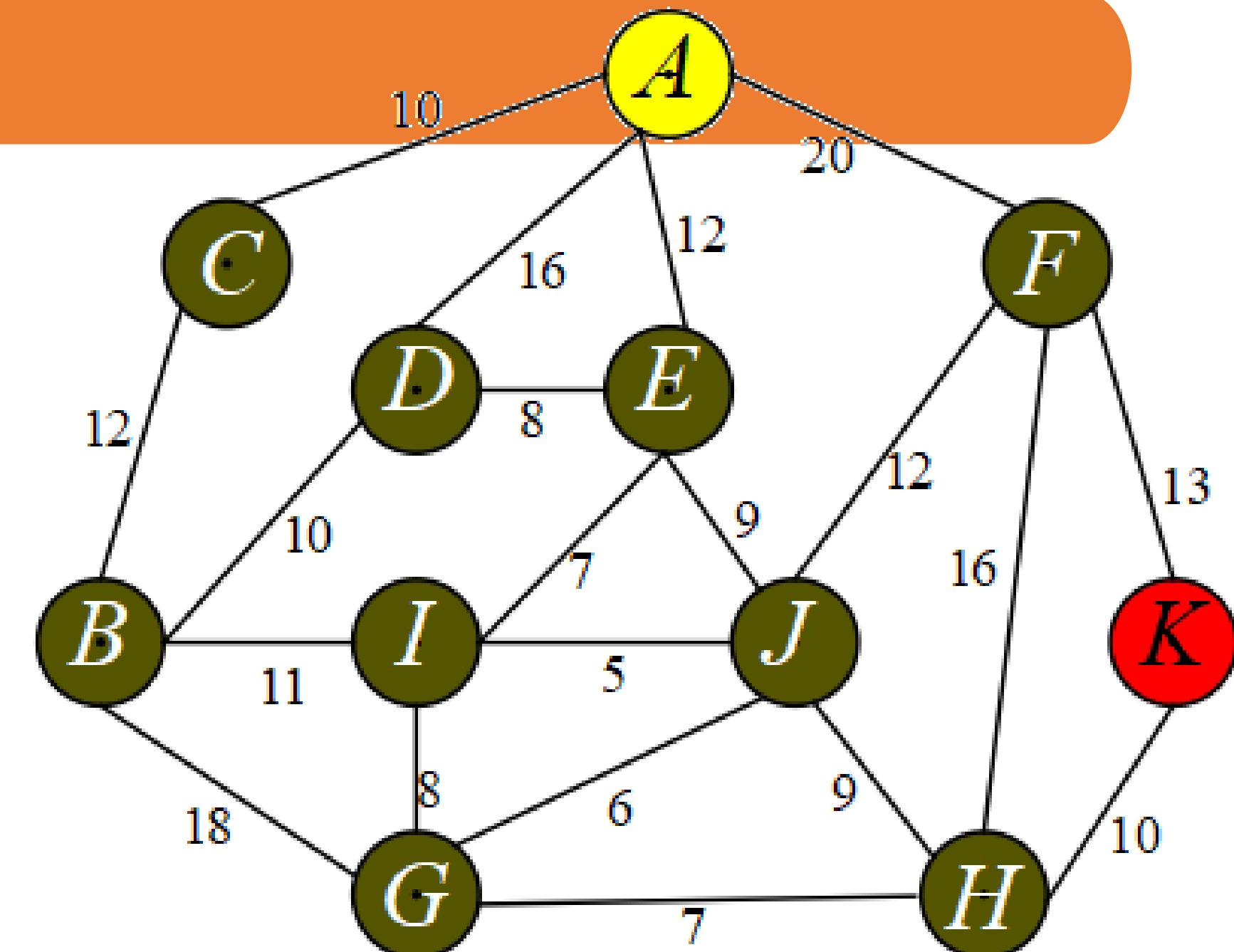
Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- A star(A*)
- OPEN = start; CLOSE = \emptyset ; $g(\text{start}) = 0$; $f(\text{start}) = h(\text{start})$
- Loop:
 - if ($\text{OPEN} = \emptyset$): Thất bại
 - $n = \text{REMOVE}(\text{OPEN})$ // phần tử đầu ds OPEN ($f(n) == \min$)
 - if ($n == \text{goal}$): Tìm thấy đường đi
 - Ngược lại:
 - Đưa $\{n\}$ vào danh sách CLOSE
 - for (mỗi trạng thái m là con của $\{n\}$):
 - Tính $g(m) = g(n) + \text{cost}(n,m)$
 - Tính $f(m) = g(m) + h(m)$
 - if ($\exists! t \in \text{OPEN} \equiv m$): *(cond 1)*
 - elseif ($\exists! t \in \text{CLOSE} \equiv m$): *(cond 2)*
 - else *INSERT(n,OPEN)// $m \notin \{\text{OPEN}, \text{CLOSE}\}$*
 - if ($\exists! t \in \text{OPEN} \equiv m$):
 - if ($f(m)_{\text{new}} < f(t)_{\text{old}}$):
 - $g(t)_{\text{old}} = g(m)_{\text{new}}$
 - $f(t)_{\text{old}} = f(m)_{\text{new}}$
 - Sắp xếp lại OPEN
 - FATHER(t) = n
 - elseif ($\exists! t \in \text{CLOSE} \equiv m$):
 - if ($f(m)_{\text{new}} < f(t)_{\text{old}}$):
 - $g(t)_{\text{old}} = g(m)_{\text{new}}$
 - $f(t)_{\text{old}} = f(m)_{\text{new}}$
 - FATHER(t) = n

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- A star(A*)

- Áp dụng giải thuật A* để tìm đường đi ngắn nhất $A \rightarrow K$, với các ước lượng heuristic của các trạng thái so với trạng thái đích được cho ở bảng sau:



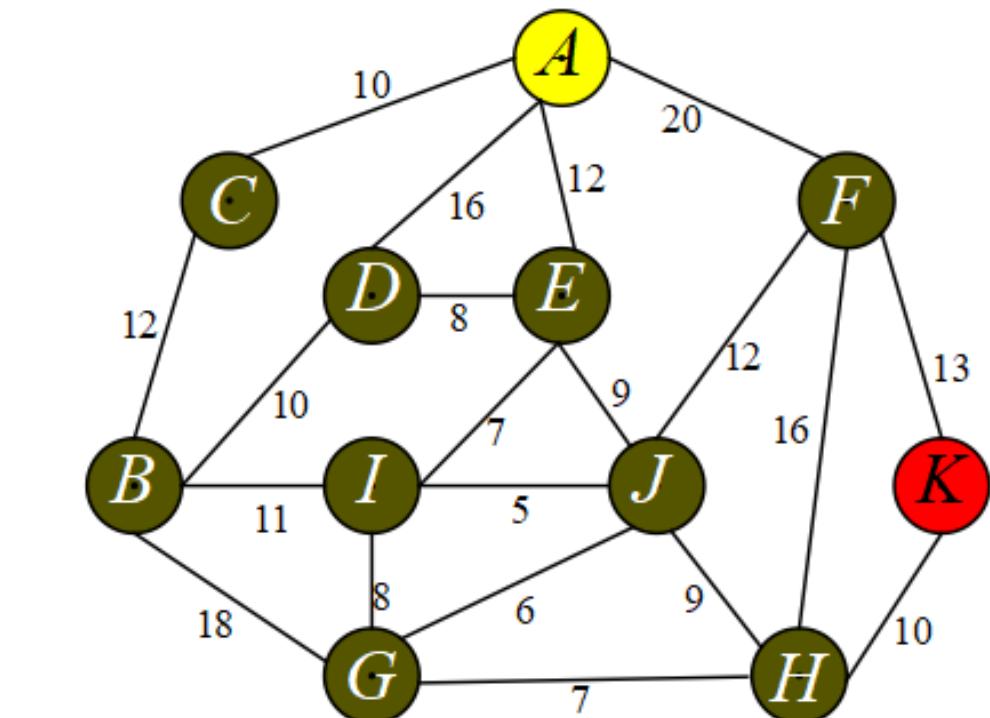
A	B	C	D	E	F	G	H	I	J	K
33	35	43	36	28	13	17	10	24	19	0

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- A star(A*)

- Start = {A}, goal = {K}, OPEN = {A}, CLOSE = \emptyset
- $g(A) = 0, f(A) = h(A) = 33$
- Bước 1. n=A, CLOSE={A}
 - $g(C) = g(A)+\text{cost}(A,C) = 0+10 = 10$
 - $f(C) = g(C)+h(C) = 10+43 = 53$
 - $g(D) = g(A)+\text{cost}(A,D) = 0+16 = 16$
 - $f(D) = g(D)+h(D) = 16+36 = 49$
 - $g(E) = g(A)+\text{cost}(A,E) = 0+12 = 12$
 - $f(E) = g(E)+h(E) = 12+28 = 40$
 - $g(F) = g(A)+\text{cost}(A,F) = 0+20 = 20$
 - $f(F) = g(F)+h(F) = 20+13 = 33$
 - $\{C,D,E,F\} \notin \text{OPEN} \text{ và } \{C,D,E,F\} \notin \text{CLOSE}$
 - OPEN = {F(33), E(40), D(49), C(53)}

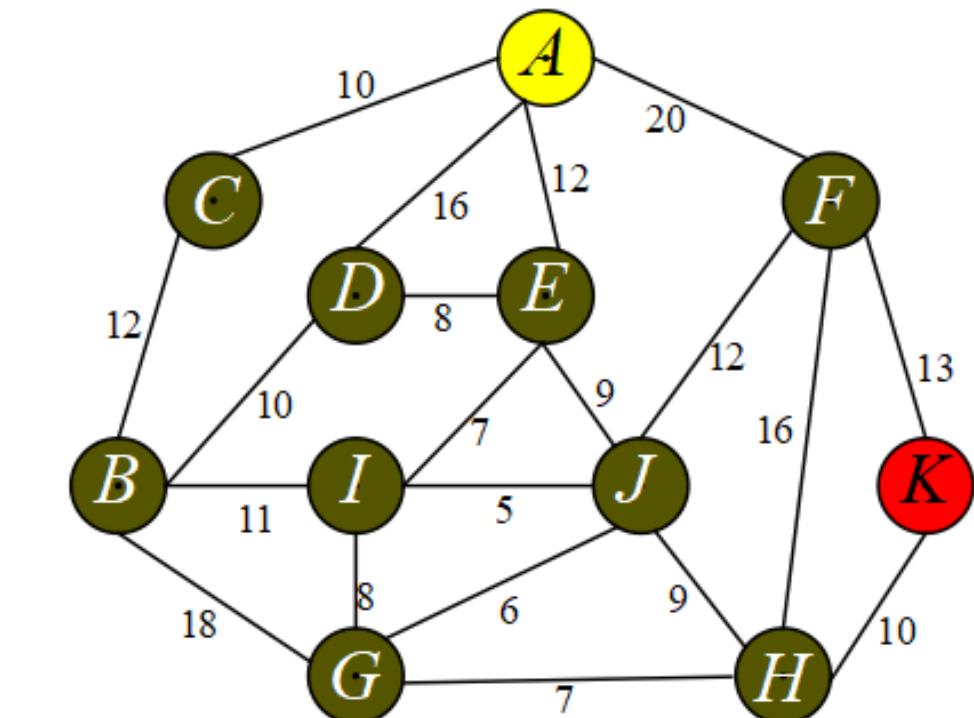
A	B	C	D	E	F	G	H	I	J	K
33	35	43	36	28	13	17	10	24	19	0



- Father(F, E, D, C) = A

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- A star(A*)
- Bước 2. n=F, CLOSE={A,F}, OPEN = {E,D,C}
- $g_{\text{new}}(A) = g(F)+\text{cost}(F,A) = 20+20 = 40$
- $f_{\text{new}}(A) = g(A)+h(A) = 40+33 = 73$
- $t=A \in \text{CLOSE}, f_{\text{new}}(A) < f(t) (73 < 55)$: FALSE
- $g(A)=0, f(A)=33$
- $g(J) = g(F)+\text{cost}(F,J) = 20+12 = 32$
- $f(J) = g(J)+h(J) = 32+19 = 51$
- $g(H) = g(F)+\text{cost}(F,H) = 20+16 = 36$
- $f(H) = g(H)+h(H) = 36+10 = 46$
- $g(K) = g(F)+\text{cost}(F,K) = 20+13 = 33$
- $f(K) = g(K)+h(K) = 33+0 = 33$
- $\{H,J,K\} \notin \text{OPEN} \text{ và } \{H,J,K\} \notin \text{CLOSE}$
- OPEN = {K(33), E(40), H(46), D(49), J(51), C(53)}



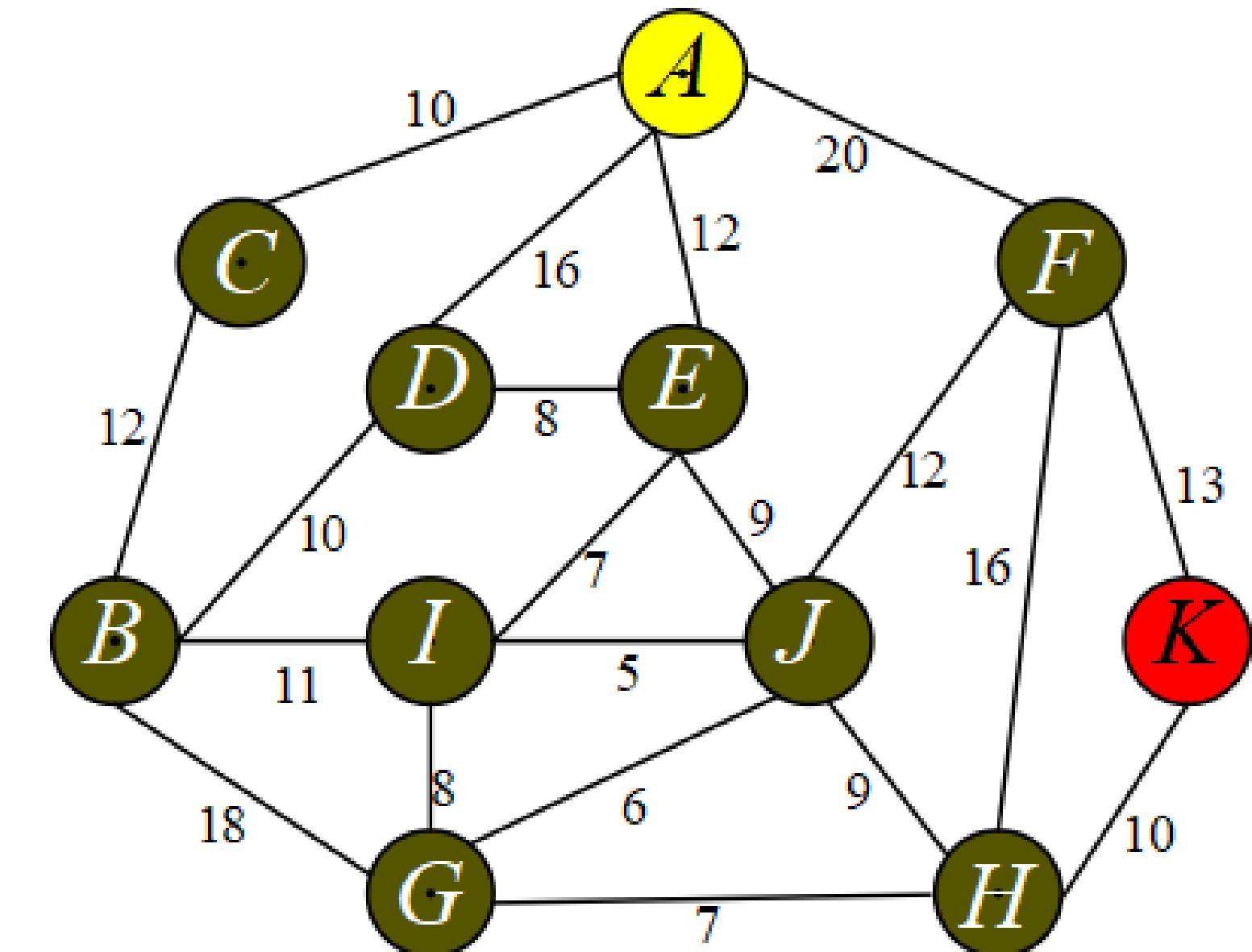
A	B	C	D	E	F	G	H	I	J	K
33	35	43	36	28	13	17	10	24	19	0

- Father(J, H, K) = F

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- A star(A*)
- Bước 3. n=K, TRUE
- Father(J, H, K) = F
- Father(F, E, D, C) = A

▪ $A \Rightarrow F \Rightarrow K$

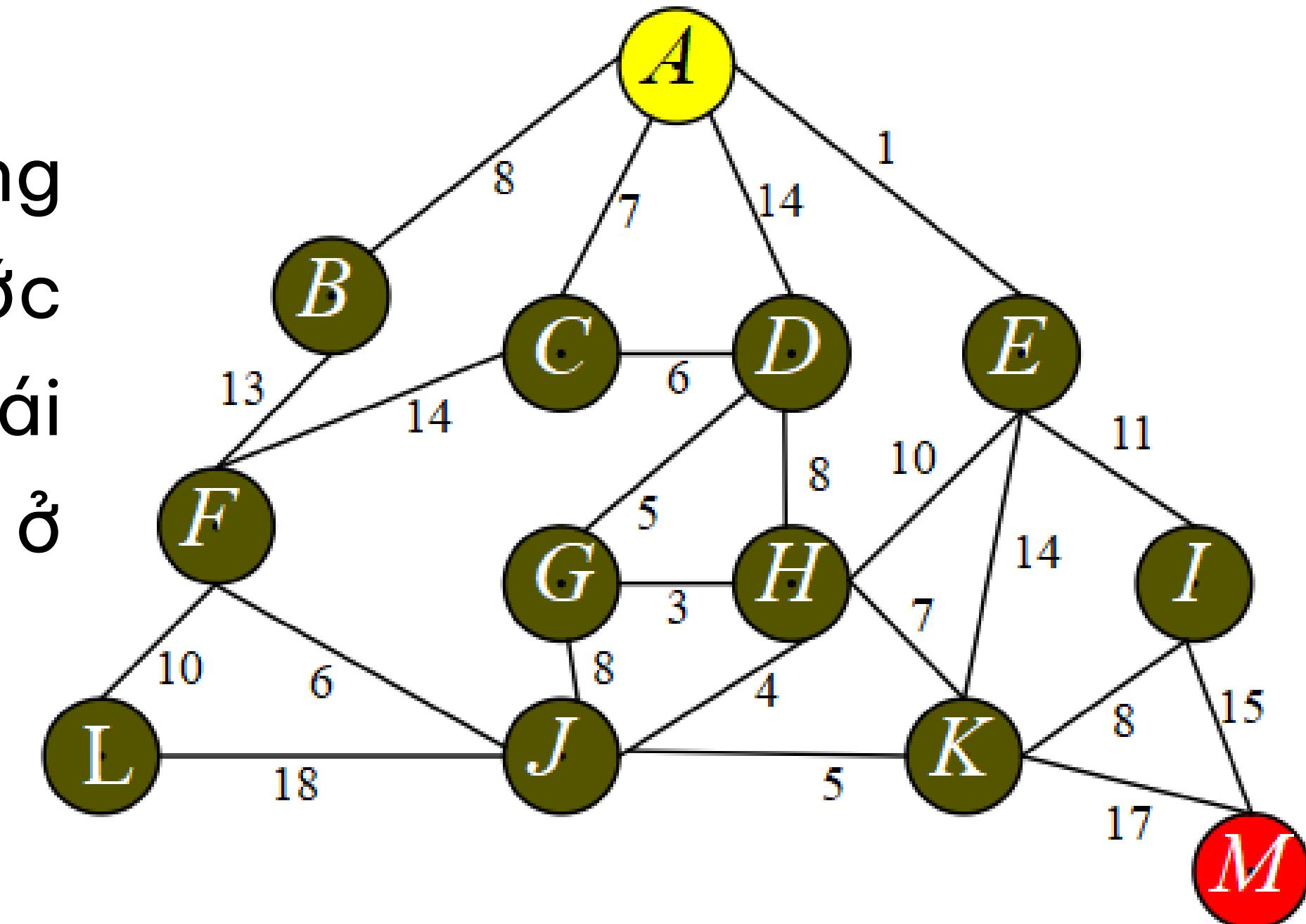


A	B	C	D	E	F	G	H	I	J	K
33	35	43	36	28	13	17	10	24	19	0

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- A star(A*)

- Áp dụng giải thuật A* để tìm đường đi ngắn nhất $A \rightarrow M$, với các ước lượng heuristic của các trạng thái so với trạng thái đích được cho ở bảng sau:

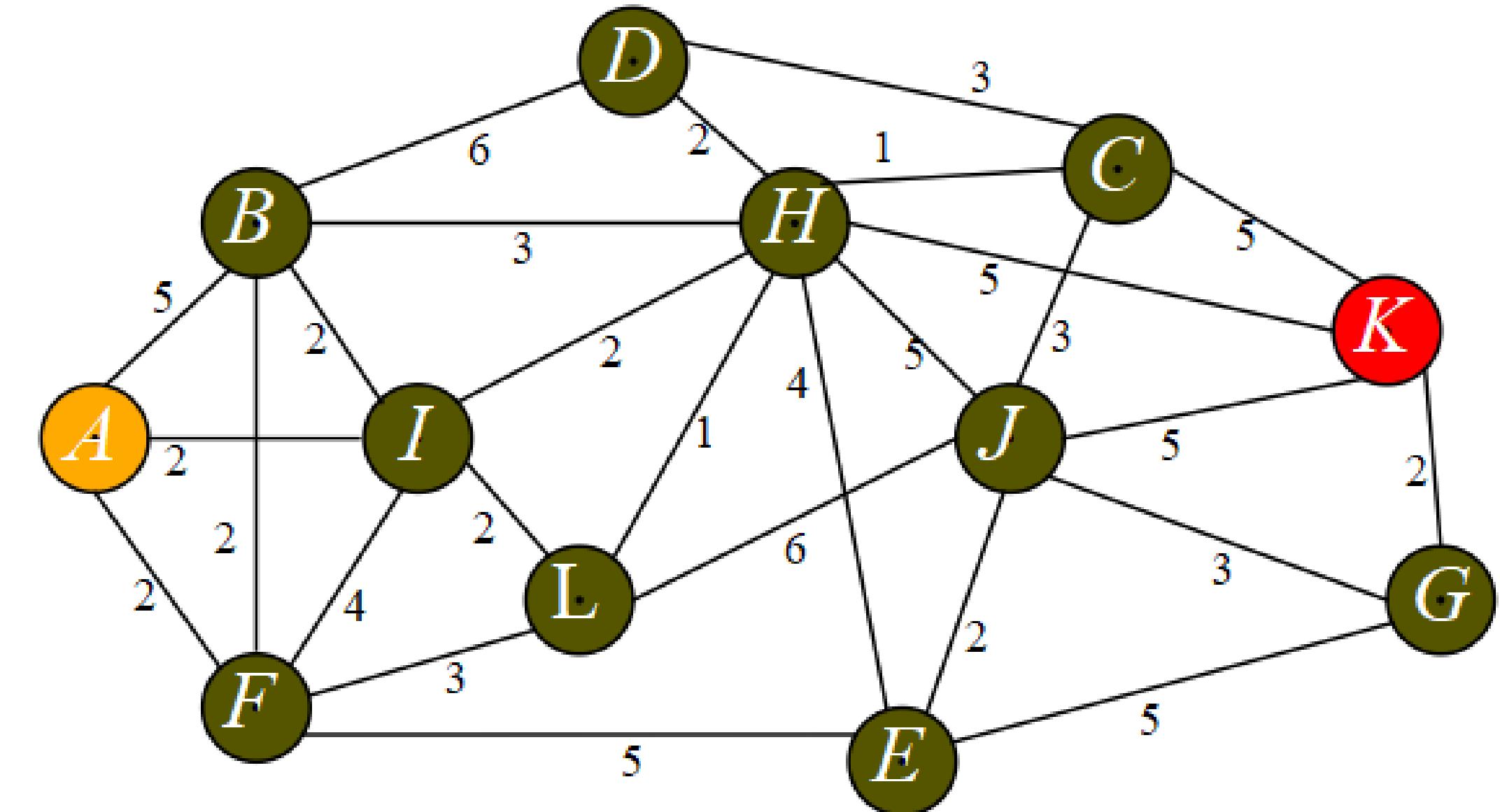


A	B	C	D	E	F	G	H	I	J	K	L	M
27	34	33	32	26	28	27	24	15	22	17	15	0

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- A star(A*)

- Áp dụng giải thuật A* để tìm đường đi ngắn nhất A → K, với các ước lượng heuristic của các trạng thái so với trạng thái đích được cho ở bảng sau:



A	B	C	D	E	F	G	H	I	J	K	L
9	8	5	7	7	9	2	5	7	5	0	6

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Nhánh - cận (Branch-and-Bound)
- Thuật toán tìm kiếm leo đồi kết hợp với hàm đánh giá $f(u)$. Tại mỗi bước, khi phát triển trạng thái u , chọn trạng thái con v tốt nhất ($f(v)$ nhỏ nhất) của u để phát triển ở bước sau.
- Quá trình tiếp tục như vậy cho đến khi gặp trạng thái w là đích, hoặc w không có đỉnh kề, hoặc w có $f(w)$ lớn hơn độ dài đường đi tối ưu tạm thời. Trong các trường hợp này, chúng ta không phát triển đỉnh w nữa, tức là cắt bỏ những nhánh xuất phát từ w , và quay lên cha của w để tiếp tục đi xuống trạng thái tốt nhất trong số những trạng thái còn lại chưa được phát triển

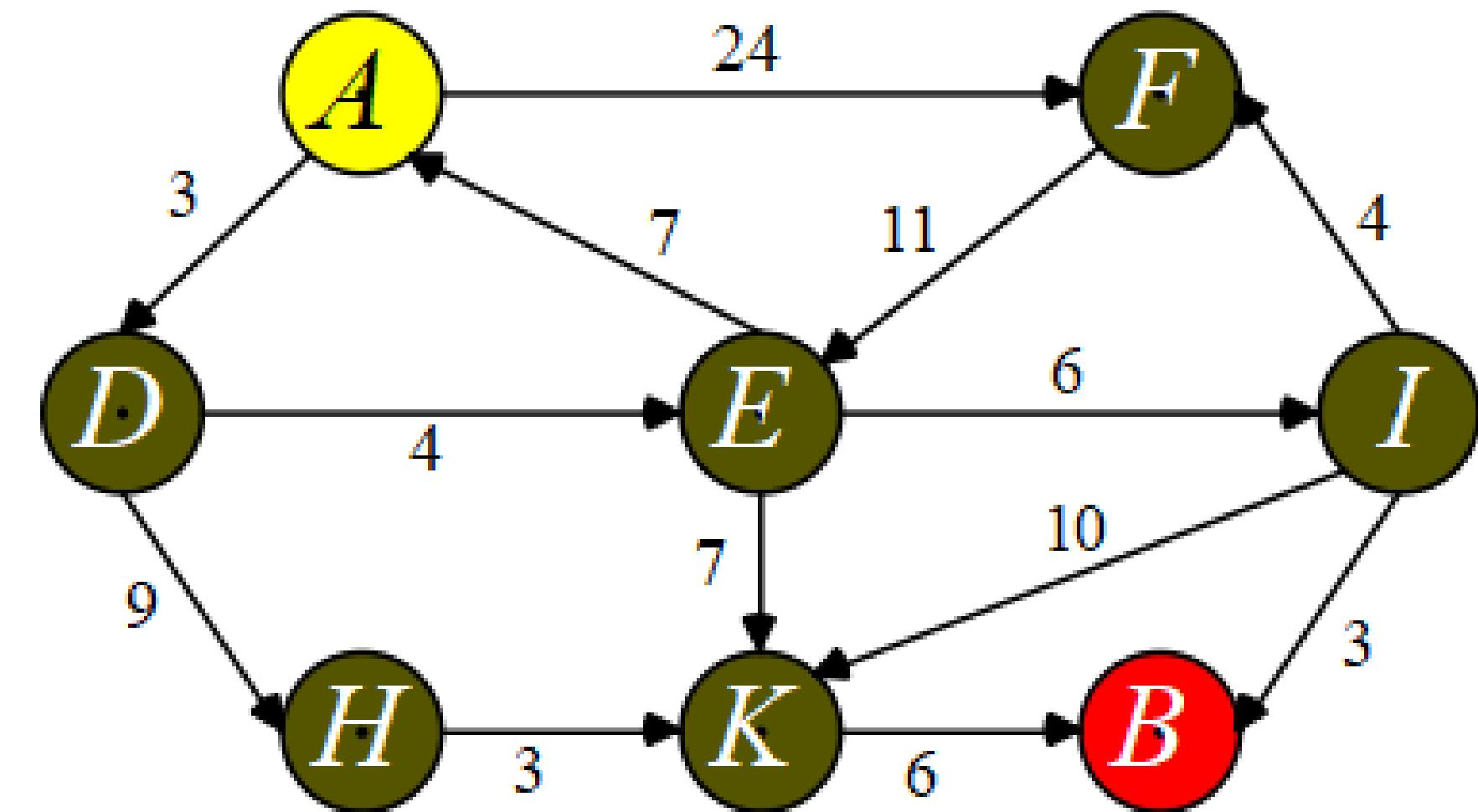
Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Nhánh - cận (Branch-and-Bound)
- Bước 1. OPEN = {start}, min = $+\infty$
- Bước 2. Loop:
 - 2.1. if(OPEN = \emptyset): STOP
 - 2.2. n:= lấy phần tử đầu danh sách OPEN
 - 2.3. if(n= goal):
 - if ($f(n) < \text{min}$): { $\text{min}=f(n)$; quay lại 2.1 //cập nhật lại min}}
 - 2.4. if ($f(n) > \text{min}$): quay lại 2.1 //cắt bỏ nhánh con
 - 2.5. for ($\forall m$: đỉnh kề n):
 - $g(m) = g(n) + \text{cost}(m,n)$
 - $f(m) = g(m)+h(m)$
 - Đưa m vào danh sách $\Gamma(n)$
 - 2.6. Sắp xếp $\Gamma(n)$ theo thứ tự tăng dần của f
 - 2.7. Chèn $\Gamma(n)$ vào đầu OPEN

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Nhánh - cận (Branch-and-Bound)

- Áp dụng thuật toán nhánh cận để tìm đường đi ngắn nhất từ đỉnh A → I, với các ước lượng heuristic của các trạng thái so với trạng thái đích được cho ở Bảng.



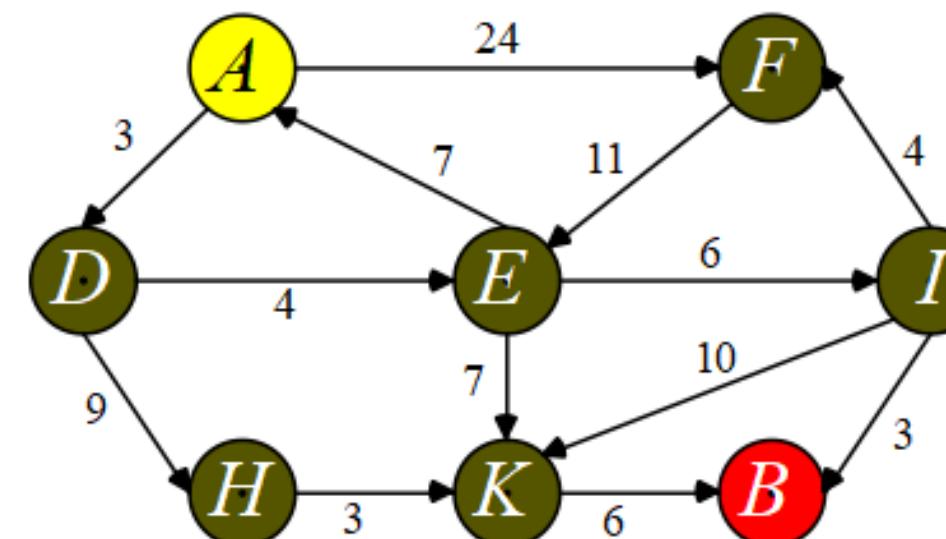
A	B	D	E	F	H	I	K
13	0	2	8	9	7	6	4

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Nhánh - cận (Branch-and-Bound)

Bước	n	m	$g(m) = g(n) + \text{cost}(n,m)$	$f(m) = g(m) + h(m)$	$\Gamma(n) \uparrow$	OPEN	min
0			$g(A) = 0$	$f(A) = h(A) = 13$	\emptyset	A	$+\infty$
1	A $f(A) < \text{min}$	D F	$g(D) = g(A) + \text{cost}(A,D)$ $= 0 + 3 = 3$ $g(F) = g(A) + \text{cost}(A,F)$ $= 0 + 24 = 24$	$f(D) = g(D) + h(D)$ $= 3 + 2 = 5$ $f(F) = g(F) + h(F)$ $= 24 + 9 = 33$	$D^4 \rightarrow F^{33}$	D^4, F^{33}	$+\infty$

- Father(D,F) = A



VET								
A	B	D	E	F	H	I	K	
1	-	-	-	-	-	-	-	-

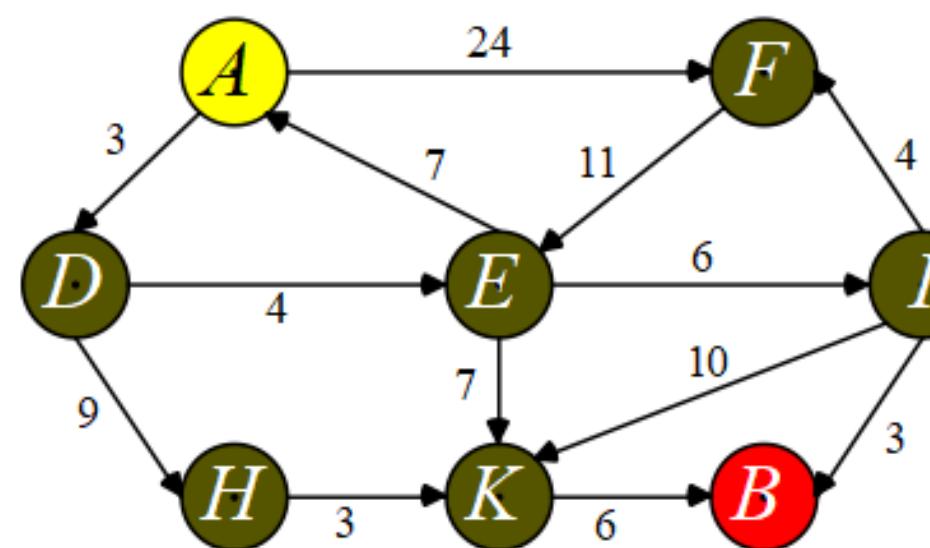
HÀM LƯỢNG GIÁ (H)								
A	B	D	E	F	H	I	K	
13	0	2	8	9	7	6	4	

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

Nhánh - cận (Branch-and-Bound)

Bước	n	m	$g(m) = g(n) + \text{cost}(n,m)$	$f(m) = g(m) + h(m)$	$\Gamma(n) \uparrow$	OPEN	min
2	D f(D)<min	E H	$\begin{aligned} g(E) &= g(D) + \text{cost}(D,E) \\ &= 3 + 4 = 7 \end{aligned}$	$\begin{aligned} f(E) &= g(E) + h(E) \\ &= 7 + 8 = 15 \end{aligned}$	$E^{15} \rightarrow H^{19}$	E^{15}, H^{19}, F^{33}	$+\infty$

- Father(D,F) = A
- Father(E,H) = D



VET							
A	B	D	E	F	H	I	K
1	-	2	-	-	-	-	-

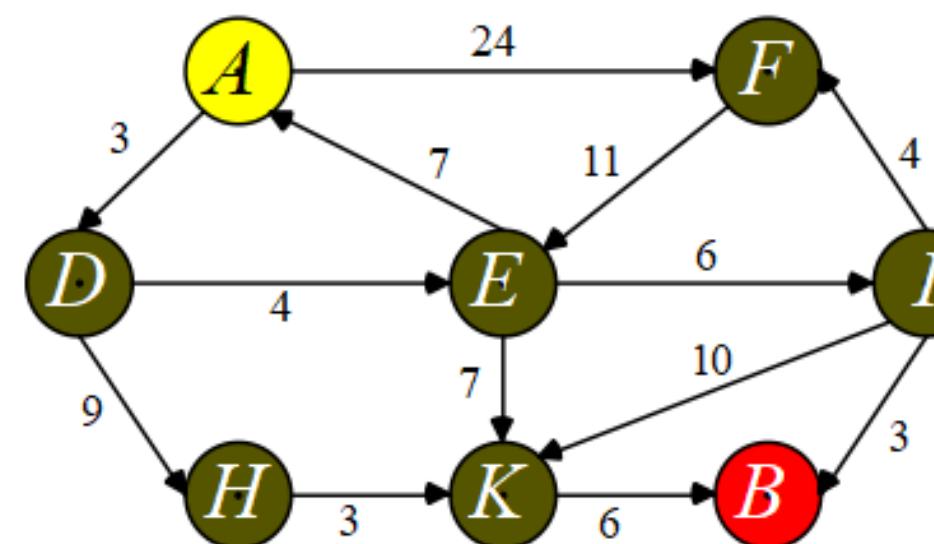
HÀM LƯỢNG GIÁ (H)							
A	B	D	E	F	H	I	K
13	0	2	8	9	7	6	4

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

Nhánh - cận (Branch-and-Bound)

Bước	n	m	$g(m) = g(n) + \text{cost}(n,m)$	$f(m) = g(m) + h(m)$	$\Gamma(n) \uparrow$	OPEN	min
3 f(E)<min	E	I K A	$g(I) = g(E) + \text{cost}(E,I)$ $= 7 + 6 = 13$ $g(K) = g(E) + \text{cost}(E,K)$ $= 7 + 7 = 14$ $g(A) = g(E) + \text{cost}(E,A)$ $= 7 + 7 = 14$	$f(I) = g(I) + h(I)$ $= 13 + 6 = 19$ $f(K) = g(K) + h(K)$ $= 14 + 4 = 18$ $f(A) = g(A) + h(A)$ $= 14 + 13 = 26$	$K^{18} \rightarrow$ $I^{19} \rightarrow$ A^{26}	$K^{18}, I^{19}, A^{26},$ H^{19}, F^{33}	$+\infty$

- Father(D,F) = A
- Father(E,H) = D
- Father(I,K,A) = E



VET							
A	B	D	E	F	H	I	K
1	-	2	3	-	-	-	-

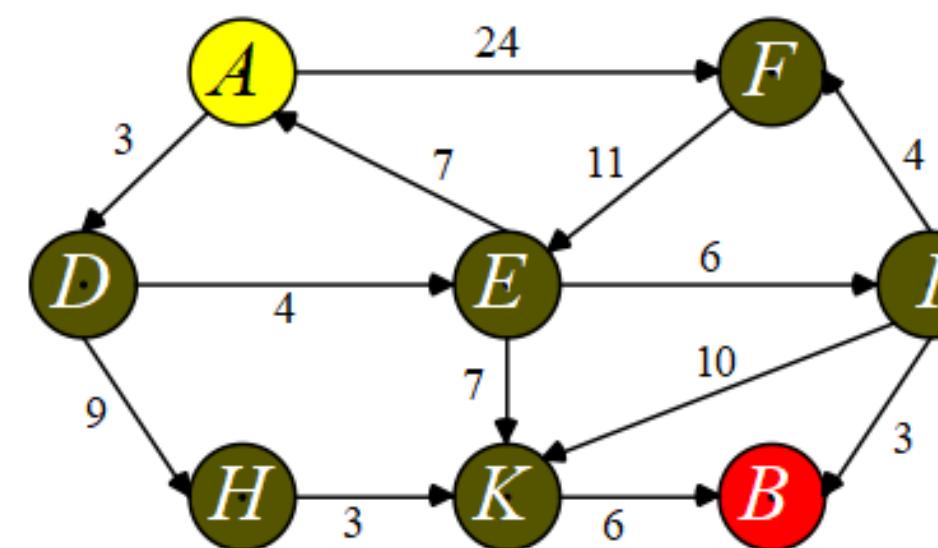
HÀM LƯỢNG GIÁ (H)							
A	B	D	E	F	H	I	K
13	0	2	8	9	7	6	4

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

Nhánh - cận (Branch-and-Bound)

Bước	n	m	$g(m) = g(n) + \text{cost}(n,m)$	$f(m) = g(m) + h(m)$	$\Gamma(n) \uparrow$	OPEN	min
4	K $f(K) < \text{min}$	B	$g(B) = g(K) + \text{cost}(K,B)$ $= 14 + 6 = 20$	$f(B) = g(B) + h(B)$ $= 20 + 0 = 20$	B^{20}	$B^{20}, I^{19}, A^{26},$ H^{19}, F^{33}	$+\infty$

- Father(D,F) = A
- Father(E,H) = D
- Father(I,K,A) = E



VET							
A	B	D	E	F	H	I	K
1	-	2	3	-	-	-	4

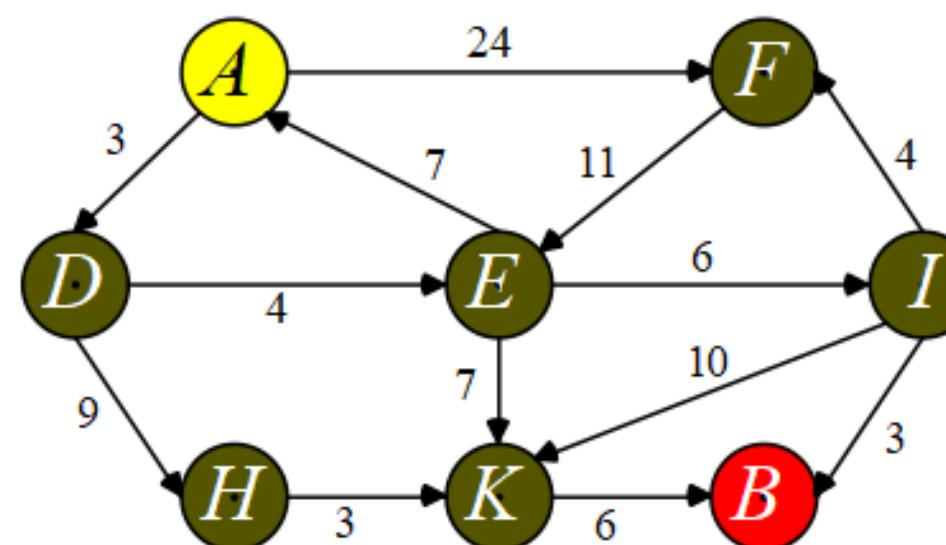
HÀM LƯỢNG GIÁ (H)							
A	B	D	E	F	H	I	K
13	0	2	8	9	7	6	4

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

Nhánh - cận (Branch-and-Bound)

Bước	n	m	$g(m) = g(n) + \text{cost}(n,m)$	$f(m) = g(m) + h(m)$	$\Gamma(n) \uparrow$	OPEN	min
5	B	B		$f(B) = 20 < \text{min}$		$I^{19}, A^{19}, H^{19}, F^{33}$	20
6 $f(I) < \text{min}$	I	B	$g(B) = g(I) + \text{cost}(I,B)$ = $13 + 3 = 16$	$f(B) = g(B) + h(B)$ = $16 + 0 = 16$	$B^{16} \rightarrow$ $F^{26} \rightarrow$ K^{27}	$B^{16}, F^{26}, K^{27},$ A^{19}, H^{19}, F^{33}	20
		F	$g(F) = g(I) + \text{cost}(I,F)$ = $13 + 4 = 17$	$f(F) = g(F) + h(F)$ = $17 + 19 = 26$			
		K	$g(K) = g(I) + \text{cost}(I,K)$ = $13 + 10 = 23$	$f(K) = g(K) + h(K)$ = $23 + 4 = 27$			

- Father(D,F) = A
- Father(E,H) = D
- Father(I,K,A) = E



VET							
A	B	D	E	F	H	I	K
1	-	2	3	-	-	-	4

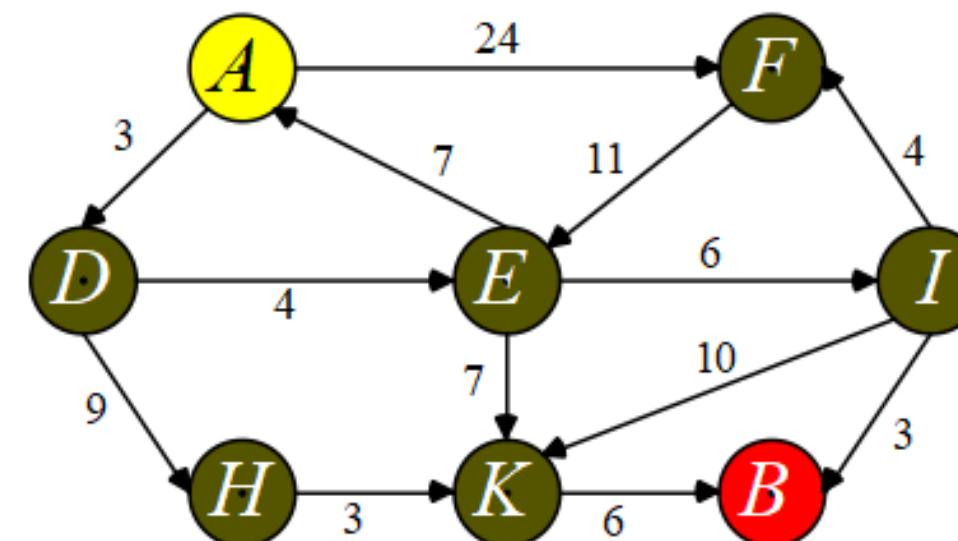
HÀM LƯỢNG GIÁ (H)							
A	B	D	E	F	H	I	K
13	0	2	8	9	7	6	4

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

Nhánh - cận (Branch-and-Bound)

Bước	N	m	$g(m) = g(n) + \text{cost}(n,m)$	$f(m) = g(m) + h(m)$	$\Gamma(n) \uparrow$	OPEN	min
7	B	B		$f(B) = 16 < \text{min}$		$F^{26}, K^{27}, A^{19}, H^{19}, F^{33}$	16
8	F		quay lại 2.1 (cắt bỏ nhánh con)	$f(F) = 26 > \text{min}$		$F^{26}, K^{27}, A^{19}, H^{19}, F^{33}$	16

- Father(D,F) = A
- Father(E,H) = D
- Father(I,K,A) = E



VET							
A	B	D	E	F	H	I	K
1	-	2	3	-	-	-	4

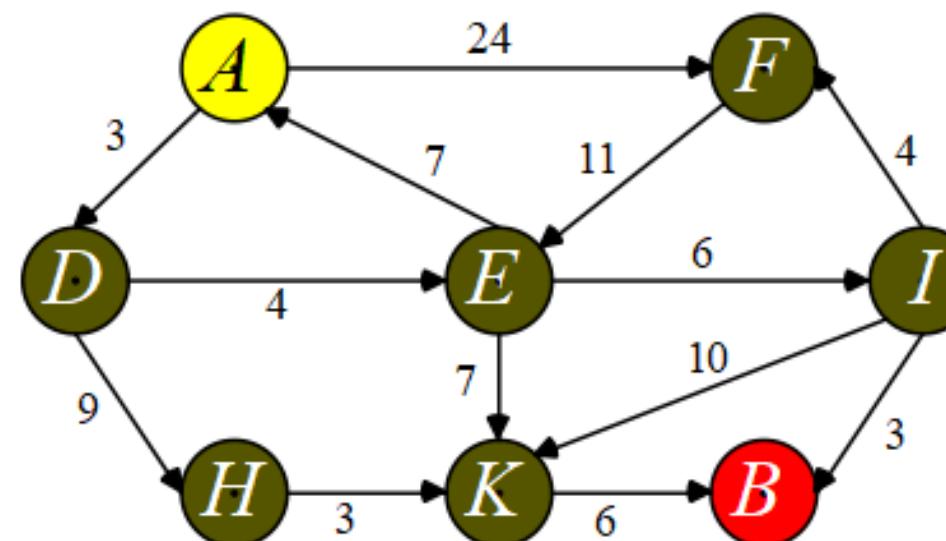
HÀM LƯỢNG GIÁ (H)							
A	B	D	E	F	H	I	K
13	0	2	8	9	7	6	4

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

Nhánh - cận (Branch-and-Bound)

Bước	N	m	$g(m) = g(n) + \text{cost}(n,m)$	$f(m) = g(m) + h(m)$	$\Gamma(n) \uparrow$	OPEN	min
9	K		quay lại 2.1 (cắt bỏ nhánh con)	$f(K) = 27 > min$		$K^{27}, A^{19}, H^{19}, F^{33}$	16
10	A		quay lại 2.1 (cắt bỏ nhánh con)	$f(A) = 19 > min$		A^{19}, H^{19}, F^{33}	16

- Father(D,F) = A
- Father(E,H) = D
- Father(I,K,A) = E



VET							
A	B	D	E	F	H	I	K
1	-	2	3	-	-	-	4

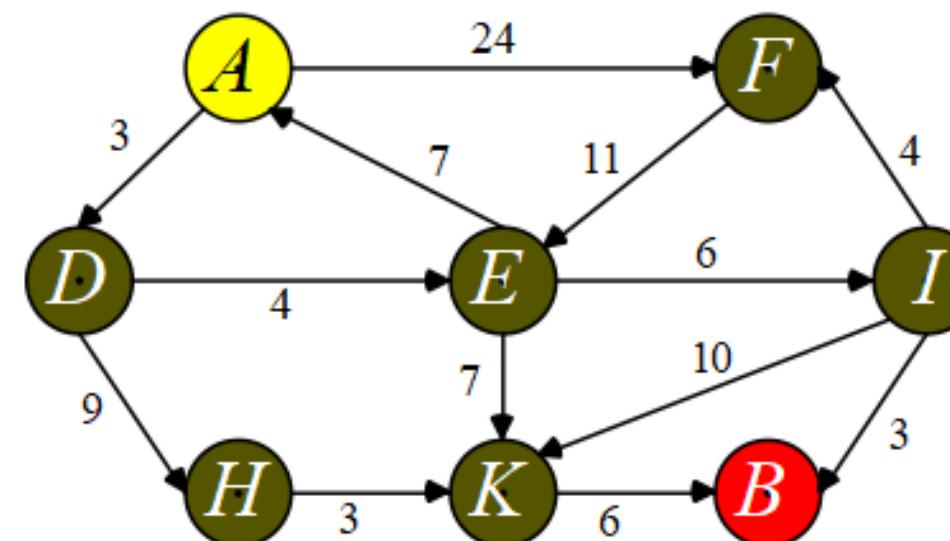
HÀM LƯỢNG GIÁ (H)							
A	B	D	E	F	H	I	K
13	0	2	8	9	7	6	4

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Nhánh - cận (Branch-and-Bound)

Bước	N	m	$g(m) = g(n) + \text{cost}(n,m)$	$f(m) = g(m) + h(m)$	$\Gamma(n) \uparrow$	OPEN	min
11	H		quay lại 2.1 (cắt bỏ nhánh con)	$f(H) = 19 > min$		F^{33}	16
12	F		quay lại 2.1 (cắt bỏ nhánh con)	$f(F) = 33 > min$		\emptyset	16
13	TRUE						

- Father(D,F) = A
- Father(E,H) = D
- Father(I,K,A) = E



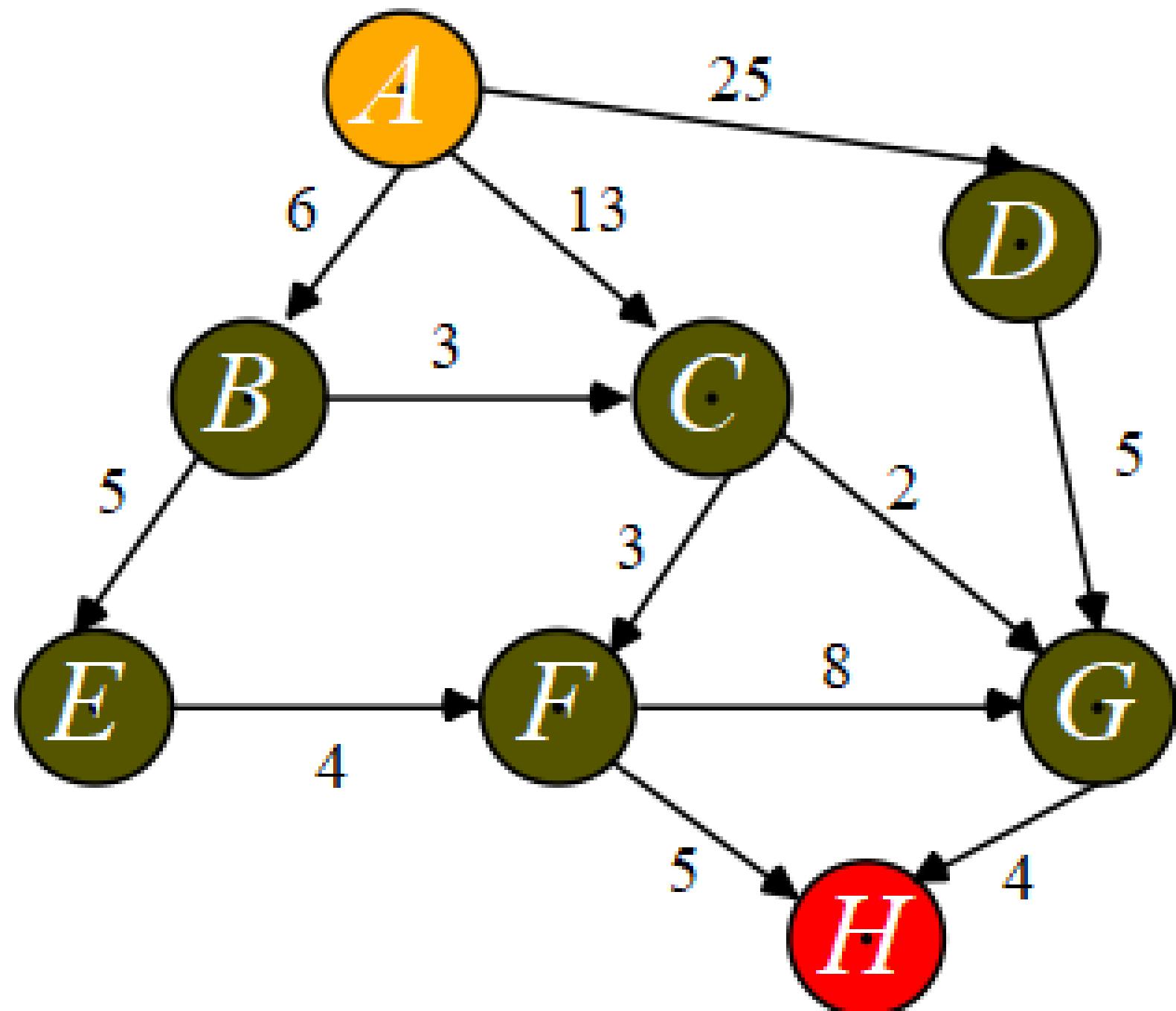
VET							
A	B	D	E	F	H	I	K
1	-	2	3	-	-	-	4

HÀM LƯỢNG GIÁ (H)							
A	B	D	E	F	H	I	K
13	0	2	8	9	7	6	4

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM HEURISTIC

- Nhánh - cận (Branch-and-Bound)

- Áp dụng thuật toán nhánh cận để tìm đường đi ngắn nhất từ đỉnh A → I, với các ước lượng heuristic của các trạng thái so với trạng thái đích được cho ở Bảng.



A	B	C	D	E	F	G	H
20	5	7	6	9	1	3	0

HẾT CHƯƠNG 4

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI



TRÍ TUỆ NHÂN TẠO *Artificial Intelligence*

Đoàn Vũ Thịnh
Khoa Công nghệ Thông tin
Đại học Nha Trang
Email: thinhdv@ntu.edu.vn

Nha Trang, 06-2023

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Thuật giải A* giải bài toán Puzzle 8

Khởi tạo

1	2	3
0	4	6
7	5	8

Kết thúc

1	2	3
4	5	6
7	8	0

Xây dựng hàm lượng giá: $f(n) = g(n) + h(n)$

Trong đó:

- $f(n)$: Hàm lượng giá heuristic tại trạng thái n
- $g(n)$: Khoảng cách từ n đến trạng thái bắt đầu
- $h(n)$: Khoảng cách từ trạng thái n đến mục tiêu

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Thuật giải A* giải bài toán Puzzle 8

Khởi tạo
src[]

1	2	3
0	4	6
7	5	8

Kết thúc
des[]

1	2	3
4	5	6
7	8	0

Hàm lượng giá h được tính:

- $h1+ = \text{abs}(\text{src}[i][j] - \text{des}[i][j])$: $h = |0 - 4| + |4 - 5| + |5 - 8| + |8 - 0| = 16$
- $h2$: số ô sai vị trí

Khoảng cách từ n đến trạng thái bắt đầu: $g++$ (sau mỗi bước giá trị khoảng cách (g) tự động tăng thêm 1 đơn vị. Do vậy giá trị của hàm f chỉ còn phụ thuộc vào giá trị của h.

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Thuật giải A* giải bài toán Puzzle 8

Khởi tạo
src[]

1	2	3
0	4	6
7	5	8

Kết thúc
des[]

1	2	3
4	5	6
7	8	0

0	2	3
1	4	6
7	5	8

$$\begin{aligned} h1 &= 16 \\ h2 &= 5 \end{aligned}$$

1	2	3
4	0	6
7	5	8

$$\begin{aligned} h1 &= 16 \\ h2 &= 3 \end{aligned}$$

1	2	3
7	4	6
0	5	8

$$\begin{aligned} h1 &= 22 \\ h2 &= 5 \end{aligned}$$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Thuật giải A* giải bài toán Puzzle 8

Khởi tạo
src[]

1	2	3
4	0	6
7	5	8

Kết thúc
des[]

1	2	3
4	5	6
7	8	0

1	2	3	1	2	3
0	4	6	4	0	6
7	5	8	7	5	8

CLOSE

∈CLOSE

1	0	3
4	2	6
7	5	8

h1=16
h2=3

1	2	3	1	2	3
0	4	6	4	6	0
7	5	8	7	5	8

h1=16
h2=4

1	2	3
4	5	6
7	0	8

h1=18
h2=4

h1=16
h2=2

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Thuật giải A* giải bài toán Puzzle 8

Khởi tạo
src[]

1	2	3
4	5	6
7	0	8

Kết thúc
des[]

1	2	3
4	5	6
7	8	0

1	2	3	1	2	3	CLOSE
0	4	6	4	0	6	
7	5	8	7	5	8	
1	2	3	1	2	3	
4	5	6	4	5	6	
7	0	8	7	0	8	

∈CLOSE

1	2	3	1	2	3
4	0	6	4	5	6
7	5	8	7	8	0

$h_1=16$
 $h_2=2$

$h_1=h_2 = 0$

Heuristic 1: Tổng số miếng sai vị trí
Heuristic 2: Tổng khoảng cách sai vị trí
của từng miếng.

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Thuật giải A* giải bài toán Puzzle 8

Khởi tạo

1	2	3
0	4	6
7	5	8

Kết thúc

1	2	3
4	5	6
7	8	0

Trò chơi Puzzle 8 số với số tổ hợp các trường hợp có thể say ra là $(3 \times 3)! = 362.880$ trường hợp.

Ta có thể giới hạn độ sâu của các nhánh sinh ra bằng phương pháp cắt cựt alpha-beta cho phép cắt bỏ những nhánh không cần thiết nhằm giảm bớt số trường hợp phải xét mà không ảnh hưởng đến kết quả đánh giá.

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Trò chơi 2 người đối kháng

- Xét một trò chơi trong đó 2 người thay phiên nhau đi nước của mình như cờ vua, cờ tướng, cờ caro
- Trò chơi có một trạng thái bắt đầu và mỗi nước đi sẽ biến đổi trạng thái hiện hành thành một trạng thái mới.
- Trò chơi sẽ kết thúc theo một quy định nào đó, nghĩa là cuộc chơi sẽ dẫn đến một trạng thái phản ánh có một người chơi thắng cuộc hoặc một trạng thái không có ai thắng cuộc (trạng thái hòa).
- Phân tích xem từ một trạng thái nào đó sẽ dẫn đến đấu thủ nào sẽ thắng với điều kiện cả 2 đấu thủ đều sẽ đi những nước đi tốt nhất.

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Trò chơi 2 người đối kháng

- Mỗi trò chơi có thể biểu diễn bởi một cây trò chơi
- Mỗi nút cây biểu diễn cho một trạng thái
- Nút gốc biểu diễn trạng thái bắt đầu của trò chơi
- Mỗi nút lá biểu diễn cho một trạng thái kết thúc của trò chơi (thắng, thua, 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 nước đi có thể xuất phát từ trạng thái X.

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Trò chơi 2 người đối kháng

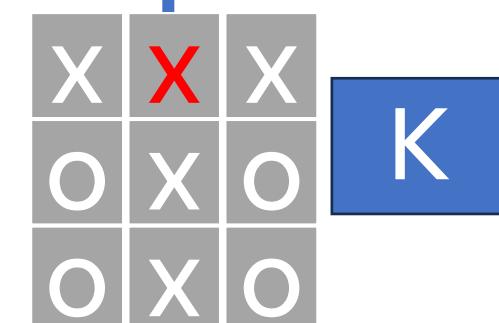
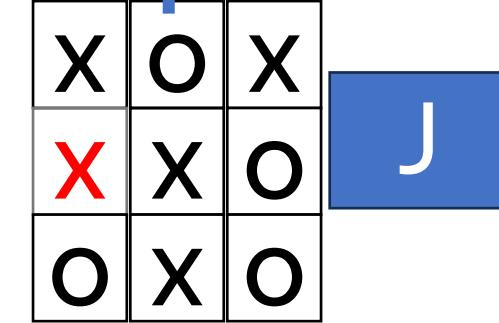
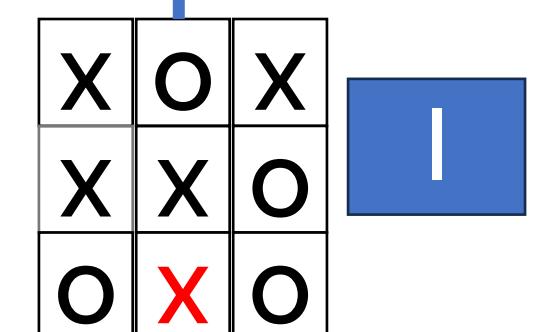
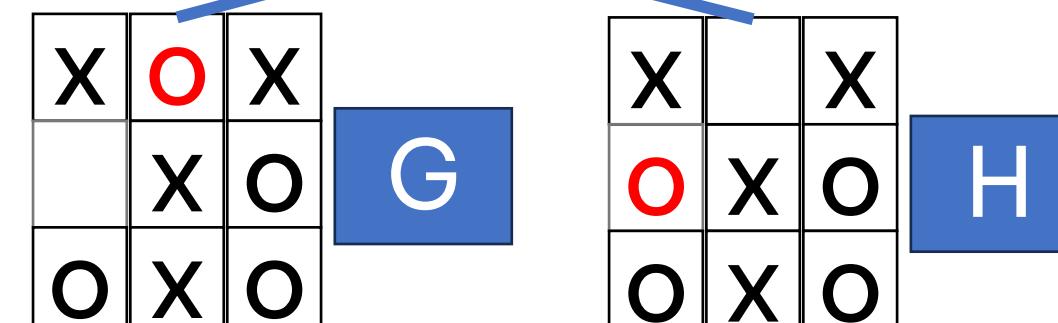
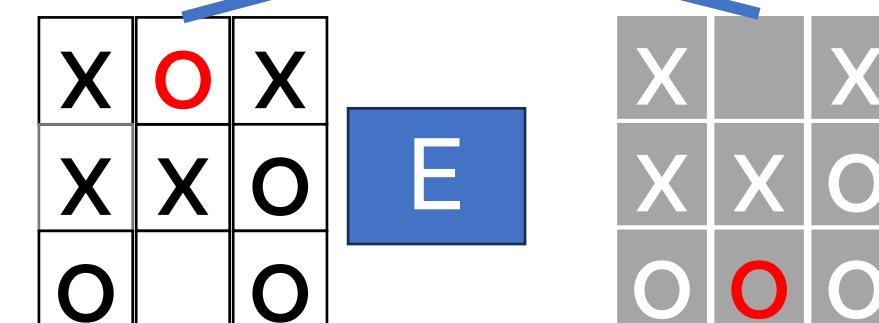
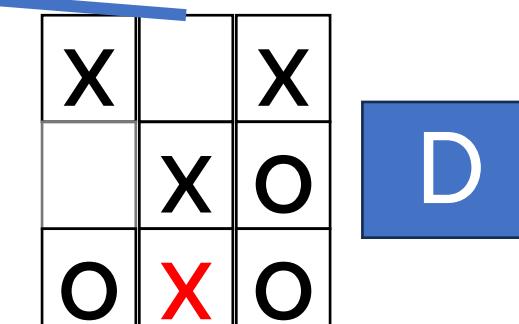
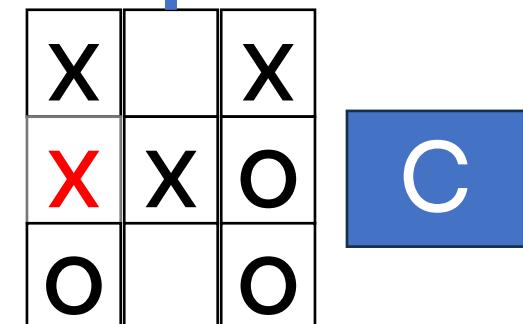
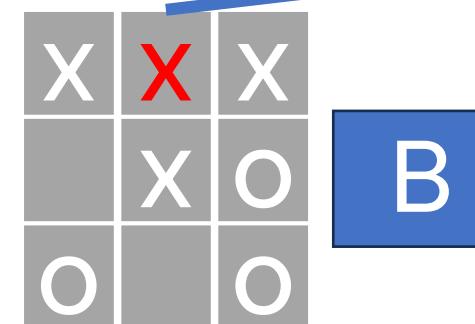
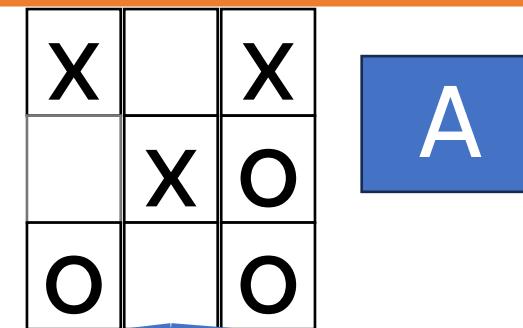
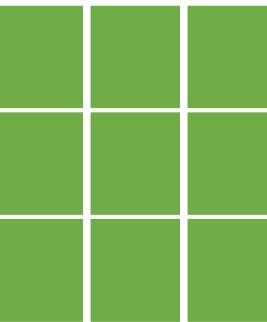
X-đi
MAX

O-đi
MIN

X-đi
MAX

O-đi
MIN

ROOT



Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

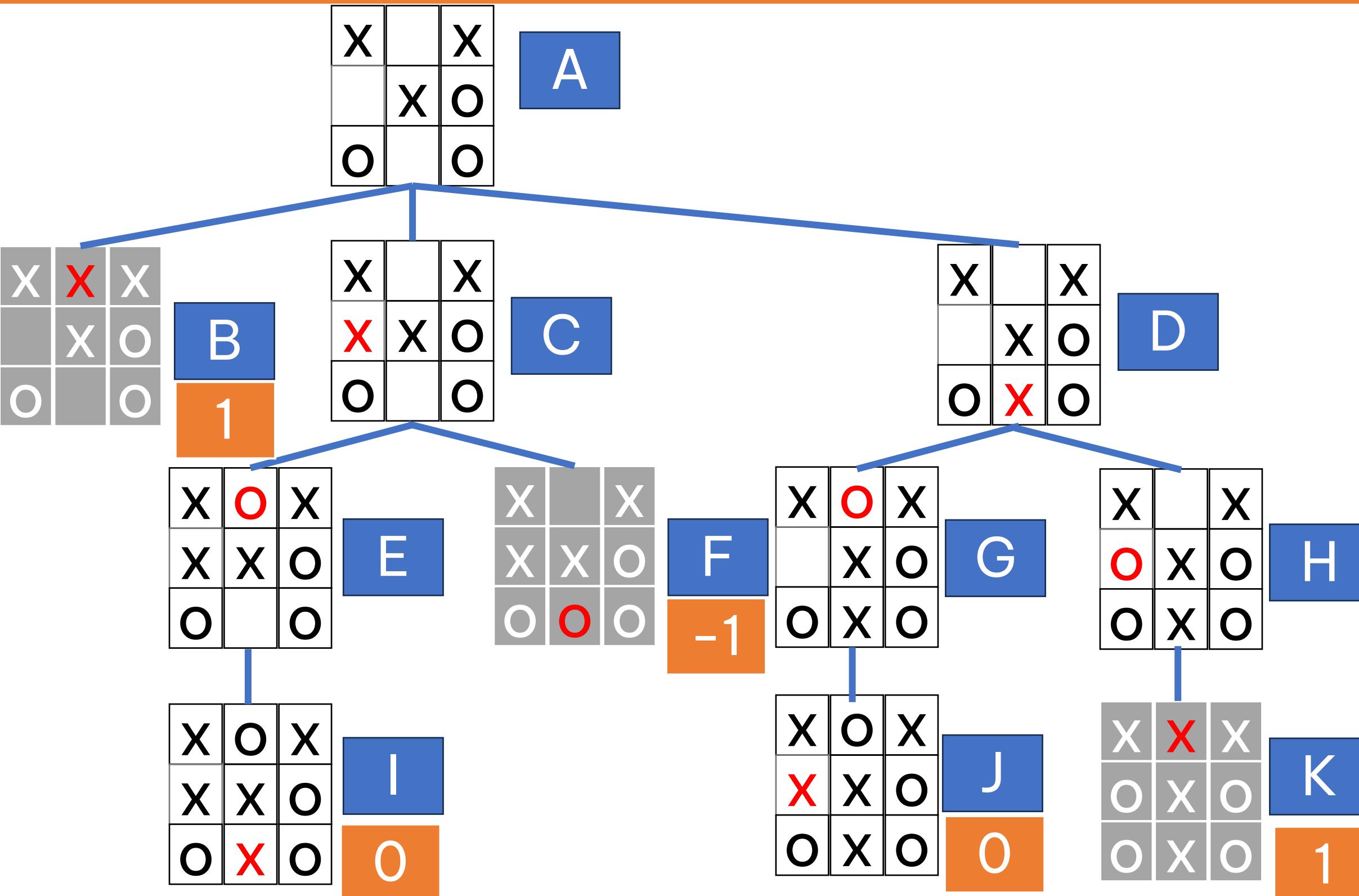
Trò chơi 2 người đối kháng

X-đi
MAX

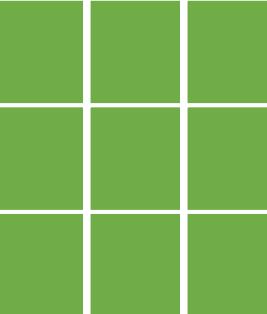
O-đi
MIN

X-đi
MAX

O-đi
MIN



ROOT



- X: nút cha: 9 k.năng
- O: nút con: 8 k.năng
- X: nút cha: 7 k.năng
- ...
- Tổng số: $9!$ k.năng
- Nguyên tắc gán giá trị cho nút:
 - Nút lá nào thắng: 1
 - Nút lá nào thua: -1
 - Nút lá nào hòa: 0

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

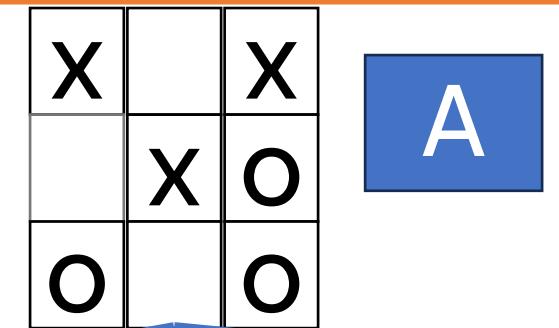
Trò chơi 2 người đối kháng

X-đi
MAX

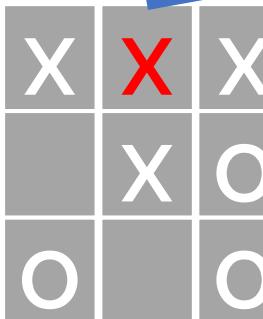
O-đi
MIN

X-đi
MAX

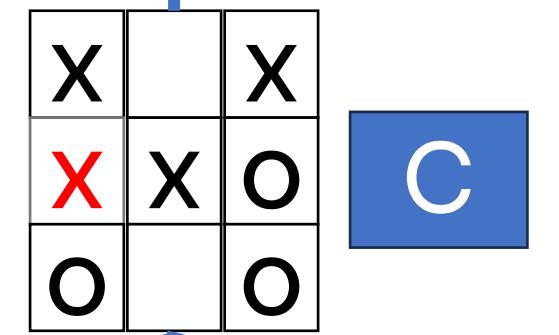
O-đi
MIN



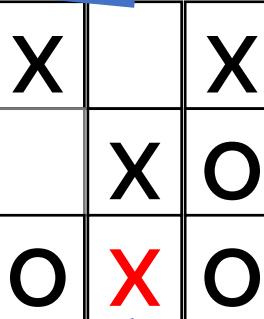
A



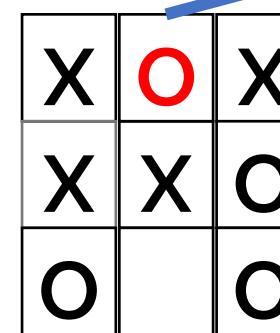
B



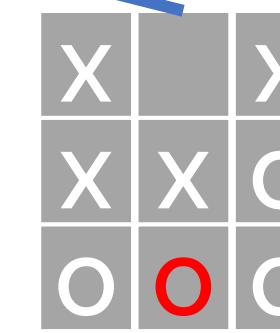
C



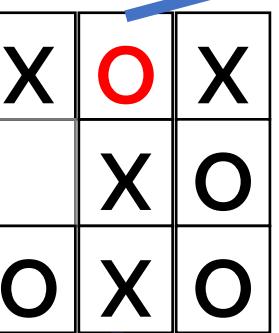
D



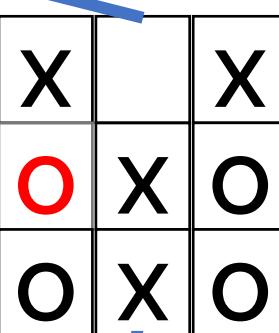
E



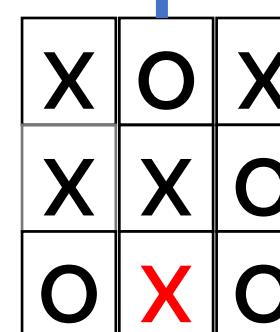
F



G



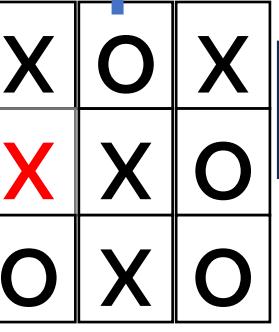
H



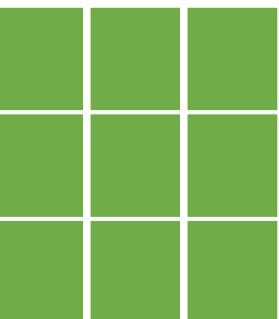
I



J



K



ROOT

- X: muốn thắng cần hướng đến các nút có giá trị 1 hay MAX
- O: muốn thắng cần hướng đến các nút có giá trị -1 hay MIN

- Nguyên tắc định trị:
- Nút lá: luôn có g.trị

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

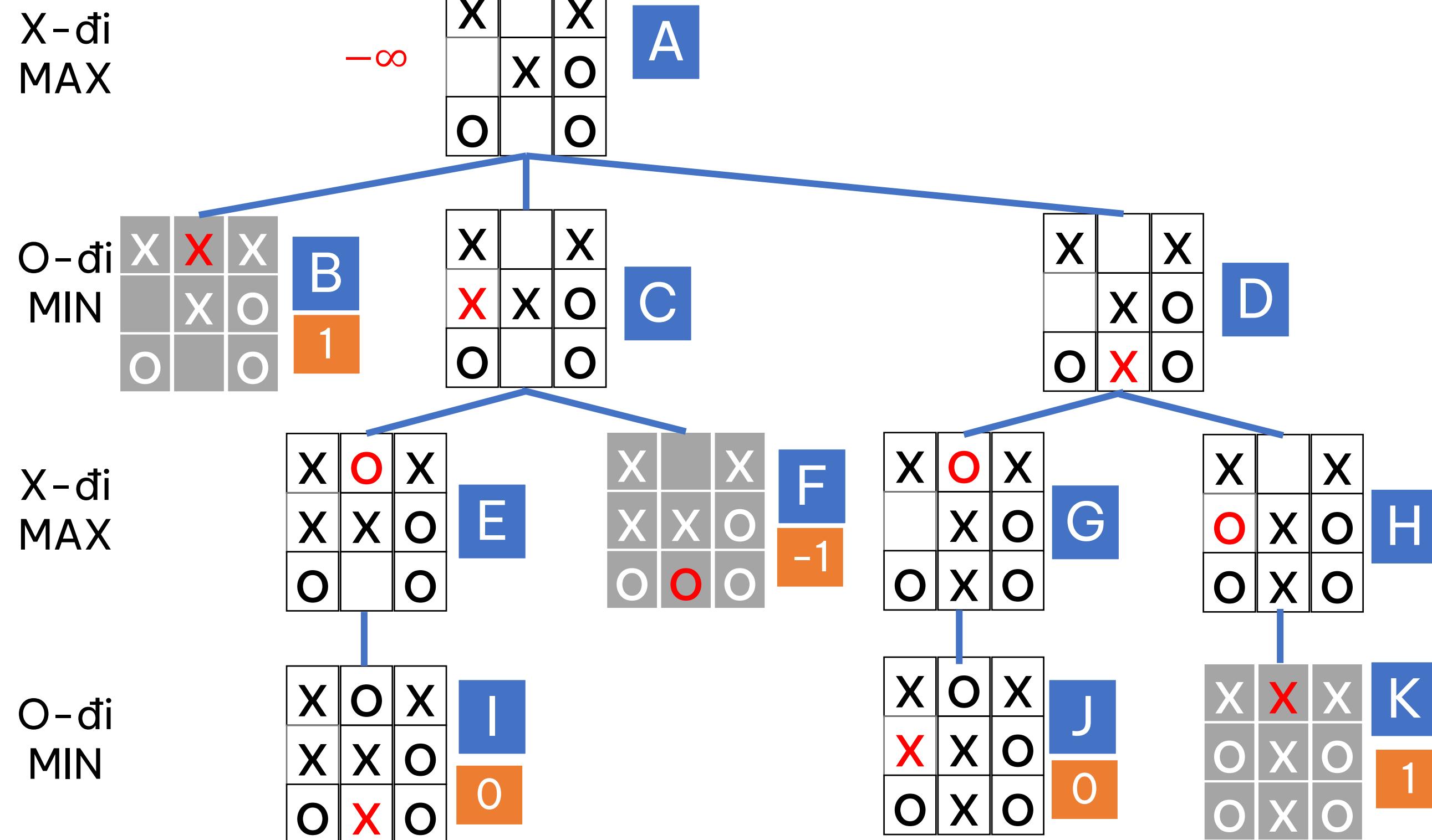
Trò chơi 2 người đối kháng

Kỹ thuật vét cạn (MIN-MAX)

- Giá trị nút MAX là giá trị lớn nhất của các nút con
- Giá trị nút MIN là giá trị nhỏ nhất của các nút con
- Bài toán tìm MAX hoặc MIN của dãy số (giải thuật vét cạn)
- Có 2 cách giải:
 - Cách 1: MAX = MIN = A0, duyệt A1-An tìm MAX, MIN
 - Cách 2: MAX = $-\infty$ và MIN = $+\infty$
- Áp dụng cách 2 cho bài toán: Mỗi nút sẽ có một giá trị tạm: Nút MAX có giá trị tạm = $-\infty$ và nút MIN có giá trị tạm $+\infty$
- Khi tìm thấy giá trị của một nút con thì tính lại giá trị tạm của nút cha: giá trị tạm mới của nút MAX = LN(giá trị tạm cũ, giá trị nút con) VÀ giá trị tạm mới của nút MIN = NN(giá trị tạm cũ, giá trị của con)

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

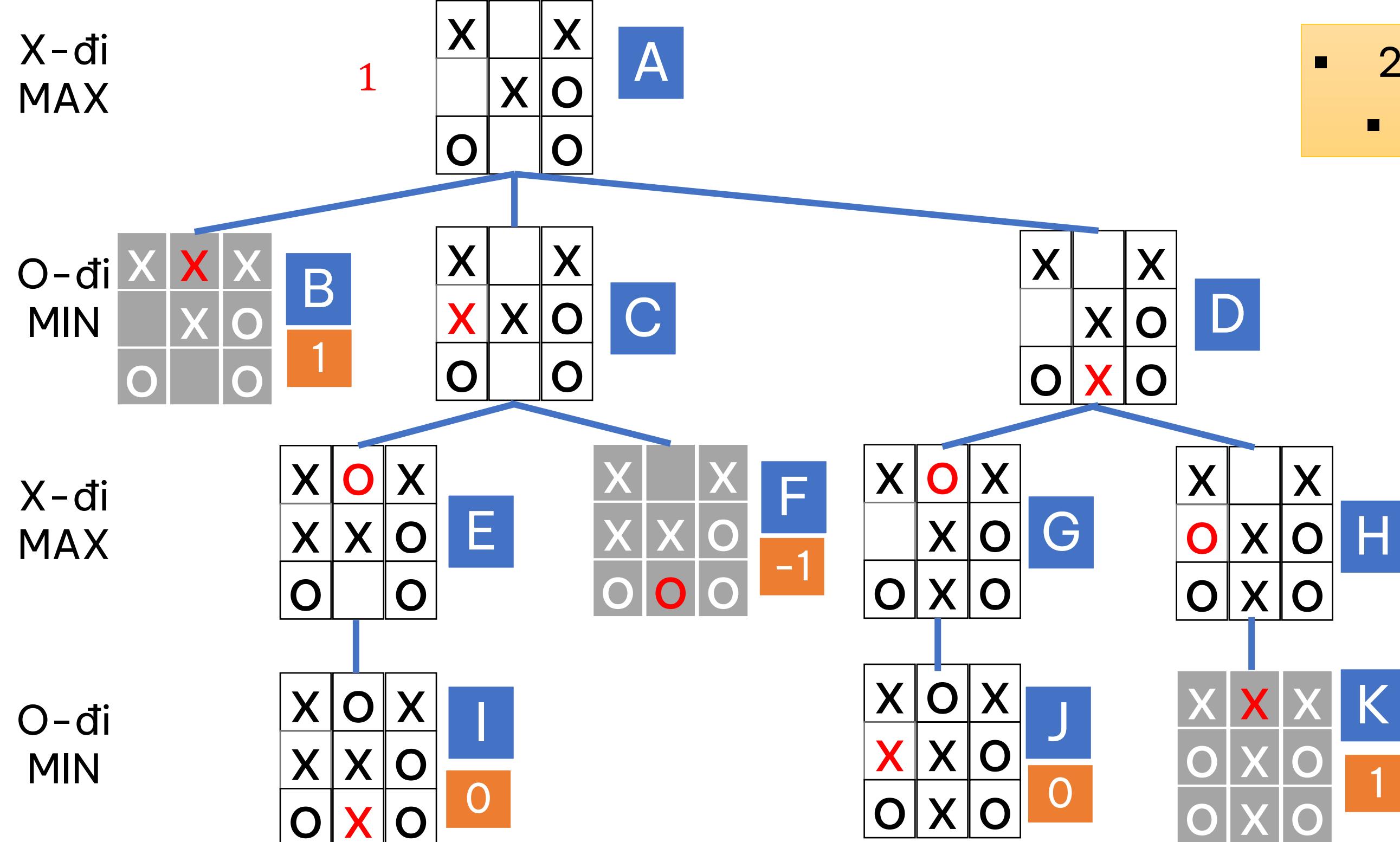
Kỹ thuật vét cạn (MIN-MAX)



- 1. Xét nút A: tạm A = $-\infty$ (vì A là MAX)

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

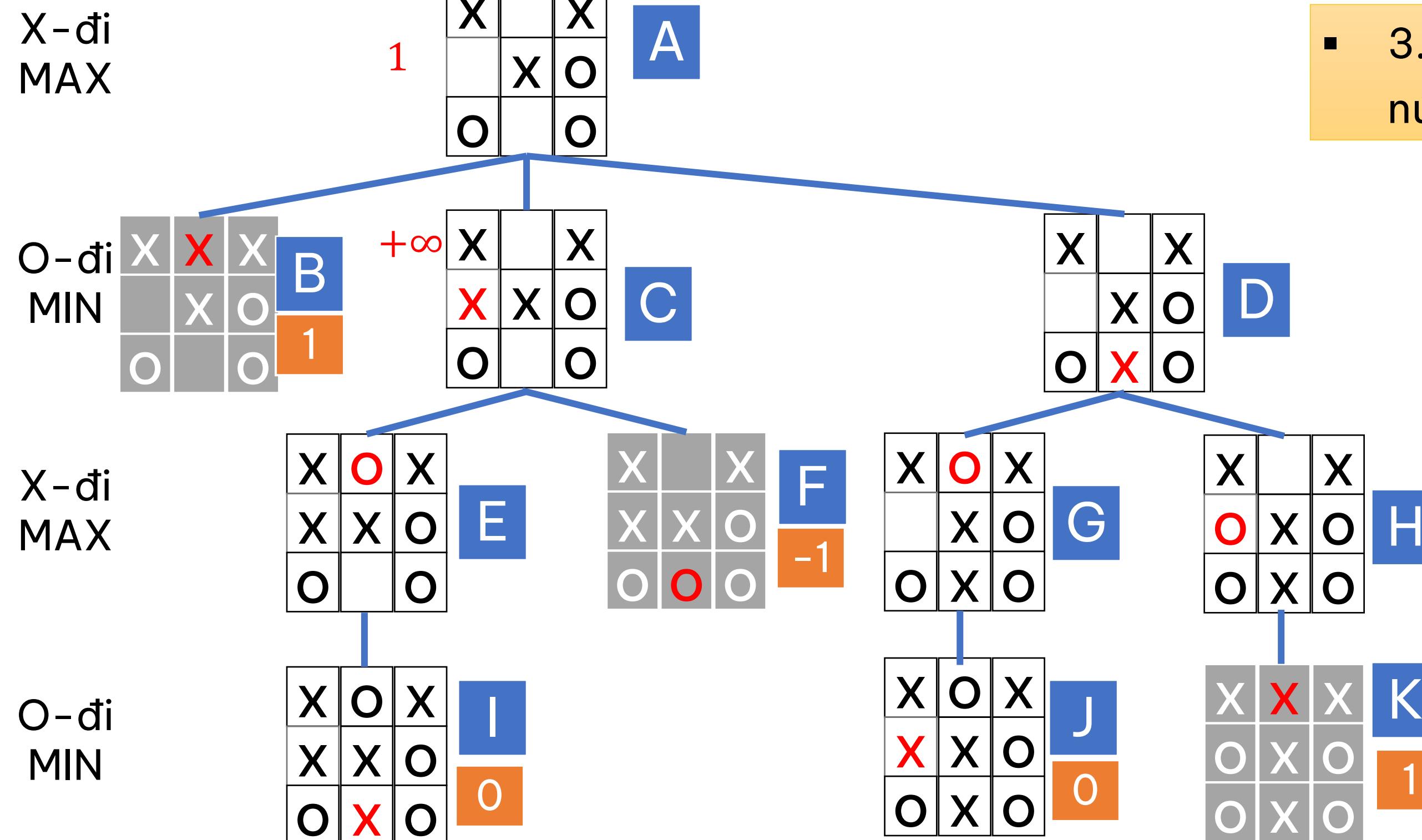
Kỹ thuật vét cạn (MIN-MAX)



- 2. Xét nút B (con A, nút lá): $\text{valB} = 1$
 - tạm A = $\text{MAX}(\text{tạm A}, 1) = 1$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

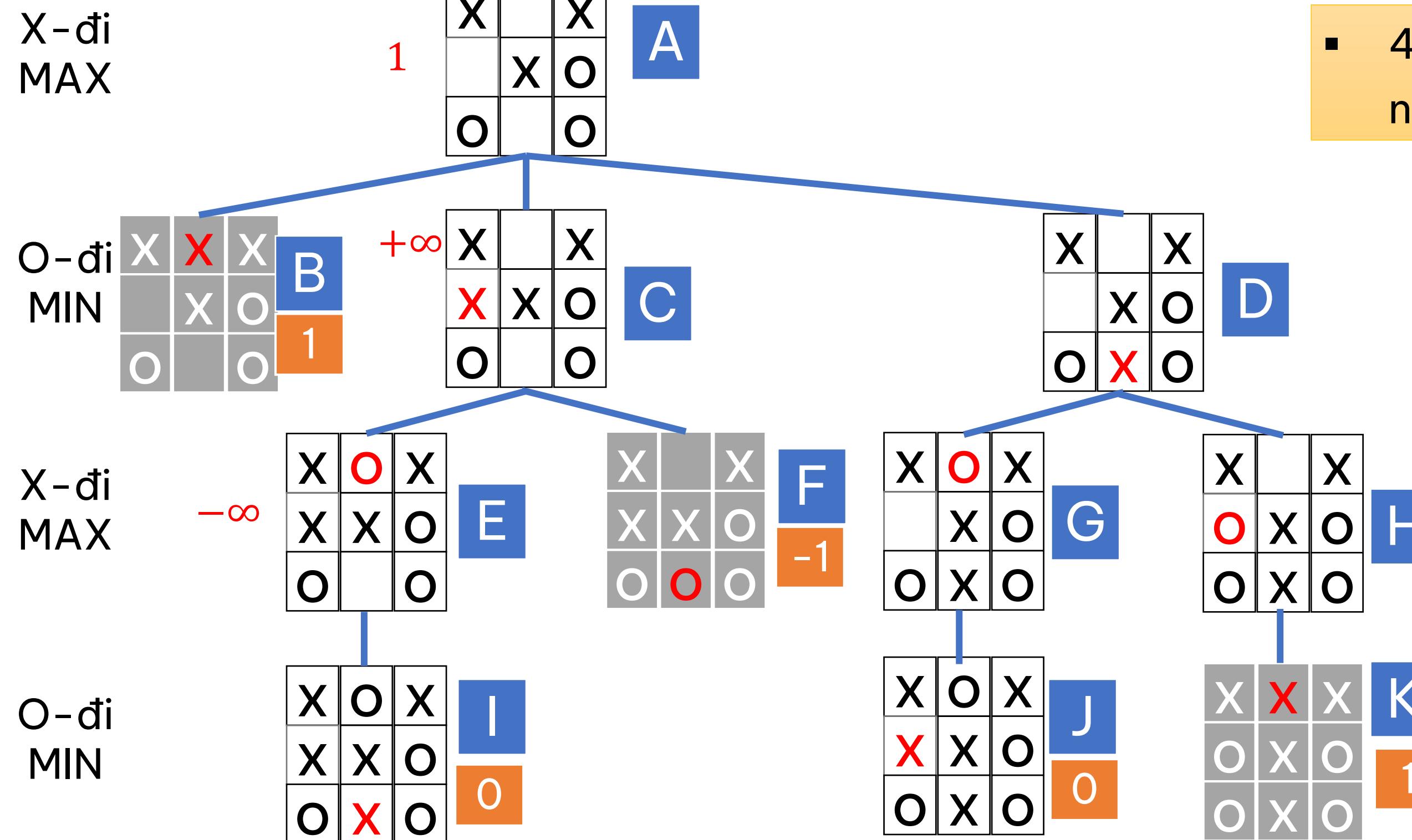
Kỹ thuật vét cạn (MIN-MAX)



- 3. Xét nút C (con A, chưa phải là lá, nút MIN): tạm C = $+\infty$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

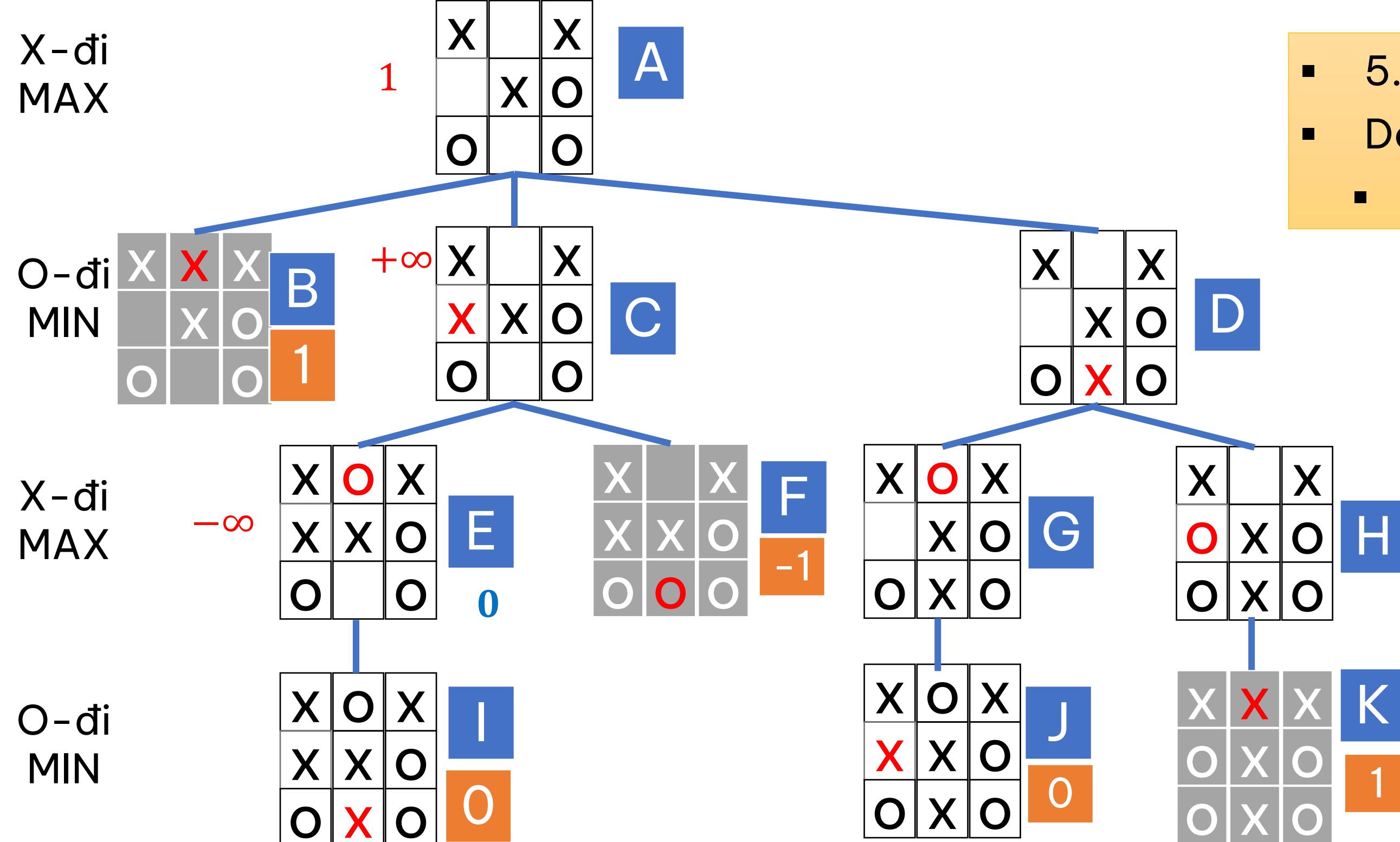
Kỹ thuật vét cạn (MIN-MAX)



- 4. Xét nút E (con C, chưa phải là lá, nút MAX): tạm $E = -\infty$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

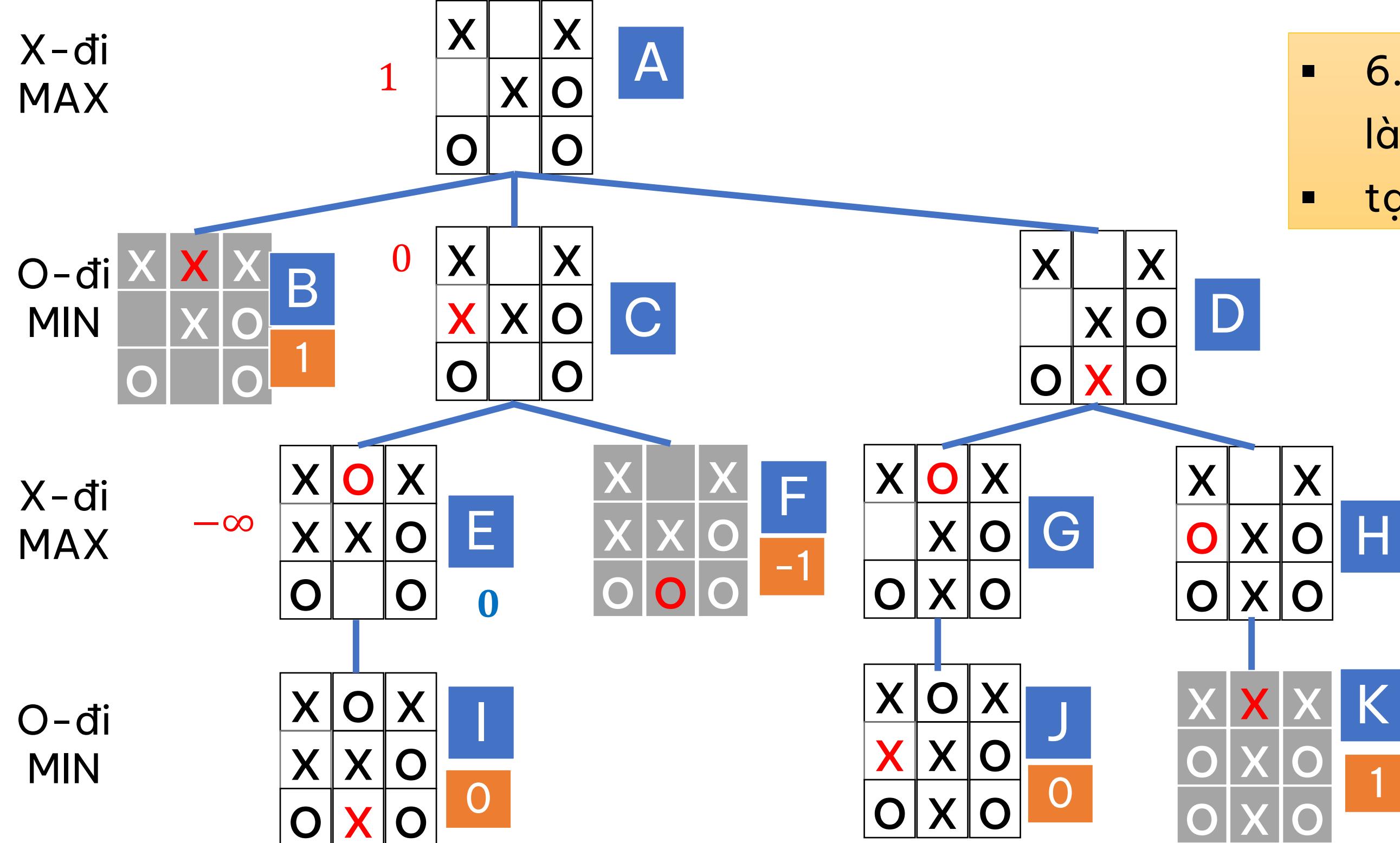
Kỹ thuật vét cạn (MIN-MAX)



- 5. Xét nút I (nút lá, con E): 0
- Do E có 1 nút con (I): xét xong
- valE = max(tạm E, 0) = 0

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

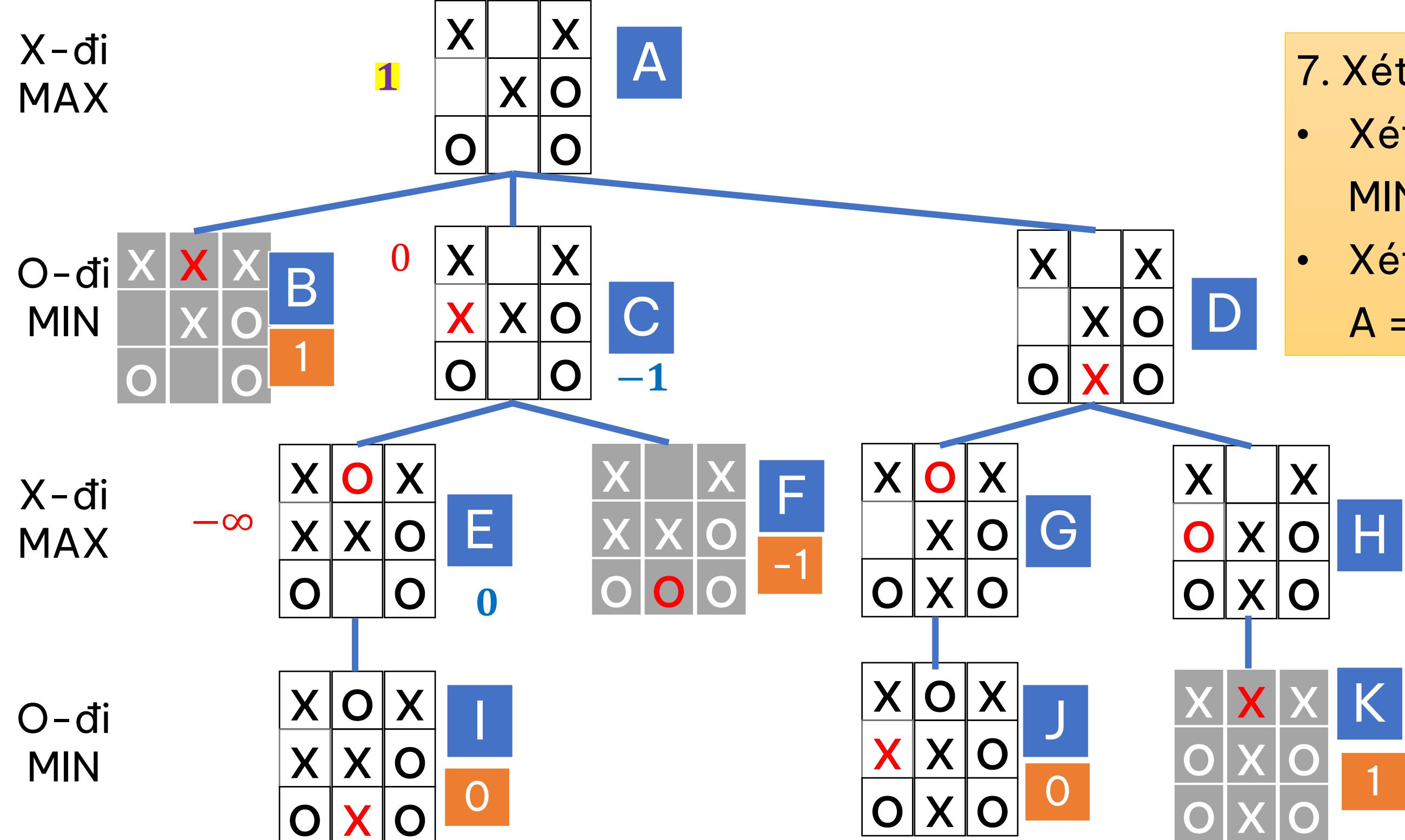
Kỹ thuật vét cạn (MIN-MAX)



- 6. Xét nút C (nút MIN, E con C, chưa là nút lá):
- tạm C = MIN(tạm C, 0) = 0

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

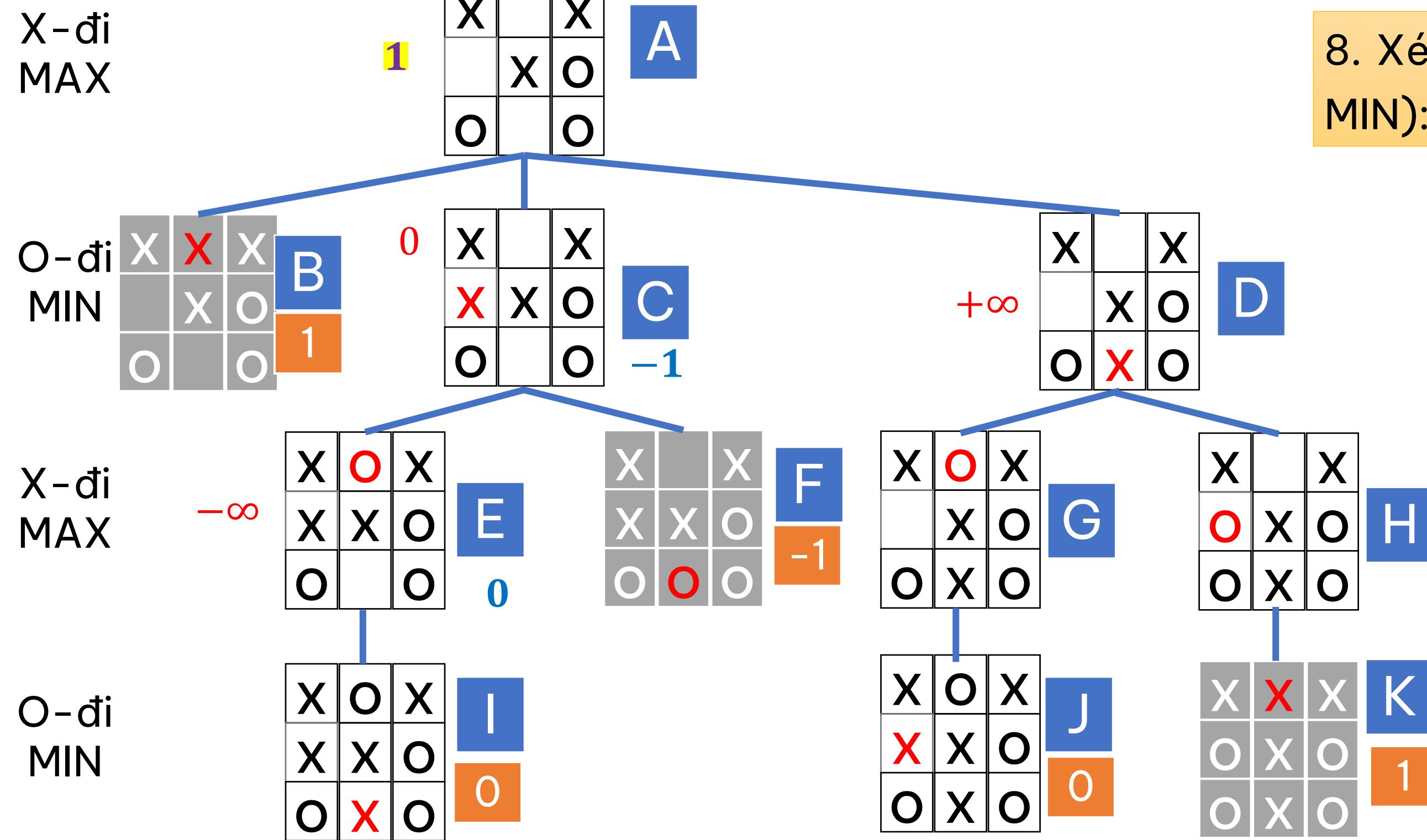
Kỹ thuật vét cạn (MIN-MAX)



7. Xét nút F (nút lá, F con C): -1
- Xét nút cha của F=C, có giá trị = $\text{MIN}(\text{tạm C}, -1) = -1$
 - Xét nút cha của C=A, có giá trị tạm A = $\text{MAX}(\text{tạm A}, -1) = 1$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

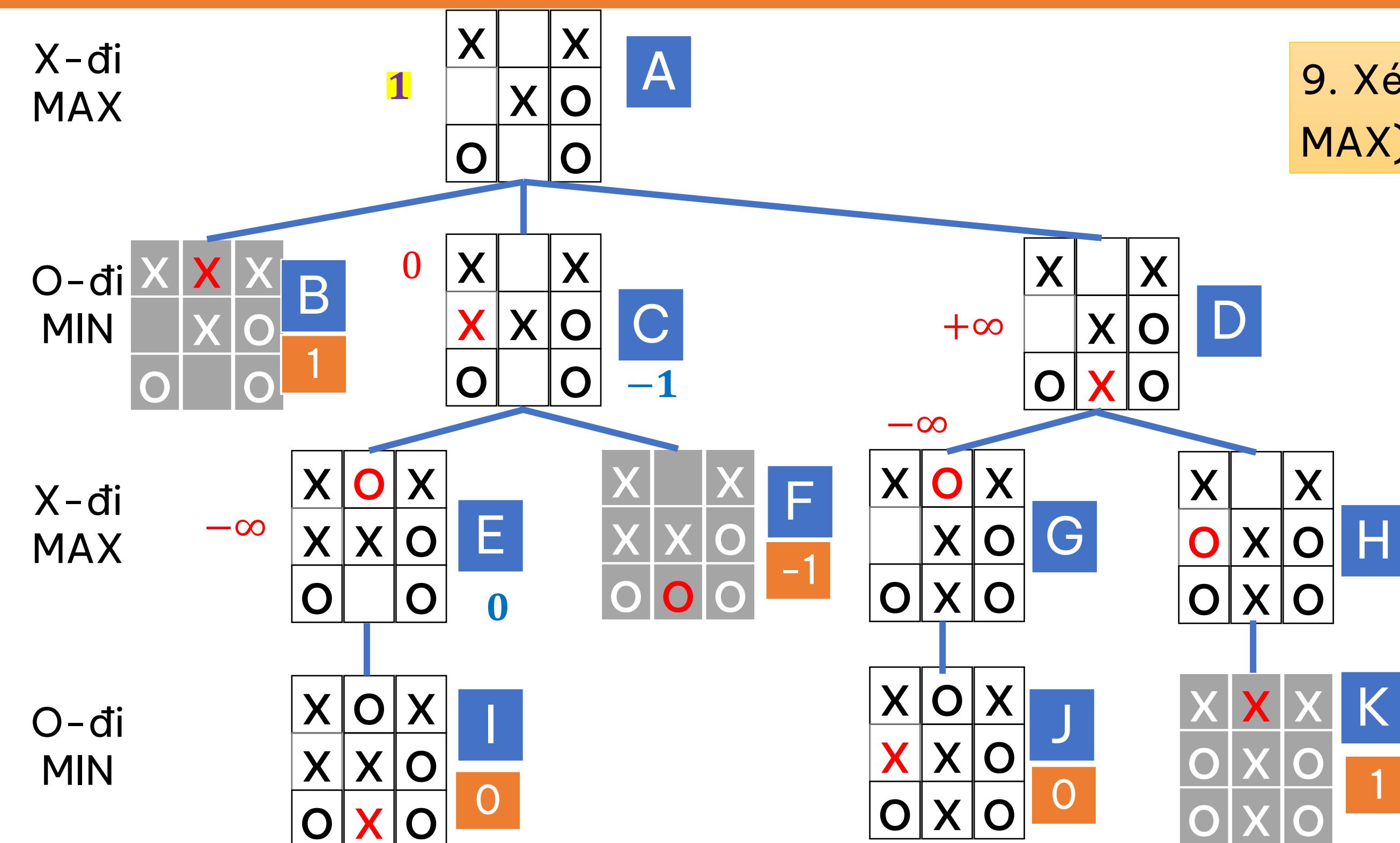
Trò chơi 2 người đối kháng



8. Xét nút D (con A, chưa là nút lá, nút MIN): tạm $D = +\infty$

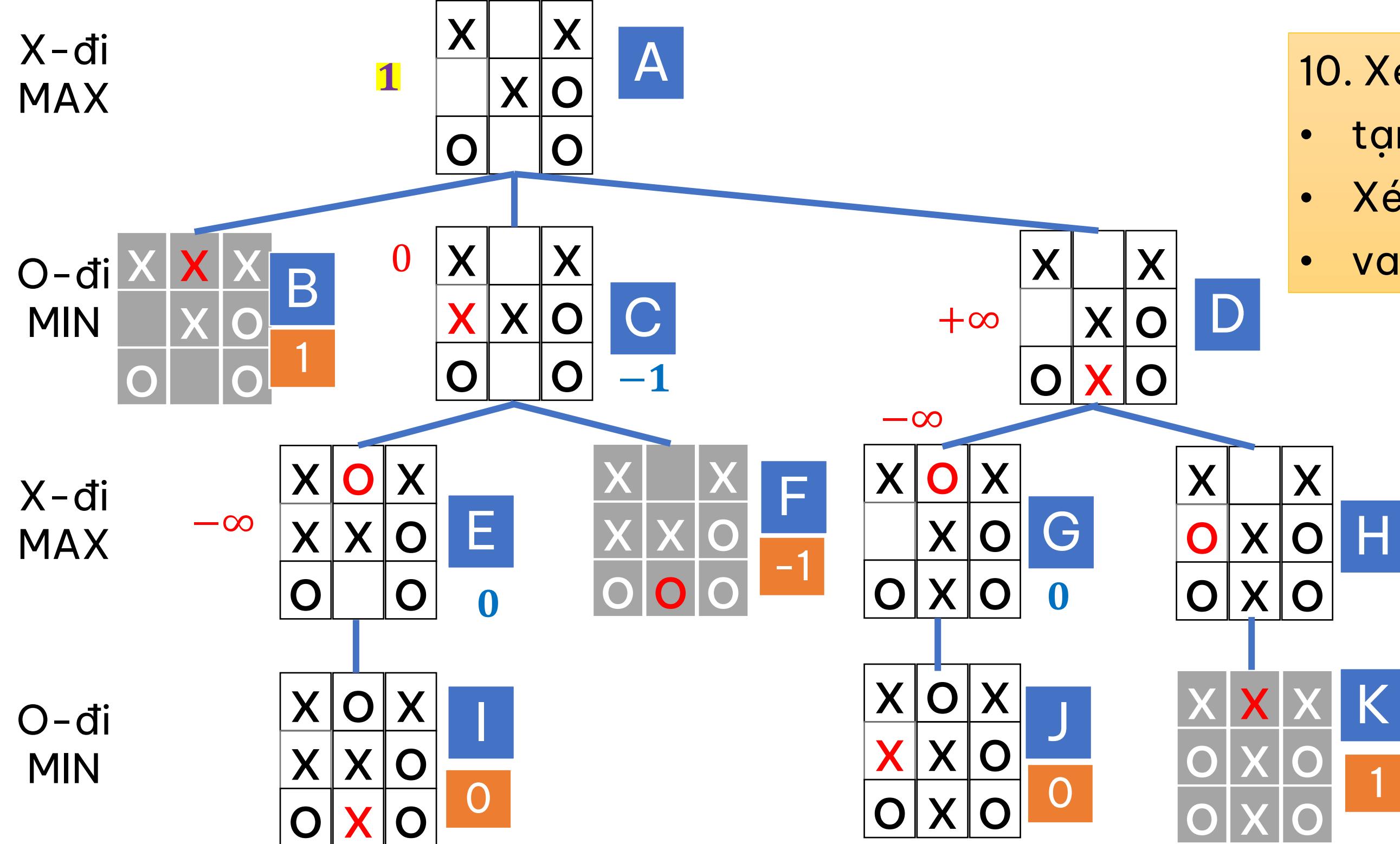
Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Kỹ thuật vét cạn (MIN-MAX)



Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Kỹ thuật vét cạn (MIN-MAX)



10. Xét nút J (con G, nút lá): 0

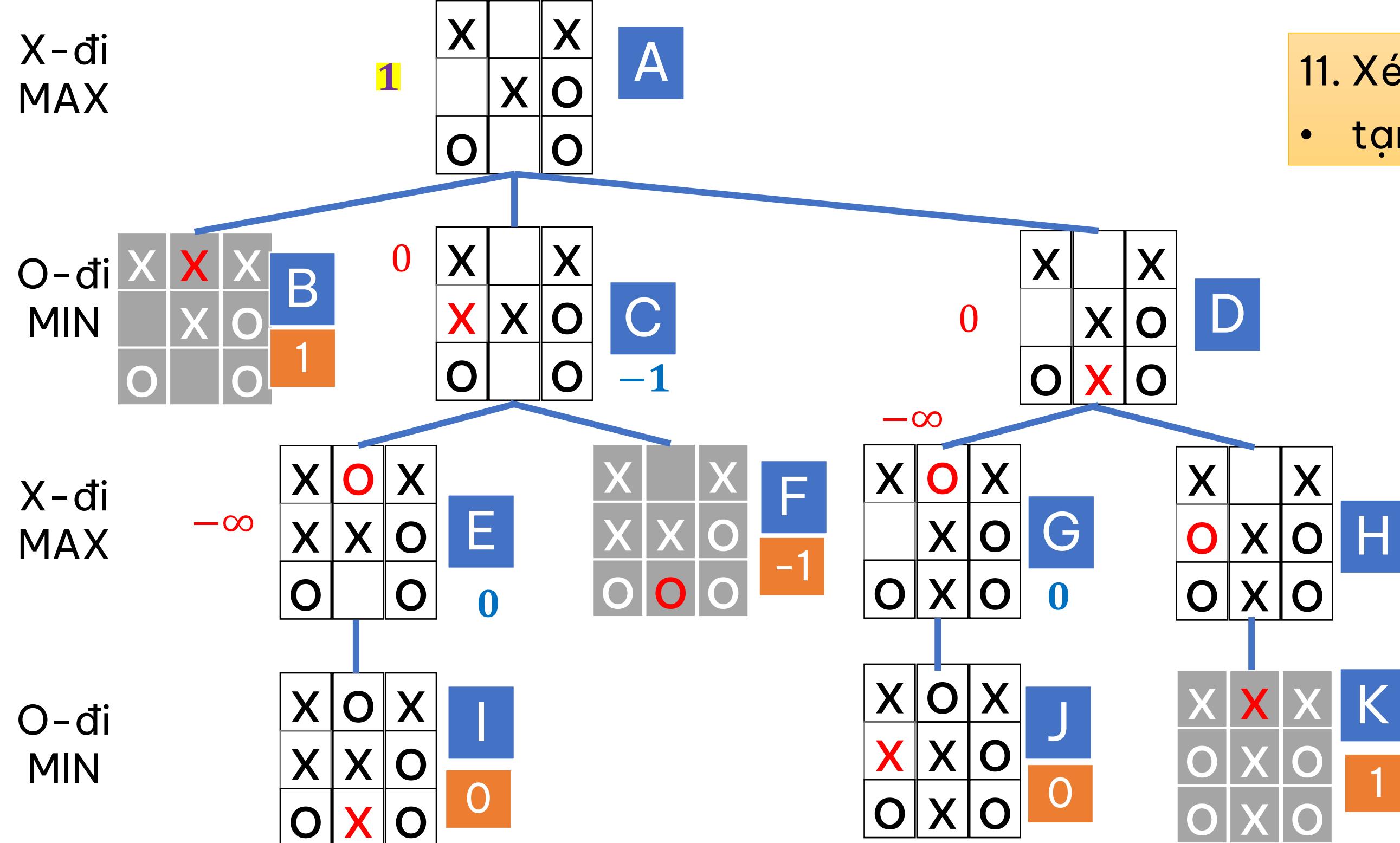
- tạmG=MAX(tạm G, 0) = 0

- Xét nút G cha J: xét xong (hết con)

- valG = tạm G

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

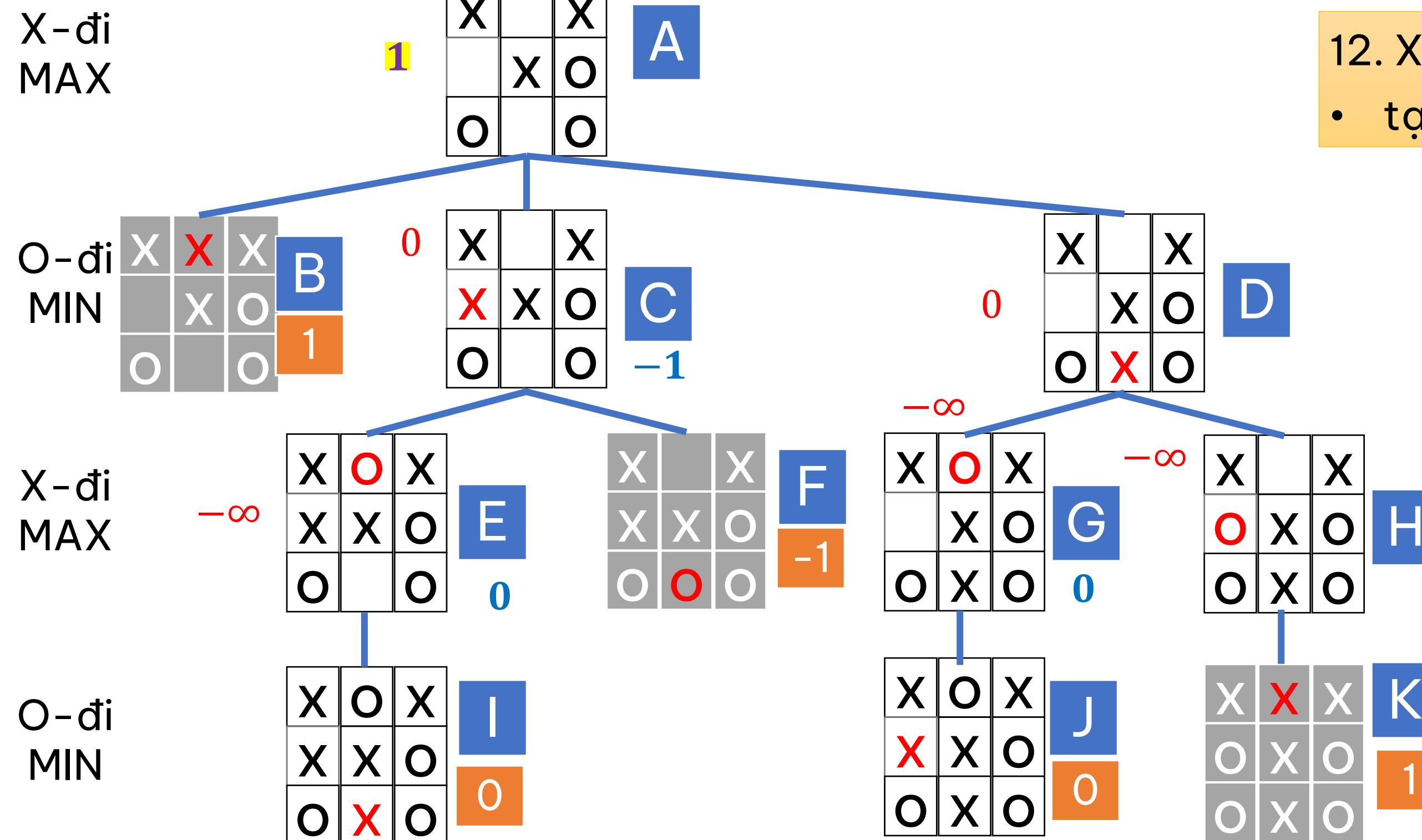
Kỹ thuật vét cạn (MIN-MAX)



11. Xét nút D (chưa là nút lá, nút MIN):
• tạmD=MIN(tạm D, 0) = 0

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

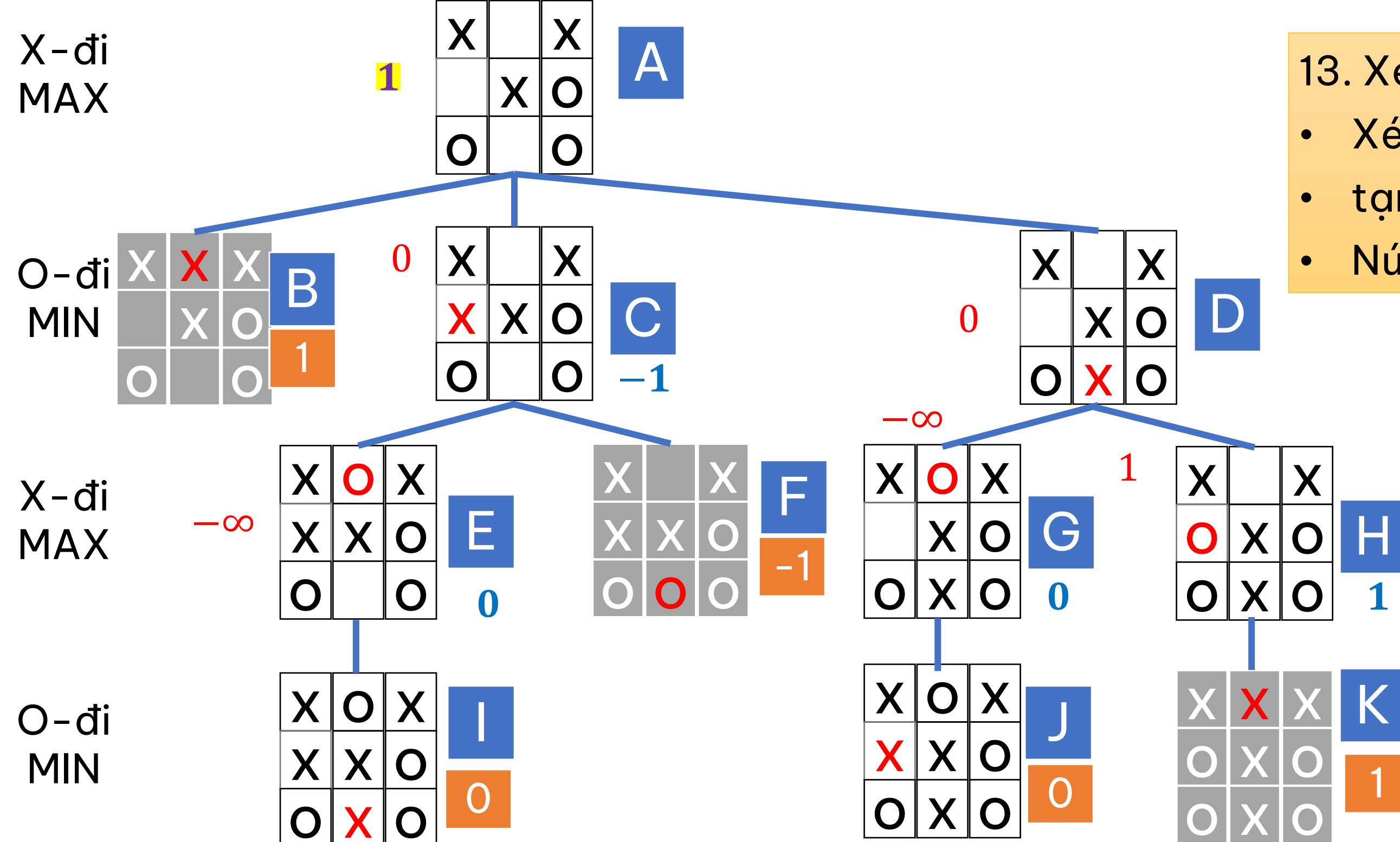
Kỹ thuật vét cạn (MIN-MAX)



12. Xét nút H (chưa là nút lá, nút MAX):
• tạmH = -∞

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Kỹ thuật vét cạn (MIN-MAX)

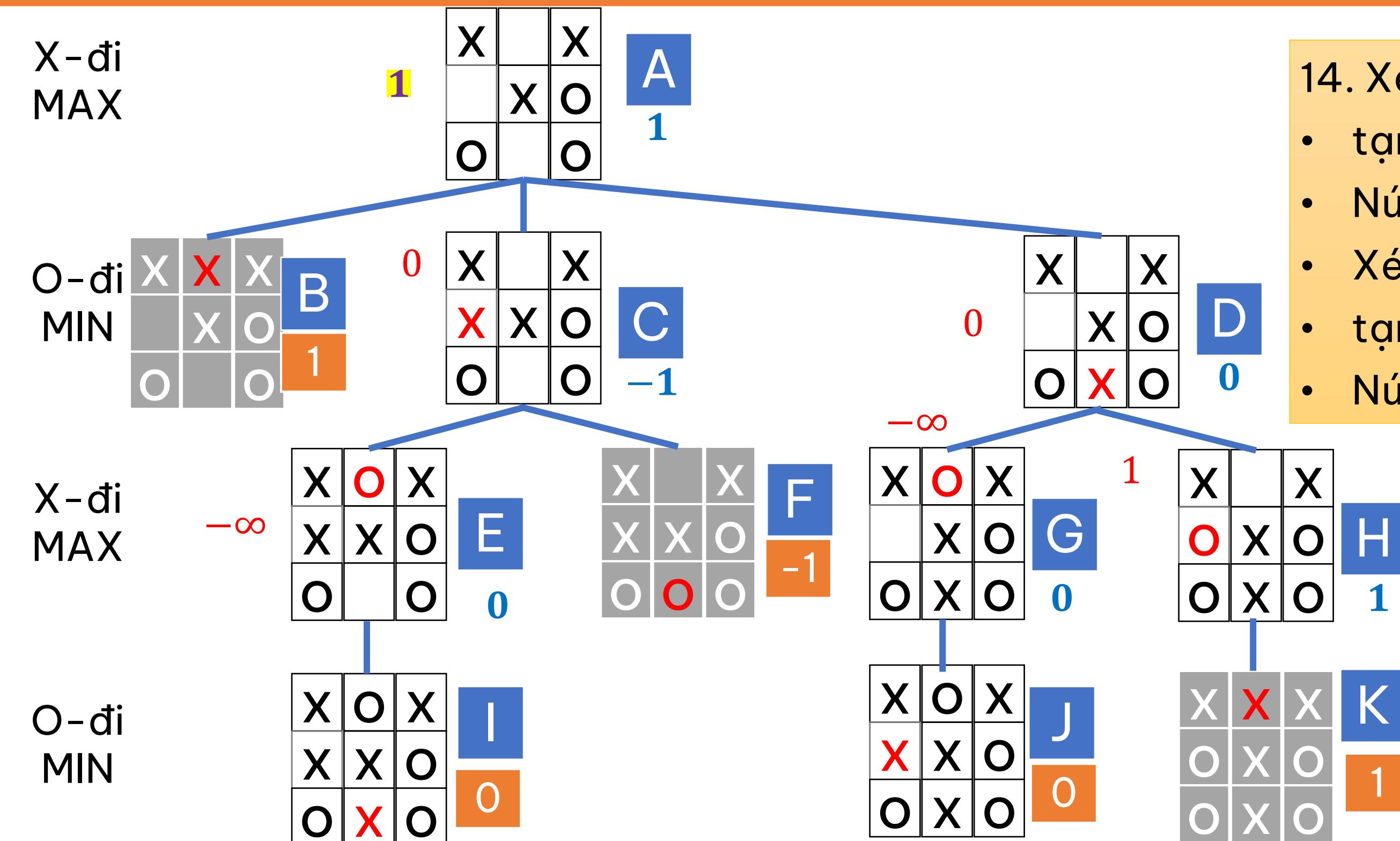


13. Xét nút K (nút lá): 1

- Xét nút H cha K:
- tạm $H = \text{MAX}(\text{tạm } H, 1) = 1$
- Nút H hết con: $\text{val } H = \text{tạm } H = 1$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

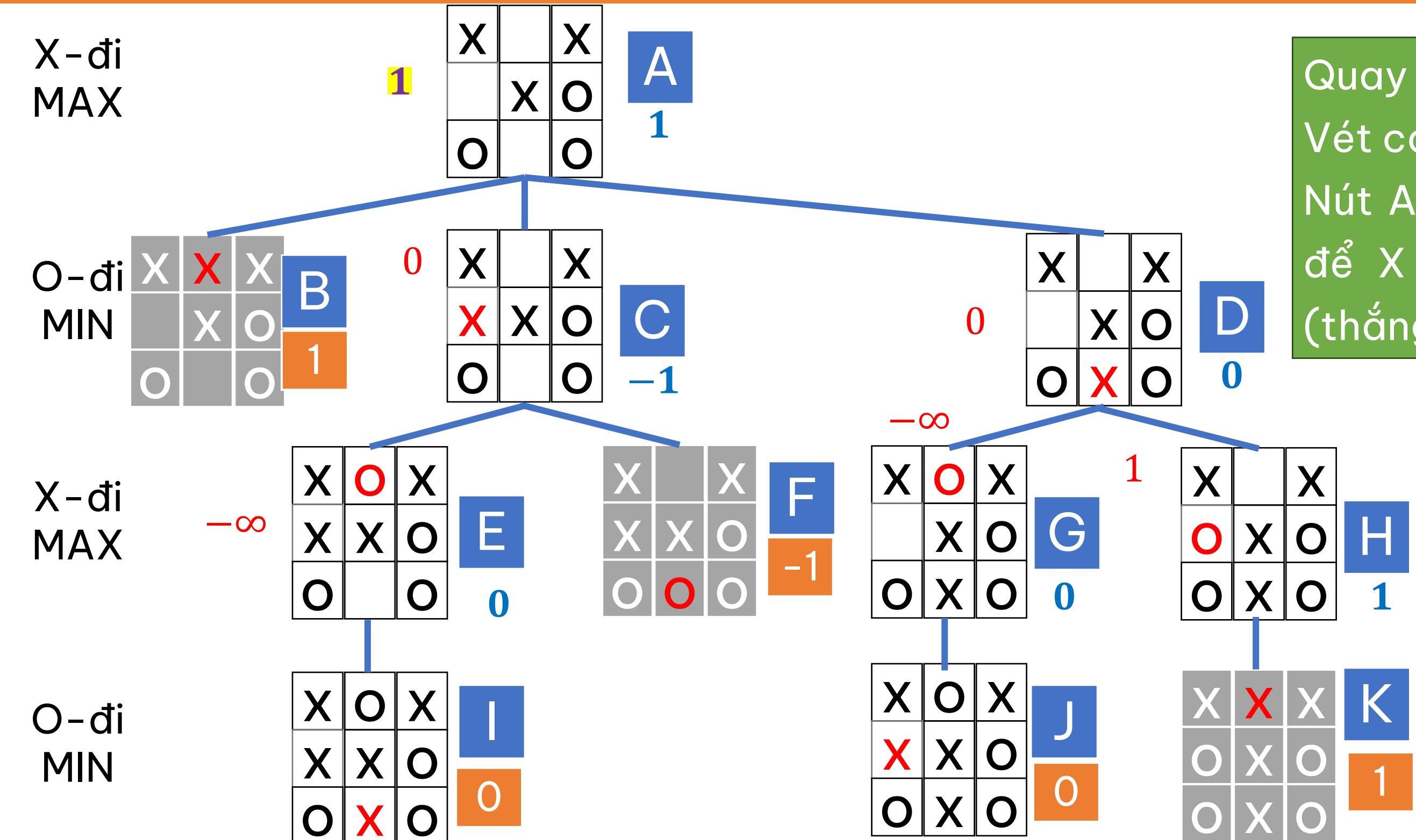
Kỹ thuật vét cạn (MIN-MAX)



14. Xét nút D (chưa là lá, nút MIN):
- tạm D = MIN(tạm D, 1)=0
 - Nút D hết con: val D = tạm D = 0
 - Xét nút A (chưa là lá, nút MAX):
 - tạm A = MAX(tạm A, 1) = 1
 - Nút A hết con: val A = tạm A

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Kỹ thuật vét cạn (MIN-MAX)



Quay lui: Xét nút lá → xét nút cha

Vét cạn: Xét hết tất cả các nút

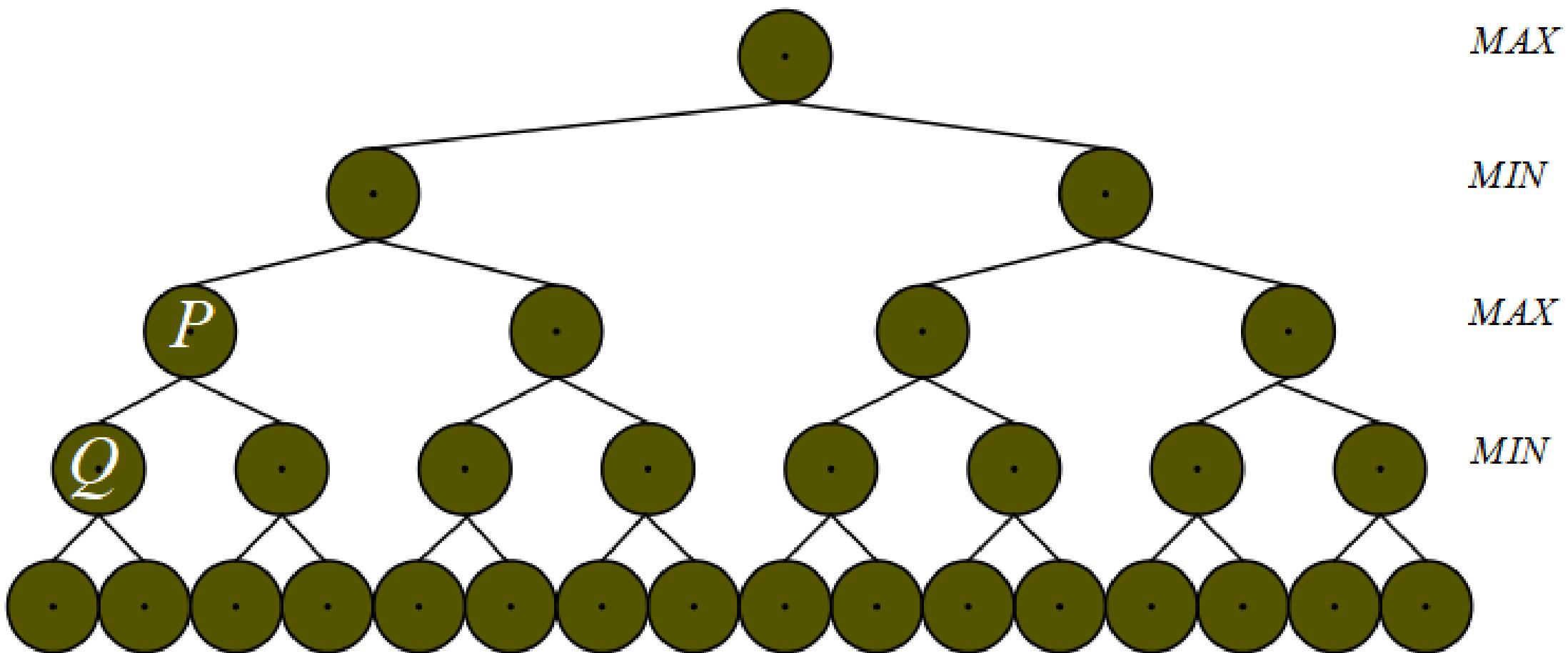
Nút A ban đầu có giá trị là 1 (hướng đi để X chiến thắng). Đường đi theo B (thắng), D(hòa), C(thua)

Từ kết quả này ta nhận thấy việc xét tất cả các nút là không cần thiết.
Có cách nào để hạn chế các nút/nhánh

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Xém tỉa $\alpha - \beta$ (Alpha - Beta pruning)

- P là nút đang xét và đã có 1 số nút con đã xét và một số con chưa xét.
- Q là nút đang xét ($Q \subset P$)
- Gọi V_P, V_Q : giá trị tạm P, Q
- Nếu P là MAX thì Q là MIN (MIN - MAX đan xen nhau)



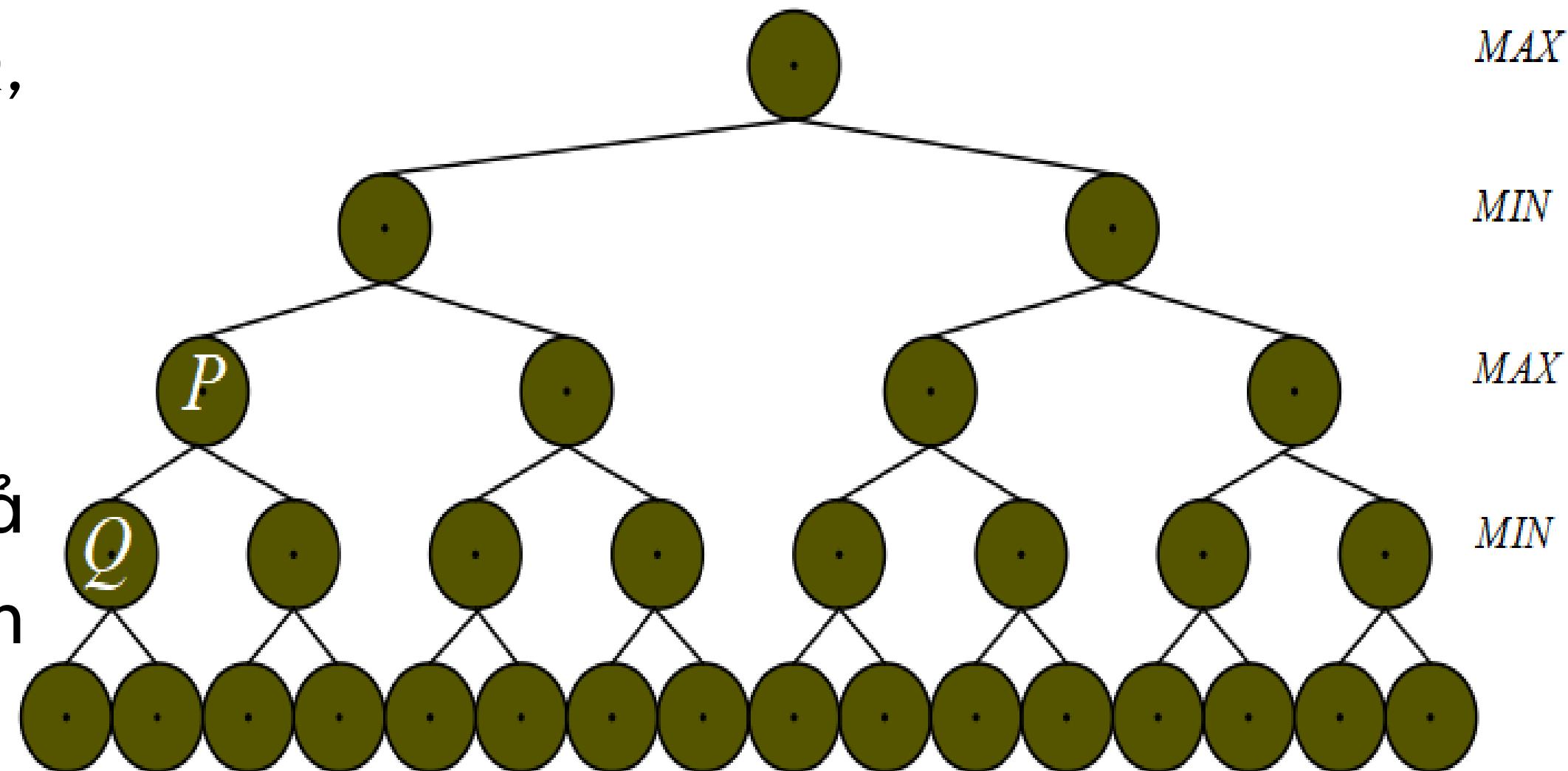
- Nếu $V_P \geq V_Q$: cắt tỉa các con chưa xét của Q (cắt tỉa α)
- Nếu P là nút MIN thì Q sẽ là nút MAX: $V_P \leq V_Q$: cắt tỉa các con chưa xét của Q (cắt tỉa β)

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Xém tỉa $\alpha - \beta$ (Alpha - Beta pruning)

- Nếu $VP \geq VQ$: cắt tỉa các con chưa xét của Q (cắt tỉa α) – vì sao?

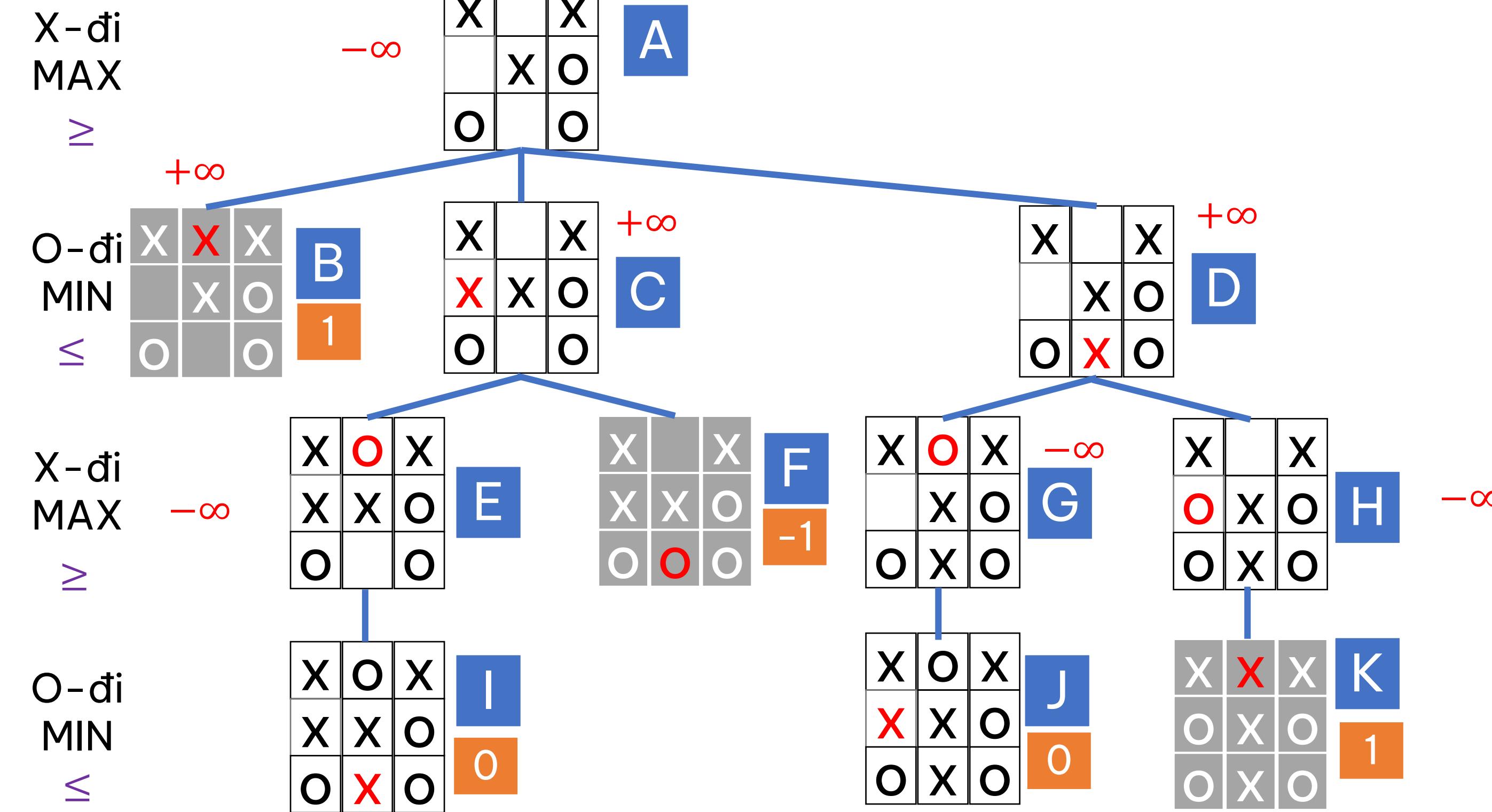
- Nếu xét hết tất cả các con của Q, ta sẽ tìm được giá trị của Q là V
- Vì Q là nút MIN: $V \leq VQi \leq VP$ (*)
- Vì P là nút MAX(V_P, V)
- Từ (*) ta nhận thấy có xét tất cả các nút con của Q cũng không làm ảnh hưởng đến giá trị của nút P



- Nếu tất cả các nút con của Q đã xét hoặc bị cắt thì giá trị tạm của Q (V_Q) sẽ là giá trị của nút Q

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

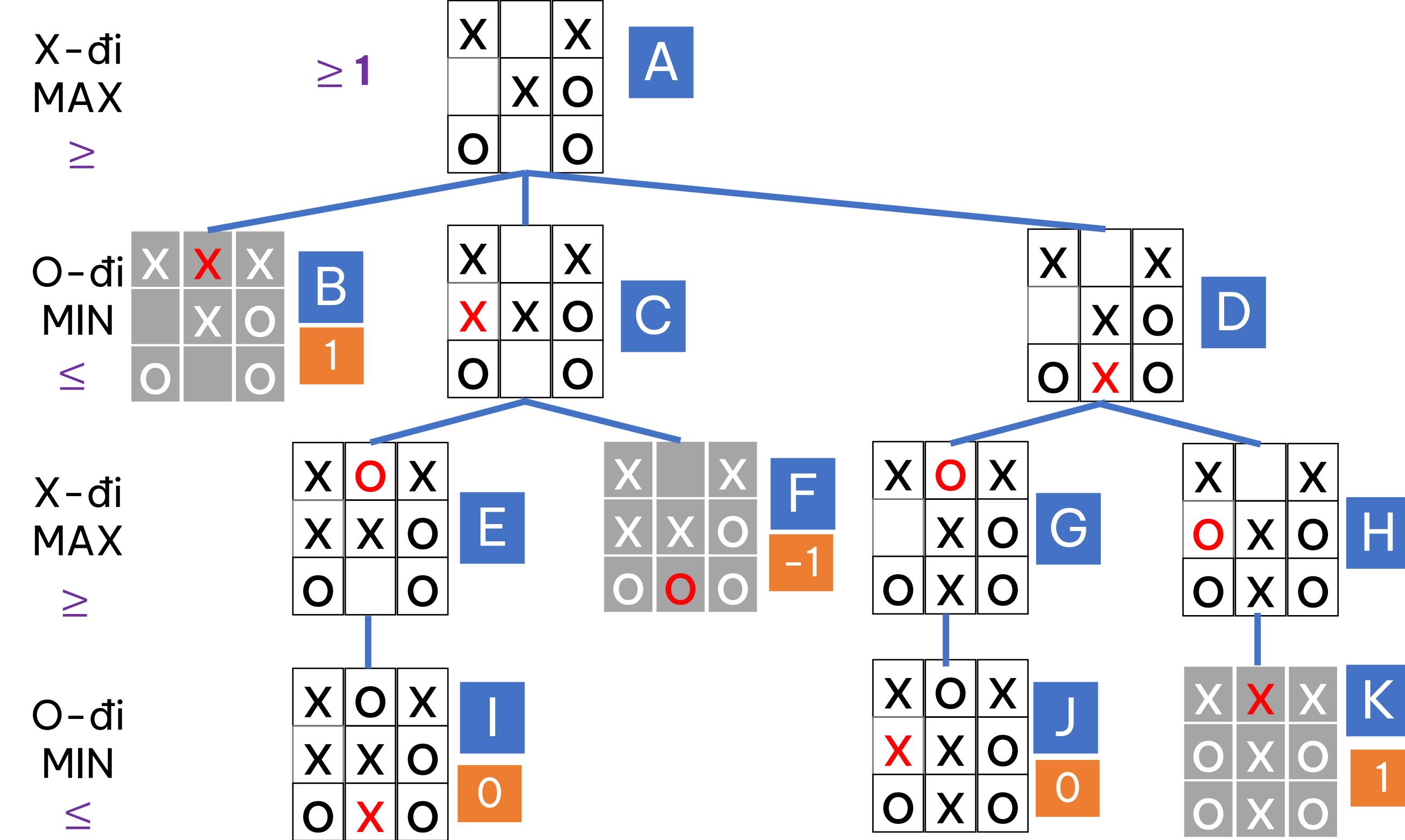
Xém tǎa $\alpha - \beta$ (Alpha - Beta pruning)



- Ghi chú: MAX \geq và MIN \leq

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Xém tǐa $\alpha - \beta$ (Alpha - Beta pruning)

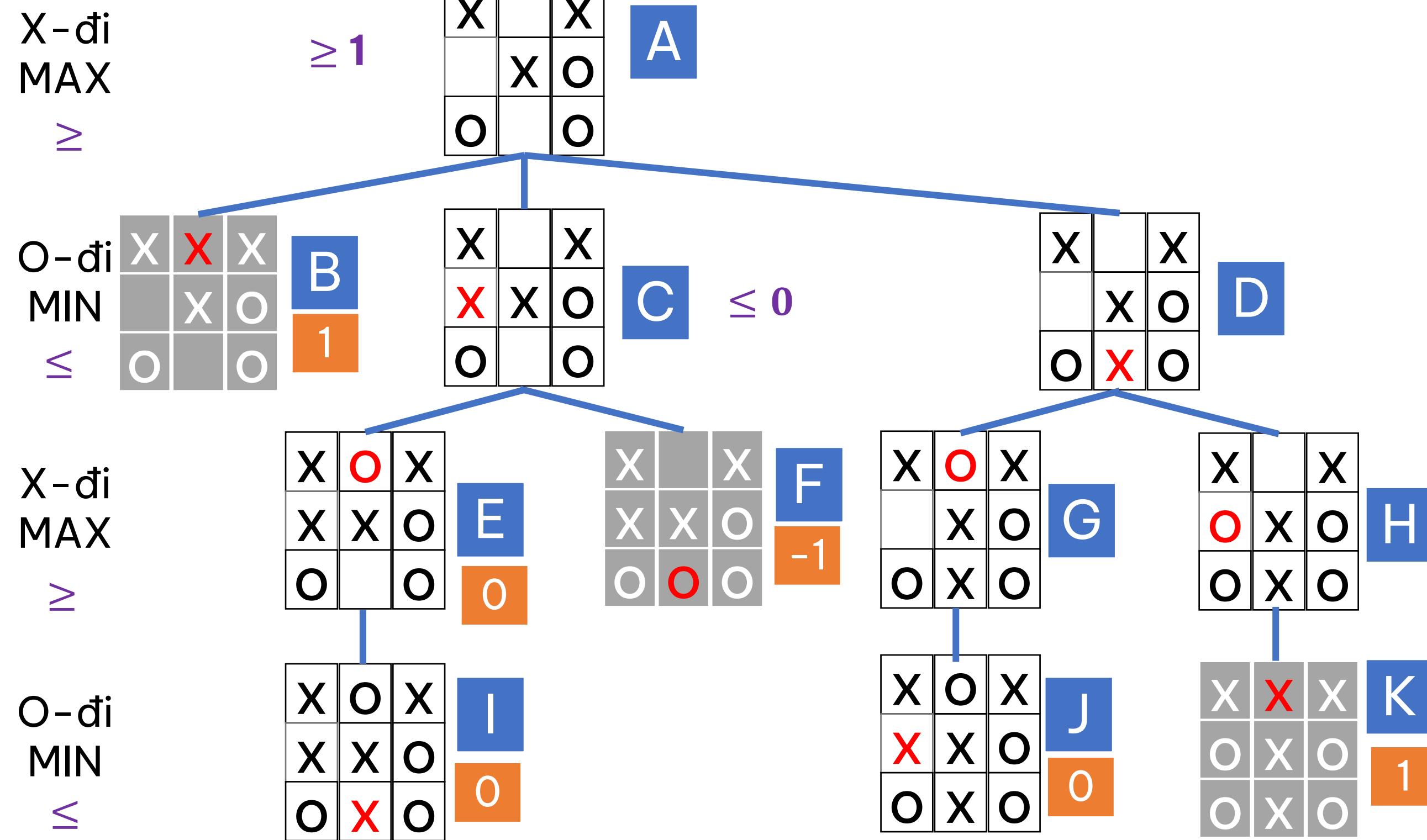


- Xét nút A có nút con B là nút lá = 1 nên tạm $A \geq 1$

- Ghi chú: MAX \geq và MIN \leq

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Xém tǐa $\alpha - \beta$ (Alpha - Beta pruning)

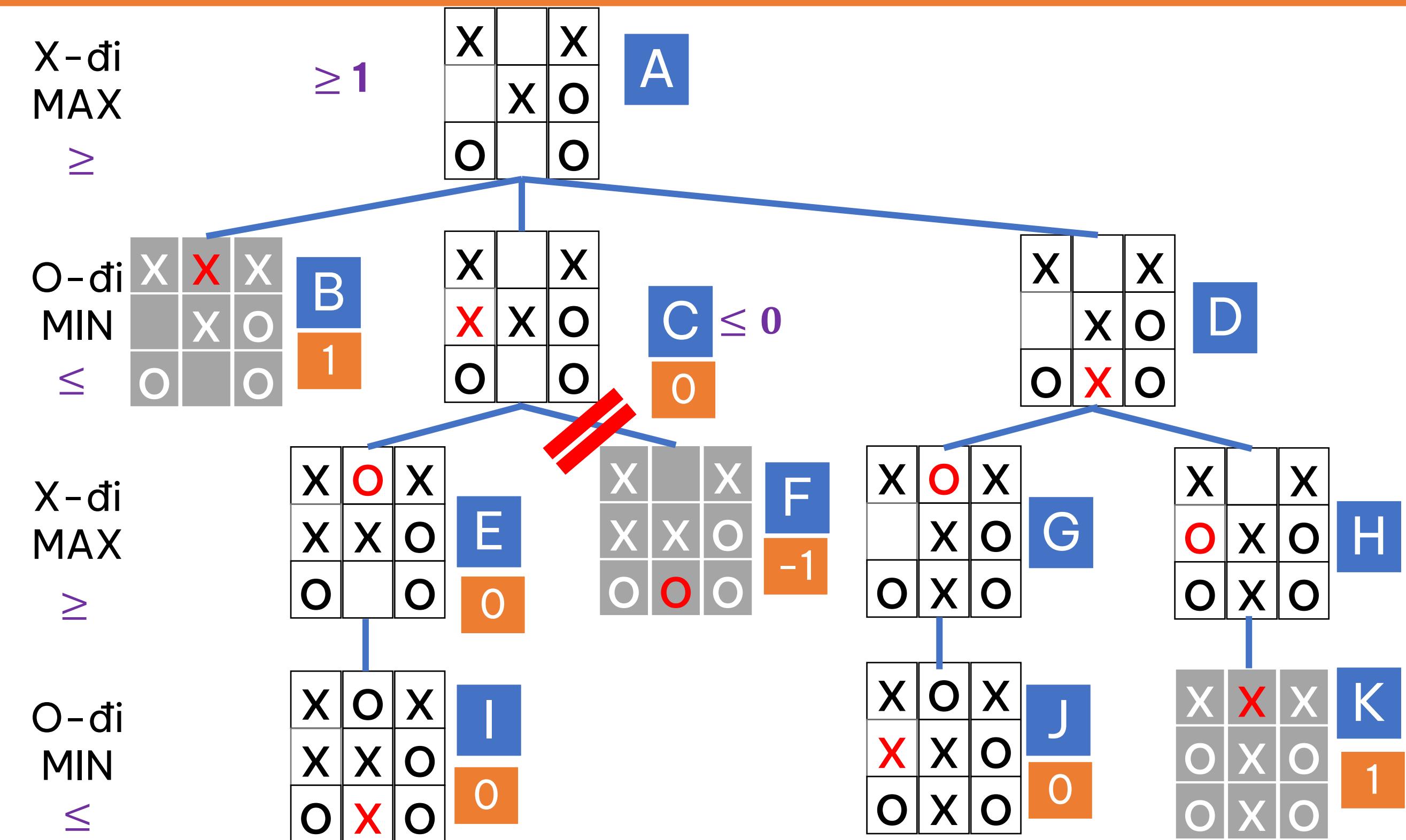


- Xét nút C có nút con E chưa là nút lá = 1 nên
- Xét nút I là con E có giá trị là 0 nên $\text{valE} \geq 0$
- **Nên tạm $C \leq 0$**

- Ghi chú: $\text{MAX} \geq \text{và } \text{MIN} \leq$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Xém tỉa $\alpha - \beta$ (Alpha - Beta pruning)

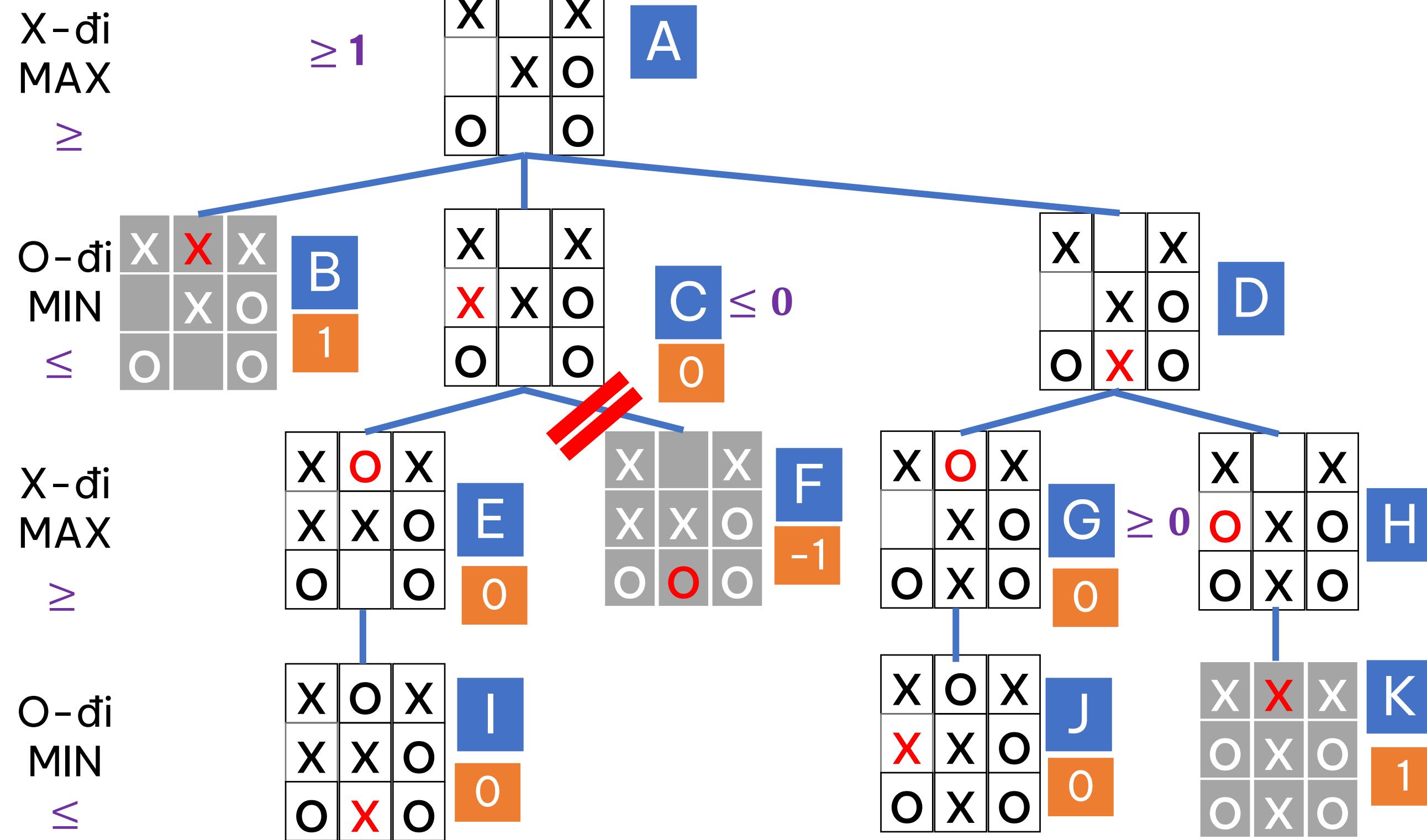


- A là nút MAX ($>=1$), C là MIN ($<=0$). Một số $X \leq 0$ và $X \geq 1$ thì không tồn tại X
- Nên nút còn lại của C (nút F) không cần xét
- Xén tỉa alpha
- Nút C sẽ nhận giá trị valC=0

▪ Ghi chú: MAX \geq và MIN \leq

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Xém tǐa $\alpha - \beta$ (Alpha - Beta pruning)

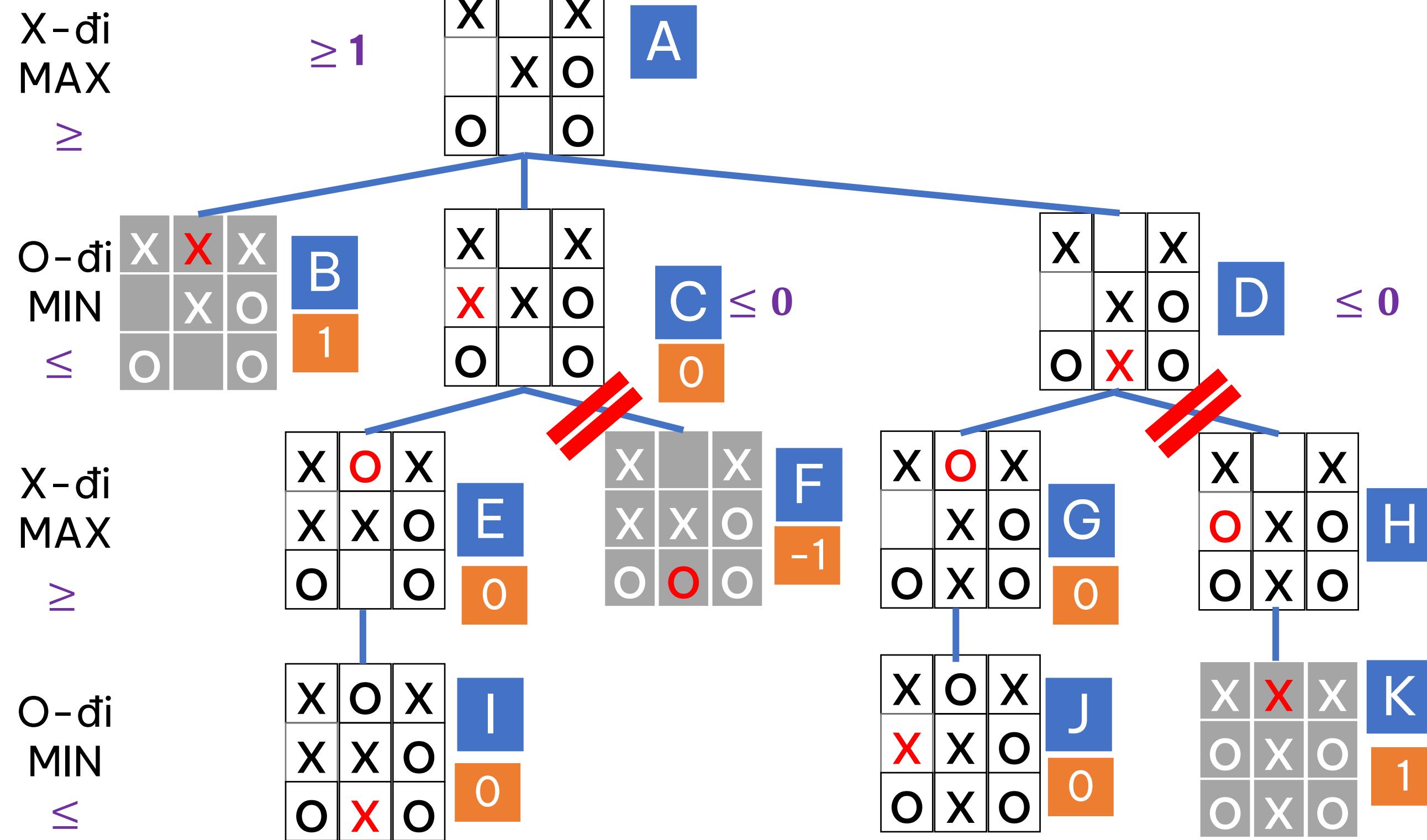


- Xét nút D có 2 con G và H chưa có giá trị nên
- Xét nút G, có con là nút J có giá trị là 0 nên tạm $G \geq 0$
- Vì nút G có 1 con đã xét nên $valG=0$

- Ghi chú: $MAX \geq$ và $MIN \leq$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Xém tỉa $\alpha - \beta$ (Alpha - Beta pruning)

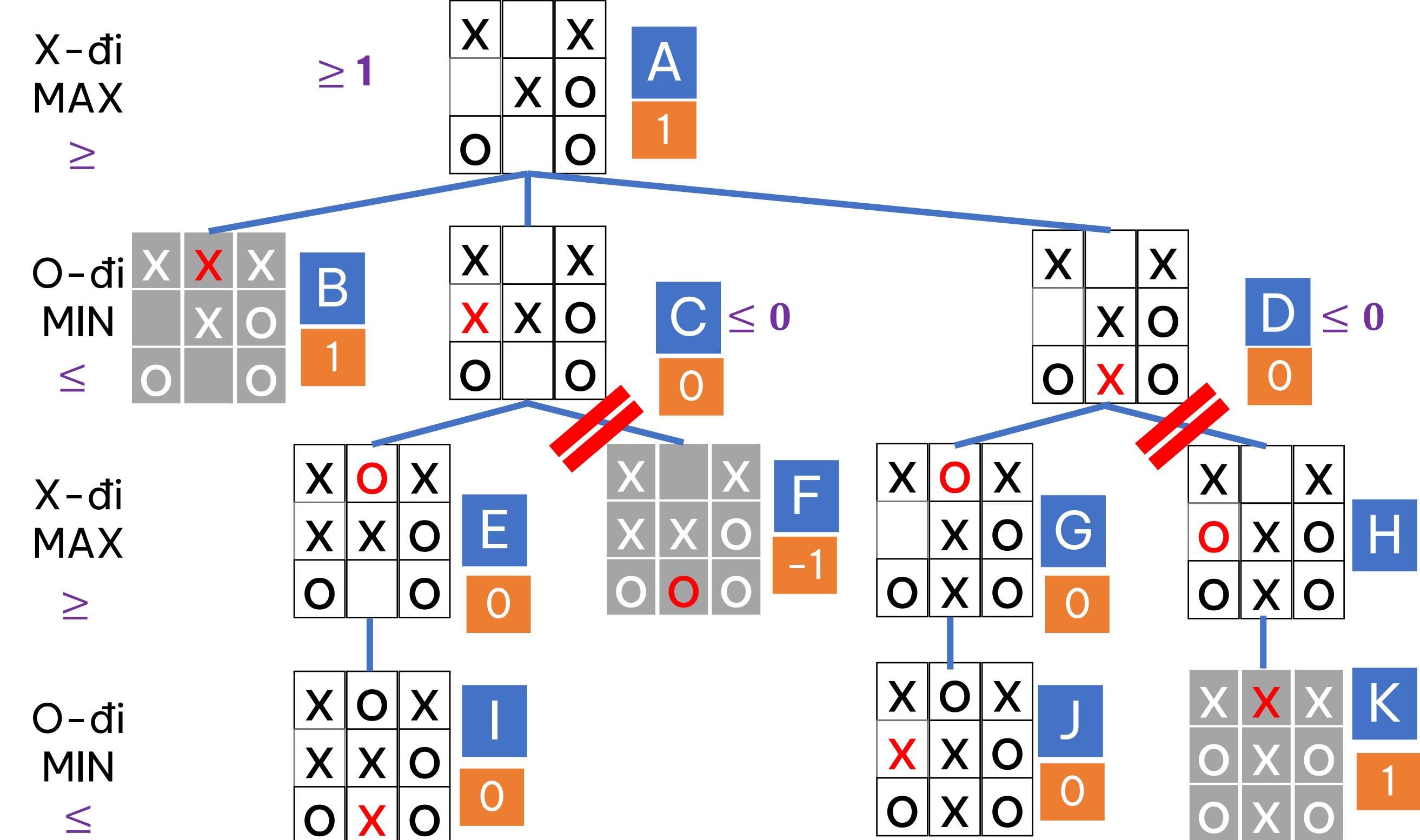


- D là nút MIN có tạm $D \leq 0$
- Nút A là nút MAX ≥ 1
- D là nút MIN ≤ 0
- $X \leq 0$ và $X \geq 1$ thì $\exists! X$
- Cắt tỉa nhánh con của D là H**

- Ghi chú: MAX \geq và MIN \leq

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Xém tǐa $\alpha - \beta$ (Alpha - Beta pruning)



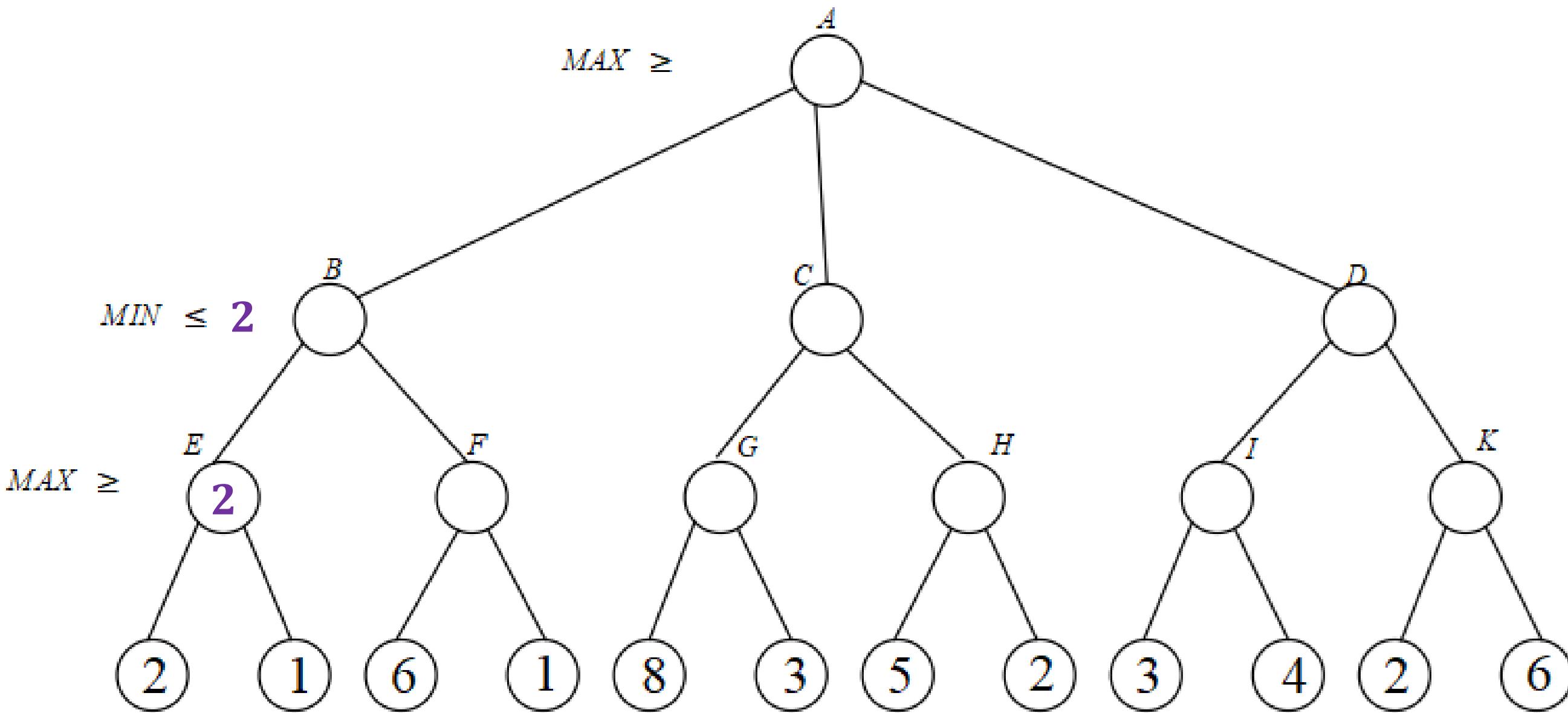
- Nút D có các nhánh đã xét nên valD= 0
- Các nút con B,C,D của D đã xét. MAX(tamamjAA, valD)=1 nên valA=1

- Kết quả nút A so với PP vét cạn là không đổi nhưng số trường hợp phải xét ít hơn nhiều so với PP trước.

- Ghi chú: MAX \geq và MIN \leq

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

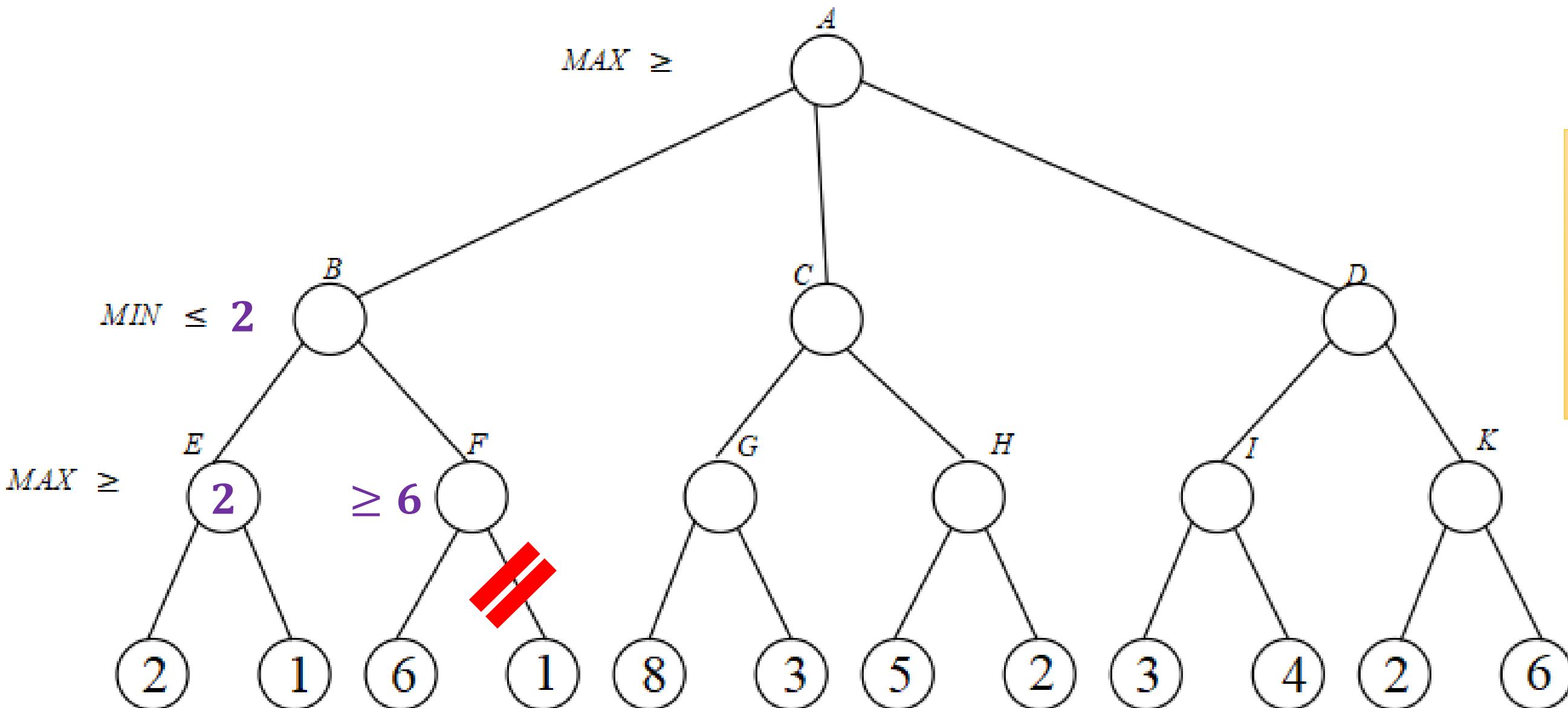
Xém tǎa $\alpha - \beta$ (Alpha - Beta pruning)



- Xét nút E có 2 nút con đã biết nên $valE = MAX(2,1)=2$
- Nút B có tạm B: ≤ 2

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

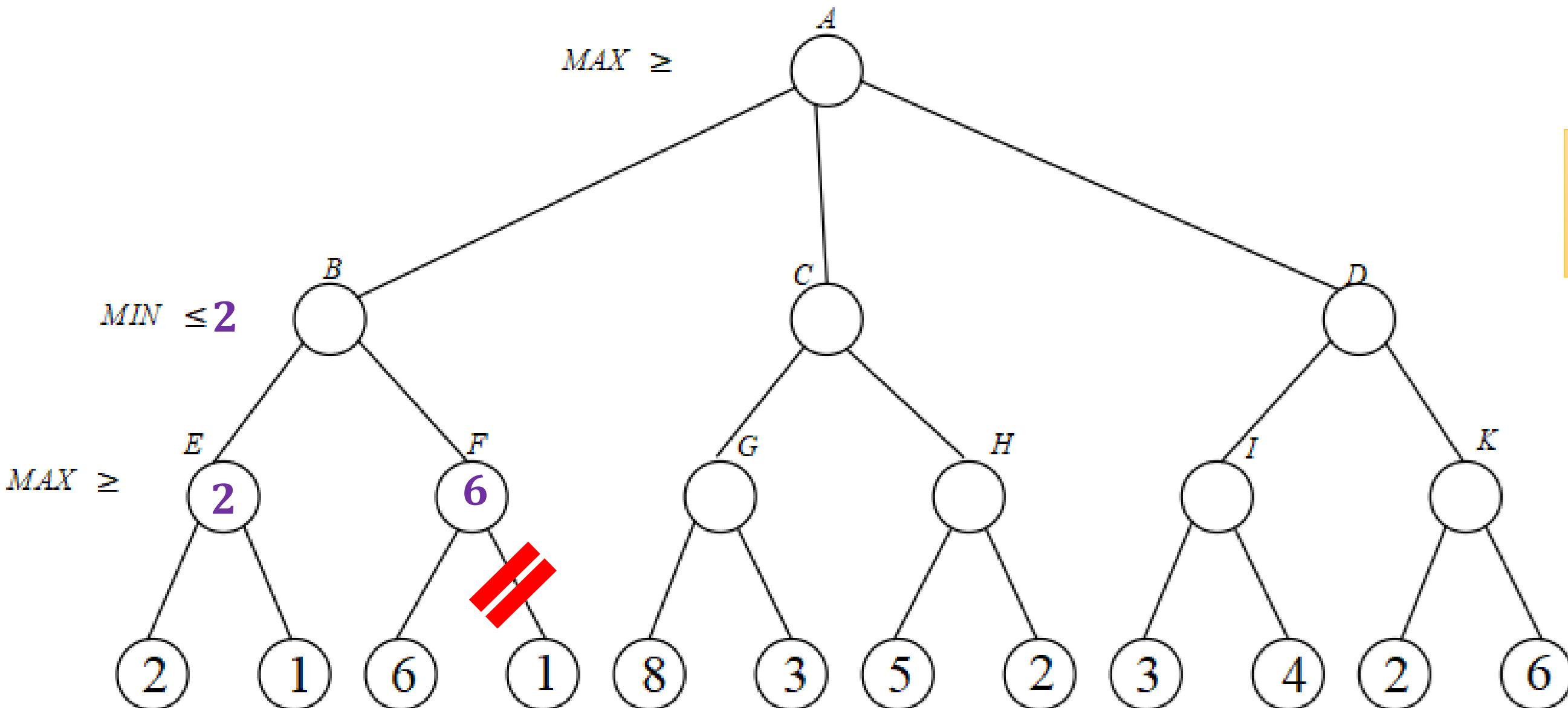
Xém tỉa $\alpha - \beta$ (Alpha - Beta pruning)



- Xét nút F có nút con trái = 6 nên tạm $F >= 6$
- $\nexists X (X \geq 6 \text{ và } X \leq 2)$
- *cắt tỉa nhánh con của F bên phải*

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

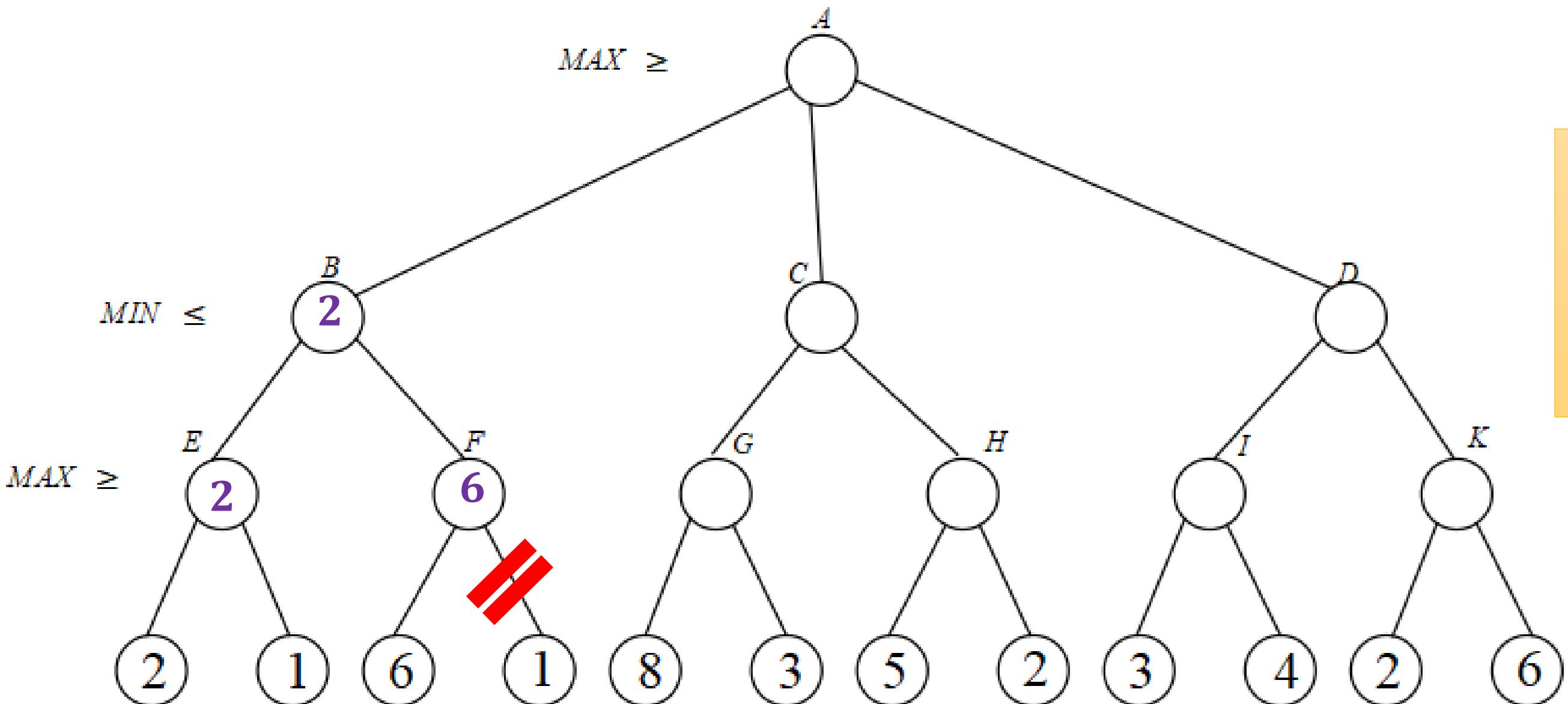
Xém tǎa $\alpha - \beta$ (Alpha - Beta pruning)



- Nút F có 2 nhánh đều xác định nên $valF = 6$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

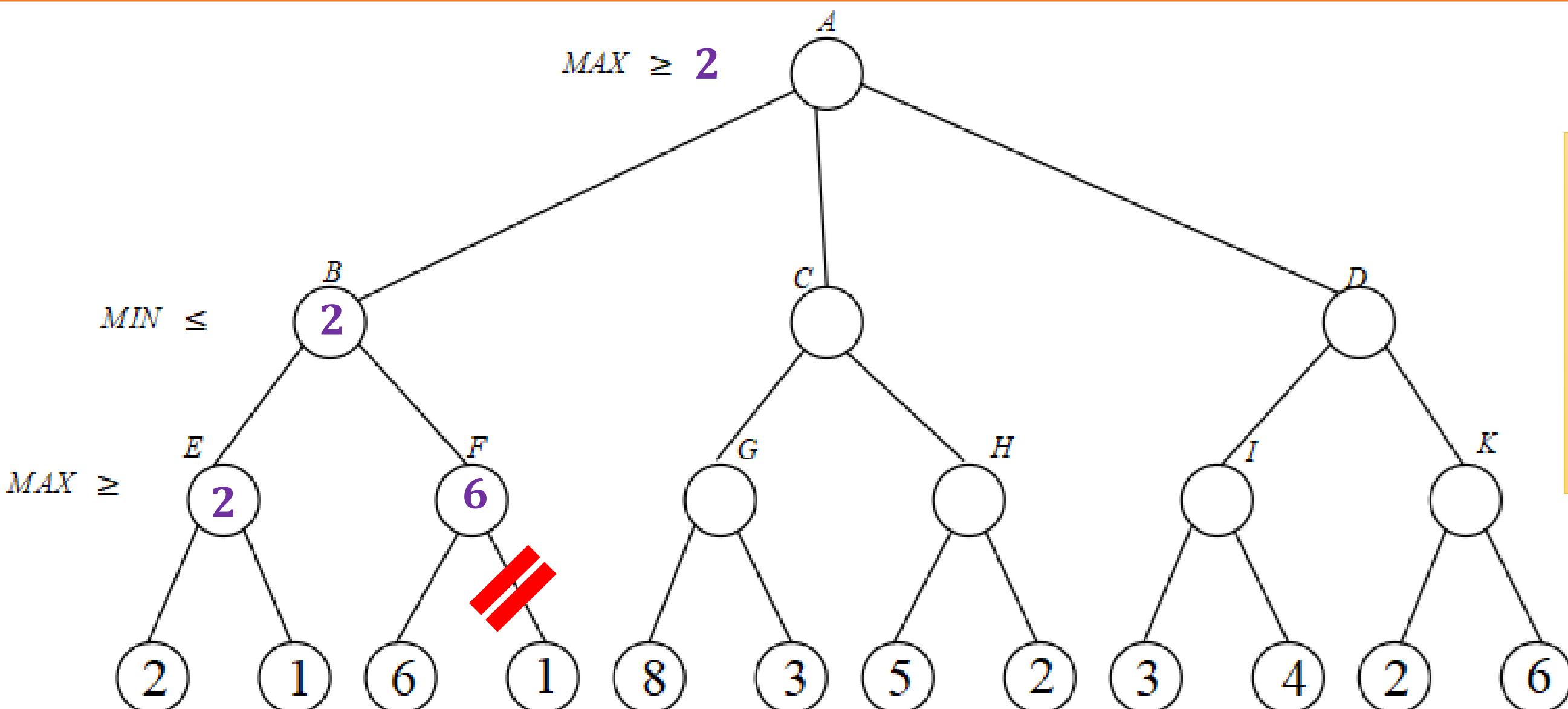
Xém tia $\alpha - \beta$ (Alpha - Beta pruning)



- Nút F có 2 nhánh đều xác định nên $\text{valF} = 6$
- Nút B có 2 nhánh đều đã xét nên $\text{valB} = 2$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

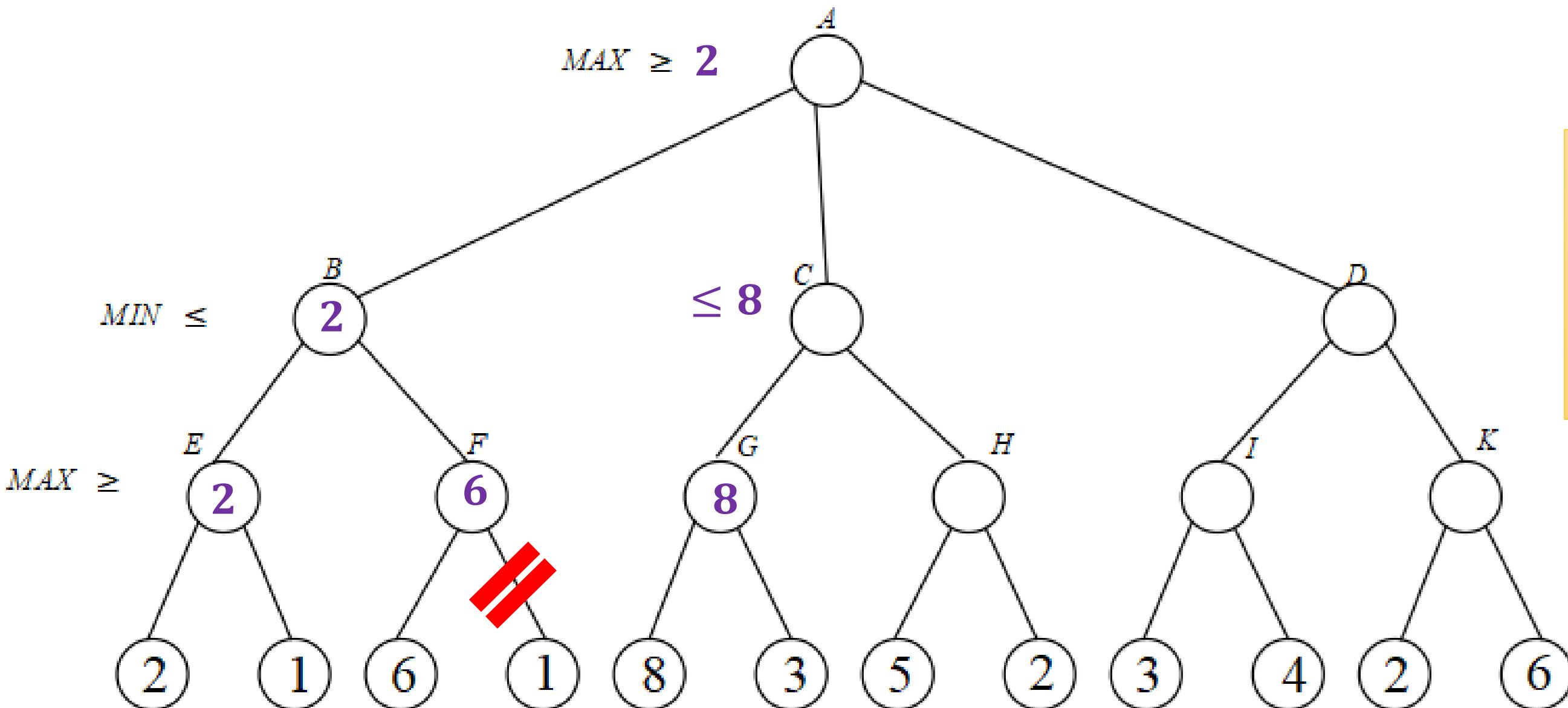
Xém tǎa $\alpha - \beta$ (Alpha - Beta pruning)



- Nút F có 2 nhánh đều xác định nên $\text{valF} = 6$
- Nút B có 2 nhánh đều đã xét nên $\text{valB} = 2$
- Nút A có tạm $\text{A} \geq 2$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

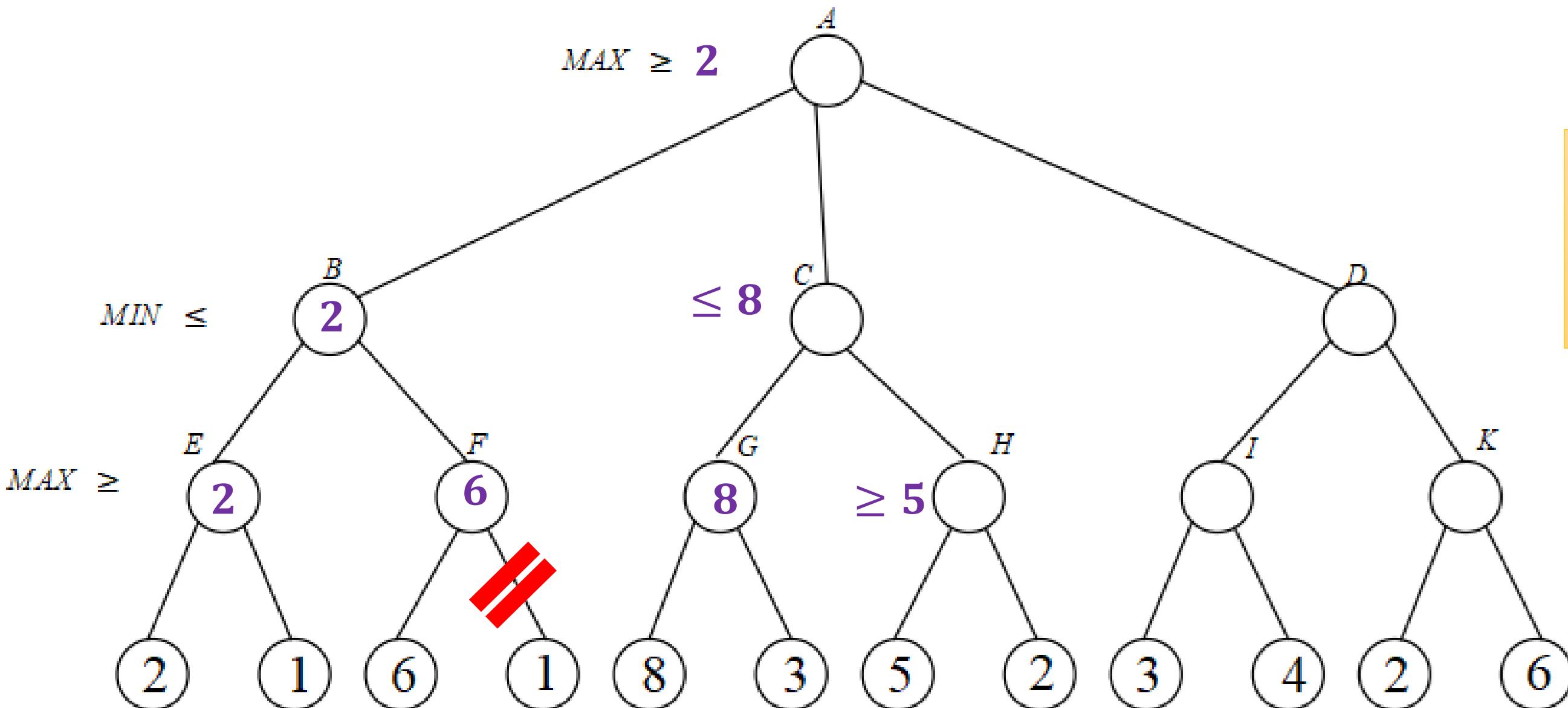
Xém tỉa $\alpha - \beta$ (Alpha - Beta pruning)



- Xét nút C có 2 con G và H đều chưa có giá trị
- Xét nút G có $valG = MAX(8,3)$
- Tạm $C \leq 8$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

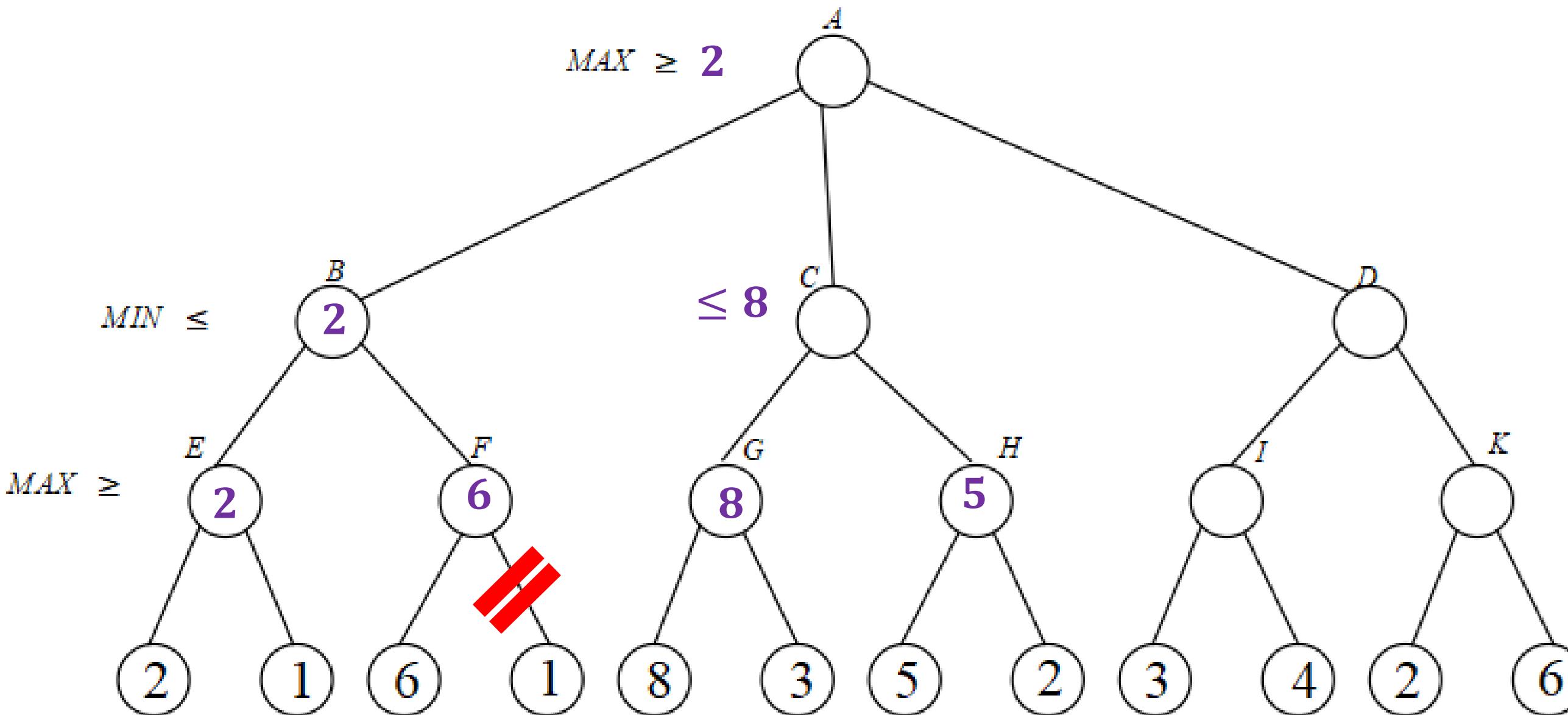
Xém tǐa $\alpha - \beta$ (Alpha - Beta pruning)



- Xét nút C là con của A
- $\exists X (X \leq 8 \text{ và } X \geq 2)$ nên
- Xét nút H có tạm $H \geq 5$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

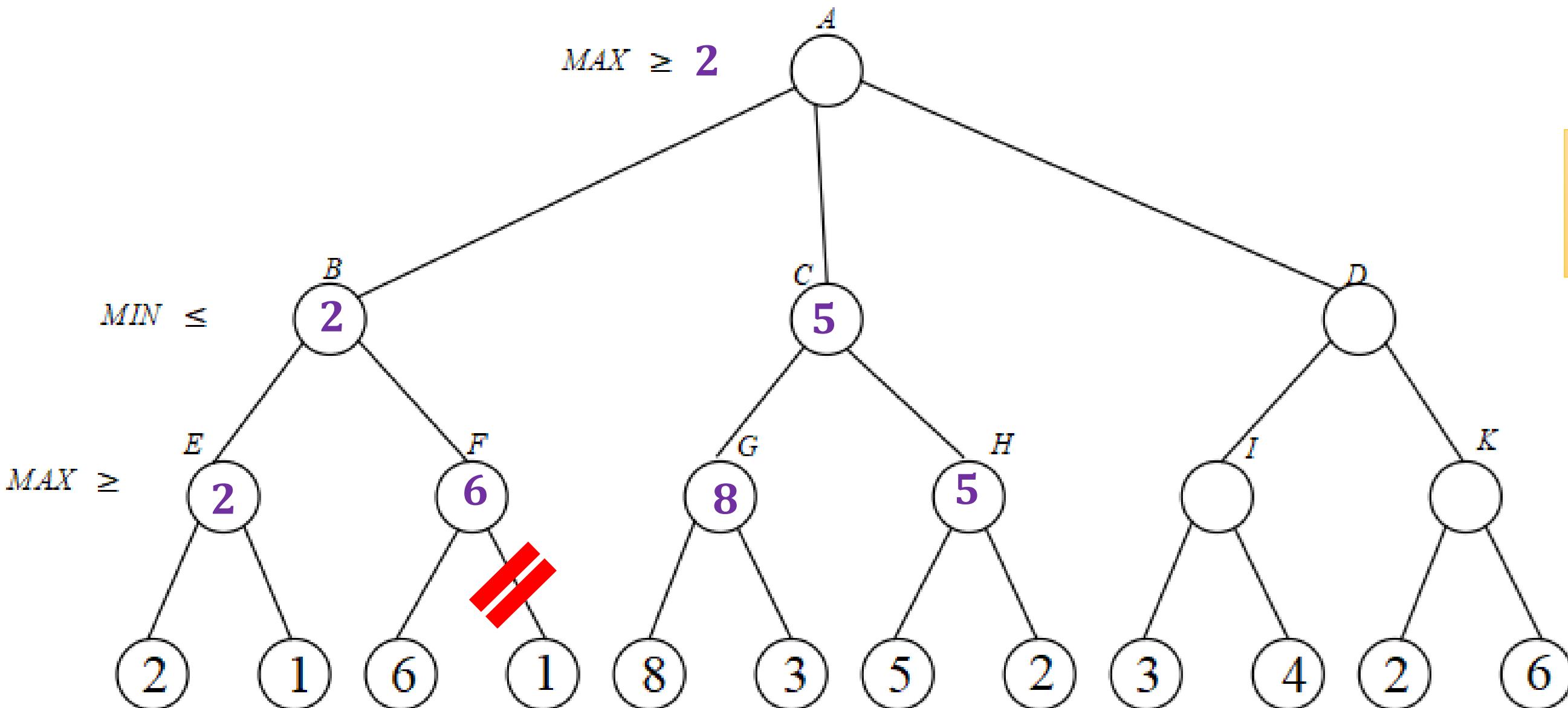
Xém tỉa $\alpha - \beta$ (Alpha - Beta pruning)



- Xét nút H là con của C
- $\exists X (X \leq 8 \text{ và } X \geq 5)$ nên
- Xét nút con bên phải H và tính lại $valH = MAX(5, 2) = 5$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

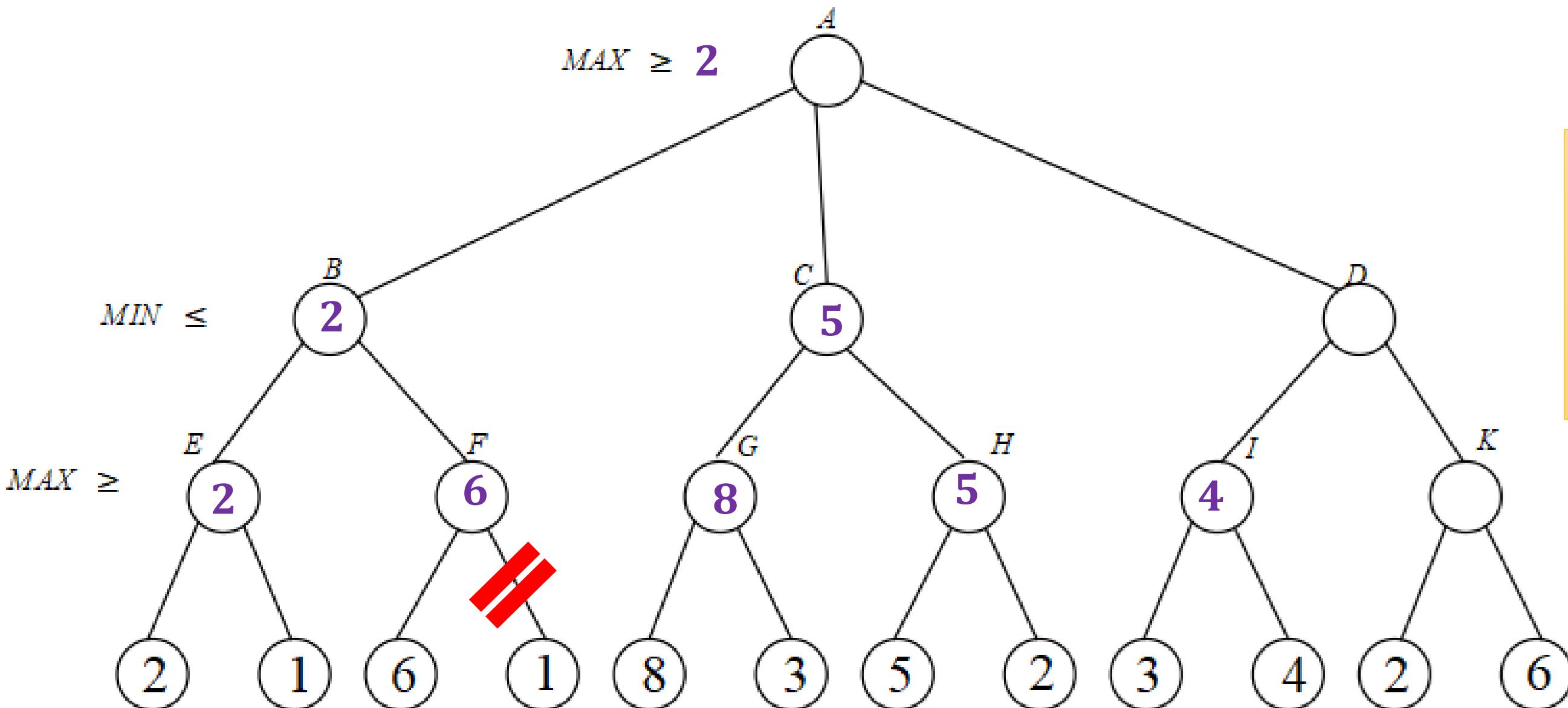
Xém tǎa $\alpha - \beta$ (Alpha - Beta pruning)



- Xét nút C có 2 nút con đã biết
- $valC = MIN(tạm C, 5) = 5$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

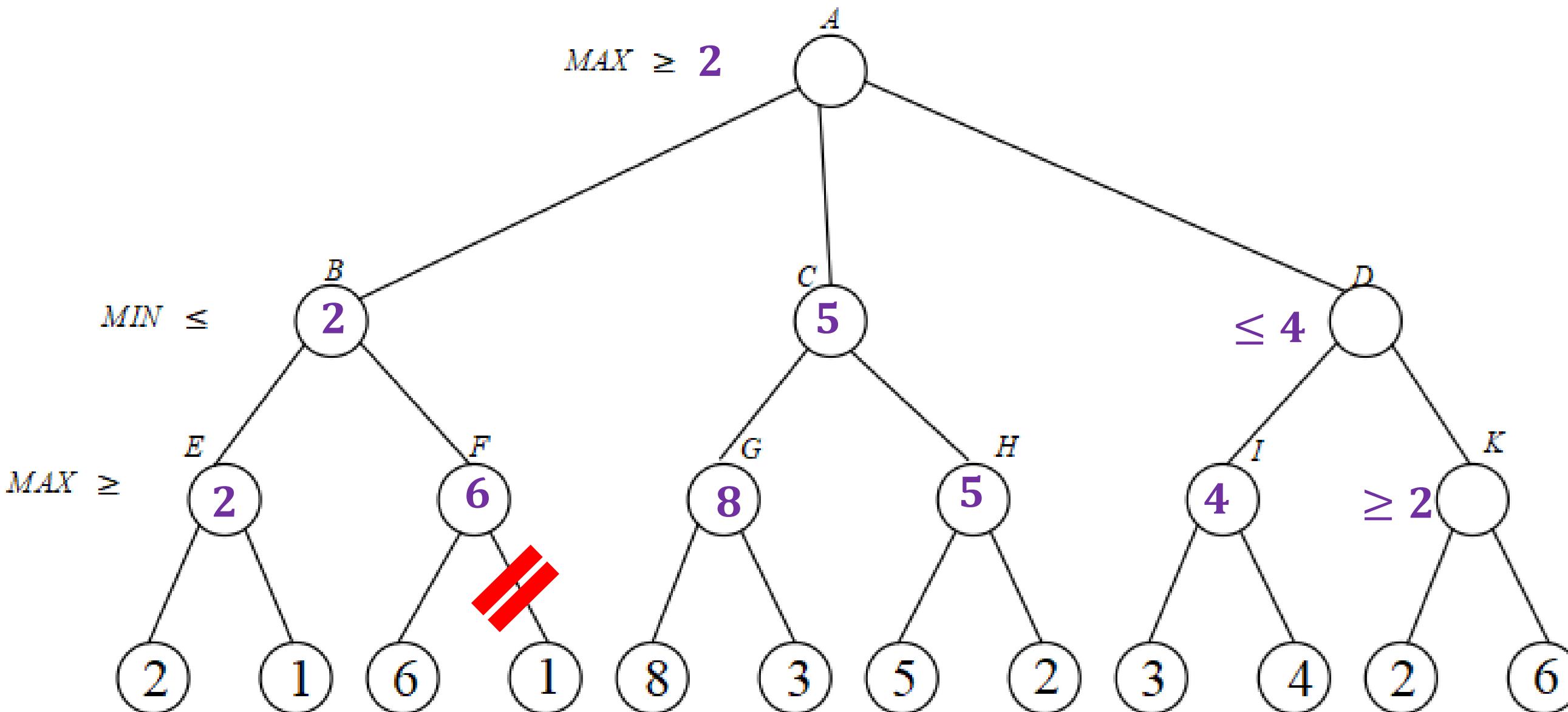
Xém tỉa $\alpha - \beta$ (Alpha - Beta pruning)



- Xét nút D là con của A chưa có giá trị và có 2 con I,E chưa có giá trị.
- Xét nút I: $I = \text{MAX}(3,4)$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

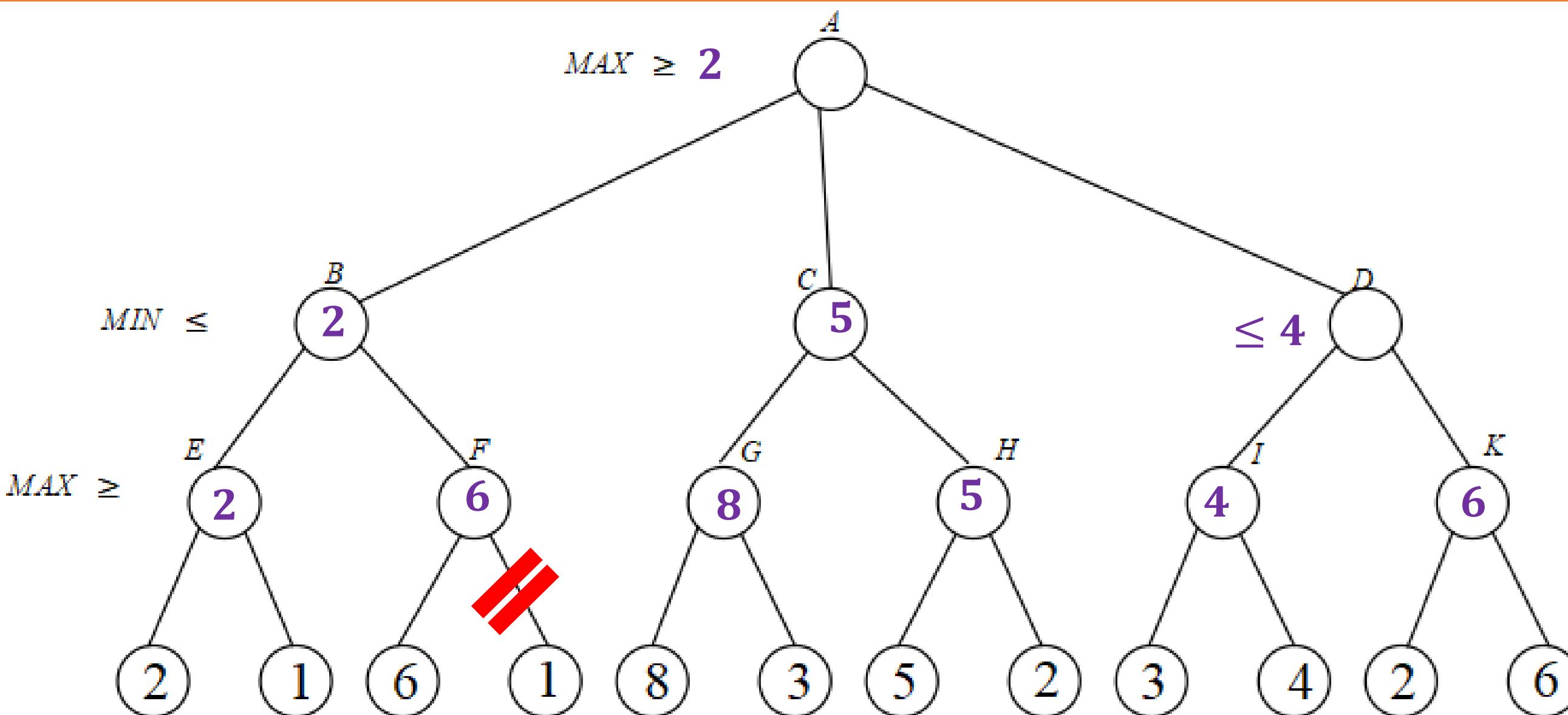
Xém tǎa $\alpha - \beta$ (Alpha - Beta pruning)



- Xét nút D có con là I: tạm $D \leq 4$
- Xét nút A có con là D
- $\exists X (X \leq 4 \text{ và } X \geq 2)$ nên xét tiếp con của D (nút K)
- Tạm $K \geq 2$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

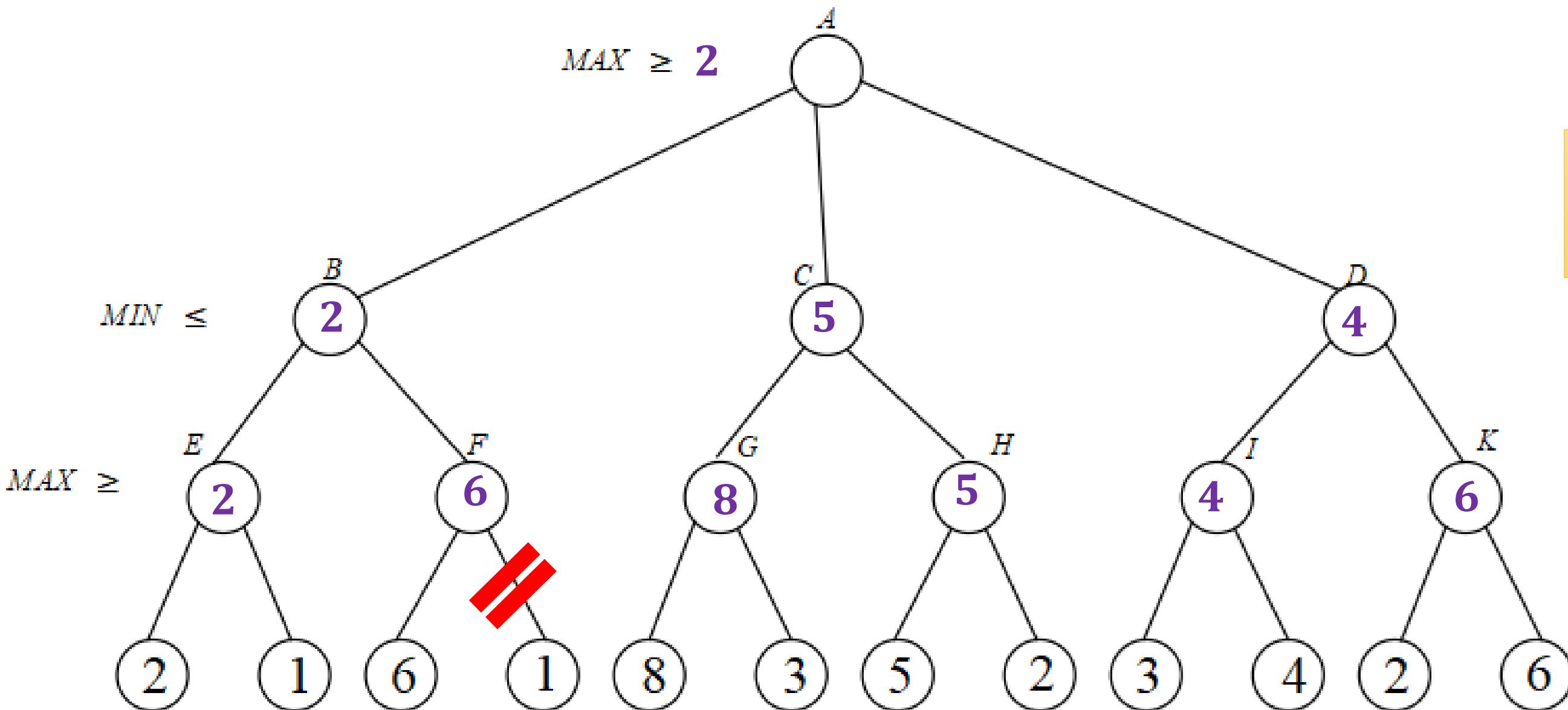
Xém tǎa $\alpha - \beta$ (Alpha - Beta pruning)



- Xét nút K
- $\exists X (X \leq 4 \text{ và } X \geq 2)$ nên xét tiếp con của K (nút bên phải)
- val K $>= MAX(2,6)=6$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

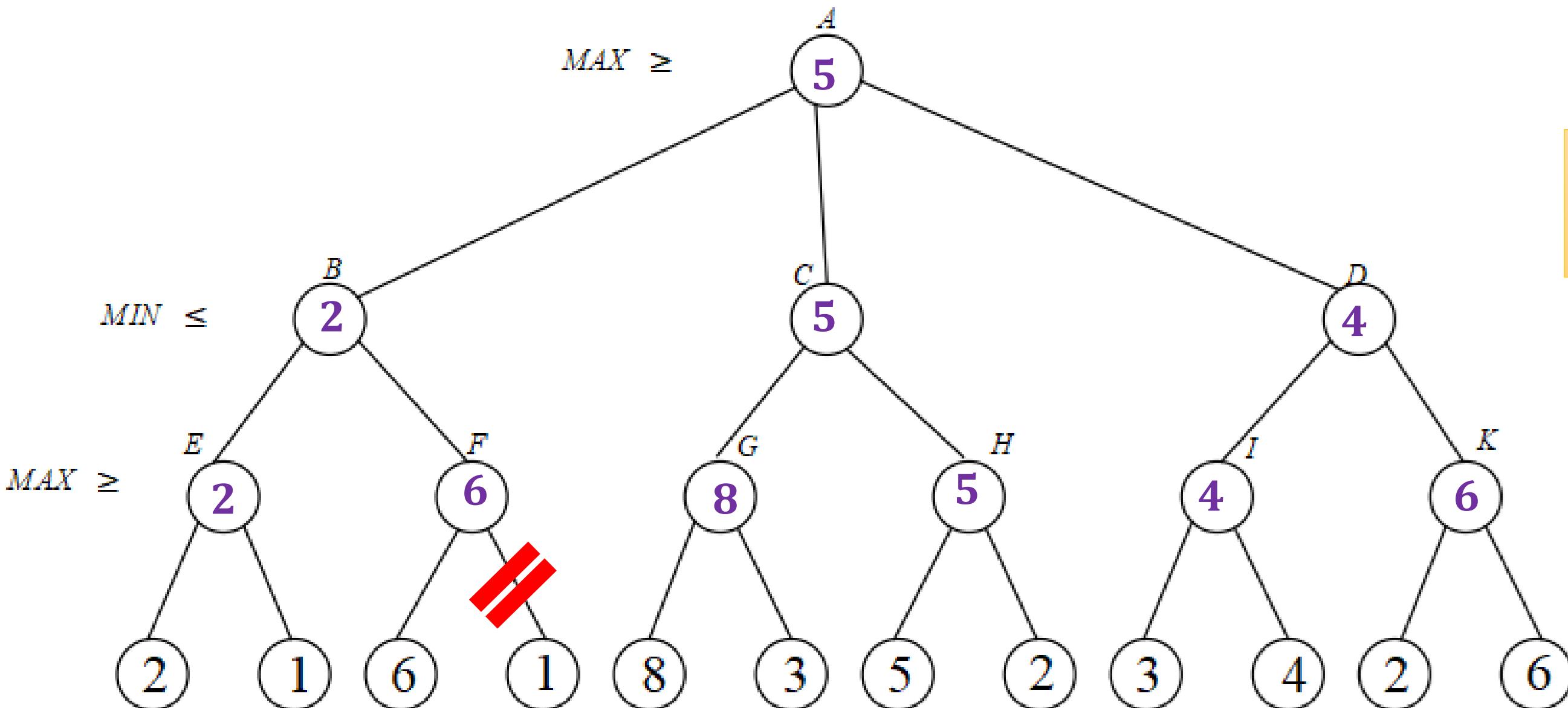
Xém tǎa $\alpha - \beta$ (Alpha - Beta pruning)



- Xét nút D có 2 con đã biết
- $valD = MIN(\text{tạm D}, 6) = 4$

Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

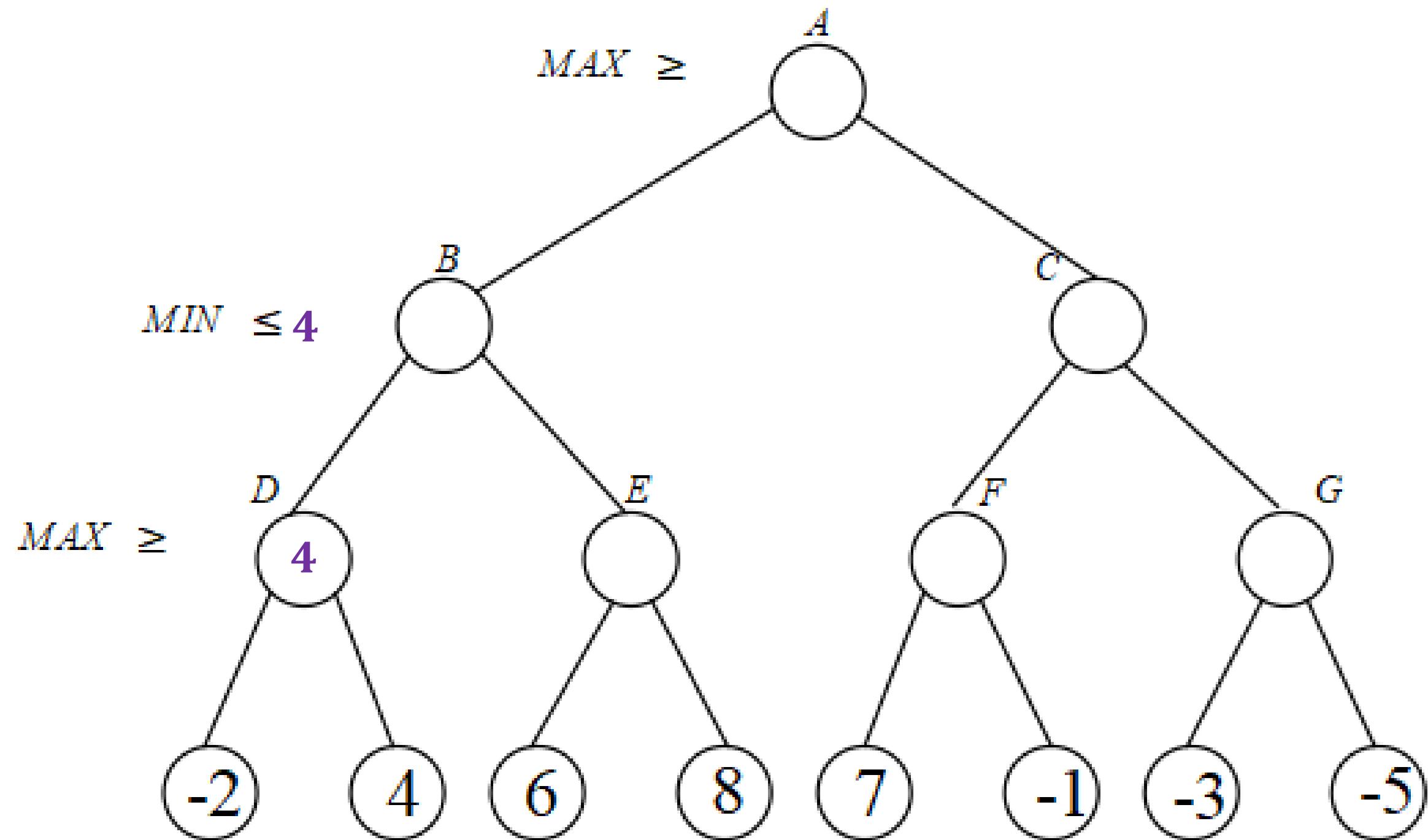
Xém tǎa $\alpha - \beta$ (Alpha - Beta pruning)



- Xét nút A có 3 con đã biết
- $valA = MAX(tạm A, 5, 4) = 5$

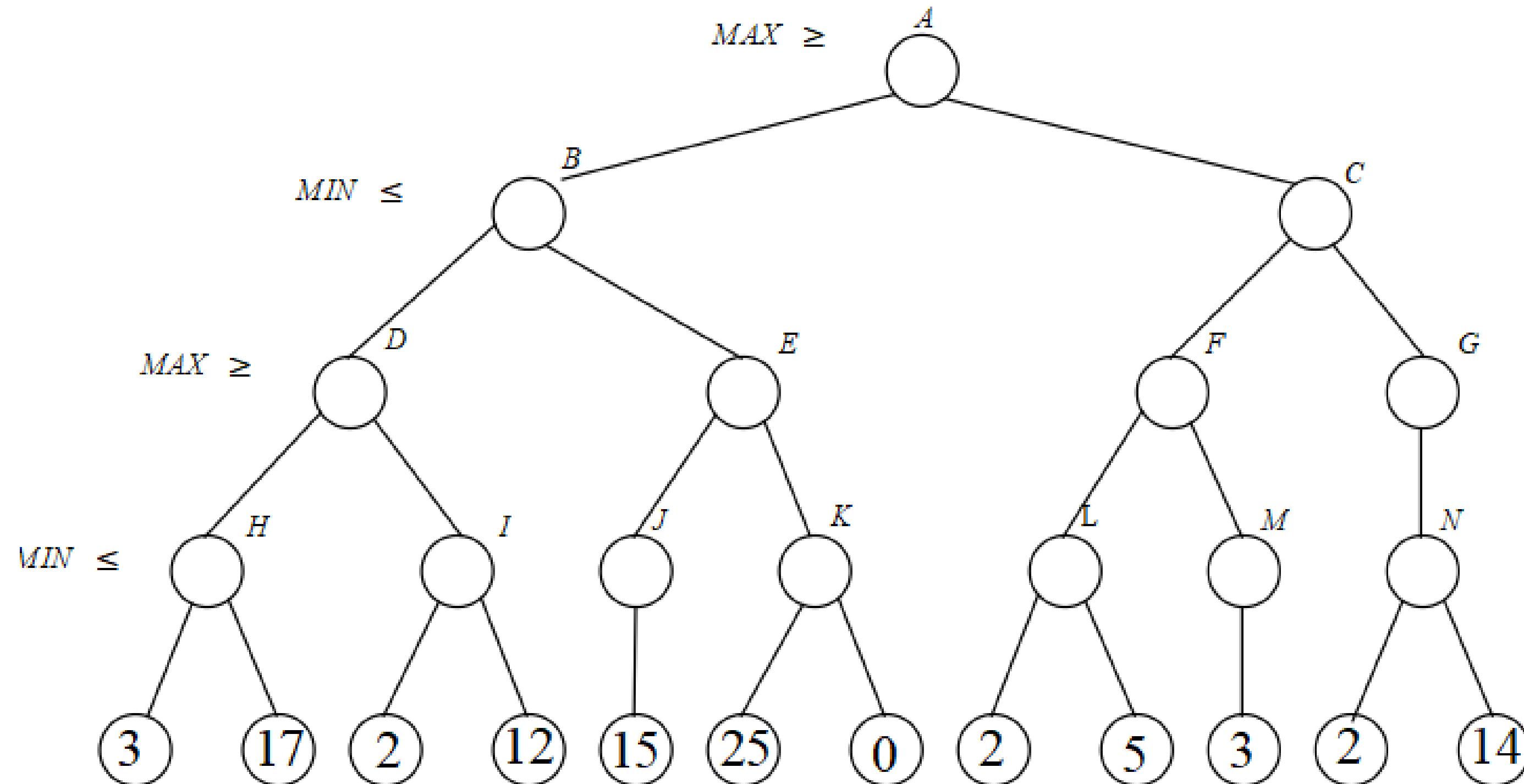
Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Xém tia $\alpha - \beta$ (Alpha - Beta pruning)



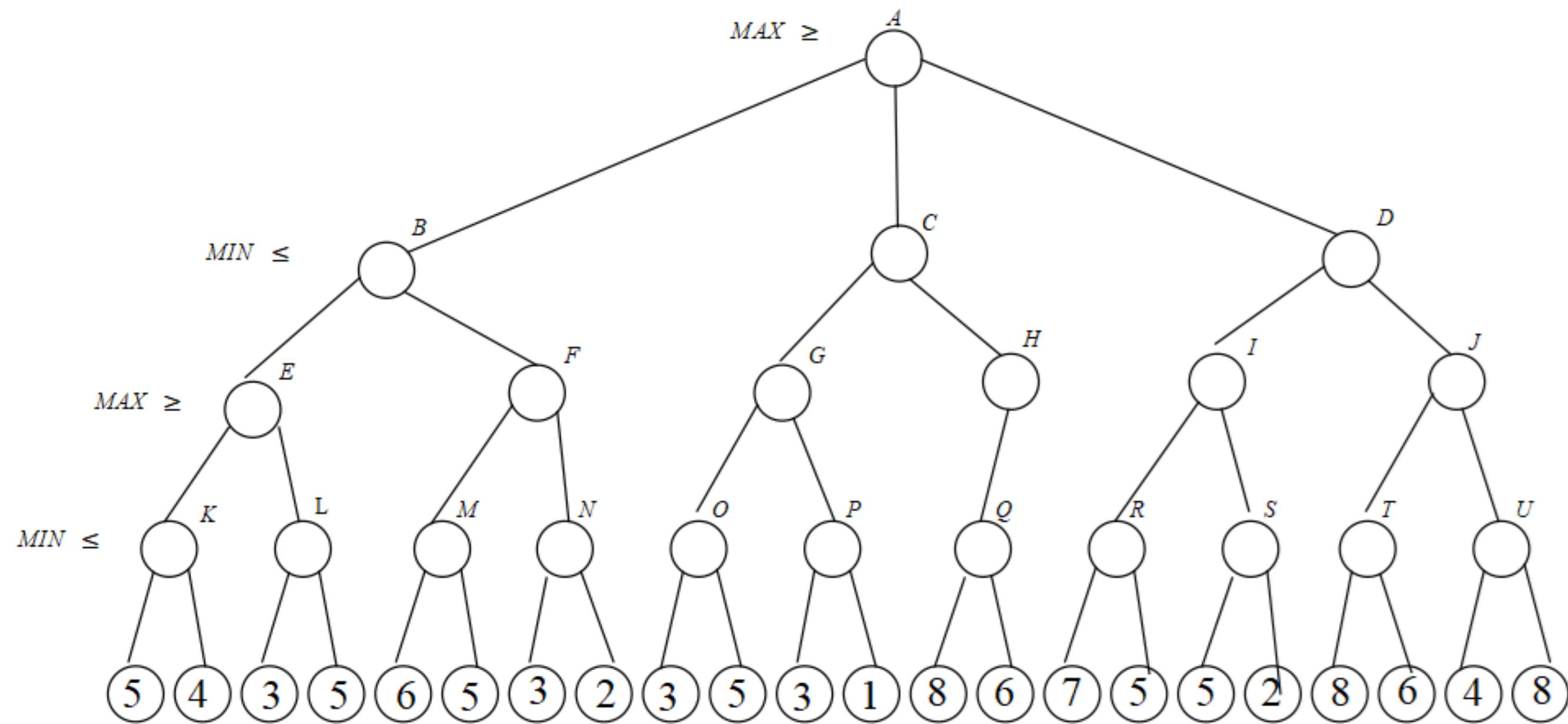
Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Xém tia $\alpha - \beta$ (Alpha - Beta pruning)



Chương 5. GIẢI THUẬT TÌM KIẾM LỜI GIẢI CHO TRÒ CHƠI

Xém tia $\alpha - \beta$ (Alpha - Beta pruning)



HẾT CHƯƠNG 5

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC



TRÍ TUỆ NHÂN TẠO *Artificial Intelligence*

Đoàn Vũ Thịnh
Khoa Công nghệ Thông tin
Đại học Nha Trang
Email: thinhdv@ntu.edu.vn

Nha Trang, 06-2023

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Dữ liệu thô chưa được xử lý

Dữ liệu

Thông tin

Tri thức

Siêu tri thức

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

- Dữ liệu:
 - Biểu diễn dưới dạng chuỗi số, ký tự hay hỗn hợp cả hai loại mà tự bản thân nó không có ý nghĩa độc lập.
- Thông tin:
 - Dữ liệu được tổ chức có đầy đủ ý nghĩa đối với người nhận nó.
- Tri thức (Knowledge):
 - Là những thông tin liên quan đến một vấn đề nào đó được tổ chức để có thể giải quyết được vấn đề.
- Siêu tri thức (Meta Knowledge):
 - Là loại tri thức ở mức cao, nó có tính khái quát và trừu tượng hơn tri thức.

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

- Cơ sở tri thức (Knowledge Base):
 - là tập hợp các tri thức liên quan đến một vấn đề được sử dụng trong một hệ thống trí tuệ nhân tạo. Nó không chỉ bao gồm các sự kiện mà còn chứa các luật suy diễn và nó có thể được sắp xếp lại khi có thêm một tri thức mới làm thay đổi mối liên hệ giữa chúng.
- Hệ chuyên gia (Expert System):
 - là một hệ cơ sở tri thức (Knowledgebased system) được xây dựng từ tri thức của các chuyên gia trong một lĩnh vực nào đó.

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

- Tri thức được chia thành 5 loại:
 - Tri thức thủ tục: mô tả cách thức giải quyết một vấn đề. Loại tri thức này đưa ra giải pháp để thực hiện một công việc nào đó. Các dạng tri thức thủ tục tiêu biểu thường là các luật, chiến lược, lịch trình và thủ tục.
 - Tri thức khai báo: cho biết một vấn đề được thấy như thế nào. Loại tri thức này bao gồm các phát biểu đơn giản, dưới dạng các khẳng định logic đúng sai. Tri thức khai báo cũng có thể là một danh sách các khẳng định nhằm mô tả đầy đủ hơn về đối tượng hay một khái niệm nào đó.

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

- Siêu tri thức: mô tả tri thức về tri thức. Loại tri thức này giúp lựa chọn các tri thức hợp nhất trong số các tri thức khi giải quyết một vấn đề. Các chuyên gia sử dụng tri thức này để hiệu chỉnh hiệu quả giải quyết vấn đề bằng cách lập luận về miền tri thức có khả năng hơn cả.
- Tri thức heuristic: mô tả các mẹo để dẫn dắt tiến trình lập luận. Tri thức heuristic không đảm bảo 100% chính xác về kết quả. Các chuyên gia thường dùng các tri thức khoa học như sự kiện, luật ... Sau đó chuyển chúng thành các tri thức heuristic để thuận tiện hơn trong giải quyết một số bài toán.

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

- Tri thức có cấu trúc: mô tả tri thức theo cấu trúc. Loại tri thức này mô tả mô hình tổng quan hệ thống theo quan điểm của chuyên gia, bao gồm các khái niệm, khái niệm con, các đối tượng; diễn tả chức năng và mối liên hệ giữa các tri thức dựa theo cấu trúc xác định.
- Hãy phân loại các loại tri thức sau:
 - Nha Trang là thành phố đẹp
 - Thuật toán tìm kiếm theo chiều rộng, sâu
 - Thuật toán tham lam
 - Hệ thống các khái niệm trong hình học không gian
 - Tóm lược về cuốn sách «Cẩm nang lập trình»

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Phương pháp biểu diễn tri thức

- Bộ ba Đối tượng – Thuộc tính – Giá trị
- Các luật dẫn
- Mạng ngữ nghĩa
- Frames
- Logic

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Bộ ba Đối tượng – Thuộc tính – Giá trị

- Một sự kiện có thể được dùng để xác nhận giá trị của một thuộc tính xác định của một vài đối tượng.
- Ví dụ, mệnh đề «quả bóng màu đỏ» xác nhận «đỏ» là giá trị thuộc tính «màu» của đối tượng «quả bóng».
- Kiểu sự kiện này được gọi là bộ ba Đối tượng (Object) – Thuộc tính (Attribute) – Giá trị (Value).

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Bộ ba Đối tượng – Thuộc tính – Giá trị

- Trong các sự kiện O-A-V, một đối tượng có thể có nhiều thuộc tính với các kiểu giá trị khác nhau. Hơn nữa một thuộc tính cũng có thể có một hoặc nhiều giá trị. Chúng được gọi là các sự kiện «đơn trị» hoặc «đa trị». Điều này cho phép các hệ tri thức linh động trong việc biểu diễn các tri thức cần thiết.
- Các sự kiện không phải lúc nào cũng đảm bảo là đúng hay sai với độ chắc chắn hoàn toàn. Vì thế, khi xem xét các sự kiện, người ta còn sử dụng thêm một khái niệm là độ «tin cậy».

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Các luật dẫn

- Là cấu trúc tri thức dùng để liên kết thông tin đã biết với thông tin khác giúp đưa ra các suy luận, kết luận.
- Trong hệ thống dựa trên các luật, người ta thu thập các tri thức lĩnh vực trong một tập và lưu chúng trong cơ sở tri thức của hệ thống. Hệ thống dùng các luật này cùng với các thông tin trong bộ nhớ để giải bài toán. Việc xử lý các luật trong hệ thống dựa trên các luật được quản lý bằng một module gọi là bộ suy diễn.
- 7 dạng luật cơ bản:

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Các luật dẫn

- 1. Quan hệ
 - IF Bình điện hỏng THEN Xe không khởi động được
- 2. Lời khuyên
 - IF Xe không khởi động được THEN Đi bộ
- 3. Hướng dẫn
 - IF Xe không khởi động được AND Hệ thống nhiên liệu tốt
 - THEN Kiểm tra hệ thống điện
- 4. Chiến lược
 - IF Xe không khởi động được
 - THEN Đầu tiên hãy kiểm tra hệ thống nhiên liệu, sau đó kiểm tra hệ thống điện

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Các luật dẫn

- 5. Diễn giải
 - IF Xe nổ AND Tiếng giòn THEN Động cơ hđ bình thường
- 6. Chẩn đoán
 - IF Sốt cao AND Đau họng THEN Bị covid-19
- 7. Thiết kế
 - IF là nữ AND Da sáng THEN Nên chọn SH-Mode AND Màu sáng

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

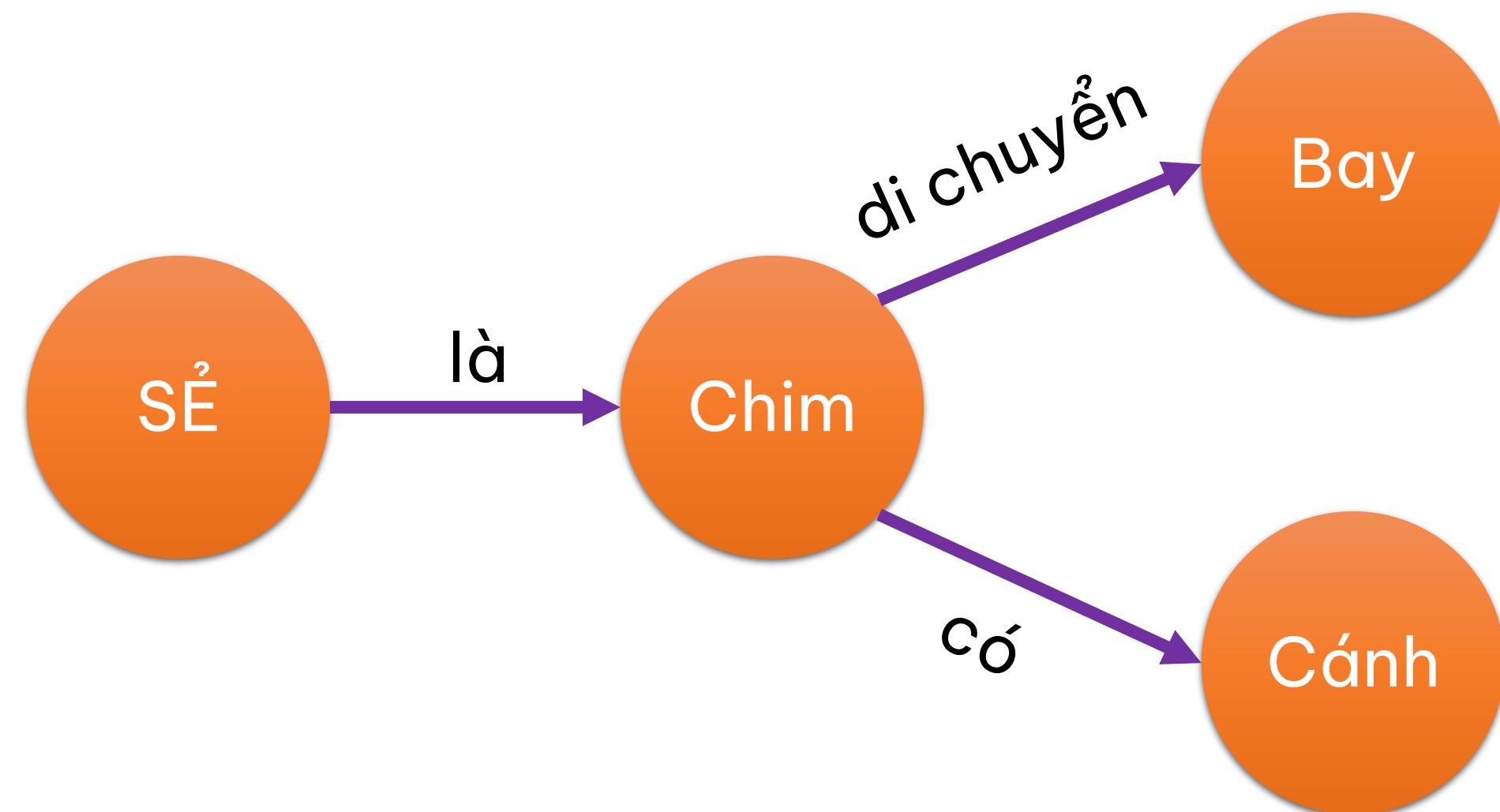
Mở rộng cho các luật

- Luật có biến: Áp dụng cần thực hiện cùng một phép toán trên một tập hay các đối tượng.
 - IF X là viên chức AND X là nam giới AND Tuổi X>63
 - THEN X có thể nghỉ hưu
- Luật không chắc chắn: Sự kiện có thể không chắc chắn (mệnh đề phát biểu - luật đưa thêm hệ số chắc chắn CF).
 - IF Giá vàng tăng THEN Giá đất tăng, CF=0,9
- Siêu luật: mô tả cách thức dùng các luật khác (Không có TT mới)
 - IF máy không khởi động AND phần cứng tốt THEN sử dụng các luật liên quan đến Hệ điều hành

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

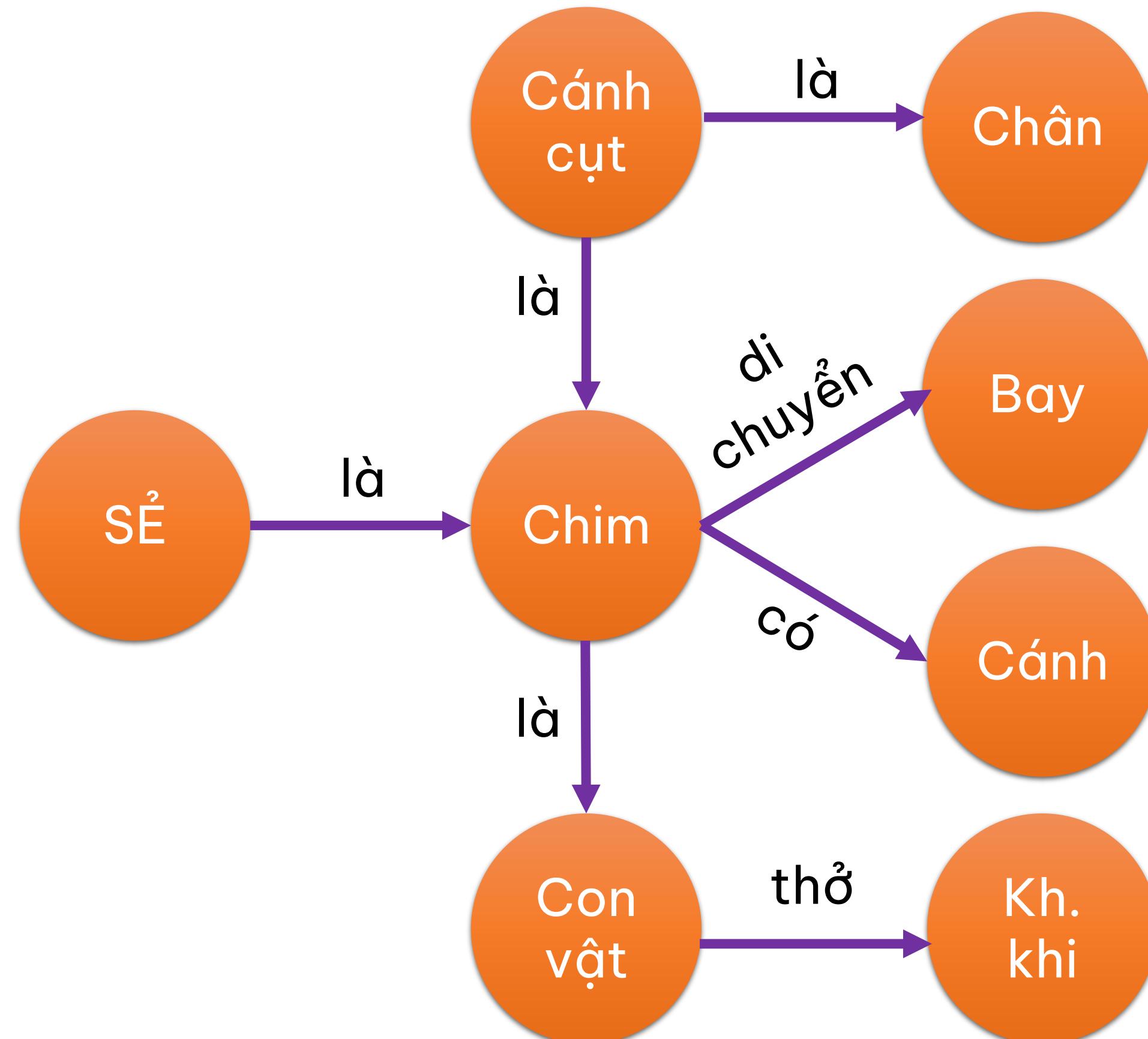
Mạng ngữ nghĩa

- Là phương pháp biểu diễn tri thức dùng đồ thị: Nút biểu diễn đối tượng; cung biểu diễn quan hệ.



Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Mạng ngữ nghĩa



- Mở rộng mạng ngữ nghĩa: Thêm các nút (đối tượng bổ sung) và nối vào đồ thị bằng các cung:
 - Thêm một đối tượng tương tự
 - Thêm đối tượng đặc biệt hơn
 - Thêm một đối tượng tổng quát hơn

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Mạng ngữ nghĩa

- Giải bài toán tam giác bằng mạng ngữ nghĩa, sử dụng ~ 200 đỉnh (chứa công thức) và 22 đỉnh (chứa yếu tố của tam giác).
- Đỉnh chứa công thức (ký hiệu hình chữ nhật)
- Đỉnh chứa yếu tố của tam giác (ký hiệu hình tròn)
- Cung: nối từ hình tròn – hình chữ nhật cho biết yếu tố tam giác xuất hiện trong công thức nào.

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

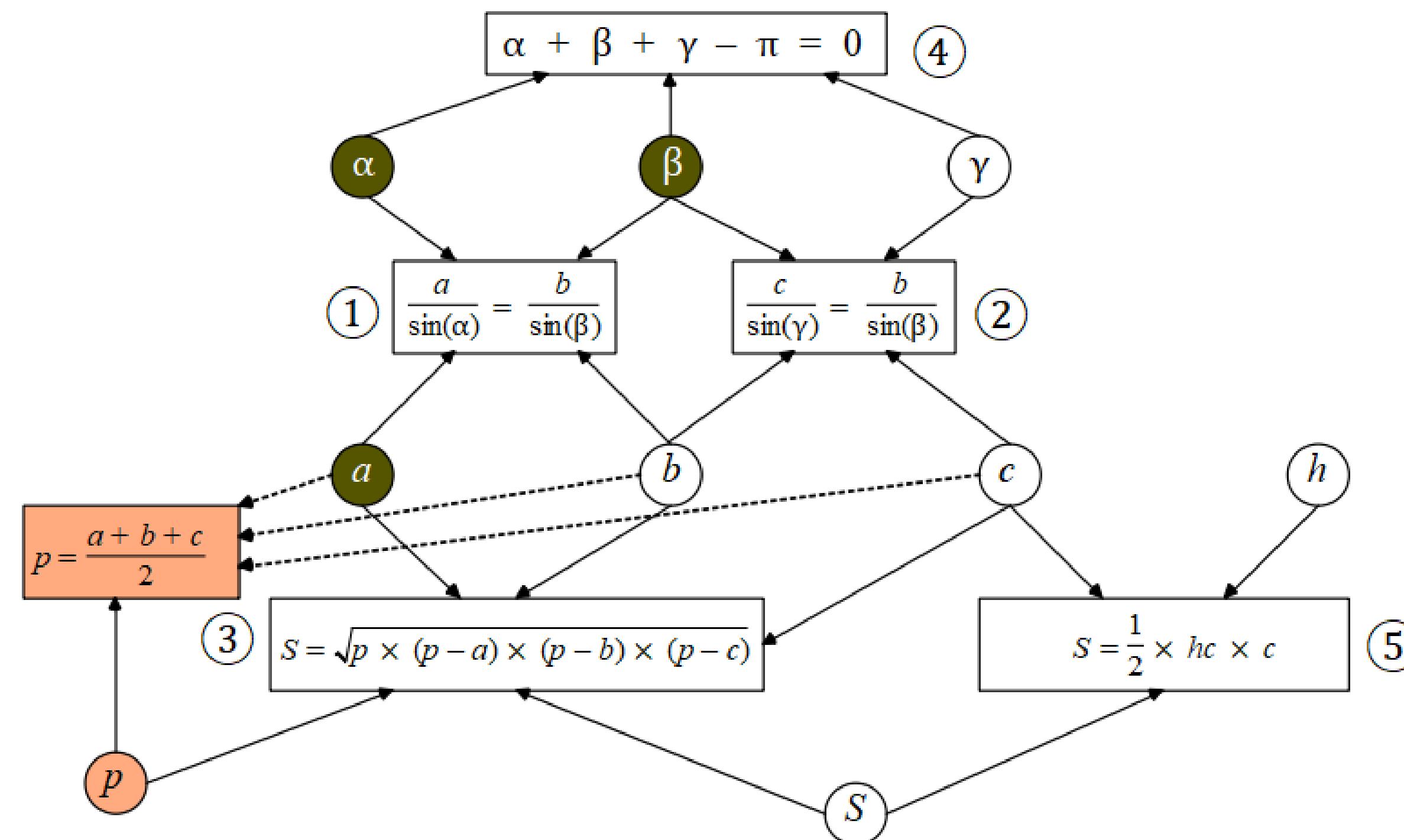
Mạng ngữ nghĩa

- Bước 1. Kích hoạt những đỉnh hình tròn đã cho ban đầu
- Bước 2. Lặp lại bước sau cho đến khi kích hoạt tất cả các đỉnh ứng với các yếu tố cần tính hoặc không thể kích hoạt được bất kỳ đỉnh nào nữa.
 - Nếu một đỉnh hình chữ nhật có cung nối với n đỉnh hình tròn mà $n-1$ đỉnh hình tròn đã được kích hoạt thì kích hoạt đỉnh hình tròn còn lại (và tính giá trị đỉnh còn lại này thông qua công thức ở đỉnh hình chữ nhật).

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Mạng ngữ nghĩa

- Cho 2 góc α, β và chiều dài cạnh a của tam giác. Tính chiều cao hc .



Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Frames

- Có hình thức như bảng mẫu, tờ khai cho phép điền vào chỗ trống.
- Cấu trúc cơ bản:
 - Tên frame
 - Lớp
 - Các thuộc tính: Biểu diễn như O - A - V

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Frames

Tên Frame

Lớp

Thuộc tính

TT	GT
Thuộc tính 1	Giá trị 1
Thuộc tính 2	Giá trị 2
...	...
Thuộc tính n	Giá trị n

PHIẾU BÁO ĐIỂM

Họ tên

MSSV

Môn

Điểm

CSDL

7

Toán rời rạc

8

Kiểm thử PM

6

CTDL & GT

7

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Logic

- Sử dụng ký hiệu để thể hiện tri thức & toán tử

Phép toán	NOT	AND	OR	Kéo theo	Tương đương
Ký hiệu	$\neg \sim$	$\cap \wedge \&$	$UV+$	$\rightarrow \supset$	$\equiv \leftrightarrow$

- Logic mệnh đề
- Logic vị từ

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Logic

- Logic mệnh đề
 - IF Xe không khởi động được (A)
 - AND Khoảng cách từ nhà đến chỗ làm là xa (B)
 - THEN Trễ giờ làm (C)
- Luật trên có thể biểu diễn lại như sau:
 - $A \cap B \rightarrow C$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Logic

- Logic vị từ
 - Phép toán mệnh đề → suy diễn tự động nhưng chưa đủ khi cần phải truy cập vào thành phần nhỏ trong câu, dùng biến số trong câu.
 - Ví dụ: “Mọi sinh viên trường ĐHNT đều có bằng tú tài. Lan không có bằng tú tài. Do vậy, Lan không là sinh viên ĐHNT”
 - “Lan” là một đối tượng cụ thể của “SV trường ĐHNT” – không thể đặc tả được “quan hệ” này trong mệnh đề được mà chỉ có thể là:
 - “LAN là sinh viên trường ĐHNT thì Lan có bằng tú tài. Lan không có bằng tú tài. Do vậy, Lan không là sinh viên trường ĐHNT”

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Logic

- Mệnh đề phải giải quyết bằng cách liệt kê tất cả các trường hợp
→ Không khả thi
- Do đó, chúng ta cần một Logic khác hơn là phép toán mệnh đề:
- **Vị từ là một phát biểu nói lên quan hệ giữa một đối tượng với các thuộc tính của nó hay quan hệ giữa các đối tượng với nhau.**
- Vị từ được biểu diễn bởi một tên được gọi là tên vị từ, theo sau nó là một danh sách các thông số.

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Logic

- Ví dụ:
- Phát biểu: “Nam là sinh viên trường ĐHNT”
- Biểu diễn: sv_NT(“Nam”)
- Ý nghĩa: đối tượng tên là “Nam” có thuộc tính là “sinh viên ĐHNT”.
- Biểu thức Vị từ: là sự kết hợp của các vị từ bởi các phép toán vị từ.

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Logic

- Các phép toán

Ký hiệu	Ý nghĩa	Số ngôi
\neg	Phủ định	Một ngôi
$\forall X$	Với mọi	Một ngôi
$\exists X$	Tồn tại	Một ngôi
\cap	Hội	Hai ngôi
\cup	Tuyển	Hai ngôi
\rightarrow	Kéo theo	Hai ngôi
\leftrightarrow	Tương đương	Hai ngôi

- Chuyển các câu sau sang biểu thức vị từ:
 - “Mọi sinh viên trường ĐHNT đều có bằng tú tài. Lan không có bằng tú tài. Do vậy, Lan không là sinh viên trường ĐHNT”.
 - $\text{sv_NT}(X)$: “ X là sinh viên trường DHNT”
 - $\text{tu_tai}(X)$: “ X có bằng tú tài”
 - Các câu trên được chuyển qua vị từ là:
 - $\forall X(\text{sv_NT}(X) \rightarrow \text{tu_tai}(X))$.
 - $\neg\text{tu_tai}(\text{"Lan"})$.
 - Do vậy, $\neg\text{sv_NT}(\text{"Lan"})$.

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Logic

- Đặt $C(x)$: “ x có 1 con mèo”, $D(x)$: “ x có 1 con chó”, $F(x)$: “ x có 1 con chồn”. Biểu diễn các phát biểu sau theo $C(x)$, $D(x)$, $F(x)$, các lượng từ và các phép nối logic.
- Xét không gian biên là các Sinh viên trong lớp.
- a. Một SV trong lớp có 1 con mèo, 1 con chó hoặc 1 con chồn:
 - $\exists x, C(x) \cup D(x) \cup F(x)$
- b. Tất cả SV trong lớp có 1 con mèo, 1 con chó hoặc 1 con chồn:
 - $\forall x, C(x) \cup D(x) \cup F(x)$
- c. Một SV nào đó có 1 con mèo và 1 con chồn nhưng không có chó:
 - $\exists x, C(x) \cap F(x) \cap \neg D(x)$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Logic

- Đặt $C(x)$: “ x có 1 con mèo”, $D(x)$: “ x có 1 con chó”, $F(x)$: “ x có 1 con chồn”. Biểu diễn các phát biểu sau theo $C(x)$, $D(x)$, $F(x)$, các lượng từ và các phép nối logic.
- Xét không gian biến là các Sinh viên trong lớp.
- d. Không có SV nào trong lớp có 1 con mèo, 1 con chó và 1 con chồn:
 - $\neg \exists x, C(x) \cap D(x) \cap F(x)$
- e. Mỗi SV trong lớp có ít nhất 1 con mèo, 1 con chó hoặc 1 con chồn:
 - $\forall x, y, z, C(x) \cup D(y) \cup F(z)$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Logic

- Đặt $L(x)$: “ x là 1 nhà Logic”, $C(x)$: “ x uống cafe”, $W(x)$: “ x làm việc chăm chỉ”. $T(x)$: “ x phát biểu định lý”.
- Hàm $f(x)$: trả về giá trị là bạn của x (mỗi người có đúng 1 bạn).
- Phát biểu các câu sau dưới dạng logic bậc 1.
 - a. Không nhà logic nào uống cafe:
 - $\forall x, L(x) \rightarrow \neg C(x)$
 - b. Bất kỳ nhà Logic nào cũng là bạn của ai đó:
 - $\forall x, L(x) \rightarrow \exists y, x = f(y)$
 - c. Không ai phát biểu được định lý lại có một người bạn uống cafe:
 - $\forall x, T(x) \rightarrow \exists y, y = f(x) \cap C(y)$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Logic

- Đặt $L(x)$: “ x là 1 nhà Logic”, $C(x)$: “ x uống cafe”, $W(x)$: “ x làm việc chăm chỉ”. $T(x)$: “ x phát biểu định lý”.
- Hàm $f(x)$: trả về giá trị là bạn của x (mỗi người có đúng 1 bạn).
- Phát biểu các câu sau dưới dạng logic bậc 1.
 - d. Bất kỳ ai có một người bạn làm việc chăm chỉ hoặc là 1 nhà Logic hoặc cũng là một người làm việc chăm chỉ:
 - $\forall x, \exists y, y = f(x) \cap W(y) \rightarrow L(x) \cup W(x)$
 - b. Bất kỳ người bạn nào cũng là nhà Logic:
 - $\forall x, y, y = f(x) \rightarrow L(y)$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

Các luật Logic mệnh đề

$$1. A \rightarrow B \equiv \neg A \cup B$$

$$2. A \cap (B \cup C) \equiv A \cap B \cup A \cap C$$

$$3. A \cup (B \cap C) \equiv A \cup B \cap A \cup C$$

$$4. \overline{A \cup B} \equiv \neg A \cap \neg B \text{ (De Morgan)}$$

$$5. \overline{A \cap B} \equiv \neg A \cup \neg B \text{ (De Morgan)}$$

$$6. A \leftrightarrow B \equiv A \cap B \cup \neg A \cap \neg B$$

$$7. A \cap (A \cup B) \equiv A$$

$$8. A \cup A \cup B \equiv A$$

$$9. A \cap B \cup \neg B \equiv A \cup \neg B$$

$$10. (A \cup B) \cap \neg B \equiv A \cup \neg B$$

$$11. A \cap B \equiv B \cap A$$

$$12. A \cup B \equiv B \cup A$$

$$13. (A \cap B) \cap C \equiv A \cap (B \cap C)$$

$$14. (A \cup B) \cup C \equiv A \cup (B \cup C)$$

$$15. A \cap A \equiv A$$

$$16. A \cup A \equiv A$$

$$17. A \cap \neg A \equiv 0$$

$$18. A \cup \neg A \equiv 1$$

$$19. A \cap 0 \equiv 0$$

$$20. A \cup 0 \equiv 0$$

$$21. A \cap 1 \equiv A$$

$$22. A \cup 1 \equiv 1$$

$$23. \neg \neg A \equiv A$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Vương Hao (Wang Hao)

Bước 1: Phát biểu lại giả thiết và kết luận của vấn đề dưới dạng chuẩn như sau: $GT_1, GT_2, \dots, GT_n \rightarrow KL_1, KL_2, \dots, KL_m$

Trong đó các GT_i và KL_j được xây dựng từ các biến mهن đề và các phép toán $\cup \cap \neg$

Bước 2: Chuyển về các GT_i và KL_j có dạng phủ định.

$$(p \rightarrow q, \neg(r \rightarrow s), \neg q, p \cup r) \rightarrow (s, \neg p) \equiv (p \rightarrow q, p \cup r, p) \rightarrow (s, r \rightarrow s, q)$$

Bước 3. Thay dấu \cap trong GT_i và dấu \cup trong KL_j bằng dấu ,

$$p \cap q, r \cap (p \cup s) \rightarrow q \cup r \equiv p, q, r, (p \cup s) \rightarrow q, r$$

Bước 4. Nếu dòng hiện hành có 1 trong 2 dạng sau thì tiến hành tách dòng

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Thuật giải Vương Hao (Wang Hao)

Dạng 1: $GT_1, GT_2, \dots, a \cup b, \dots, GT_n \rightarrow KL_1, KL_2, \dots, KL_m$ thì thay bằng 2 dòng:

$$\begin{cases} GT_1, GT_2, \dots, a, \dots, GTn \rightarrow KL_1, KL_2, \dots, KLM \\ GT_1, GT_2, \dots, b, \dots, GTn \rightarrow KL_1, KL_2, \dots, KLM \end{cases}$$

Dạng 2: $GT_1, GT_2, \dots, GT_n \rightarrow KL_1, KL_2, \dots, a \cap b, \dots, KL_m$ thì thay bằng 2 dòng:

$$\begin{cases} GT_1, GT_2, \dots, a, \dots, GTn \rightarrow KL_1, KL_2, \dots, KLM \\ GT_1, GT_2, \dots, b, \dots, GTn \rightarrow KL_1, KL_2, \dots, KLM \end{cases}$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Thuật giải Vương Hao (Wang Hao)

Bước 5. Nếu một dòng được chứng minh: nếu tồn tại chung một mệnh đề ở cả 2 vế thì coi như đúng

$$p, q \rightarrow p: \text{mệnh đề đúng}$$

Bước 6. Nếu một dòng không còn dấu liên kết tuyển "U" và hội "n" mà cả 2 vế đều không có chung biến mệnh đề nào thì dòng đó không được CM.

$$p, \neg q \rightarrow q$$

Một vấn đề được giải quyết một cách trọn vẹn nếu mọi dòng dẫn xuất từ dạng chuẩn được chứng minh.

TỪ BƯỚC 2 – 4 KHÔNG CẦN THỰC HIỆN THEO THỨ TỰ

Chương 6. BIỂU ĐIỂN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Vương Hao (Wang Hao)

Cho logic mệnh đề: $\{p \rightarrow q, q \rightarrow r\}. CMR \{p \rightarrow r\}$

Bước 0. Chuyển tất cả biểu thức về dạng $\cup \cap \neg$ dựa vào các luật Logic

$$p \rightarrow q \equiv \neg p \cup q \text{ (Logic 1)}$$

$$q \rightarrow r \equiv \neg q \cup r \text{ (Logic 1)}$$

$$p \rightarrow r \equiv \neg p \cup r \text{ (Logic 1)}$$

Bước 1. Biểu diễn bài toán dưới dạng $GT_i \rightarrow KL_j$

$$(\neg p \cup q, \neg q \cup r) \rightarrow (\neg p \cup r)$$

Bước 3. Thay dấu \cap trong GT_i và dấu \cup trong KL_j thành dấu ,

$$(\neg p \cup q, \neg q \cup r) \rightarrow (\neg p, r)$$

Bước 2. Chuyển về từ $KL \leftrightarrow GT$ nếu trong biểu thức có dấu \neg

$$(\neg p \cup q, \neg q \cup r, p) \rightarrow (r)$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Thuật giải Vương Hao (Wang Hao)

Biểu thức: $(\neg p \cup q, \neg q \cup r, p) \rightarrow (r)$ thỏa dạng 1 nên có thể tách thành 2 dòng

$GT_1, GT_2, \dots, a \cup b, \dots, GT_n \rightarrow KL_1, KL_2, \dots, KL_m$

Bước 4.

Dòng 1: $(\neg p, \neg q \cup r, p) \rightarrow (r)$ (bt.1)

Dòng 2: $(q, \neg q \cup r, p) \rightarrow (r)$ (bt.2)

Xét biểu thức bt.1:

$(\neg p, \neg q \cup r, p) \rightarrow (r) \equiv (\neg q \cup r, p) \rightarrow (p, r)$: ĐPCM

Xét biểu thức bt.2:

$(q, \neg q \cup r, p) \rightarrow (r)$

Biểu thức thỏa dạng chuẩn 1: $GT_1, GT_2, \dots, a \cup b, \dots, GT_n \rightarrow KL_1, KL_2, \dots, KL_m$ nên có thể tách thành 2 dòng.

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Vương Hao (Wang Hao)

Biểu thức $(q, \neg q \cup r, p) \rightarrow (r)$ được tách thành 2 dòng

$(q, \neg q, p) \rightarrow (r)$: (bt.3)

$(q, r, p) \rightarrow (r)$: (bt.4)

Xét biểu thức bt.3:

$(q, \neg q, p) \rightarrow (r)$: thực hiện Bước 2

$(\textcolor{yellow}{q}, p) \rightarrow (\textcolor{yellow}{q}, r)$ (ĐPCM)

Xét biểu thức bt.4:

$(q, \textcolor{yellow}{r}, p) \rightarrow (\textcolor{yellow}{r})$: (ĐPCM)

Chương 6. BIỂU ĐIỂN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Vương Hao (Wang Hao)

Biểu thức $\{(a \cap b) \rightarrow c, (b \cap c) \rightarrow d, \neg d\}$ CMR $\{a \rightarrow b\}$

Bước 0. Chuyển tất cả biểu thức về dạng $\cup \cap \neg$ dựa vào các luật Logic

$$(a \cap b) \rightarrow c \equiv \neg(a \cap b) \cup c \text{ (Logic 1)} \equiv \neg a \cup \neg b \cup c \text{ (Logic 5)}$$

$$(b \cap c) \rightarrow d \equiv \neg(b \cap c) \cup d \text{ (Logic 1)} \equiv \neg b \cup \neg c \cup d \text{ (Logic 5)}$$

$$(a \rightarrow b) \equiv \neg a \cup b \text{ (Logic 1)}$$

Bước 1. Biểu diễn bài toán dưới dạng $GT_i \rightarrow KL_j$

$$(\neg a \cup \neg b \cup c, \neg b \cup \neg c \cup d, \neg d) \rightarrow (\neg a \cup b)$$

Bước 3. Thay dấu \cap trong GT_i và dấu \cup trong KL_j thành dấu ,

$$(\neg a \cup \neg b \cup c, \neg b \cup \neg c \cup d, \neg d) \rightarrow (\neg a, b)$$

Bước 2. Chuyển về từ $KL \leftrightarrow GT$ nếu trong biểu thức có dấu \neg

$$(\neg a \cup \neg b \cup c, \neg b \cup \neg c \cup d) \rightarrow (\neg a, b, d)$$

Chương 6. BIỂU ĐIỂN VÀ SUY LUẬN TRI THỨC

▪ Thuật giải Vương Hao (Wang Hao)

Biểu thức $(\neg a \cup \neg b \cup c, \neg b \cup \neg c \cup d) \rightarrow (\neg a, b, d)$ thỏa dạng 1 nên có thể tách thành 2 dòng

$$GT_1, GT_2, \dots, a \cup b, \dots, GT_n \rightarrow KL_1, KL_2, \dots, KL_m$$

Bước 4.

Dòng 1: $(\neg a \cup \neg b \cup c, \neg b \cup \neg c) \rightarrow (\neg a, b, d)$ (bt.1)

Dòng 2: $(\neg a \cup \neg b \cup c, d) \rightarrow (\neg a, b, d)$ (ĐPCM)

Xét biểu thức bt.1: $(\neg a \cup \neg b \cup c, \neg b \cup \neg c) \rightarrow (\neg a, b, d)$ thỏa dạng 1 $GT_1, GT_2, \dots, a \cup b, \dots, GT_n \rightarrow KL_1, KL_2, \dots, KL_m$ nên có thể tách thành 2 dòng

Bước 4.

Dòng 1: $(\neg a, \neg b \cup \neg c, \neg b \cup \neg c) \rightarrow (\neg a, b, d)$ (ĐPCM)

Dòng 2: $(\neg b \cup c, \neg b \cup \neg c) \rightarrow (\neg a, b, d)$ (bt.2)

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Vương Hao (Wang Hao)

Xét bt.2: $(\neg b \cup c, \neg b \cup \neg c) \rightarrow (\neg a, b, d) \equiv (\neg b \cup c, \neg(b \cap c)) \rightarrow (\neg a, b, d)$ (Logic 5)

$\equiv (\neg b \cup c \rightarrow (\neg a, b, d, b \cap c)$ thỏa dạng 1 $GT_1, GT_2, \dots, a \cup b, \dots, GT_n \rightarrow KL_1, KL_2, \dots$

KL_m nên có thể tách thành 2 dòng

Dòng 1: $\neg b \rightarrow \neg a, b, d, b \cap c$ (bt.3)

Dòng 2: $c \rightarrow \neg a, b, d, b \cap c$ (bt.4)

Xét bt.3 thỏa dạng 2: $GT_1, GT_2, \dots, GT_n \rightarrow KL_1, KL_2, \dots, a \cap b, \dots, KL_m$ thì thay bằng 2 dòng:

Dòng 1: $\neg b \rightarrow \neg a, b, d, b$ (bt.3.1): (KĐCM)

Dòng 2: $\neg b \rightarrow \neg a, b, d, c$ (bt.3.2): (KĐCM)

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Thuật giải Vương Hao (Wang Hao)

$$c \rightarrow \neg a, b, d, b \cap c$$

Xét bt.4 thoả dạng 2: $GT_1, GT_2, \dots, GT_n \rightarrow KL_1, KL_2, \dots, a \cap b, \dots, KL_m$ thì thay bằng 2 dòng:

Dòng 1: $c \rightarrow \neg a, b, d, b$ (bt.4.1): (KĐCM)

Dòng 2: $c \rightarrow \neg a, b, d, c$ (bt.4.2): (ĐPCM)

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Thuật giải Vương Hao (Wang Hao)

Cho biểu thức $\{p \rightarrow q, q \rightarrow r, r \rightarrow s, p\}$ CMR $\{p \cap s\}$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Vương Hao (Wang Hao)

Cho biểu thức $\{a \cap b \rightarrow c, b \cap c \rightarrow d, a \cap b\}$ Kết luận $\{d\}$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Thuật giải Vương Hao (Wang Hao)

Cho biểu thức $\{a \rightarrow c \cup b, c \rightarrow d\}$ Kết luận $\{a \rightarrow d\}$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Robinson

Bước 1: Phát biểu lại giả thiết và kết luận của vấn đề dưới dạng chuẩn như sau: $GT_1, GT_2, \dots, GT_n \rightarrow KL_1, KL_2, \dots, KL_m$

Trong đó các GT_i và KL_j được xây dựng từ các biến mệnh đề và các phép toán $\cup \cap \neg$

Bước 2: Biến đổi dòng trên thành danh sách các mệnh đề.

$GT_1, GT_2, \dots, GT_n, \neg KL_1, \neg KL_2, \dots, \neg KL_m$

Bước 3: Nếu trong danh sách các mệnh đề ở Bước 2 có các mệnh đề đối ngẫu ($a, \neg a$) thì vấn đề đã được giải quyết và chuyển sang Bước 4.

Bước 4: Xây dựng mệnh đề mới bằng cách tuyển một cặp mệnh đề từ danh sách mệnh đề ở Bước 2, Nếu mệnh đề mới các biến mệnh đề đối ngẫu thì các biến mệnh đề đó được loại bỏ. $(p \cup \neg q), (q \cup r) \equiv (p \cup \neg q \cup q \cup r) \equiv p \cup r$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Robinson

Bước 5: Bổ sung mệnh đề mới vào danh sách và loại bỏ 2 mệnh đề cũ vừa tạo thành mệnh đề mới ra khỏi danh sách.

Bước 6: Nếu không xây dựng thêm mệnh đề mới nào và trong danh sách có các mệnh đề không có 2 mệnh đề nào đối ngẫu nhau thì vấn đề đã phát biểu ở Bước 1 là sai.

Một số công thức cơ bản:

$$1. A \rightarrow B \equiv \neg A \cup B$$

$$2. \overline{A \cup B} \equiv \neg A \cap \neg B \text{ (De Morgan)}$$

$$3. \overline{A \cap B} \equiv \neg A \cup \neg B \text{ (De Morgan)}$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Robinson

Sử dụng giải thuật Robison, kiểm tra mệnh đề sau:

$$\{p \rightarrow q, q \rightarrow r\} \rightarrow \{p \rightarrow r\}$$

Ta có: $p \rightarrow q \equiv \neg p \cup q$

$q \rightarrow r \equiv \neg q \cup r$

$p \rightarrow r \equiv \neg p \cup r$

Bước 1: $(\neg p \cup q, \neg q \cup r) \rightarrow (\neg p \cup r)$

Bước 2: $\neg p \cup q, \neg q \cup r, \neg(\neg p \cup r)$

$\equiv \neg p \cup q, \neg q \cup r, \neg\neg p \cap \neg r$ (Logic 5, De Morgan)

$\equiv \neg p \cup q, \neg q \cup r, p, \neg r$ (Biến đổi dấu \cap thành dấu $,$)

$\equiv \neg p \cup r, p, \neg r \equiv r, \neg r \equiv 0$ (ĐPCM)

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Robinson

Sử dụng giải thuật Robison, kiểm tra mệnh đề sau:

$\{a \rightarrow b, a \rightarrow c \cup e, b \cap c \rightarrow d, e \rightarrow f, f \cup d \rightarrow g, a\}$. Chứng minh $\{g\}$ được dẫn nhập

Ta có: $a \rightarrow b \equiv \neg a \cup b$

$a \rightarrow c \cup e \equiv \neg a \cup c \cup e$

$b \cap c \rightarrow d \equiv \neg(b \cap c) \cup d \equiv \neg b \cup \neg c \cup d$ (*De Morgan*)

$e \rightarrow f \equiv \neg e \cup f$

$f \cup d \rightarrow g \equiv \neg(f \cup d) \cup g \equiv \neg f \cap \neg d \cup g$ (*De Morgan*)

Bước 1: $(\neg a \cup b, \neg a \cup c \cup e, \neg b \cup \neg c \cup d, \neg e \cup f, \neg f \cap \neg d \cup g, a) \rightarrow (g)$

Bước 2: $\neg a \cup b, \neg a \cup c \cup e, \neg b \cup \neg c \cup d, \neg e \cup f, \neg f \cap \neg d \cup g, a, \neg g$

$\equiv \neg b, \neg a \cup c \cup e, \neg b \cup \neg c \cup d, \neg e \cup f, \neg f \cap \neg d \cup g, \neg g$

$\equiv \neg a \cup \neg c \cup e, \neg b \cup \neg d, \neg e \cup f, \neg f \cap \neg d \cup g, \neg g$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Robinson

$$\equiv \neg a \cup \textcolor{red}{c} \cup e, \neg \textcolor{red}{c} \cup d, \neg e \cup f, \neg f \cap \neg d \cup g, \neg g$$

$$\equiv \neg a \cup \textcolor{red}{e} \cup d, \neg \textcolor{red}{e} \cup f, \neg f \cap \neg d \cup g, \neg g$$

$$\equiv \neg a \cup d \cup f, \neg f \cap \neg d \cup \textcolor{red}{g}, \neg g$$

$$\equiv \neg a \cup d \cup f, \neg f \cap \neg d \equiv \neg a \cup \textcolor{red}{d} \cup f, \neg(f \cup d) \text{ (De Morgan)}$$

$$\equiv \neg a \text{ (KĐCM)}$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Robinson

Sử dụng giải thuật Robison, kiểm tra mệnh đề sau:

$$\{p \rightarrow q, q \rightarrow r, r \rightarrow s, p\} \rightarrow \{p \cap s\}$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Robinson

Sử dụng giải thuật Robison, kiểm tra mệnh đề sau:

$$\{a \cap b \rightarrow c, b \cap c \rightarrow d, a \cap b\}. Hỏi \{d\}$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Robinson

Sử dụng giải thuật Robison, kiểm tra mệnh đề sau:

$$\{\neg p \cup q, \neg q \cup r, \neg r \cup s, \neg u \cup \neg s\} \rightarrow \{\neg p, \neg u\}$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Robinson

Sử dụng giải thuật Robison, kiểm tra mệnh đề sau:

$$\{a \rightarrow b \cup d, d \rightarrow e \cap f, e \cap a \rightarrow \neg b\} \rightarrow \{a \rightarrow \neg d\}$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Robinson

Sử dụng giải thuật Robison, kiểm tra mệnh đề sau:

$$\{a \cap b \rightarrow c, b \cap c \rightarrow d\} \rightarrow \{a \cap b \rightarrow d\}$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Thuật giải Robinson

Sử dụng giải thuật Robison, kiểm tra mệnh đề sau:

$$\{a \rightarrow b \cap c, c \rightarrow e \cup f, b \rightarrow \neg e, a\} \rightarrow \{f\}$$

Chương 6. BIỂU ĐIỂN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn tiến

Câu chuẩn Horn có dạng:

$$Q \cup \neg P_1 \cup \neg P_2 \cup \dots \cup P_m$$

$$\text{Hay } P_1 \cap P_2 \cap \dots \cap P_m \rightarrow Q$$

Câu Horn còn gọi là luật nếu – thì (if–then) và viết lại dưới dạng:

Nếu P_1 và P_2 và ... và P_m thì Q

Trong đó:

P_1, P_2, \dots, P_m : Các điều kiện của luật

Q : Kết luận của luật

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn tiến

Hay ta có thể biểu diễn:

Vào: - Tập các mệnh đề đã cho $GT = \{gt_1, gt_2, \dots, gtm\}$

- Tập các luật $R = \{r_1, r_2, \dots, r_m\}$, với $r_i = p_1 \cap p_2 \cap \dots \cap p_n \rightarrow q$ ($i = 1, n$)

- Tập kết luận $KL = \{q_1, q_2, \dots, q_k\}$

Ra: Thông báo thành công nếu $\forall q_i$ ($i = 1, k$) đều được suy diễn từ GT và R

Lưu ý:

Để thực hiện suy diễn tiến thì bắt buộc các luật phải có dạng chuẩn Horn.

Tức là Về trái có thể có nhiều sự kiện nhưng chỉ có điều kiện “và”. Còn Về phải của mỗi luật chỉ có 1 sự kiện duy nhất.

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn tiến

Cho tập các luật sau:

- | | |
|-----------------------------|------------------------------|
| 1. $x \cap b \rightarrow c$ | 7. $e \cap k \rightarrow f$ |
| 2. $x \cap h \rightarrow d$ | 8. $f \rightarrow n$ |
| 3. $a \cap c \rightarrow e$ | 9. $c \rightarrow k$ |
| 4. $a \cap d \rightarrow l$ | 10. $a \cap k \rightarrow e$ |
| 5. $e \cap b \rightarrow n$ | 11. $e \cap p \rightarrow n$ |
| 6. $n \cap e \rightarrow l$ | 12. $e \rightarrow q$ |

Dùng suy diễn tiến chứng minh:

$$GT = \{a \cap b \cap x\}, KL = \{l\}$$

Hoặc có thể viết $GT = \{a, b, x\}$ hỏi có a,b,x có suy ra được l hay không?

r: luật được lấy ra để sử dụng

TG: Tập trung gian chứa các sự kiện nhận giá trị là đúng tới thời điểm hiện tại

SAT: chứa các luật thỏa điều kiện (r thỏa khi vế trái của r là tập con của TG)

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

r	TG	SAT	R	Ghi chú
Khởi tạo	a,b,x	r1	r1,r2,...,r11,r12	VT(a,b,x): r1 trong khi r2 (h), r3(c)
r1	a,b,x,c	r3,r9	r2,r3,...,r11,r12	r1: bổ sung c ở vế phải vào TG. Chỉ có r3, r9 thỏa c ở vế trái Ri
r3	a,b,x,c,e	r9,r5,r12	r2,r4,..r9..,r11,r12	r3: bổ sung e ở vế phải vào TG Chỉ có r5, r12 thỏa e ở vế trái Ri
r9	a,b,x,c,e,k	r5,r12,r7,r10	r2,r4,r5,..,r11,r12	r9: bổ sung k vào vế phải của TG chỉ có r7,r10 thỏa k ở vế trái Ri
r5	a,b,x,c,e,k,n	r12,r7,r10,r6	r2,r4,..,r11,r12	r6: bổ sung n vào vế phải của TG chỉ có r6 thỏa n ở vế trái Ri
r12	a,b,x,c,e,k,n,q	r7,r10,r6	r2,r4,..,r7,r11	r12: bổ sung q vào vế phải của TG Không có R thỏa q ở vế trái Ri
r7	a,b,x,c,e,k,n,q,f	r10,r6,r8	r2,r4,..,r10,r11	r7: bổ sung f vào vế phải của TG chỉ có r8 thỏa f ở vế trái Ri
r10	a,b,x,c,e,k,n,q,f,e	r6,r8	r2,r4,r6,..,r10,r11	r10: bổ sung e vào vế phải của TG Không có luật thỏa e ở vế trái Ri mà chưa thêm
r6	a,b,x,c,e,k,n,q,f,e,l	r8	r2,r4,r6,..,r10,r11	r6: bổ sung l vào vế phải của TG l đã có ở Kết luận: DÙNG

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn tiến

Cho tập các luật sau:

$$1. \quad a \cap b \rightarrow c$$

$$2. \quad b \cap c \rightarrow d$$

Dùng suy diễn tiến chứng minh:

$$GT = \{a \cap b\}, KL = \{d\}$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn tiến

Cho tập các luật sau:

- | | | |
|----------------------|-----------------------------|-----------------------------|
| 1. $a \rightarrow c$ | 3. $c \rightarrow e$ | 5. $b \cap c \rightarrow f$ |
| 2. $b \rightarrow d$ | 4. $a \cap d \rightarrow e$ | 6. $e \cap f \rightarrow g$ |

Dùng suy diễn tiến chứng minh:

$$GT = \{a \cap b\}, KL = \{g\}$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

■ Suy diễn tiến

Cho tập các luật sau:

- | | |
|-----------------------------|----------------------|
| 1. $a \cap c \rightarrow f$ | 3. $b \rightarrow e$ |
| 2. $a \cap e \rightarrow g$ | 4. $a \rightarrow d$ |

Dùng suy diễn tiến chứng minh:

$$GT = \{a, b\}, KL = \{a \cap d \rightarrow d\}$$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn tiến

Cho tập các luật sau:

$$1. \quad a \rightarrow c$$

$$4. \quad c \cap p \rightarrow f$$

$$7. \quad t \rightarrow i$$

$$2. \quad a \rightarrow p$$

$$5. \quad d \cap p \cap f \rightarrow h$$

$$8. \quad c \cap k \rightarrow i$$

$$3. \quad a \cap b \rightarrow d$$

$$6. \quad d \cap p \cap f \rightarrow k$$

$$9. \quad q \cap b \rightarrow i$$

Dùng suy diễn tiến chứng minh:

GT: $\{a, b\}$, KL: $\{i = 1\}$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn lùi

Suy diễn lùi: Bắt đầu từ Kết luận và suy diễn ngược lại cho đến khi tồn tại Giải thiết thì kết luận đó là đúng.

GOAL: tập các mệnh đề cần phải chứng minh đến thời điểm đang xét

VET: Tập các luật đã sử dụng để chứng minh các đích (kể cả đích trung gian)

first: biến theo dõi số lần quay lui

$Timluat(f, i, RULE)$: Tìm từ i trở đi xem có luật nào sinh ra f hay không

$$GT = \{a, b\}, KL = \{d\}, RULE = \{r1: a \cap b \rightarrow c, r2: b \cap c \rightarrow d\}$$

GOAL={d}, suy diễn bắt đầu từ $r1 \rightarrow r2$, có d là kết luận của $r2$ mà b thuộc GT nên nạp c vào GOAL.

GOAL={c}, suy diễn các rule sinh ra c. Ta có $r1$ thỏa đk và a,b cùng thuộc GT đã biết nên c cũng đã biết $\rightarrow d$ cũng đã biết \rightarrow ĐPCM

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn lùi

$$GT = \{a, b\}, KL = \{d\}, RULE = \{r1: a \cap b \rightarrow c, r2: b \cap c \rightarrow d\}$$

Bước 0. $GOAL = \emptyset, VET = \emptyset, first = 0, KL \neq GT$.

nạp KL vào GOAL: $GOAL = \{d, 0\}$

Bước 1: Lấy $f = (d, 0)$ từ GOAL, $first = 1, f \notin GT$

$Timluat(d, 0, RULE, j) \rightarrow j = 2 // r2: b \cap c \rightarrow d$

$VET = \{d, 2\}$

Mà $b \in GT$ nên $GOAL = left_2 \setminus GT = \{c, 0\}$

$// left_2: vế trái rule 2(b,c) | GT \leftrightarrow \{b,c\} - b \in GT = \{c, 0\}$

Bước 2: Lấy $f = (c, 0)$ từ GOAL, $first = 2, f \notin GT$

$Timluat(c, 0, RULE, j) \rightarrow j = 1 // r1: a \cap b \rightarrow c$

$VET = \{c, 1\}$

Vì $a, b \in GT$ nên $GOAL = left_1 \setminus GT = \{\emptyset\}$ (Bài toán được chứng minh)

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn lùi

$GT = \{a\}, KL = \{m\}, RULE = \{r1: a \cap b \rightarrow c, r2: a \cap h \rightarrow d, r3: b \cap c \rightarrow e, r4: a \cap d \rightarrow m, r5: a \cap b \rightarrow o, r6: o \cap e \rightarrow m\}$

Bước 0. $GOAL = \emptyset, VET = \emptyset, first = 0, KL \neq GT$.

nạp KL vào GOAL: $GOAL = \{m, 0\}$

Bước 1: Lấy $f = (m, 0)$ từ GOAL, $first = 1, f \notin GT$

$Timluat(m, 0, RULE, j) \rightarrow j = 4 // r4: a \cap d \rightarrow m, r6: o \cap e \rightarrow m$. Chọn rule đầu tiên

$VET = \{m, 4\}$

Mà $a \in GT$ nên $GOAL = left_4 \setminus GT = \{d, 0\}$

//left₄: vẽ trái rule 4(a,d) | GT ← {a,d} - a ∈ GT = {d,0}

Bước 2: Lấy $f = (d, 0)$ từ GOAL, $first = 2, f \notin GT$

$Timluat(d, 0, RULE, j) \rightarrow j = 2 // r2: a \cap h \rightarrow d$

$VET = \{(d, 2), (m, 4)\}$

Vì $a \in GT$ nên $GOAL = left_2 \setminus GT = \{h\}$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn lùi

$GT = \{a\}, KL = \{m\}, RULE = \{r1: a \cap b \rightarrow c, r2: a \cap h \rightarrow d, r3: b \cap c \rightarrow e, r4: a \cap d \rightarrow m, r5: a \cap b \rightarrow o, r6: o \cap e \rightarrow m\}$

Bước 3: Lấy $f = (h, 0)$ từ GOAL, $first = 3, f \notin GT$

$Timluat(h, 0, RULE, j) \rightarrow j = 7$ //không có rule nào thỏa mãn

Quay trở lại VET tìm lại

Từ VET, ta có $g = \{d, 2\}$ //vào sau ra trước

$GOAL = \{\emptyset\}$ //do đã lấy h

$VET = \{m, 4\}$

$Timluat(d, 2, RULE, j) \rightarrow j = 7$ //không có rule nào thỏa mãn **vì rule 2 đã lấy trước đó**

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn lùi

$GT = \{a\}, KL = \{m\}, RULE = \{r1: a \cap b \rightarrow c, r2: a \cap h \rightarrow d, r3: b \cap c \rightarrow e, r4: a \cap d \rightarrow m, r5: a \cap b \rightarrow o, r6: o \cap e \rightarrow m\}$

Từ VET, ta có $g = \{m, 4\}$ //vào sau ra trước

$GOAL = \{\emptyset\}$ //do đã lấy d

$VET = \{\emptyset\}$

$Timluat(m, 4, RULE, j) \rightarrow j = 6$ //ta chỉ xét các luật từ $4+1=5$ trở đi

$VET = \{m, 6\}$

Vì $o, e \notin GT$ nên $GOAL = left_6 \setminus GT = \{(o, 0), (e, 0)\}$

Bước 4: Lấy $f = (o, 0)$ từ GOAL, $first = 4, f \notin GT$

$Timluat(o, 0, RULE, j) \rightarrow j = 5$ // $r5: a \cap b \rightarrow o$

$VET = \{(o, 5), (m, 6)\}$ //vào sau ra trước

Vì $a \in GT$ nên $GOAL = left_5 \setminus GT = \{(b, 0), (e, 0)\}$

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn lùi

$GT = \{a\}, KL = \{m\}, RULE = \{r1: a \cap b \rightarrow c, r2: a \cap h \rightarrow d, r3: b \cap c \rightarrow e, r4: a \cap d \rightarrow m, r5: a \cap b \rightarrow o, r6: o \cap e \rightarrow m\}$

Bước 5: Lấy $f = (\textcolor{red}{b}, 0)$ từ GOAL, $first = 5, f \notin GT$

$Timluat(\textcolor{red}{b}, 0, RULE, j) \rightarrow j = 7 // không có rule nào có KL = b$

Quay trở lại VET tìm lại

Từ VET, ta có $g = \{\textcolor{red}{o}, 5\} //$ vào sau ra trước

$GOAL = \{e, 0\} //$ do đã lấy (o,5)

$VET = \{m, 6\}$

$Timluat(\textcolor{red}{o}, 0, RULE, j) \rightarrow j = 7 //$ không có rule nào thỏa mãn **vì rule 2 đã lấy trước đó**

Quay trở lại VET tìm lại

Chương 6. BIỂU DIỄN VÀ SUY LUẬN TRI THỨC

▪ Suy diễn lùi

$GT = \{a\}, KL = \{m\}, RULE = \{r1: a \cap b \rightarrow c, r2: a \cap h \rightarrow d, r3: b \cap c \rightarrow e, r4: a \cap d \rightarrow m, r5: a \cap b \rightarrow o, r6: o \cap e \rightarrow m\}$

Từ VET, ta có $g = \{m, 6\}$ //vào sau ra trước

$GOAL = \{\emptyset\}$ //do đã lấy (m,6)

$VET = \{\emptyset\}$

$Timluat(m, 6, RULE, j) \rightarrow j = 7$ //không có rule nào thỏa mãn

Quay trở lại VET tìm lại

Từ VET, ta có $g = \emptyset = f$ (**KĐCM**)

HẾT CHƯƠNG 6

Chương 7. MÁY HỌC



TRÍ TUỆ NHÂN TẠO

Artificial Intelligence

Đoàn Vũ Thịnh
Khoa Công nghệ Thông tin
Đại học Nha Trang
Email: thinhdv@ntu.edu.vn

Nha Trang, 06-2023

Chương 7. MÁY HỌC

Trong chương 6 đã thảo luận về biểu diễn và suy luận tri thức. Trong trường hợp này giả định đã có sẵn tri thức và có thể biểu diễn tường minh tri thức.

Tuy vậy, trong nhiều tình huống sẽ không có sẵn tri thức như:

- Kỹ sư tri thức cần thu nhận tri thức từ chuyên gia cùng lĩnh vực
- Cần biết các luật mô tả lĩnh vực cụ thể
- Bài toán không được biểu diễn tường minh theo luật, sự kiện hay các quan hệ.

Chương 7. MÁY HỌC

- Hệ thống được gọi là có khả năng học (có dáng vẻ học như con người) là hệ thống có khả năng tìm ra một sự khái quát hoặc mô hình cho các dữ liệu huấn luyện (dữ liệu có gán nhãn nhận diện hoặc phân loại).
- Đặc trưng khái quát hoặc mô hình đó có thể được sử dụng để nhận diện hoặc phân loại dữ liệu mới.
- Hai hướng tiếp cận cho hệ thống học:
 - Học từ ký hiệu
 - Học từ dữ liệu số

Chương 7. MÁY HỌC

Các hình thức học

- **Học vẹt:** Hệ tiếp nhận các khẳng định của các quyết định đúng. Khi hệ tạo ra một quyết định không đúng, hệ sẽ đưa ra các luật hay quan hệ đúng mà hệ đã sử dụng. Hình thức học vẹt nhằm cho phép chuyên gia cung cấp tri thức theo kiểu tương tác.
- **Học bằng cách chỉ dẫn:** Thay vì đưa ra 1 luật cụ thể cần áp dụng vào tình huống cho trước, hệ thống sẽ được cung cấp bằng các chỉ dẫn tổng quát. Ví dụ “Sinh viên đạt học bổng khi có trung bình học kỳ từ 7.0 trở lên”. Hệ thống phải tự mình đề ra các biến đổi trừu tượng đến các luật khả dụng.

Chương 7. MÁY HỌC

Các hình thức học

- **Học bằng quy nạp:** Hệ thống được cung cấp một tập các ví dụ và kết luận được rút ra từ từng ví dụ. Hệ liên tục lọc ra các luật và quan hệ nhằm xử lý từng ví dụ mới.
- **Học bằng tương tự:** Hệ thống được cung cấp đáp ứng đúng cho các tác vụ tương tự nhưng không giống nhau. Hệ thống cần làm thích ứng đáp ứng trước đó nhằm tạo ra một luật mới có khả năng áp dụng cho tình huống mới.
- **Học dựa trên giải thích:** Hệ thống phân tích tập các lời giải ví dụ nhằm ấn định khả năng đúng/sai và tạo ra các giải thích dùng để hướng dẫn cách giải bài toán trong tương lai.

Chương 7. MÁY HỌC

Các hình thức học

- **Học dựa trên tình huống:** Bất kỳ tình huống nào được hệ thống lập luận đều được lưu trữ cùng với kết quả cho dù đúng hay sai. Khi gặp tình huống mới, hệ thống sẽ làm thích nghi hành vi đã lưu trữ với tình huống mới.
- **Học khám phá hay học không giám sát:** Thay vì có mục tiêu tường minh, hệ khám phá liên tục tìm kiếm các mẫu và quan hệ trong dữ liệu nhấp. Các ví dụ về học không giám sát bao gồm gom cụm dữ liệu, học để nhận dạng các đặc tính cơ bản từ các điểm ảnh.

Chương 7. MÁY HỌC

Cây định danh

- Xây dựng cây định danh dựa trên sự phân hoạch của các thuộc tính. Trong đó, phân hoạch là:
 - Nút cha: là thuộc tính được phân hoạch
 - Các nút con: Các giá trị phân biệt ứng với thuộc tính được phân hoạch.
- Là công cụ phổ biến trong một số ứng dụng

Chương 7. MÁY HỌC

Cây định danh

- Xây dựng các quy luật để có thể kết luận 1 người như thế nào thì khi tắm biển sẽ bị cháy nắng.

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Chương 7. MÁY HỌC

Cây định danh

- Ta gọi tính chất cháy nắng hay không cháy nắng là thuộc tính quan tâm (thuộc tính mục tiêu)
- Như vậy, trong trường hợp này tập R chỉ bao gồm 2 phần tử {nám, không}
- Ta gọi tập P là tất cả những người tham gia khảo sát ($n=8$)
- Chúng ta quan sát các hiện tượng cháy nắng dựa trên 4 thuộc tính: chiều cao (TB, cao, thấp), cân nặng (nhẹ, TB, nặng), Màu tóc (vàng, nâu, đỏ), dùng kem chống nắng (có, không). Ta gọi các thuộc tính này là dẫn xuất.

Tên	Màu tóc	C.caо	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Chương 7. MÁY HỌC

Cây định danh

- Bước 1: Phân hoạch tập P ban đầu thành các tập P_i sao cho tất cả các phần tử trong tất cả các tập P_i có cùng thuộc tính mục tiêu.

$$P = P_1 \cup P_2 \cup \dots \cup P_n \text{ và}$$

$$\forall (i, j), i \neq j \text{ thì } (P_i \cap P_j) = \emptyset \text{ và}$$

$$\forall (i, k, l): P_k \in P_i \text{ và } P_l \in P_i \text{ thì } f(P_k) = f(P_l)$$

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhé	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhé	Có	Không

Chương 7. MÁY HỌC

Cây định danh

- Bước 2: Sau khi đã phân hoạch xong tập P thành các tập P_i đặc trưng bởi thuộc tính đích r_i ($r_i \in R$), bước tiếp theo là ứng với mỗi phân hoạch P_i ta xây dựng các luật $L_i: GT_i \rightarrow r_i$

Trong đó, GT_i là mệnh đề được hình thành bằng cách kết hợp các thuộc tính dẫn xuất.

Cụ thể, có 2 cách phân hoạch dễ thấy nhất.

- Cách 1: Cho mỗi người vào một danh sách phân hoạch:
 $P_1=\{Sarah\}, P_2=\{Dana\}, \dots, P_8=\{Kartie\}$
- Cách 2: Tập 1={Nám}, Tập 2 = {Không}

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Chương 7. MÁY HỌC

Cây định danh

■ Thủ cách khác:

Đầu tiên quan sát màu tóc, ta có

$$P_{\text{vàng}} = \{\text{Sarah}, \text{Dana}, \text{Amie}, \text{Kartie}\}$$

$$P_{\text{đỏ}} = \{\text{Emilie}\}$$

$$P_{\text{nâu}} = \{\text{Alex}, \text{John}, \text{Peter}\}$$

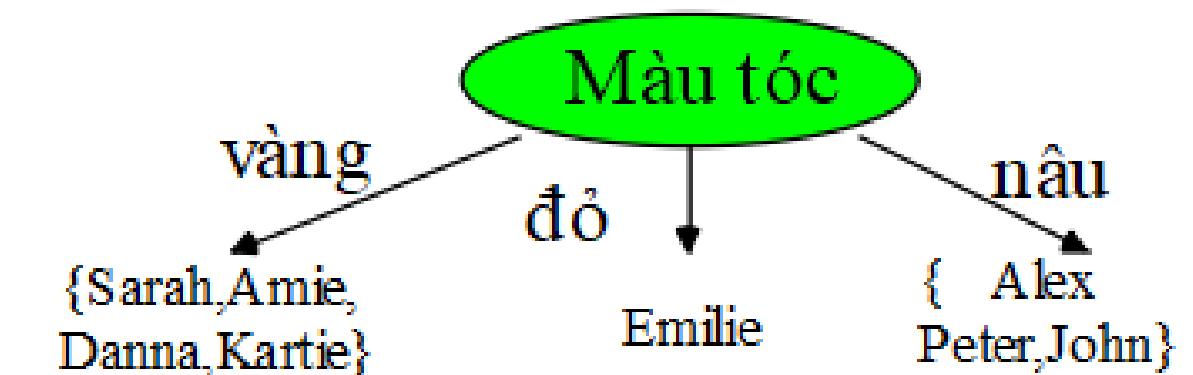
Nhìn vào bảng ta thấy:

$$P_{\text{nâu}} = \{\text{tùy người không bị cháy nắng}\}$$

$$P_{\text{đỏ}} = \{\text{tùy người bị cháy nắng}\}$$

tức là P_i có cùng chung thuộc tính mục tiêu. Còn lại $P_{\text{vàng}}$ có 2 trường hợp bị nám và không bị nám nên cần phải phân hoạch.

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

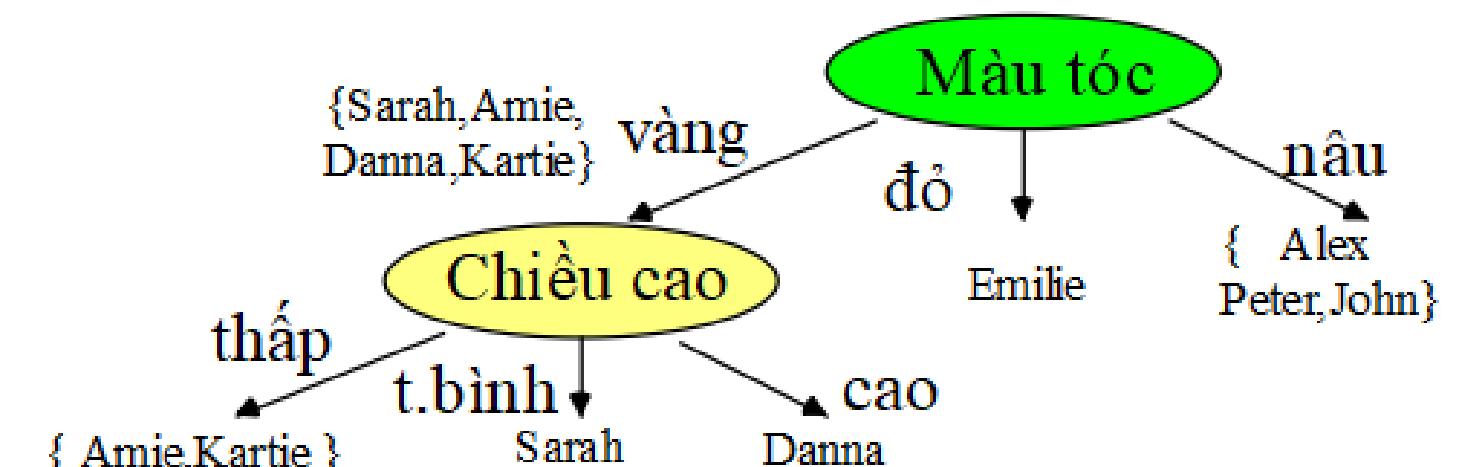


Chương 7. MÁY HỌC

Cây định danh

- Tiếp theo, từ tập $P_{nâu}$, ta quan sát thuộc tính chiều cao và phân hoạch dựa theo thuộc tính này:
 - $P_{vàng, thấp} = \{Amie, Kartie\}$
 - $P_{vàng, cao} = \{Dana\}$
 - $P_{vàng, TB} = \{Sarah\}$
 - Quá trình này cứ tiếp tục cho đến khi tất cả các nút lá của cây không còn lẫn lộn giữa nám và không bị nám.

Tên	Màu tóc	C.caو	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

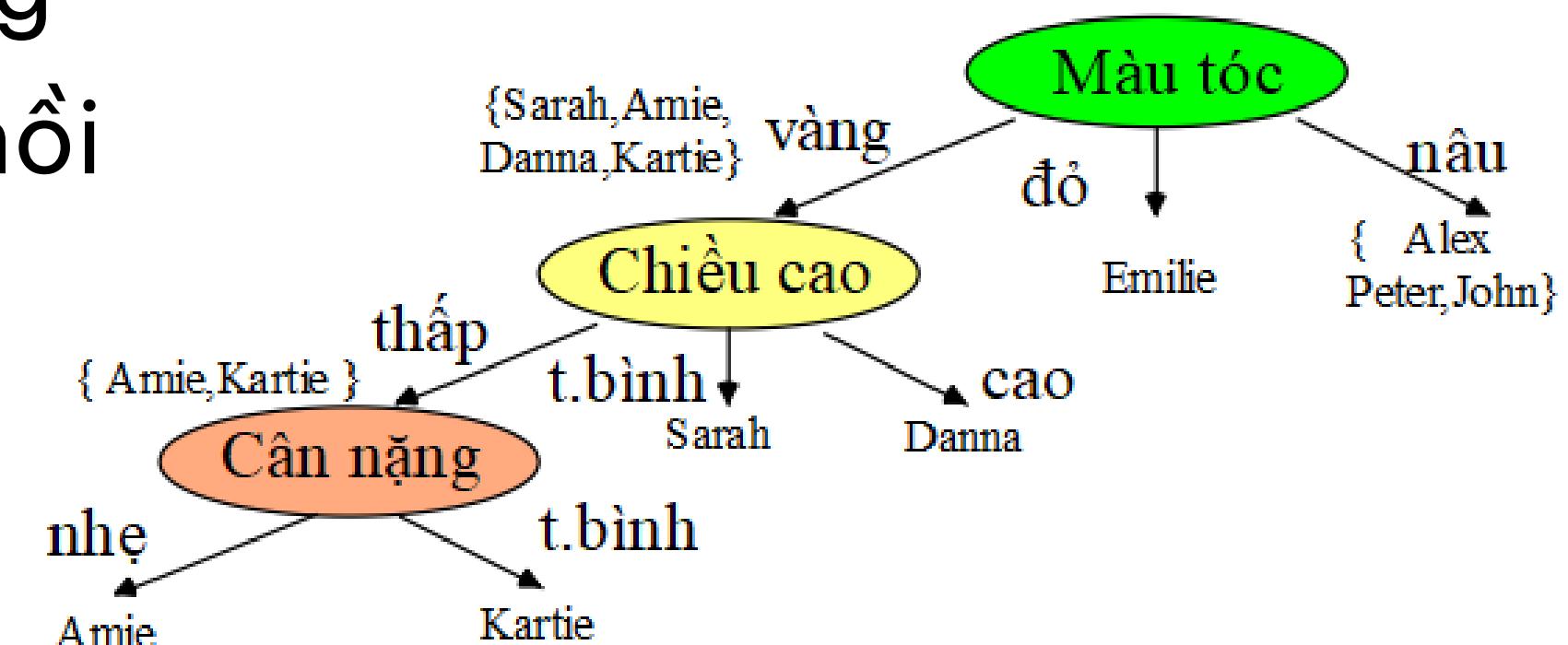


Chương 7. MÁY HỌC

Cây định danh

- Tiếp theo, từ tập $P_{nâu, thấp}$, ta quan sát thuộc tính **cân nặng** và **phân hoạch** dựa theo thuộc tính này:
- $P_{vàng, thấp, nhẹ} = \{Amie\}$
- $P_{vàng, thấp, trung bình} = \{Kartie\}$
- Có thể thấy rằng, qua mỗi bước phân hoạch thì cây phân hoạch ngày càng phình ra, quá trình này gọi là **đâm chồi** hay còn gọi là **cây định danh**.

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

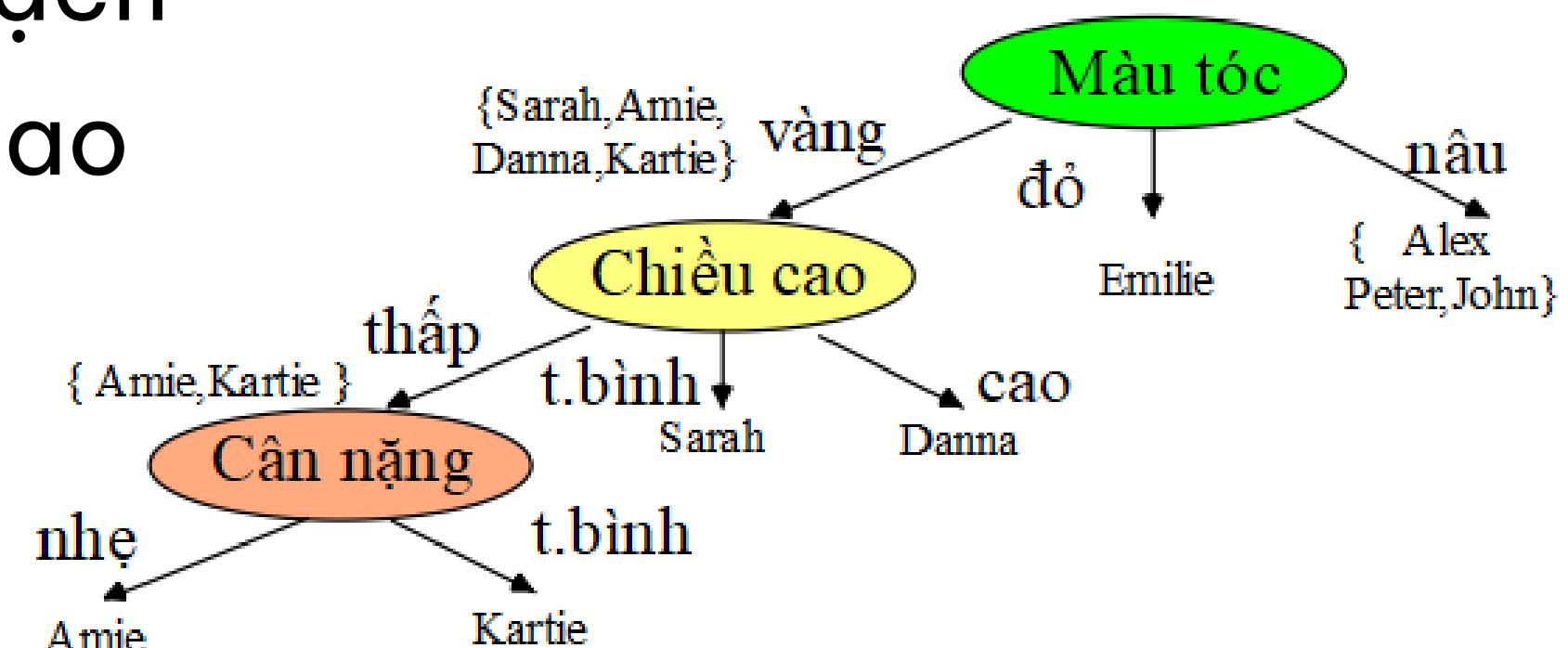


Chương 7. MÁY HỌC

Cây định danh

- Để chuyển từ tri thức học thành luật, đi đi từ nút gốc đến lá lấy nút gốc làm GT, nút lá làm KL
- $if(Màu\ tóc = vàng)and(chiều\ cao = trung\ bình)then "Nám"$
- Câu hỏi đặt ra: Nếu từ đầu ta không chọn thuộc tính màu tóc để phân hoạch mà chọn thuộc tính khác như chiều cao thì cách nào sẽ tốt hơn?

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không



Chương 7. MÁY HỌC

Cây định danh

- Ví dụ 2: Xây dựng cây định danh cho kết quả trời mưa với bảng dữ liệu sau

STT	Trời	Áp suất	Gió	Kết quả
1	Trong	Cao	Bắc	Không mưa
2	Mây	Cao	Nam	Mưa
3	Mây	Trung bình	Bắc	Mưa
4	Trong	Thấp	Bắc	Không mưa
5	Mây	Thấp	Bắc	Mưa
6	Mây	Cao	Bắc	Mưa
7	Mây	Thấp	Nam	Không mưa
8	Trong	Cao	Nam	Không mưa

Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Giải thuật Quinland quyết định thuộc tính phân hoạch bằng cách xây dựng các vector đặc trưng cho mỗi giá trị của từng thuộc tính dẫn xuất và mục tiêu.
- Với mỗi thuộc tính dẫn xuất A có thể sử dụng để phân hoạch, tính
- $V_{A(j)} = (Tj, r_1), T(j, r_2), \dots, T(j, rn))$, với
- A: thuộc tính dẫn xuất r_1, r_2, \dots, rn : thuộc tính mục tiêu
- $T(j, rj) = TAij / TAj$
 - $TAij$: tổng số phần tử trong phân hoạch có thuộc tính A là j và thuộc tính mục tiêu là r
 - TAj : Tổng số phần tử trong phân hoạch có thuộc tính A là j
- Lưu ý: $T(j, r_1) + T(j, r_2) + \dots + T(j, rn) = 1$

Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Như vậy nếu 1 thuộc tính A có thể nhận 1 trong 5 giá trị khác nhau thì nó sẽ có 5 vector đặc trưng.
- Một vector V_{Aj} được gọi là vector đơn vị nếu chỉ có duy nhất 1 thành phần có giá trị là 1, các thành phần khác có giá trị là 0.
- Thuộc tính được chọn để phân hoạch là thuộc tính có nhiều vector đơn vị nhất.

Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Xây dựng các quy luật để có thể kết luận 1 người như thế nào thì khi tắm biển sẽ bị cháy nắng theo giải thuật Quinland.

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Màu tóc có 3 giá trị (vàng, đỏ, nâu): có 3 vector đặc trưng
- $V_{tóc(vàng)} = T_{(vàng, nám)}, T_{(vàng, không)}$
 - Số người tóc vàng: 4
 - Số người tóc vàng, nám: 2
 - Số người tóc vàng, không nám: 2
 - Do đó, $V_{tóc(vàng)} = 0.5, 0.5$

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Màu tóc có 3 giá trị (vàng, đỏ, nâu): có 3 vector đặc trưng
- $V_{tóc(nâu)} = T_{(nâu, nám)}, T_{(nâu, không)}$
 - Số người tóc nâu: 3
 - Số người tóc nâu, nám: 0
 - Số người tóc vàng, không nám: 3
 - Do đó, $V_{tóc(vàng)} = 0, 1$ là một vector đơn vị

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Màu tóc có 3 giá trị (vàng, đỏ, nâu): có 3 vector đặc trưng
- $V_{tóc(\text{đỏ})} = T_{(\text{đỏ}, \text{nám})}, T_{(\text{đỏ}, \text{không})}$
 - Số người tóc đỏ: 1
 - Số người tóc đỏ, nám: 1
 - Số người tóc đỏ, không nám: 0
 - Do đó, $V_{tóc(\text{đỏ})} = 1, 0$ là một vector đơn vị
- Tổng số vector đơn vị của màu tóc: 2

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Chiều cao có 3 giá trị (cao, t.bình, thấp): có 3 vector đặc trưng
 - $V_{chiều\ cao(cao)} = 0, 1$
 - $V_{chiều\ cao(trung\ bình)} = 0.67, 0.33$
 - $V_{chiều\ cao(thấp)} = 0.33, 0.67$
- Tổng số vector đơn vị của chiều cao: 1

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Cân nặng có 3 giá trị (nhẹ, t.bình, nặng): có 3 vector đặc trưng
 - $V_{cân\ nặng(nhẹ)} = 0.5, 0.5$
 - $V_{cân\ nặng(trung\ bình)} = 0.33, 0.67$
 - $V_{cân\ nặng(nặng)} = 0.33, 0.67$
- Tổng số vector đơn vị của cân nặng: 0

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Dùng kem có 2 giá trị (có, không): có 2 vector đặc trưng
 - $V_{dùng\ kem(có)} = 0, 1$
 - $V_{dùng\ kem(không)} = 0.6, 0.4$
- Tổng số vector đơn vị của dùng kem: 1

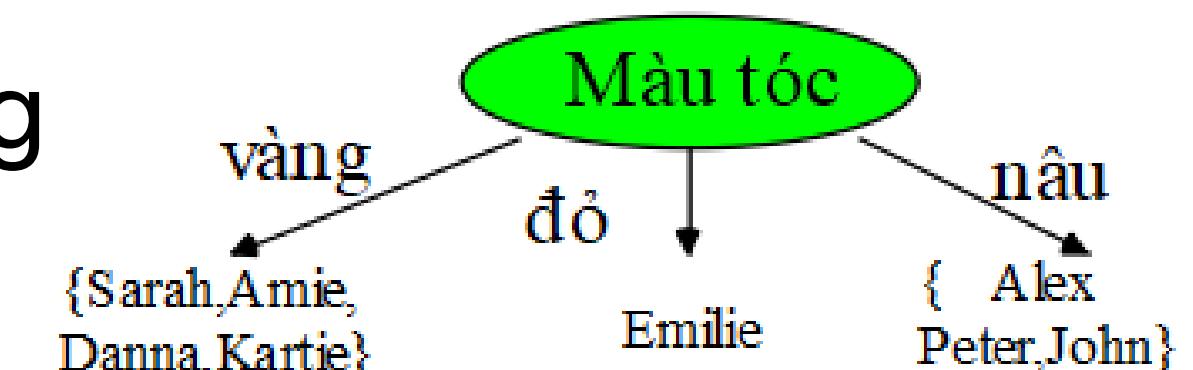
Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Ta thấy:
 - Tổng số vector đơn vị của màu tóc: 2
 - Tổng số vector đơn vị của chiều cao: 1
 - Tổng số vector đơn vị của cân nặng: 0
 - Tổng số vector đơn vị của dùng kem: 1
- Từ đó, ta chọn màu tóc để phân hoạch
- Sau khi phân hoạch ta thấy còn màu vàng chưa lỗn bị nám và không bị nám.
- Tiếp tục phân hoạch cho tập này.

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không



Chương 7. MÁY HỌC

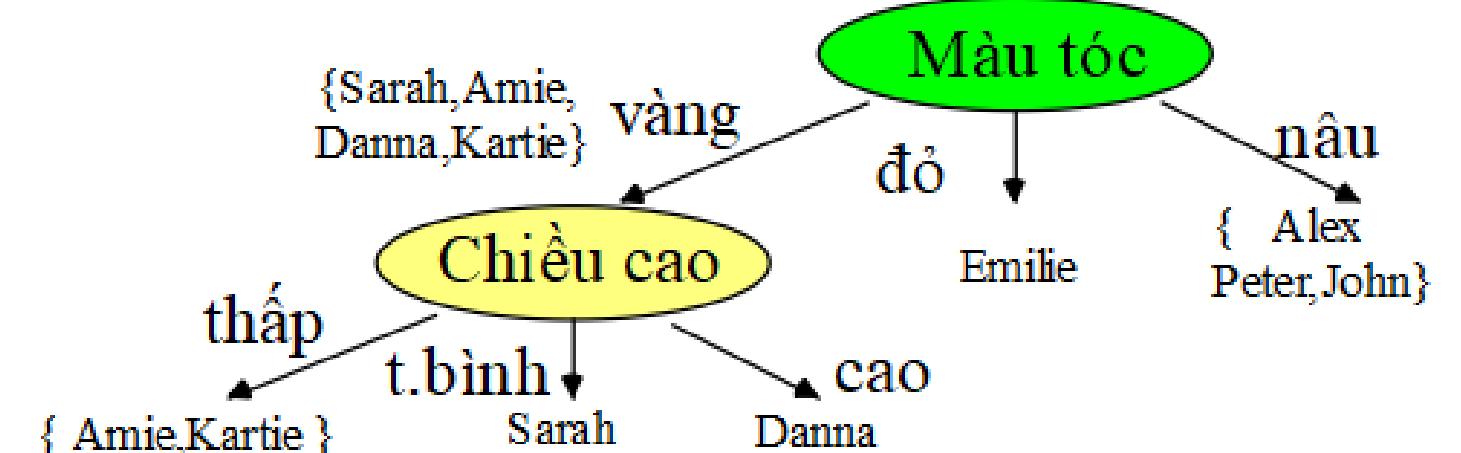
Cây định danh - Giải thuật Quinland

- Tiếp tục tính vector đặc trưng cho các thuộc tính: chiều cao, cân nặng, dùng kem cho màu tóc = vàng.

- $V_{chiều\ cao(cao)} = 0, 1$
- $V_{chiều\ cao(trung\ bình)} = 1, 0$
- $V_{chiều\ cao(thấp)} = 0.5, 0.5$
- $V_{cân\ nặng(nhẹ)} = 0.5, 0.5$
- $V_{cân\ nặng(trung\ bình)} = 0.5, 0.5$
- $V_{cân\ nặng(nặng)} = 0, 0$
- $V_{dùng\ kem(có)} = 0, 1$
- $V_{dùng\ kem(không)} = 1, 0$

Tên	Màu tóc	C.caо	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Kartie	Vàng	Thấp	Nhẹ	Có	Không

- Tổng số vector đơn vị của dùng kem = chiều cao
- Chiều cao có 3 phân hoạch > dùng kem (2) nên sử dụng chiều cao cho đợt phân hoạch tiếp theo.

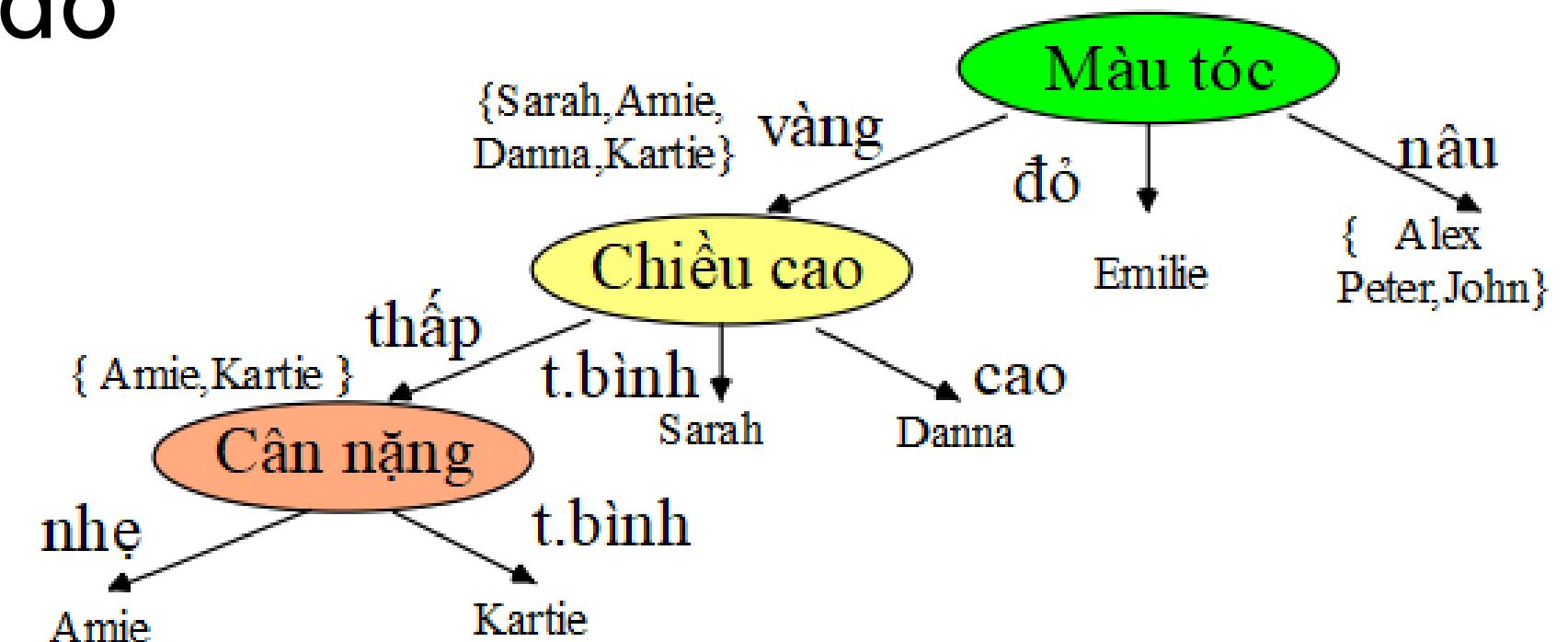


Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Sau khi phân hoạch ta thấy thuộc tính chiều cao = thấp còn lẫn lộn kết quả bị nám và không bị nám.
- Tiếp tục sử dụng thuộc tính chiều cao có giá trị là thấp để phân hoạch.
 - $V_{cân\ nặng(nhẹ)} = 0,1$
 - $V_{cân\ nặng(trung\ bình)} = 1,0$
 - $V_{cân\ nặng(nặng)} = 0,0$
 - $V_{dùng\ kem(có)} = 0,1$
 - $V_{dùng\ kem(không)} = 1,0$

Tên	Màu tóc	C.caо	C.nặng	Dùng kem	Kết quả
Amie	Vàng	Thấp	TB	Không	Nám
Kartie	Vàng	Thấp	Nhẹ	Có	Không



- Tổng số vector đơn vị của dùng kem = cân nặng
- Cân nặng có 3 phân hoạch > dùng kem (2) nên sử dụng cân nặng cho đợt phân hoạch tiếp theo.

Chương 7. MÁY HỌC

Cây định danh - Giải thuật Quinland

- Ví dụ 2: Xây dựng cây định danh bằng giải thuật Quinland cho kết quả trời mưa với bảng dữ liệu sau

STT	Trời	Áp suất	Gió	Kết quả
1	Trong	Cao	Bắc	Không mưa
2	Mây	Cao	Nam	Mưa
3	Mây	Trung bình	Bắc	Mưa
4	Trong	Thấp	Bắc	Không mưa
5	Mây	Thấp	Bắc	Mưa
6	Mây	Cao	Bắc	Mưa
7	Mây	Thấp	Nam	Không mưa
8	Trong	Cao	Nam	Không mưa

Chương 7. MÁY HỌC

Cây định danh - Entropy

- Thay vì phải xây dựng các vector đặc trưng như phương pháp của Quinland, ứng với mỗi thuộc tính dẫn xuất ta chỉ cần tính độ đo hỗn loạn và lựa chọn thuộc tính nào có độ đo hỗn loạn bé nhất để phân hoạch. Lặp lại cho đến khi hết các thuộc tính.
- Độ đo bất định cho thuộc tính X:

$$E(x) = \sum_b \left(\frac{n_b}{n_t} * \sum_c - \frac{n_{bc}}{n_b} * \log_a^{\frac{n_{bc}}{n_b}} \right)$$

- Trong đó:
 - n_b : số mẫu nhánh b
 - n_t : tổng số mẫu
- n_{bc} : tổng số mẫu trong nhánh b của lớp c
- a: số lượng giá trị của thuộc tính mục tiêu

Chương 7. MÁY HỌC

Cây định danh - Entropy

- Xây dựng các quy luật để có thể kết luận 1 người như thế nào thì khi tắm biển sẽ bị cháy nắng theo độ bất định Entropy.

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

- Số lượng giá trị thuộc tính mục tiêu: $a = 2$ (nám/không).

Chương 7. MÁY HỌC

Cây định danh - Entropy

- Xét thuộc tính màu tóc (vàng, nâu, đỏ)
 - Vàng: $\frac{2}{4}$ (nám) + $\frac{2}{4}$ (không)
 - Nâu: $\frac{3}{3}$ (nám) + $\frac{0}{3}$ (không)
 - Đỏ: $\frac{1}{1}$ (nám) + $\frac{0}{1}$ (không)
- Entropy của màu tóc:
 - $E(\text{màu tóc}) = \frac{4}{8} * \left(-\frac{2}{4} \log_2^{\frac{2}{4}} - \frac{2}{4} \log_2^{\frac{2}{4}} \right) : \text{vàng}$
 $+ \frac{3}{8} * \left(-\frac{0}{3} \log_2^{\frac{0}{3}} - \frac{3}{3} \log_2^{\frac{3}{3}} \right) : \text{nâu}$
 $+ \frac{1}{8} * \left(-\frac{1}{1} \log_2^{\frac{1}{1}} - \frac{0}{1} \log_2^{\frac{0}{1}} \right) : \text{đỏ}$

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

$$E(\text{màu tóc}) = 0.5$$

Chương 7. MÁY HỌC

Cây định danh - Entropy

- Xét thuộc tính chiều cao (cao, trung bình, thấp)

- Cao: 0/2 (nám) + 2/2 (không)
- TB: 2/3(nám) + 1/3 (không)
- Thấp: 1/3 (nám) + 2/3 (không)
- Entropy của chiều cao:

$$E(\text{chiều cao}) = \frac{2}{8} * \left(-\frac{0}{2} \log_2^{\frac{0}{2}} - \frac{2}{2} \log_2^{\frac{2}{2}} \right) : \text{cao}$$

$$+ \frac{3}{8} * \left(-\frac{2}{3} \log_2^{\frac{2}{3}} - \frac{1}{3} \log_2^{\frac{1}{3}} \right) : \text{trung bình}$$

$$+ \frac{3}{8} * \left(-\frac{1}{3} \log_2^{\frac{1}{3}} - \frac{2}{3} \log_2^{\frac{2}{3}} \right) : \text{thấp}$$

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

$$E(\text{chiều cao}) = 0.69$$

Chương 7. MÁY HỌC

Cây định danh - Entropy

▪ Xét thuộc tính cân nặng (nhẹ, trung bình, nặng)

- Nhẹ: $\frac{1}{2}$ (nám) + $\frac{1}{2}$ (không)
- TB: $\frac{1}{3}$ (nám) + $\frac{2}{3}$ (không)
- Nặng: $\frac{1}{3}$ (nám) + $\frac{2}{3}$ (không)

▪ Entropy của cân nặng:

$$E(\text{cân nặng}) = \frac{2}{8} * \left(-\frac{1}{2} \log_2^{\frac{1}{2}} - \frac{1}{2} \log_2^{\frac{1}{2}} \right) : \text{nhẹ}$$

$$+ \frac{3}{8} * \left(-\frac{1}{3} \log_2^{\frac{1}{3}} - \frac{2}{3} \log_2^{\frac{2}{3}} \right) : \text{trung bình}$$

$$+ \frac{3}{8} * \left(-\frac{1}{3} \log_2^{\frac{1}{3}} - \frac{2}{3} \log_2^{\frac{2}{3}} \right) : \text{nặng}$$

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

$$E(\text{cân nặng}) = 0.94$$

Chương 7. MÁY HỌC

Cây định danh - Entropy

- Xét thuộc tính dùng kem (có, không)
 - Có: $0/3$ (nám) + $3/3$ (không)
 - Không: $3/5$ (nám) + $2/5$ (không)
- Entropy của dùng kem:
$$E(\text{dùng kem}) = \frac{3}{8} * \left(-\frac{0}{3} \log_2^{\frac{0}{3}} - \frac{3}{3} \log_2^{\frac{3}{3}} \right) : \text{có}$$
$$+ \frac{5}{8} * \left(-\frac{3}{5} \log_2^{\frac{3}{5}} - \frac{2}{5} \log_2^{\frac{2}{5}} \right) : \text{không}$$

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

$$E(\text{dùng kem}) = 0.61$$

Chương 7. MÁY HỌC

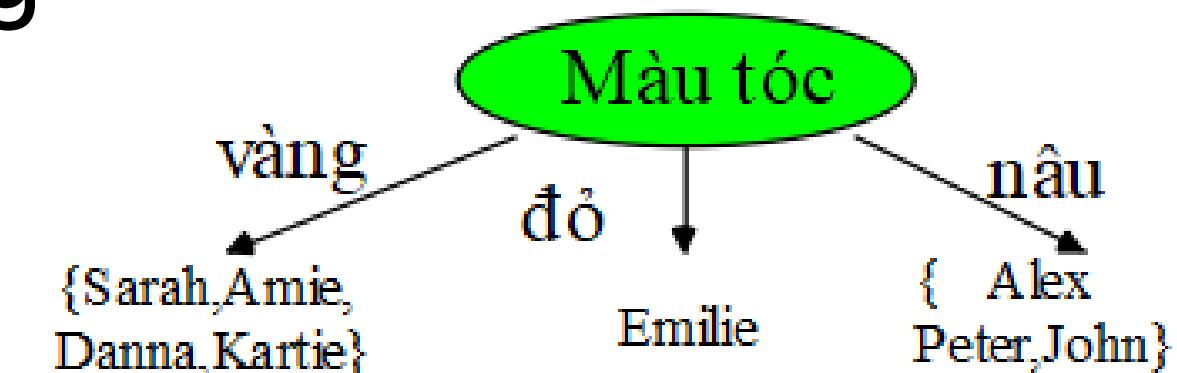
Cây định danh - Entropy

- Ta thấy:

- $E(\text{màu tóc}) = 0.5$
- $E(\text{chiều cao}) = 0.69$
- $E(\text{cân nặng}) = 0.94$
- $E(\text{dùng kem}) = 0.61$

- Từ đó, ta chọn màu tóc để phân hoạch
- Sau khi phân hoạch ta thấy còn màu vàng
chứa lẫn lộn bị nám và không bị nám.
- Tiếp tục phân hoạch cho tập này.

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không



Chương 7. MÁY HỌC

Cây định danh - Entropy

- Xét thuộc tính chiều cao (cao, trung bình, thấp)

- Cao: 0/1 (nám) + 1/1 (không)
- TB: 1/1 (nám) + 0/1 (không)
- Thấp: 1/2 (nám) + 1/2 (không)

- Entropy của chiều cao:

$$E(\text{chiều cao}) = \frac{1}{4} * \left(-\frac{0}{1} \log_2^{\frac{0}{1}} - \frac{1}{1} \log_2^{\frac{1}{1}} \right) : \text{cao}$$

$$+ \frac{1}{4} * \left(-\frac{1}{1} \log_2^{\frac{1}{1}} - \frac{0}{1} \log_2^{\frac{0}{1}} \right) : \text{trung bình}$$

$$+ \frac{2}{4} * \left(-\frac{1}{2} \log_2^{\frac{1}{2}} - \frac{1}{2} \log_2^{\frac{1}{2}} \right) : \text{thấp}$$

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Kartie	Vàng	Thấp	Nhẹ	Có	Không

$$E(\text{chiều cao}) = 0.5$$

Chương 7. MÁY HỌC

Cây định danh - Entropy

- Xét thuộc tính cân nặng (nhẹ, trung bình, nặng)

- Nhẹ: $\frac{1}{2}$ (nám) + $\frac{1}{2}$ (không)
- TB: $\frac{1}{2}$ (nám) + $\frac{1}{2}$ (không)
- Entropy của cân nặng:

$$E(\text{cân nặng}) = \frac{2}{4} * \left(-\frac{1}{2} \log_2^{\frac{1}{2}} - \frac{1}{2} \log_2^{\frac{1}{2}} \right) : \text{nhẹ}$$
$$+ \frac{2}{4} * \left(-\frac{1}{2} \log_2^{\frac{1}{2}} - \frac{1}{2} \log_2^{\frac{1}{2}} \right) : \text{trung bình}$$

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Kartie	Vàng	Thấp	Nhẹ	Có	Không

$$E(\text{cân nặng}) = 1.0$$

Chương 7. MÁY HỌC

Cây định danh - Entropy

- Xét thuộc tính dùng kem (có, không)
 - Có: $0/2$ (nám) + $2/2$ (không)
 - Không: $2/2$ (nám) + $0/2$ (không)
- Entropy của dùng kem:
 - $E(\text{dùng kem}) = \frac{2}{4} * \left(-\frac{0}{2} \log_2^{\frac{0}{2}} - \frac{2}{2} \log_2^{\frac{2}{2}} \right)$: có
 - $+ \frac{2}{4} * \left(-\frac{2}{2} \log_2^{\frac{2}{2}} - \frac{0}{2} \log_2^{\frac{0}{2}} \right)$: không

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Kartie	Vàng	Thấp	Nhẹ	Có	Không

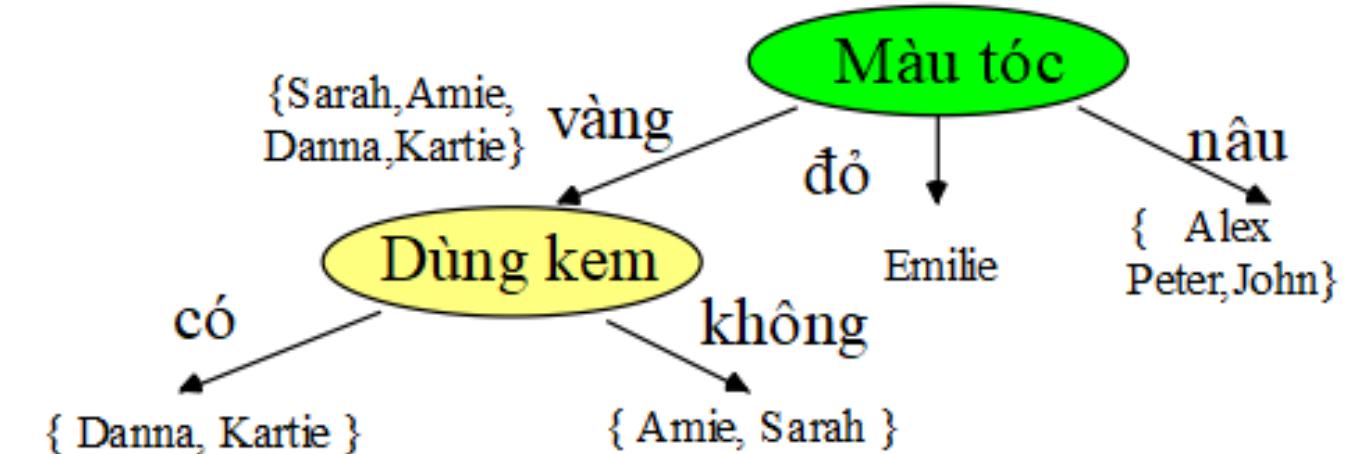
$$E(\text{dùng kem}) = 0$$

Chương 7. MÁY HỌC

Cây định danh - Entropy

- Ta thấy:
 - $E(\text{chiều cao}) = 0.5$
 - $E(\text{cân nặng}) = 1.0$
 - $E(\text{dùng kem}) = 0.0$
- Từ đó, ta chọn dùng kem để phân hoạch
- Sau khi phân hoạch ta không còn thuộc tính để phân hoạch nữa. KẾT THÚC

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Kartie	Vàng	Thấp	Nhẹ	Có	Không



Chương 7. MÁY HỌC

Cây định danh - Entropy

- Ví dụ 2: Xây dựng cây định danh bằng giải thuật độ đo hỗn loạn cho kết quả trời mưa với bảng dữ liệu sau

STT	Trời	Áp suất	Gió	Kết quả
1	Trong	Cao	Bắc	Không mưa
2	Mây	Cao	Nam	Mưa
3	Mây	Trung bình	Bắc	Mưa
4	Trong	Thấp	Bắc	Không mưa
5	Mây	Thấp	Bắc	Mưa
6	Mây	Cao	Bắc	Mưa
7	Mây	Thấp	Nam	Không mưa
8	Trong	Cao	Nam	Không mưa

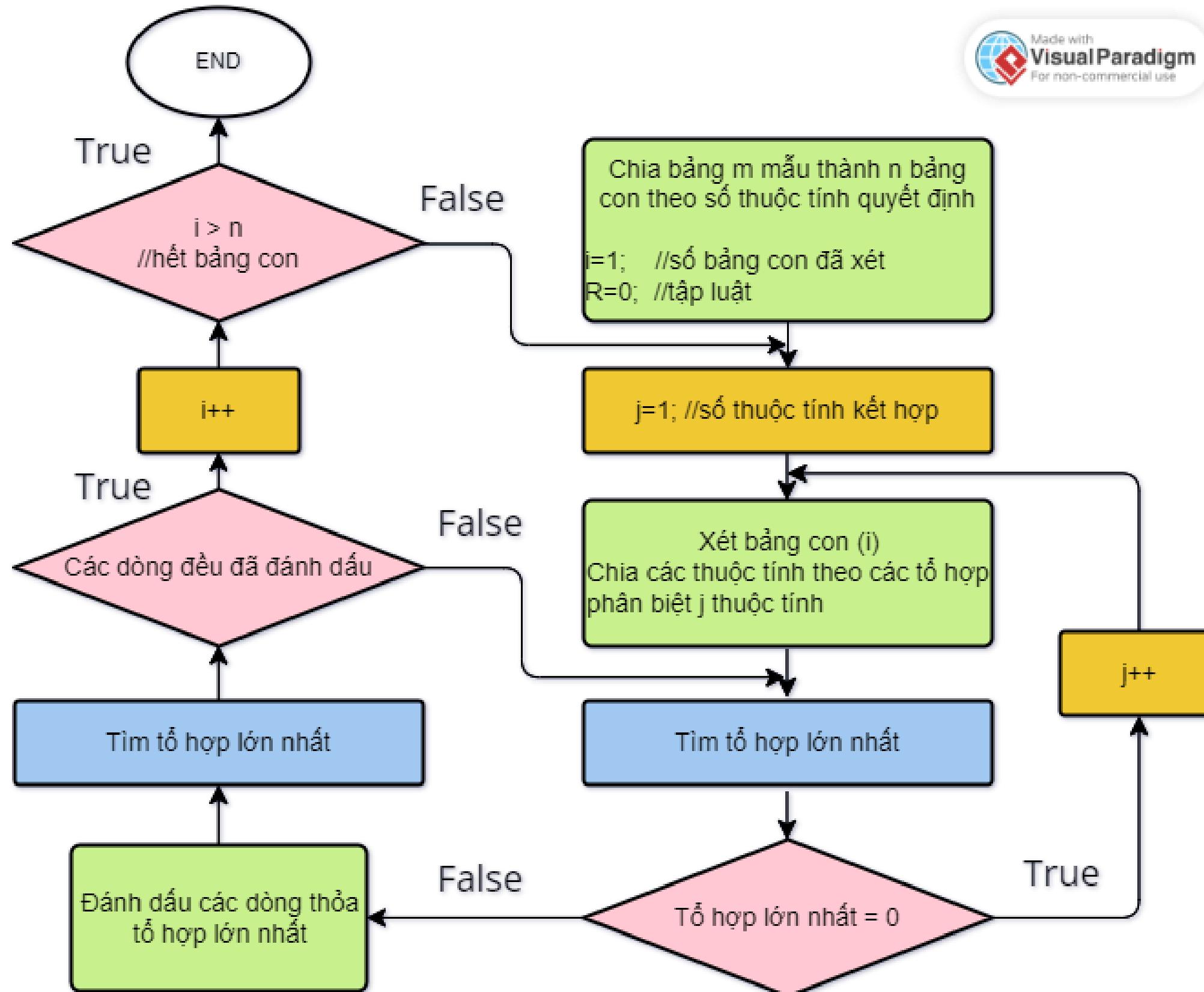
Chương 7. MÁY HỌC

Học quy nạp – Inductive learning Algorithm (ILA)

- Thuật giải ILA (Inductive Learning Algorithm) được dùng để xác định các luật phân loại cho tập hợp các mẫu học.
- Thuật giải này thực hiện theo cơ chế lặp, để tìm luật riêng đại diện cho tập mẫu của từng lớp.
- Sau khi xác định được luật, ILA loại bỏ các mẫu liên quan khỏi tập mẫu, đồng thời thêm luật mới này vào tập luật.

Chương 7. MÁY HỌC

Học quy nạp – Inductive learning Algorithm (ILA)



Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Xây dựng cây định danh bằng giải thuật ILA cho kết quả rám nắng với bảng dữ liệu sau

Chương 7. MÁY HỌC

Học quy nạp – Inductive learning Algorithm (ILA)

- Từ bảng dữ liệu có thể nhận ra số thuộc tính quyết định: n=2 nên tách thành 2 bảng con

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám

Bảng 1

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Bảng 2

- Thuộc tính kết quả (mục tiêu) có thể bỏ qua trong quá trình xét

Chương 7. MÁY HỌC

Học quy nạp – Inductive learning Algorithm (ILA)

- **Xét bảng 1**

- **j=1 //xét TH có 1 thuộc tính**

có 4 tổ hợp (màu tóc, c.cao, c.nặng, dùng kem)

- Xét thuộc tính: màu tóc

- vàng (bảng 1 và 2 đều có): 0
- đỏ (bảng 1 có, bảng 2 không): 1

- Xét thuộc tính: chiều cao

- TB (bảng 1 và 2 đều có): 0
- thấp (bảng 1 và 2 đều có): 0

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

{màu tóc}	{c.cao}	{c. nặng}	{dùng kem}
Vàng: 0 Đỏ: 1	TB: 0 thấp: 0		

Chương 7. MÁY HỌC

Học quy nạp – Inductive learning Algorithm (ILA)

- **Xét bảng 1**
- **j=1 //xét TH có 1 thuộc tính**
có 4 tổ hợp (màu tóc, c.cao, c.nặng, dùng kem)
- Xét thuộc tính: cân nặng
 - nhẹ (bảng 1 và 2 đều có): 0
 - TB (bảng 1 và 2 đều có): 0
 - nặng (bảng 1 và 2 đều có): 0
- Xét thuộc tính: dùng kem
 - không (bảng 1 và 2 đều có): 0

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

{màu tóc}	{c.cao}	{c. nặng}	{dùng kem}
Vàng: 0 Đỏ: 1	TB: 0 thấp: 0	TB: 0 nhẹ: 0 nặng: 0	không: 0

Tổ hợp max=1: Đỏ {màu tóc}

IF màu tóc = đỏ THEN kết quả = bị nám

Đánh dấu dòng màu tóc = đỏ

Chương 7. MÁY HỌC

Học quy nạp – Inductive learning Algorithm (ILA)

- **Xét bảng 1**
- **j=1 //xét TH có 1 thuộc tính**

có 4 tổ hợp (màu tóc, c.cao, c.nặng, dùng kem)

- Sau khi đánh dấu dòng đã xét, tính lại bảng tổ hợp cho các dòng chưa xét với 1 thuộc tính phân biệt

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

{màu tóc}	{c.cao}	{c. nặng}	{dùng kem}
Vàng: 0 nâu: 0	TB: 0 thấp: 0	TB: 0 nhẹ: 0 nặng: 0	không: 0

Tổ hợp max=0: không có luật mới

Chương 7. MÁY HỌC

Học quy nạp – Inductive learning Algorithm (ILA)

- **Xét bảng 1**
- **j=2 //xét TH có 2 thuộc tính**

có 6 tổ hợp: (m. tóc, c.cao), (m.tóc, c.nặng), (m.tóc, dùng kem),
 (c.cao, c.nặng), (c.cao, dùng kem)
 (c.nặng, dùng kem)

{màu tóc, c.cao}	{m.tóc, d.kem}
{vàng, TB}	{vàng, thấp}
1	0
2 (2 bản ghi # vs bảng 2)	

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	↑TB	Nặng	Không	Nám

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

{màu tóc, c.nặng}	{chiều cao, c.nặng}	{chiều cao, d.kem}	{c.nặng, d.kem}
{vàng, TB}	{vàng, nhẹ}	{TB, không}	{TB, không}
0	0	1	0
1	1	1	1
1	1	1	1

Chương 7. MÁY HỌC

Học quy nạp – Inductive learning Algorithm (ILA)

- **Xét bảng 1**
- **j=2 //xét TH có 2 thuộc tính**

có 6 tổ hợp: (m. tóc, c.cao), (m.tóc, c.nặng), (m.tóc, dùng kem),
(c.cao, c.nặng), (c.cao, dùng kem)
(c.nặng, dùng kem)

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Thêm luật vào R:
IF màu tóc = vàng AND dùng kem = không
THEN kết quả = bị nám

Tổ hợp max=2: (màu tóc, dùng kem)
= {vàng, không}

Đánh dấu dòng màu tóc = vàng và dùng kem = không

Chương 7. MÁY HỌC

Học quy nạp – Inductive learning Algorithm (ILA)

- **Xét bảng 2 (sau khi hết dòng B1)**
- **j=1 //xét TH có 1 thuộc tính**

có 4 tổ hợp: {m. tóc}, {c.cao}, {c.nặng}, {dùng kem}

{màu tóc}	{c.caو}	{c. nặng}	{dùng kem}
Vàng: 0 nâu: 3 (số dòng \notin B1)	TB: 0 thấp: 0 cao: 2	TB: 0 nhẹ: 0 nặng: 0	không: 0 có: 3

Thêm luật vào R:

IF màu tóc = nâu
THEN kết quả = không bị nám

Tên	Màu tóc	C.caو	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám

Tên	Màu tóc	C.caو	C.nặng	Dùng kem	Kết quả
Dana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

Tổ hợp max=3: (màu tóc)= {nâu} và (dùng kem) = {có}. Chọn màu tóc

Đánh dấu dòng màu tóc = nâu

Chương 7. MÁY HỌC

Học quy nạp – Inductive learning Algorithm (ILA)

- **Xét bảng 2**
- **j=1 //xét TH có 1 thuộc tính**

có 4 tổ hợp (màu tóc, c.cao, c.nặng, dùng kem)

- Sau khi đánh dấu dòng đã xét, tính lại bảng tổ hợp cho các dòng chưa xét với 1 thuộc tính phân biệt

Thêm luật vào R:

IF dùng kem = có
THEN kết quả = không bị nám

Số dòng chưa đánh dấu bảng 2 = \emptyset : END

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Sarah	Vàng	TB	Nhẹ	Không	Nám
Amie	Vàng	Thấp	TB	Không	Nám
Emilie	Đỏ	TB	Nặng	Không	Nám

Tên	Màu tóc	C.cao	C.nặng	Dùng kem	Kết quả
Đana	Vàng	Cao	TB	Có	Không
Alex	Nâu	Thấp	TB	Có	Không
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	TB	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

{màu tóc}	{c.cao}	{c. nặng}	{dùng kem}
Vàng: 0	TB: 0 thấp: 0	TB: 0 nhẹ: 0	có: 2

Tổ hợp max=2: (dùng kem) = {có}

Đánh dấu dòng dùng kem= có

Chương 7. MÁY HỌC

Xây dựng cây quyết định để đưa ra luật phù hợp

Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

TT	Ngày	Trời	Độ ẩm	Gió	Kết quả
01	D1	Nắng	Cao	Yếu	Không
02	D2	Nắng	Cao	Mạnh	Không
03	D3	U ám	Cao	Yếu	Có
04	D4	Mưa	Cao	Yếu	Có
05	D5	Mưa	Trung bình	Mạnh	Có
06	D6	Mưa	Trung bình	Mạnh	Không
07	D7	U ám	Trung bình	Yếu	Có
08	D8	Nắng	Cao	Yếu	Không
09	D9	Nắng	Trung bình	Yếu	Có
10	D10	Mưa	Trung bình	Yếu	Có

Chương 7. MÁY HỌC

Xây dựng cây quyết định để đưa ra luật phù hợp

Tên	Vị	Màu	Vỏ	Độc
A	Ngọt	Đỏ	Nhẵn	Không
B	Cay	Đỏ	Nhẵn	Không
C	Chua	Vàng	Có gai	Không
D	Cay	Vàng	Có gai	Độc
E	Ngọt	Tím	Có gai	Không
F	Chua	Vàng	Nhẵn	Không
G	Ngọt	Tím	Nhẵn	Không
H	Cay	Tím	Có gai	Độc
I	Cay	Đỏ	Có gai	Độc

Người	Dáng	Chiều cao	Giới tính	Thuộc Châu
1	To	Trung bình	Nam	Á
2	Nhỏ	Thấp	Nam	Á
3	Nhỏ	Trung bình	Nam	Á
4	To	Cao	Nam	Âu
5	Nhỏ	Trung bình	Nữ	Âu
6	Nhỏ	Cao	Nam	Âu
7	Nhỏ	Cao	Nữ	Âu
8	To	Trung bình	Nữ	Âu

Chương 7. MÁY HỌC

Xây dựng cây quyết định để đưa ra luật phù hợp

Age	Car type	Class
23	Family	High
17	Sports	High
43	Sports	High
68	Family	Low
32	Truck	Low
20	Family	High

AGE có thể phân thành 3 nhóm:

<30	≥ 30	≥ 40
-----	-----------	-----------

Chương 7. MÁY HỌC

Phân loại Naive Bayes (Naive Bayes classification)

- Thuật toán Naïve Bayes là một thuật toán học có giám sát, dựa trên định lý Bayes và được sử dụng để giải các bài toán phân loại.
- Được sử dụng chủ yếu trong phân loại văn bản.
- Naïve Bayes Classifier là một trong những thuật toán Phân loại đơn giản và hiệu quả nhất giúp xây dựng các mô hình học máy nhanh có thể đưa ra dự đoán nhanh.
 - Naïve: Giả định sự xuất hiện của một đặc điểm nào đó là độc lập với sự xuất hiện của các đặc điểm khác (độc lập ngẫu nhiên).
 - Bayes: Định lý Bayes

Chương 7. MÁY HỌC

Phân loại Naive Bayes (Naive Bayes classification)

- Công thức:

$$P(C_i/X) = \frac{P(X/C_i) * P(C_i)}{P(X)}$$

- $P(C_i/X)$: xác suất xảy ra nhãn i (mưa, không mưa) khi biết X (ít mây, áp suất thấp, gió nam...)
- $P(X/C_i)$: xác suất X khi biết nhãn i
- $P(C_i)$: xác suất xảy ra nhãn i
- $P(X)$: xác suất xảy ra nhãn X

Chương 7. MÁY HỌC

Phân loại Naive Bayes (Naive Bayes classification)

- Khả năng nào xảy ra khi trời ít mây, áp suất thấp và gió thổi từ hướng Nam?
- Bước 1: tính $P(x/c_i)$
- Bước 2: tính $P(x)$
- Bước 3: tính $P(c_i/x)$: xác suất xảy C khi biết X

Đối tượng	Mây	Áp suất	Gió	Kết quả
1	ít	cao	Bắc	Không mưa
2	nhiều	cao	Bắc	Mưa
3	ít	thấp	Bắc	Không mưa
4	nhiều	thấp	Bắc	Mưa
5	nhiều	trung bình	Bắc	Mưa
6	ít	cao	Nam	Không mưa
7	nhiều	cao	Nam	Mưa
8	nhiều	thấp	Nam	Không mưa

Chương 7. MÁY HỌC

Phân loại Naive Bayes (Naive Bayes classification)

- Khả năng nào xảy ra khi trời ít mây, áp suất thấp và gió thổi từ hướng Nam?
- Bước 1:
 - $P(X/c_i) = P(X/mưa)$, với $X = \{\text{ít, thấp, nam}\}$
 - $P(X/mưa) = P\left(\frac{\text{ít}}{\text{mưa}}\right) \cdot P\left(\frac{\text{thấp}}{\text{mưa}}\right) \cdot P\left(\frac{\text{nam}}{\text{mưa}}\right) = 0$
 - Để tránh trường hợp cho xác suất = 0 ta sử dụng biến đổi Laplace
 - $P(\text{ít}/\text{mưa}) = \frac{0+1}{4+2}$, trong đó: 1-hàng số; 4-số kết quả là mưa; 2-số thuộc tính của mây
 - $P(\text{thấp}/\text{mưa}) = \frac{1+1}{4+3} \quad P(\text{nam}/\text{mưa}) = \frac{1+1}{4+2}$

Đối tượng	Mây	Áp suất	Gió	Kết quả
1	ít	cao	Bắc	Không mưa
2	nhiều	cao	Bắc	Mưa
3	ít	thấp	Bắc	Không mưa
4	nhiều	thấp	Bắc	Mưa
5	nhiều	trung bình	Bắc	Mưa
6	ít	cao	Nam	Không mưa
7	nhiều	cao	Nam	Mưa
8	nhiều	thấp	Nam	Không mưa

Chương 7. MÁY HỌC

Phân loại Naive Bayes (Naive Bayes classification)

- Khả năng nào xảy ra khi trời ít mây, áp suất thấp và gió thổi từ hướng Nam?

- Bước 1:

- $P(X/C_i) = P(X/mưa)$, với $X = \{\text{ít, thấp, nam}\}$

- $P(X/mưa) = P\left(\frac{\text{ít}}{\text{mưa}}\right) \cdot P\left(\frac{\text{thấp}}{\text{mưa}}\right) \cdot P\left(\frac{\text{nam}}{\text{mưa}}\right) =$

$$= \frac{0+1}{4+2} * \frac{1+1}{4+3} * \frac{1+1}{4+2} = \frac{1}{63}$$

Đối tượng	Mây	Áp suất	Gió	Kết quả
1	ít	cao	Bắc	Không mưa
2	nhiều	cao	Bắc	Mưa
3	ít	thấp	Bắc	Không mưa
4	nhiều	thấp	Bắc	Mưa
5	nhiều	trung bình	Bắc	Mưa
6	ít	cao	Nam	Không mưa
7	nhiều	cao	Nam	Mưa
8	nhiều	thấp	Nam	Không mưa

Chương 7. MÁY HỌC

Phân loại Naive Bayes (Naive Bayes classification)

- Khả năng nào xảy ra khi trời ít mây, áp suất thấp và gió thổi từ hướng Nam?
- Bước 1:
 - $P(X/c_i) = P(X/không\ mưa)$, với $X = \{\text{ít, thấp, nam}\}$
 - $P(X/k.mưa) = P\left(\frac{\text{ít}}{k.mưa}\right) \cdot P\left(\frac{\text{thấp}}{k.mưa}\right) \cdot P\left(\frac{\text{nam}}{k.mưa}\right) =$
 - $P(X/k.mưa) = \frac{3+1}{4+2} * \frac{2+1}{4+3} * \frac{2+1}{4+2} = \frac{9}{63}$

Đối tượng	Mây	Áp suất	Gió	Kết quả
1	ít	cao	Bắc	Không mưa
2	nhiều	cao	Bắc	Mưa
3	ít	thấp	Bắc	Không mưa
4	nhiều	thấp	Bắc	Mưa
5	nhiều	trung bình	Bắc	Mưa
6	ít	cao	Nam	Không mưa
7	nhiều	cao	Nam	Mưa
8	nhiều	thấp	Nam	Không mưa

Chương 7. MÁY HỌC

Phân loại Naive Bayes (Naive Bayes classification)

- Khả năng nào xảy ra khi trời ít mây, áp suất thấp và gió thổi từ hướng Nam?

- Bước 2:

- $P(X) = \sum P(X/c_i)$, với $X = \{\text{ít, thấp, nam}\}$
- $P(X) = P(X/\text{mưa}) * P(\text{mưa})$
+ $P(X/k.\text{mưa}) * P(k.\text{mưa}) =$

$$= \frac{1}{63} * \frac{4+1}{8+2} + \frac{9}{63} * \frac{4+1}{8+2} = \frac{5}{63}$$

Đối tượng	Mây	Áp suất	Gió	Kết quả
1	ít	cao	Bắc	Không mưa
2	nhiều	cao	Bắc	Mưa
3	ít	thấp	Bắc	Không mưa
4	nhiều	thấp	Bắc	Mưa
5	nhiều	trung bình	Bắc	Mưa
6	ít	cao	Nam	Không mưa
7	nhiều	cao	Nam	Mưa
8	nhiều	thấp	Nam	Không mưa

Chương 7. MÁY HỌC

Phân loại Naive Bayes (Naive Bayes classification)

- Khả năng nào xảy ra khi trời ít mây, áp suất thấp và gió thổi từ hướng Nam?
- Bước 3: $P(C_i/X)$, với $X = \{\text{ít, thấp, nam}\}$
 - $P(\text{mưa}/X) = \frac{P(X/\text{mưa}) * P(\text{mưa})}{P(X)} = \frac{\frac{1}{63} * \frac{4}{8}}{\frac{5}{63}} = \frac{1}{10}$
 - $P(k.\text{mưa}/X) = \frac{P(X/k.\text{mưa}) * P(k.\text{mưa})}{P(X)} = \frac{\frac{9}{63} * \frac{4}{8}}{\frac{5}{63}} = \frac{9}{10}$
 - Hoặc $P(k.\text{mưa}/X) = 1 - P(\text{mưa}/X) = 90\%$
- Vì $P(k.\text{mưa}/X) > P(\text{mưa}/X)$ nên khả năng cao là trời sẽ không mưa.

Đối tượng	Mây	Áp suất	Gió	Kết quả
1	ít	cao	Bắc	Không mưa
2	nhiều	cao	Bắc	Mưa
3	ít	thấp	Bắc	Không mưa
4	nhiều	thấp	Bắc	Mưa
5	nhiều	trung bình	Bắc	Mưa
6	ít	cao	Nam	Không mưa
7	nhiều	cao	Nam	Mưa
8	nhiều	thấp	Nam	Không mưa

Chương 7. MÁY HỌC

Phân loại Naive Bayes (Naive Bayes classification)

- today = (Sunny, Hot, Normal, False), khả năng nào của việc chơi golf sẽ xảy ra?
- Bước 1: tính $P(X/c_i)$
- Bước 2: tính $P(X)$
- Bước 3: tính $P(c_i/X)$: xác suất C khi biết X

	Outlook	Temperature	Humidity	Windy	Play Golf
0	Rainy	Hot	High	FALSE	No
1	Rainy	Hot	High	TRUE	No
2	Overcast	Hot	High	FALSE	Yes
3	Sunny	Mild	High	FALSE	Yes
4	Sunny	Cool	Normal	FALSE	Yes
5	Sunny	Cool	Normal	TRUE	No
6	Overcast	Cool	Normal	TRUE	Yes
7	Rainy	Mild	High	FALSE	No
8	Rainy	Cool	Normal	FALSE	Yes
9	Sunny	Mild	Normal	FALSE	Yes
10	Rainy	Mild	Normal	TRUE	Yes
11	Overcast	Mild	High	TRUE	Yes
12	Overcast	Hot	Normal	FALSE	Yes
13	Sunny	Mild	High	TRUE	No

Chương 7. MÁY HỌC

Phân cụm dữ liệu K trung tâm (K-means clustering)

- Thuật toán k-means là thuật toán phân cụm từ n đối tượng ban đầu vào k cụm phân biệt, với $k < n$ được giới thiệu năm 1957 bởi Lloyd K-means.
- Thuật toán sử dụng độ đo tương tự giữa quan sát.
- Khoảng cách Euclidean là phương pháp phổ biến nhất được dùng để đánh giá khoảng cách của các quan sát.
- Cho 2 quan sát $u = \{u_1, u_2, \dots, u_q\}$ và $v = \{v_1, v_2, \dots, v_q\}$, mỗi quan sát bao gồm q biến.
 - $d_{u,v} = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \dots + (u_q - v_q)^2}$: khoảng cách giữa vectors

Chương 7. MÁY HỌC

Phân cụm dữ liệu K trung tâm (K-means clustering)

- Nếu 2 vector thẳng hàng thì khoảng cách u,v là:
$$d_{u,v} = |u_1 - v_1| + |u_2 - v_2| + \dots + |u_q - v_q|$$
- Phân cụm cho 8 điểm có tọa độ như sau thành 3 cụm:
$$A_1(2,10), A_2(2,5), A_3(8,4), A_4(5,8), A_5(7,5), A_6(6,4), A_7(1,2), A_8(4,9)$$
- Có thể nhận thấy số cụm là 3: $k=3 < n=8$
- Bước 1: chọn ngẫu nhiên 3 điểm làm giá trị trung tâm:
$$A_1(2,10), A_4(5,8), A_7(1,2)$$
- Áp dụng công thức tính khoảng cách giữa 2 điểm a(x_1, y_1) và b(x_2, y_2):
$$d(a,b) = |x_2 - x_1| + |y_2 - y_1|$$

Chương 7. MÁY HỌC

Phân cụm dữ liệu K trung tâm (K-means clustering)

- Khoảng cách của các điểm tới k-means
- Cluster: tìm MIN của mỗi Point trong mỗi dist. mean_i và gán vào cluster.
- Ví dụ: A1(2,10): min = 0 ∈ dist.mean 1
→ cluster = 1
 - cluster1: (2,10)
 - cluster2: (8,4), (5,8), (7,5),
(6,4), (4,9)
 - cluster3: (2,5), (1,2)

		(2,10)	(5,8)	(1,2)	
	Point	Dist. mean1	Dist. mean2	Dist. mean3	Cluster
A1	(2,10)	0	5	9	1
A2	(2,5)	5	6	4	3
A3	(8,4)	12	7	9	2
A4	(5,8)	5	0	10	2
A5	(7,5)	10	5	9	2
A6	(6,4)	10	5	7	2
A7	(1,2)	9	10	0	3
A8	(4,9)	3	2	10	2

Chương 7. MÁY HỌC

Phân cụm dữ liệu K trung tâm (K-means clustering)

- Cập nhật lại giá trị của các giá trị trung tâm của cluster.
- cluster1: có 1 điểm A1(2,10) nên dist.means1 = (2,10)
- cluster2: dist. means2 = $[(8+5+7+6+4)/5, (4+8+5+4+9)/5] = (6,6)$
- cluster3: dist.mean3 = $[(2+1)/2, (5+2)/2] = (1.5,3.5)$

Chương 7. MÁY HỌC

Phân cụm dữ liệu K trung tâm (K-means clustering)

- Tính lại giá trị khoảng cách các means:
 - cluster1: (2,10), (4,9)
 - cluster2: (8,4), (5,8), (7,5), (6,4)
 - cluster3: (2,5), (1,2)

		(2,10)	(5,5)	(1.5,3.5)	
	Point	Dist. mean1	Dist. mean2	Dist. mean3	Cluster
A1	(2,10)	0	8	7	1
A2	(2,5)	5	3	2	3
A3	(8,4)	12	4	7	2
A4	(5,8)	5	3	8	2
A5	(7,5)	10	2	7	2
A6	(6,4)	10	2	5	2
A7	(1,2)	9	7	2	3
A8	(4,9)	3	5	8	1

Chương 7. MÁY HỌC

Phân cụm dữ liệu K trung tâm (K-means clustering)

- Cập nhật lại giá trị của các giá trị trung tâm của cluster.
- cluster1: $\text{dist.mean1} = [(2+4)/2, (10+9)/2] = (3, 9.5)$
- cluster2: $\text{dist.means2} = [(8+5+7+6)/4, (4+8+5+4)/4] = (6.5, 5.25)$
- cluster3: $\text{dist.mean3} = [(2+1)/2, (5+2)/2] = (1.5, 3.5)$

Chương 7. MÁY HỌC

Phân cụm dữ liệu K trung tâm (K-means clustering)

- Tính lại giá trị khoảng cách các means:
 - cluster1: (2,10), (4,9), (5,8)
 - cluster2: (8,4), (7,5), (6,4)
 - cluster3: (2,5), (1,2)

		(3,9.5)	(6.5,5.25)	(1.5,3.5)	
	Point	Dist. mean1	Dist. mean2	Dist. mean3	Cluster
A1	(2,10)	1,5	9,25	7	1
A2	(2,5)	5,5	4,75	2	3
A3	(8,4)	10,5	2,75	7	2
A4	(5,8)	3,5	4,25	8	1
A5	(7,5)	8,5	0,75	7	2
A6	(6,4)	8,5	1,75	5	2
A7	(1,2)	9,5	8,75	2	3
A8	(4,9)	1,5	6,25	8	1

Chương 7. MÁY HỌC

Phân cụm dữ liệu K trung tâm (K-means clustering)

- Cập nhật lại giá trị của các giá trị trung tâm của cluster.
- cluster1: dist.mean1 = $[(2+4+5)/3, (10+9+8)/3] = (3.67, 9)$
- cluster2: dist. means2 = $[(8+7+6)/4, (4+5+4)/4] = (7, 4.3)$
- cluster3: dist.mean3 = $[(2+1)/2, (5+2)/2] = (1.5, 3.5)$

Chương 7. MÁY HỌC

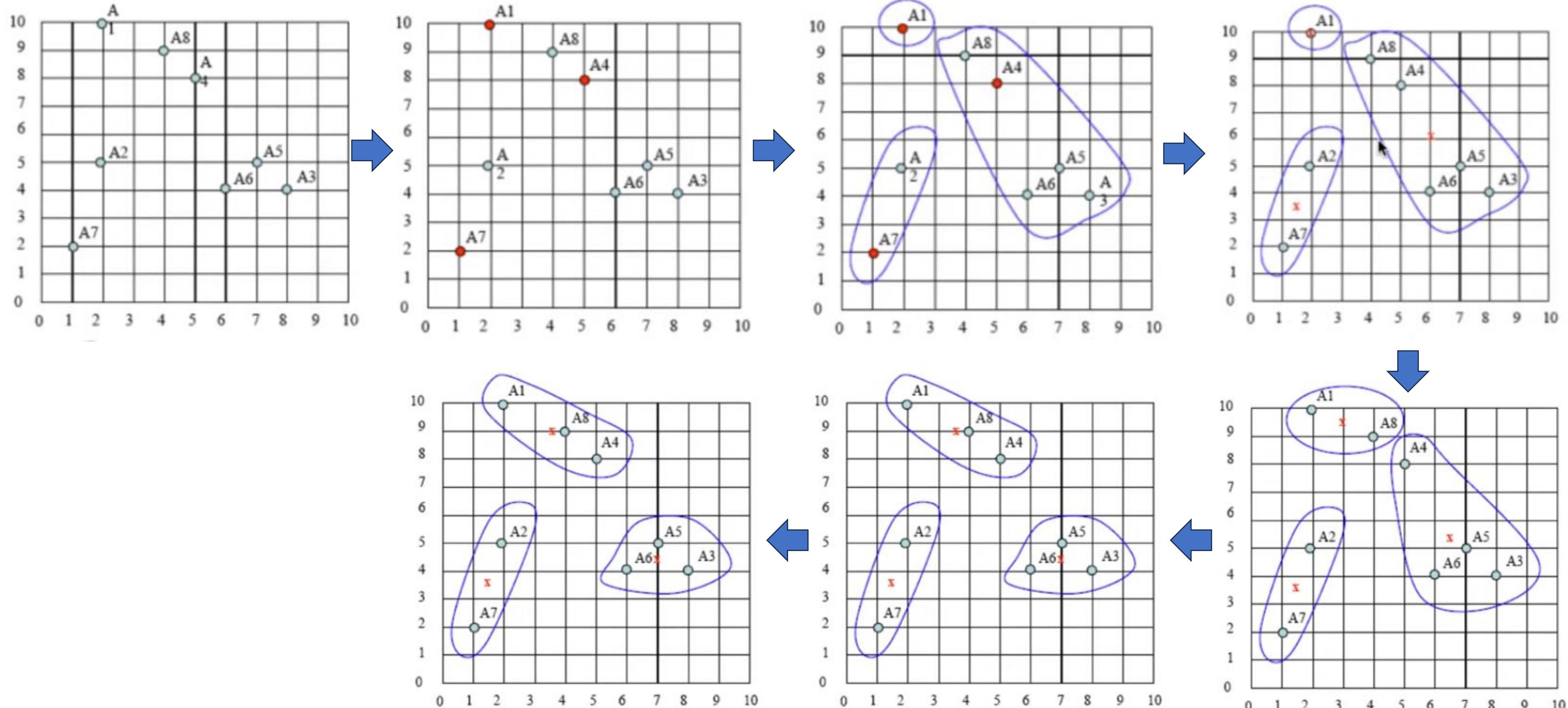
Phân cụm dữ liệu K trung tâm (K-means clustering)

- Tính lại giá trị khoảng cách các means:
 - cluster1: (2,10), (4,9), (5,8)
 - cluster2: (8,4), (7,5), (6,4)
 - cluster3: (2,5), (1,2)
- Nhận xét: số cụm không đổi so với lần lặp trước. TẠM DỪNG tính toán means

		(3.67,9)	(7,4.3)	(1.5,3.5)	
	Point	Dist. mean1	Dist. mean2	Dist. mean3	Cluster
A1	(2,10)	2,67	10,7	7	1
A2	(2,5)	5,67	5,7	2	3
A3	(8,4)	9,33	1,3	7	2
A4	(5,8)	2,33	5,7	8	1
A5	(7,5)	7,33	0,7	7	2
A6	(6,4)	7,33	1,3	5	2
A7	(1,2)	9,67	8,3	2	3
A8	(4,9)	0,33	7,7	8	1

Chương 7. MÁY HỌC

Phân cụm dữ liệu K trung tâm (K-means clustering)



Chương 7. MÁY HỌC

Phân cụm dữ liệu K trung tâm (K-means clustering)

- Phân cụm cho 8 điểm có tọa độ như sau thành 2 cụm:
 $A_1(1,1), A_2(2,1), A_3(2,3), A_4(3,2), A_5(4,3), A_6(5,5)$

HẾT CHƯƠNG 7