

BỘ MÔN KHOA HỌC DỮ LIỆU

NHẬP MÔN TRÍ TUỆ NHÂN TẠO

CHƯƠNG 2 – BÀI TOÁN TÌM KIẾM RÀNG BUỘC



TS. TRẦN NGỌC VIỆT
TH.S PHAN HỒ VIỆT TRƯỜNG



HỌC KỲ 3 – NĂM HỌC 2024-2025



KHÓA K29 & K28

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

1. Giải thuật minimax

Hai người thay phiên nhau đưa ra các nước đi tuân theo các luật của trò chơi.

- Ví dụ : cờ vua, cờ tướng, cờ ca rô (tic-tắc-toe)

Biểu diễn trong không gian trạng thái, mỗi trạng thái là một tình thế của cuộc chơi

- Trạng thái xuất phát là sự sắp xếp các quân cờ của hai bên khi bắt đầu cuộc chơi
- Các toán tử biến đổi trạng thái là các nước đi hợp lệ
- Các trạng thái kết thúc là các tình thế mà cuộc chơi dừng, được xác định bởi một số điều kiện dừng.
- Hàm kết cuộc: mang giá trị tương ứng với mỗi trạng thái kết thúc.

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

1. Giải thuật minimax

- Chiến lược Minimax

Hai đối thủ trong trò chơi có tên là **MAX** và **MIN**

- Max: biểu diễn cho mục đích của đối thủ này là làm lớn tối đa lợi thế của mình
- Min: biểu diễn cho mục đích của đối thủ này là làm nhỏ tối đa lợi thế của đối phương.

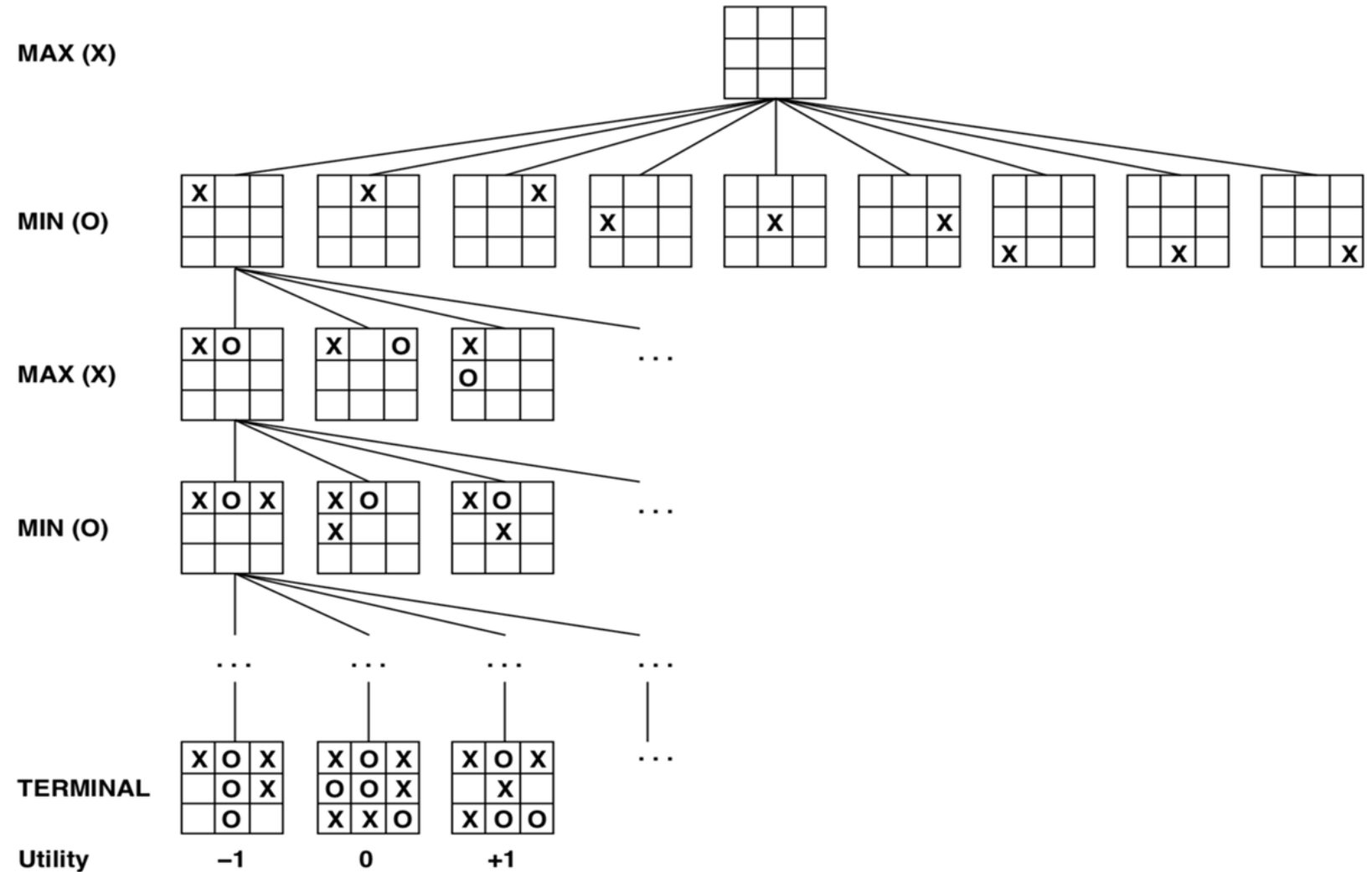
Trên cây tìm kiếm sẽ phân lớp thành các lớp Max và Min.

Với một node **n** bất kỳ,

- Nếu nó thuộc lớp Max thì gán cho nó giá trị Max của các node con
- Nếu nó thuộc lớp Min thì gán cho nó giá trị nhỏ nhất của các node con.

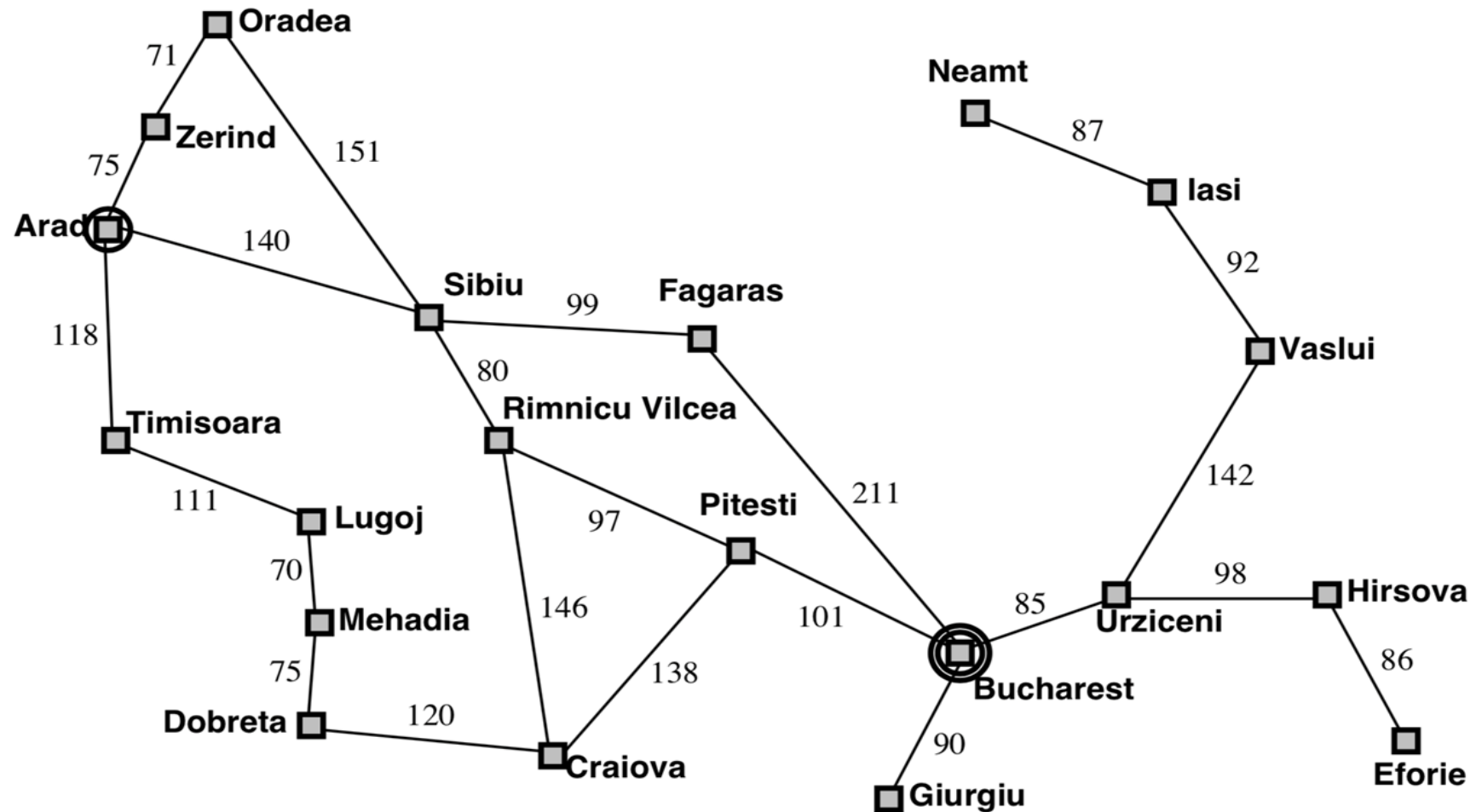
CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

1. Giải thuật minimax



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi



ARTIFICIAL INTELLIGENCE

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

- Bài toán tìm đường có thể phát biểu theo 5 thành phần như sau:
 - Trạng thái: vị trí của ô tô (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 trong phép chuyển trạng thái

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

**Giải thuật tổng quát tìm kiếm lời giải*

- Không gian trạng thái của bài toán: Không gian trạng thái có thể là vô hạn hoặc hữu hạn
- Không gian trạng thái nhỏ \longrightarrow thuật toán tìm đường đi trong lý thuyết đồ thị
- Không gian trạng thái lớn \longrightarrow ???

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

**Chiến lược tìm kiếm mù (Uninformed search strategies)*

Chiến lược tìm kiếm mù chỉ sử dụng thông tin có sẵn được định nghĩa trong bài toán.

Các thuật toán tìm kiếm mù:

- Tìm kiếm theo chiều rộng (Breadth-first search)
- Tìm kiếm theo chiều sâu (Depth-first search)

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

**Đánh giá giải thuật tìm kiếm*

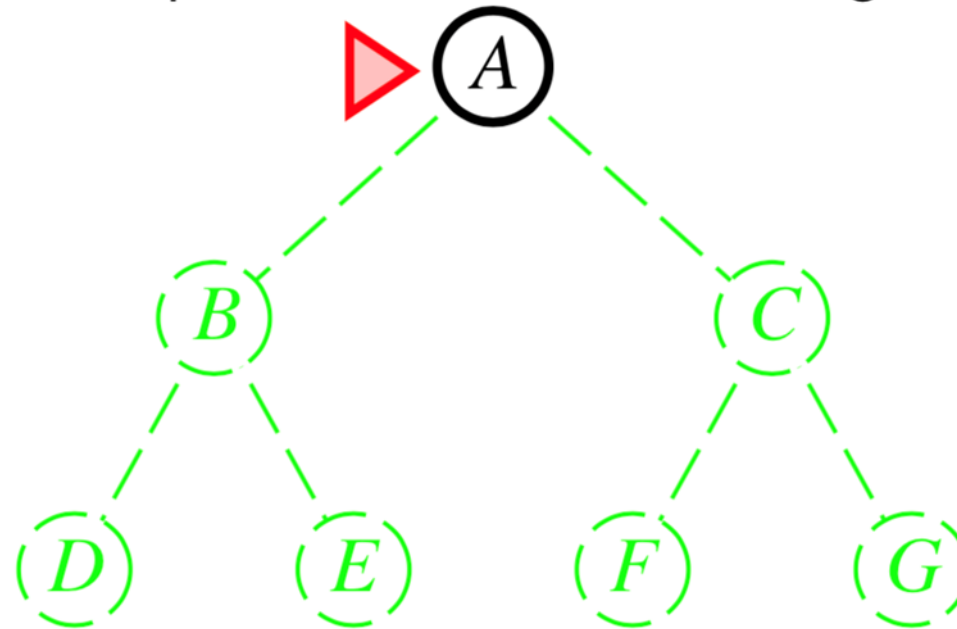
- **Tính đầy đủ:** giải thuật có tìm được lời giải của bài toán không nếu bài toán tồn tại lời giải?
- **Độ phức tạp thời gian:** thời gian của giải thuật có kích cỡ như thế nào đối với bài toán?
- **Độ phức tạp không gian:** Kích cỡ của bộ nhớ cần cho giải thuật? Trong giải thuật tổng quát ở trên, kích cỡ bộ nhớ chủ yếu phụ thuộc vào 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)?

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.1. Tìm kiếm theo chiều rộng

Mở rộng bằng hình thức vét cạn, sử dụng hàng đợi FIFO

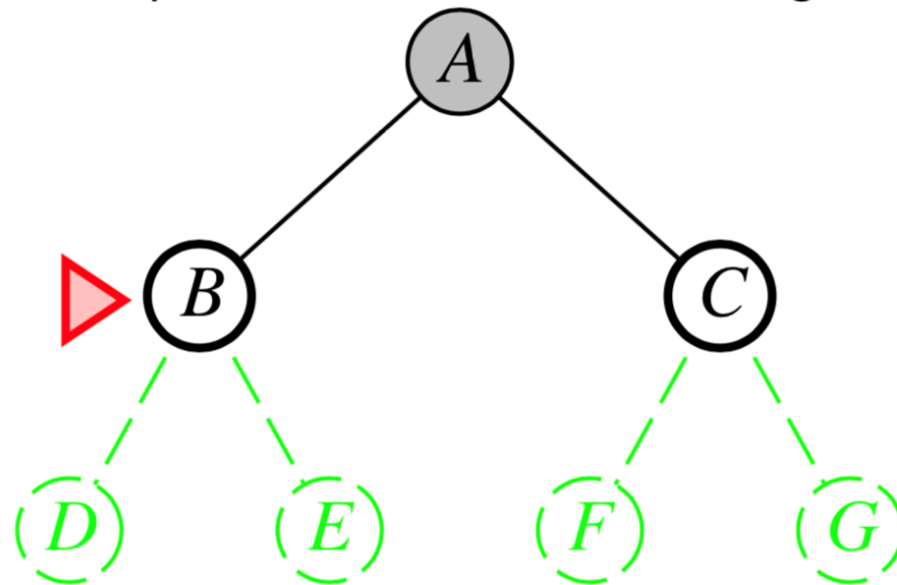


CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.1. Tìm kiếm theo chiều rộng

Mở rộng bằng hình thức vét cạn, sử dụng hàng đợi FIFO

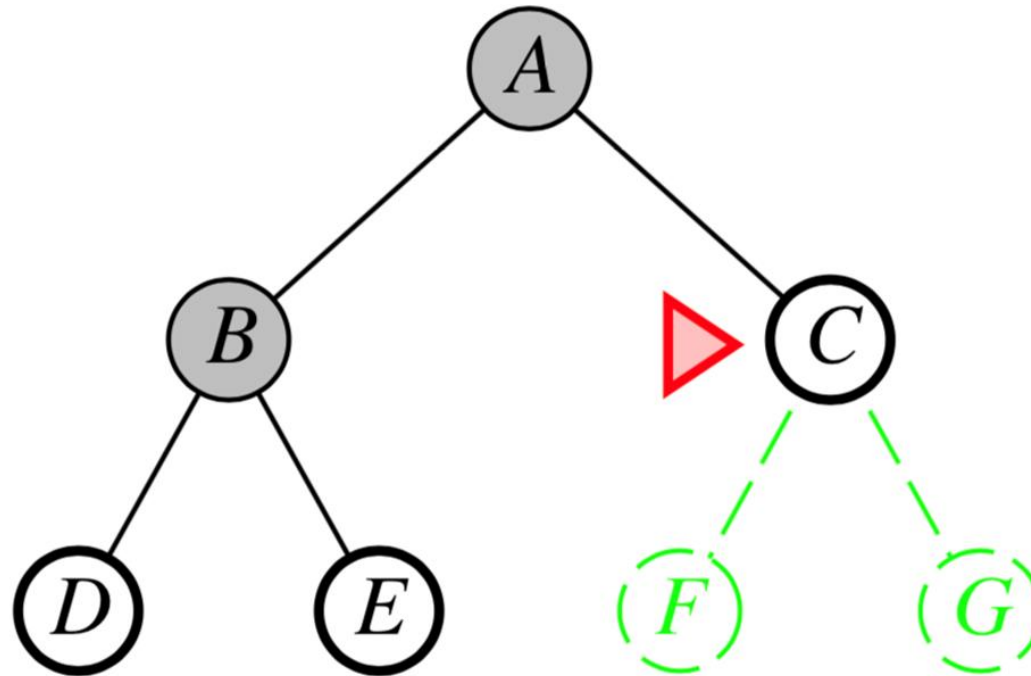


CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.1. Tìm kiếm theo chiều rộng

Mở rộng bằng hình thức vét cạn, sử dụng hàng đợi FIFO

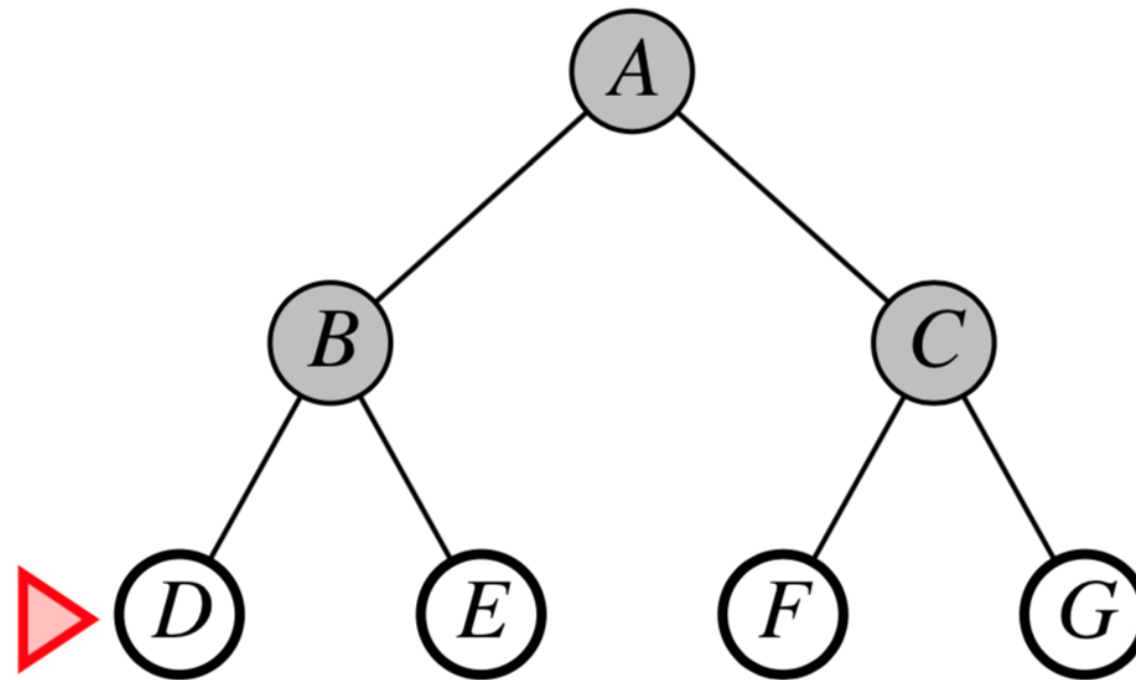


CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.1. Tìm kiếm theo chiều rộng

Mở rộng bằng hình thức vét cạn, sử dụng hàng đợi FIFO



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

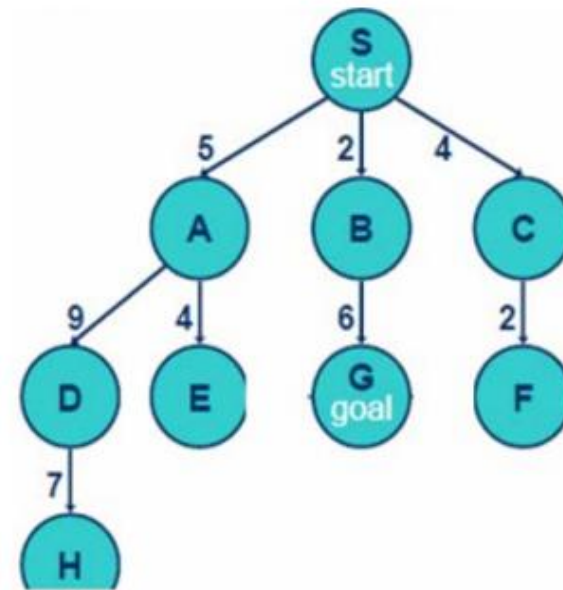
2. Bài toán tìm đường đi

2.1. Tìm kiếm theo chiều rộng

Mở rộng bằng hình thức vét cạn, sử dụng hàng đợi FIFO

node	Queue	Father
	S	
S	A, B, C	Father[A,B,C]=S
A	B, C, D, E	Father[D,E]=A
B	C,D,E,G	Father[G]=B
C	D, E, G, F	Father[F]=C
D	E,G, F, H	Father[H]=D
E	G, F, H	
G	F, H	

Giá trị các biến trong
giải thuật theo chiều rộng



Cây tìm kiếm của giải thuật theo chiều rộng

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.1. Tìm kiếm theo chiều rộng

***Đánh giá giải thuật:**

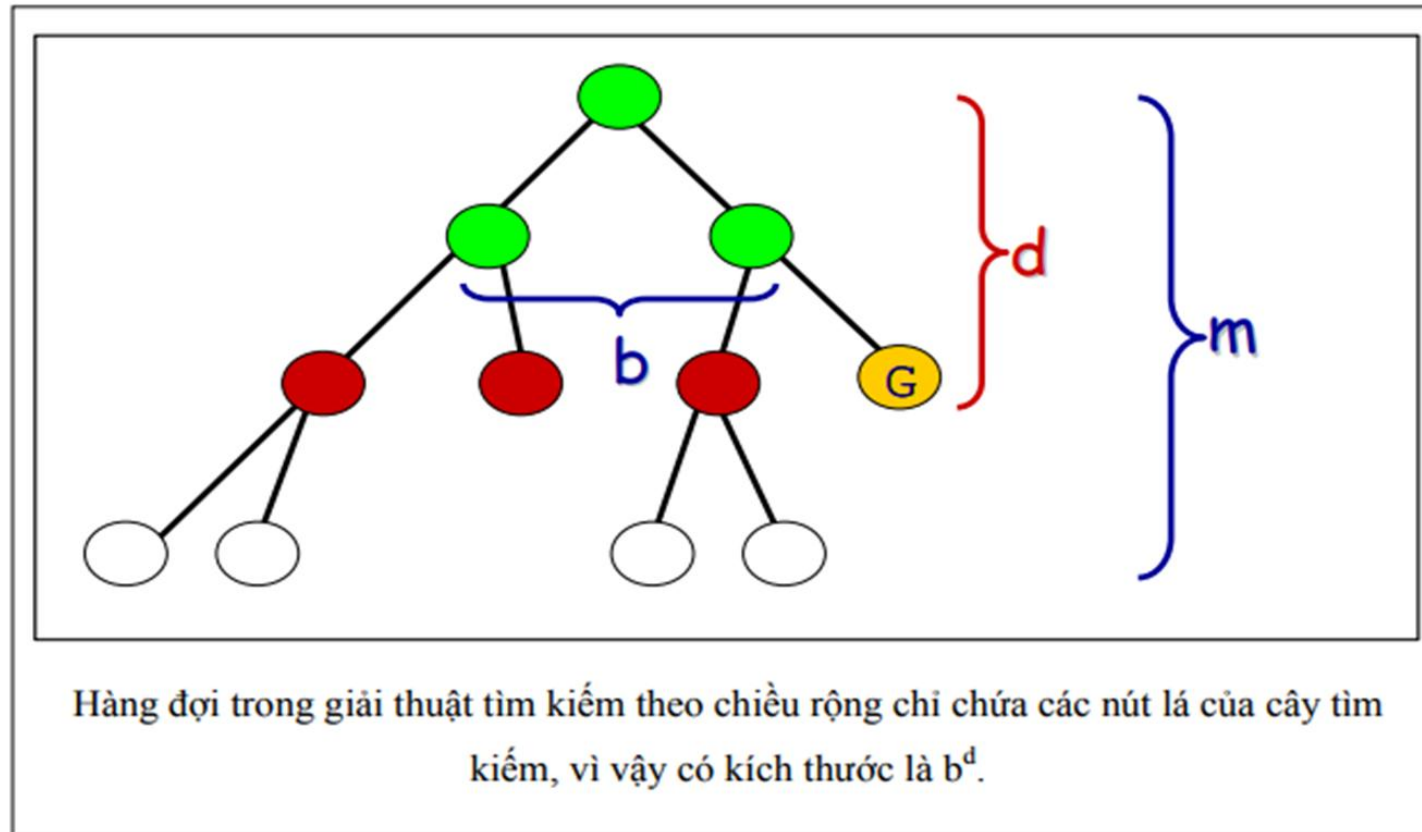
- **Tính đầy đủ:** giải thuật sẽ cho lời giải của bài toán nếu bài toán tồn tại lời giải và nhân tố nhánh b là hữu hạn
- **Độ phức tạp thời gian:** $1 + b + b^2 + \dots + b^d = O(b^d)$ (số vòng lặp khi gặp trạng thái đích $O(b^d)$)
- **Độ phức tạp không gian:** số lượng ô nhớ tối đa sử dụng trong giải thuật (chủ yếu là biến Queue): $O(b^d)$
- **Tính tối ưu:** giải thuật tìm kiếm theo chiều rộng sẽ tìm ra lời giải với ít trạng thái trung gian nhất.

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.1. Tìm kiếm theo chiều rộng

*Đánh giá giải thuật:

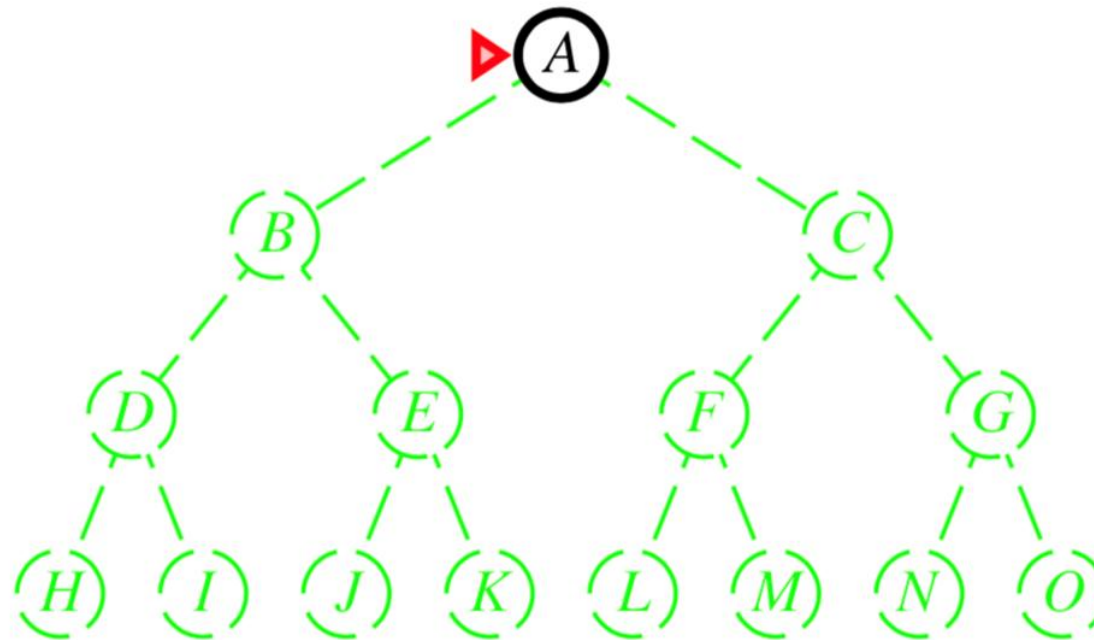


CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.2. Tìm kiếm theo chiều sâu

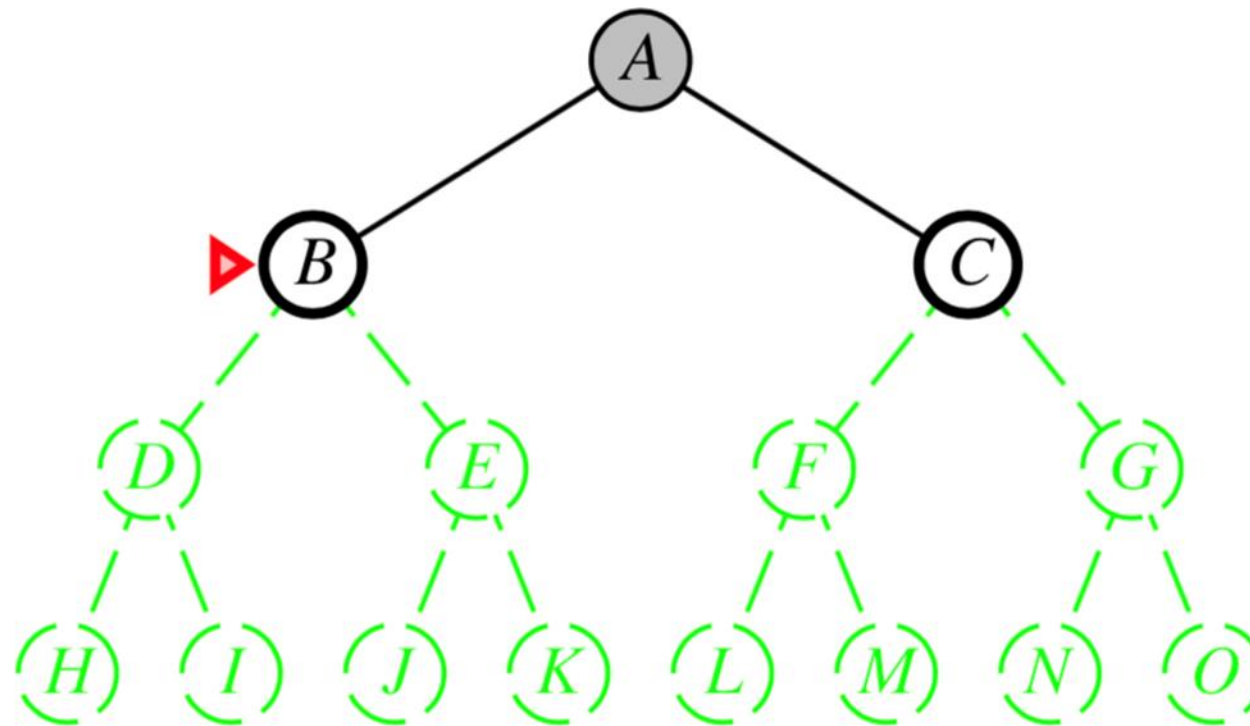
Sử dụng cấu trúc dữ liệu ngăn xếp (Stack) LIFO để lưu giữ các trạng thái lá của cây tìm kiếm



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

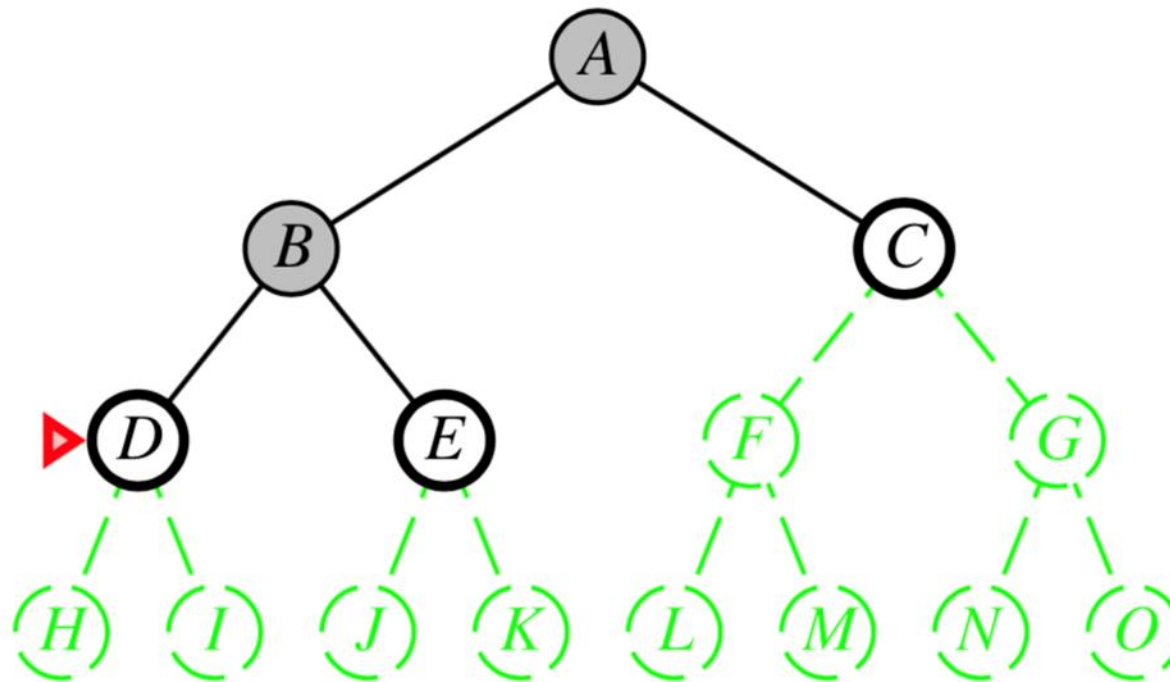
2.2. Tìm kiếm theo chiều sâu



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

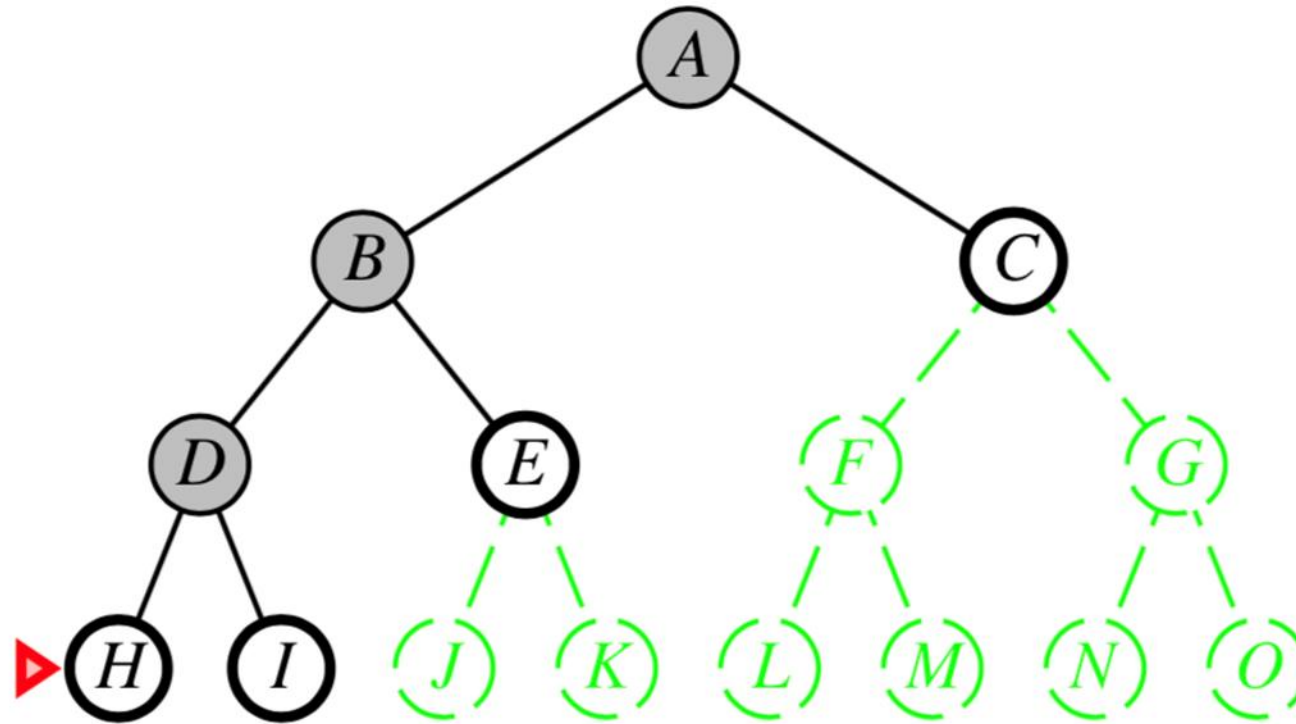
2.2. Tìm kiếm theo chiều sâu



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

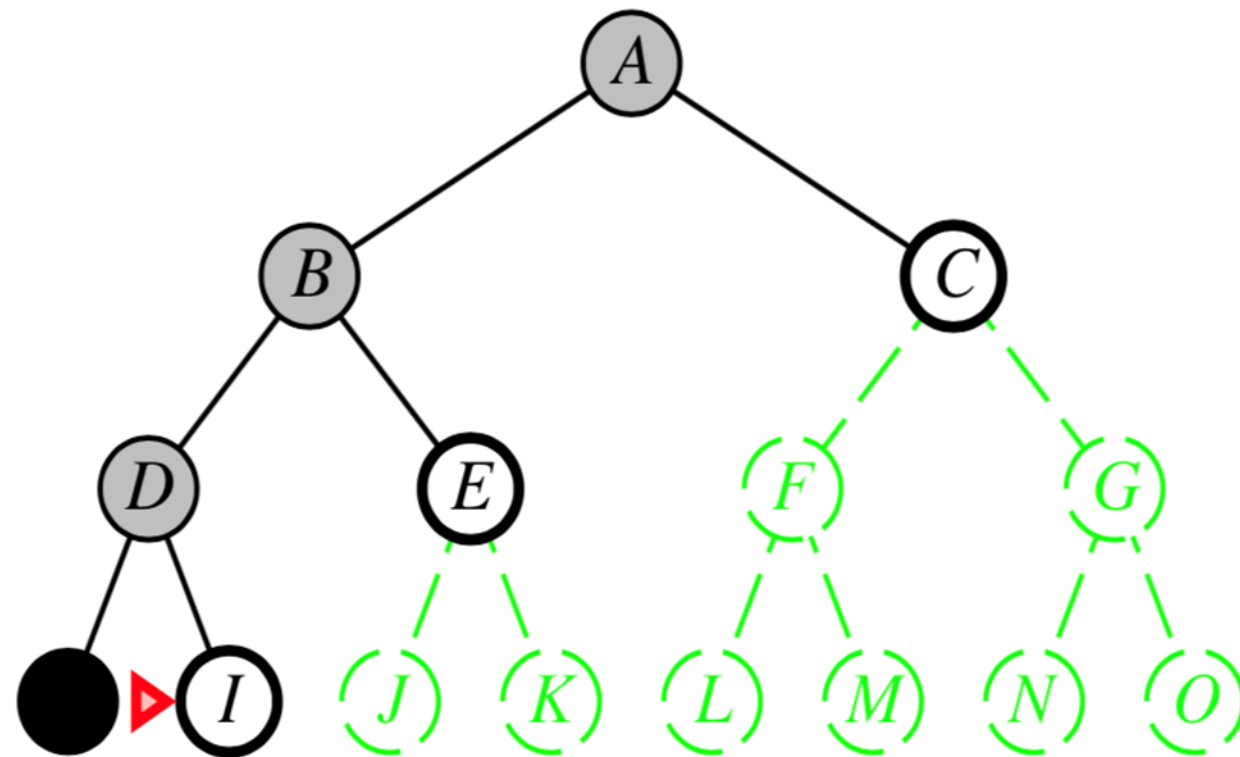
2.2. Tìm kiếm theo chiều sâu



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

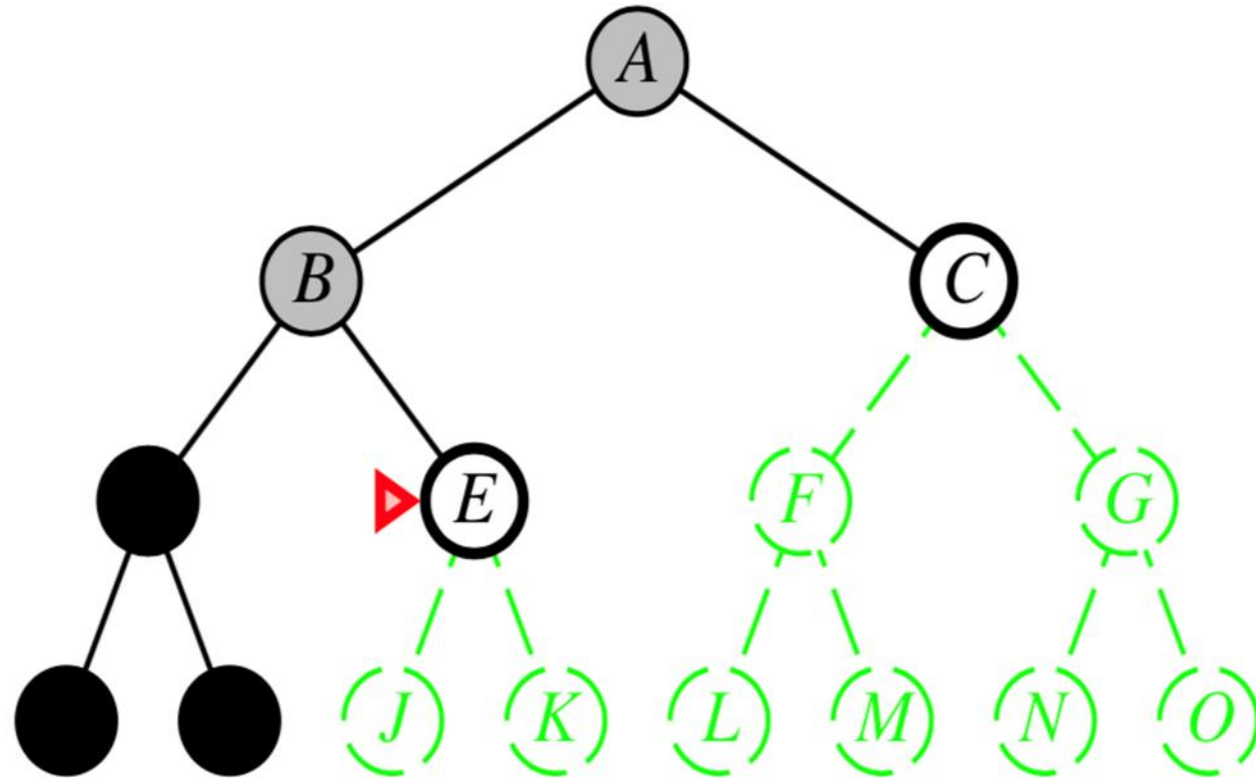
2.2. Tìm kiếm theo chiều sâu



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

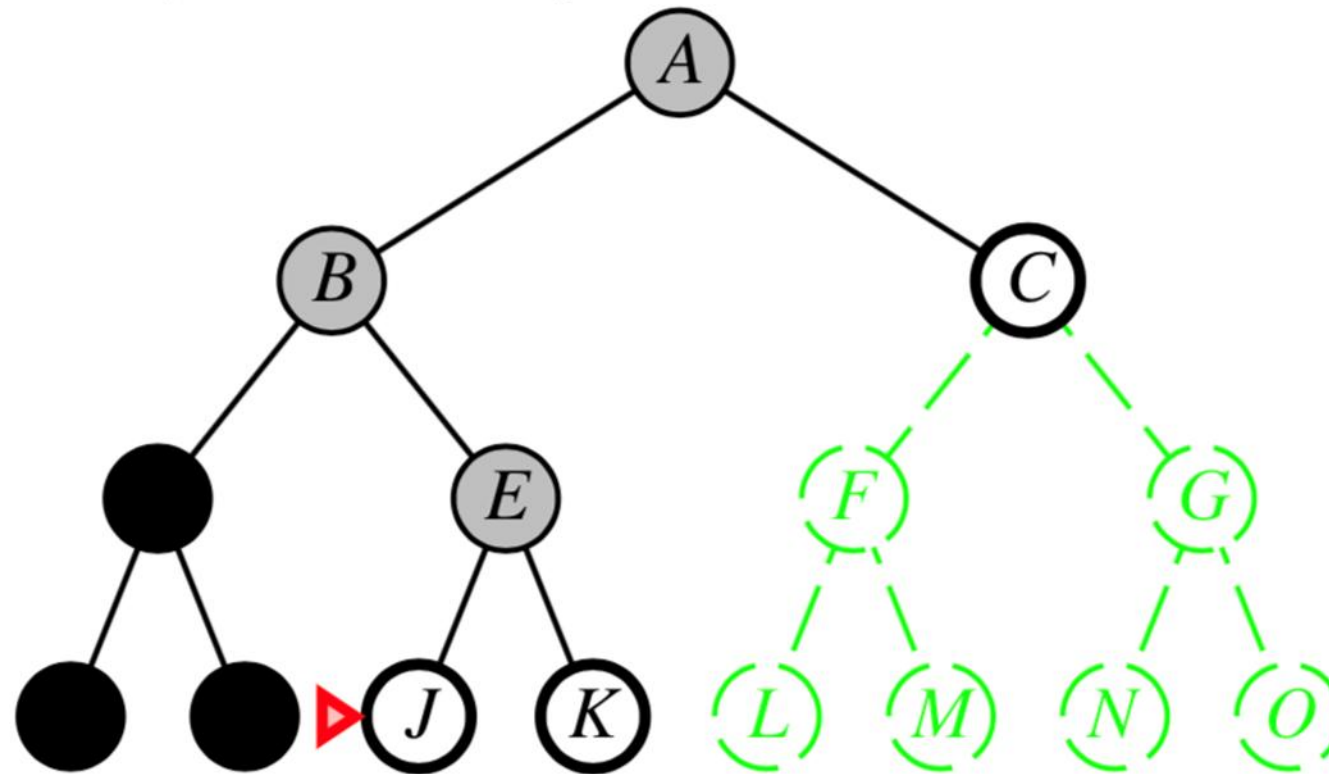
2.2. Tìm kiếm theo chiều sâu



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

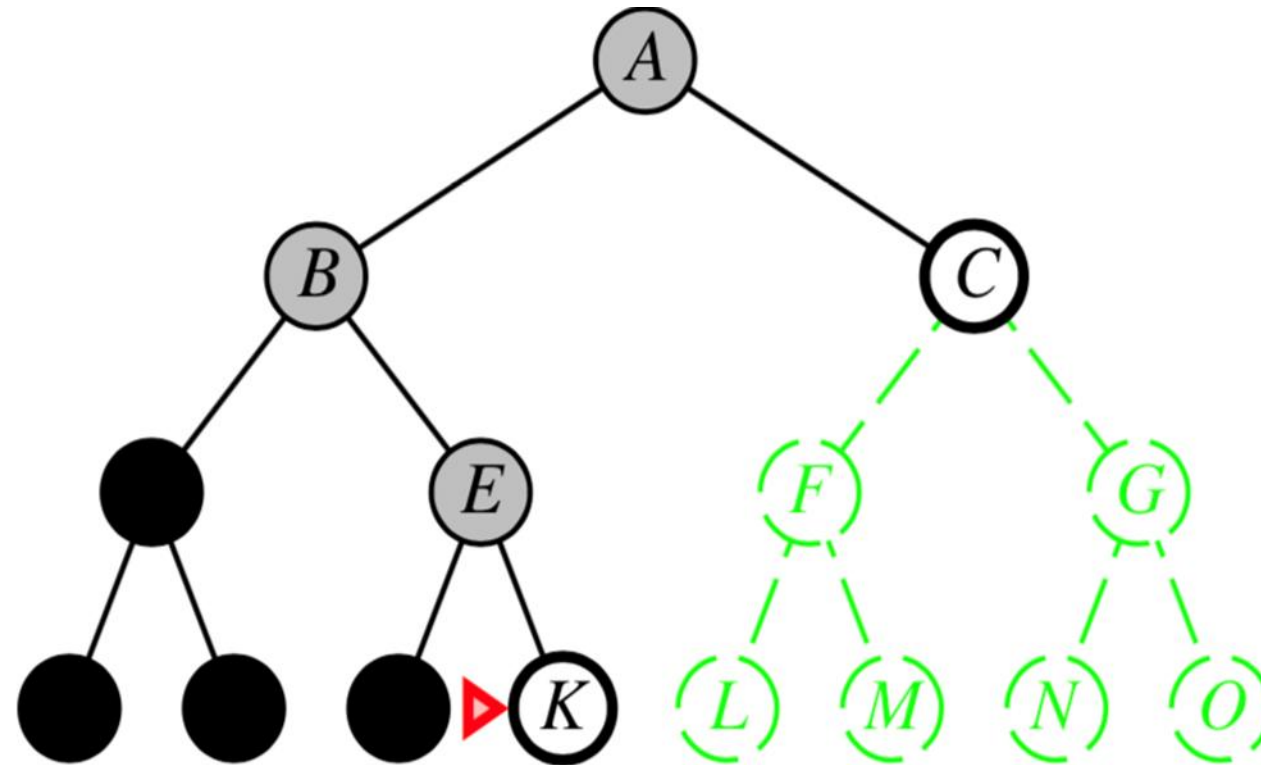
2.2. Tìm kiếm theo chiều sâu



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

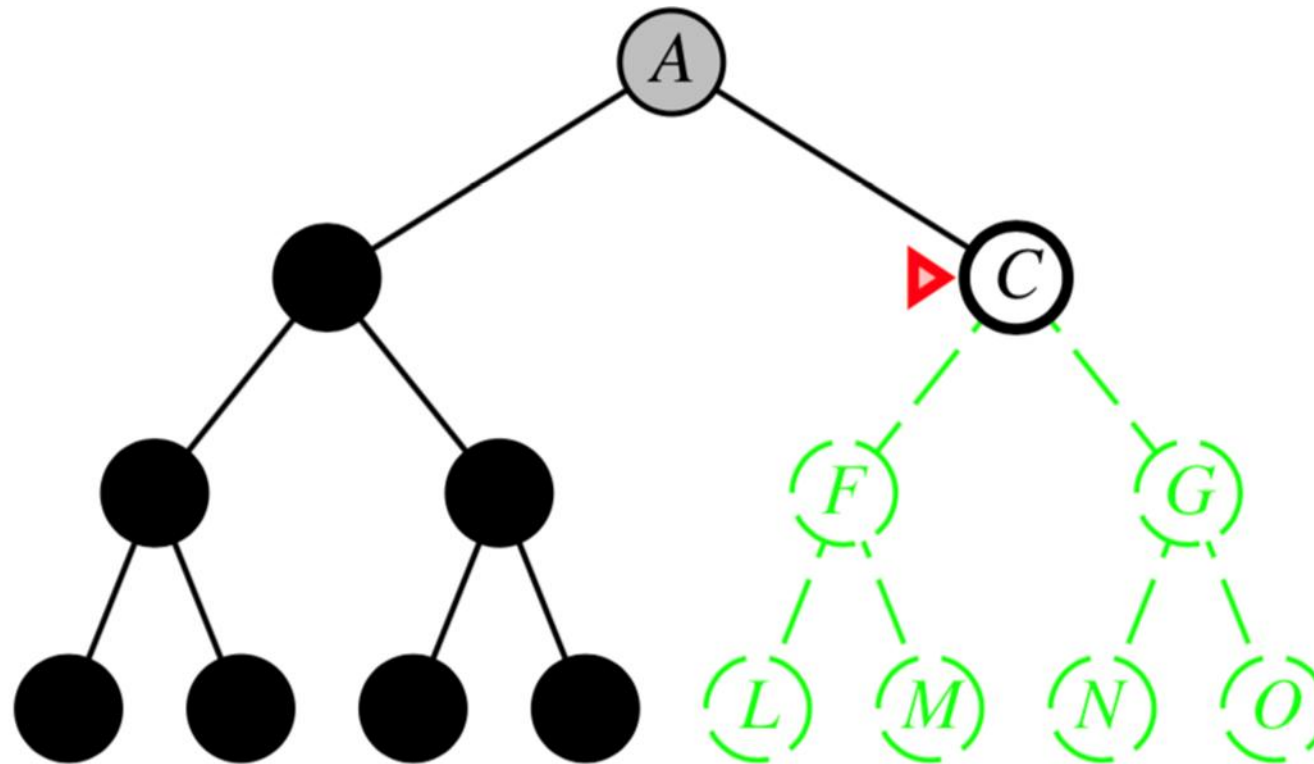
2.2. Tìm kiếm theo chiều sâu



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

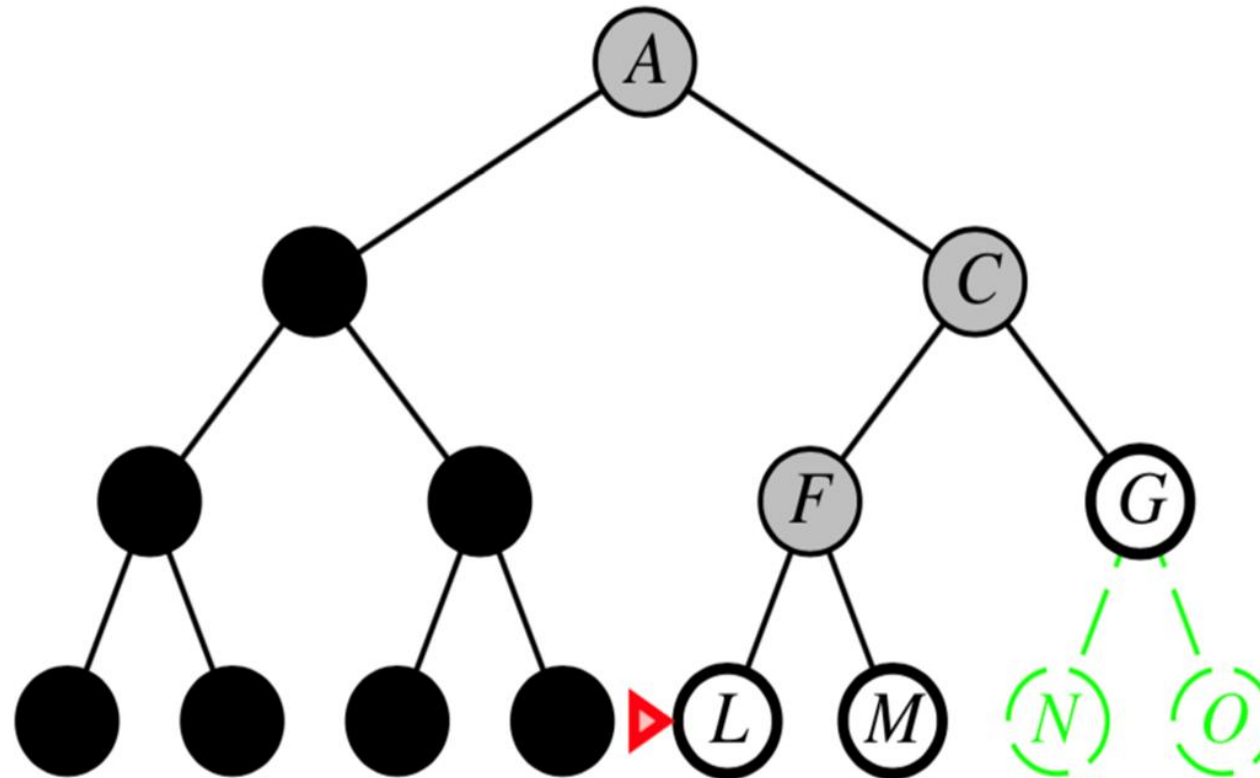
2.2. Tìm kiếm theo chiều sâu



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

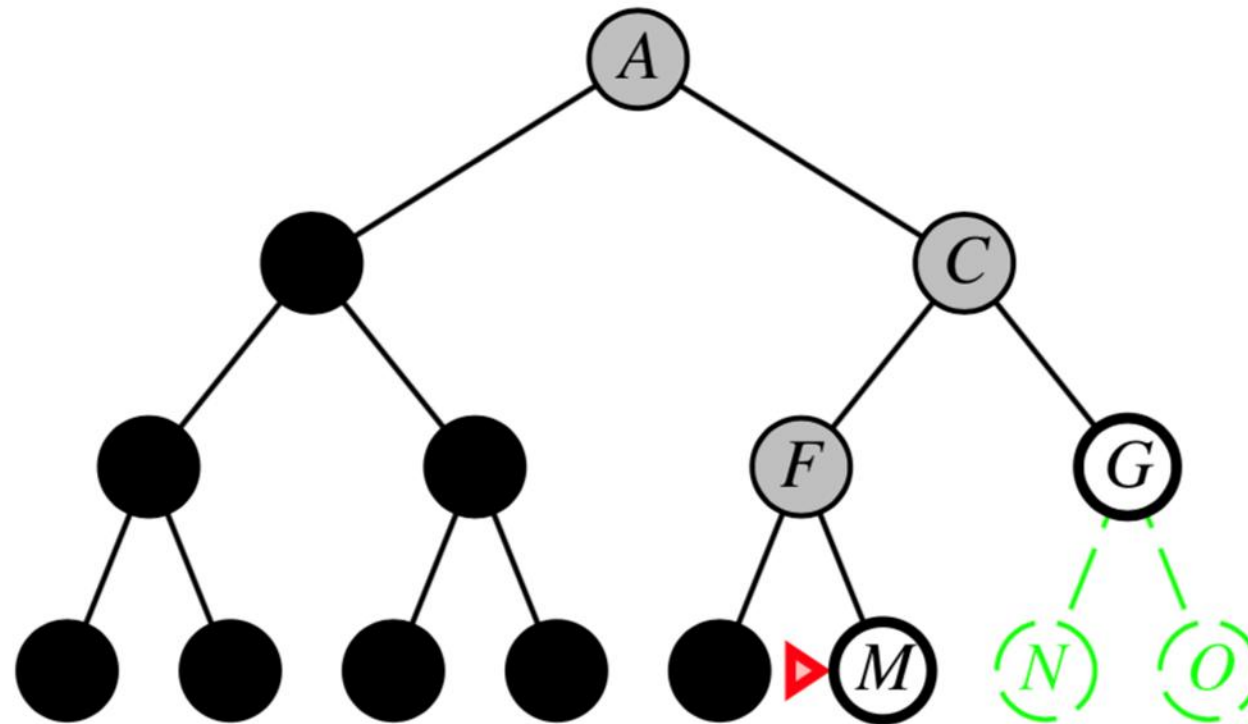
2.2. Tìm kiếm theo chiều sâu



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.2. Tìm kiếm theo chiều sâu



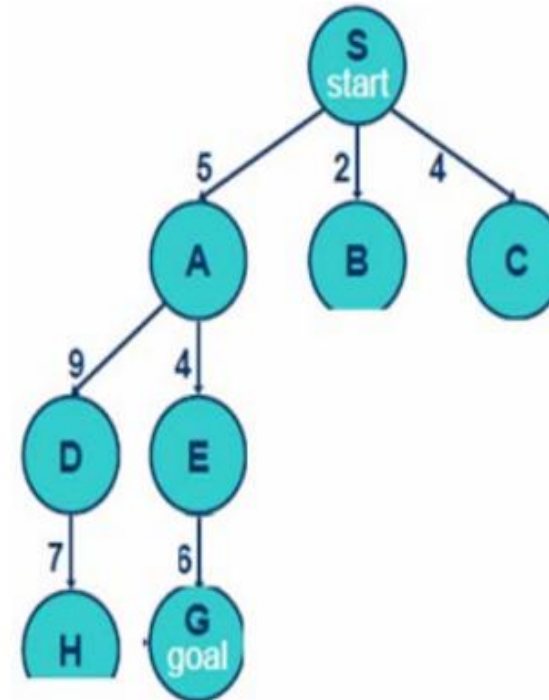
CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.2. Tìm kiếm theo chiều sâu

node	Stack	father
	S	
S	A, B, C	Father[A,B,C]=S
A	D, E, B, C	Father[D,E]=A
D	H, E, B, C	Father[H]=D
H	E, B, C	
E	G, B, C	Father[G]=E
G		

Giá trị các biến trong
giải thuật theo chiều sâu



Cây tìm kiếm của giải thuật theo chiều

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.2. Tìm kiếm theo chiều sâu

***Đánh giá giải thuật:**

- **Tính đầy đủ:** giải thuật không chắc chắn cho lời giải của bài toán trong trường hợp không gian trạng thái của bài toán là vô hạn
- **Độ phức tạp thời gian:** $O(b^m)$
- **Độ phức tạp không gian:** $O(bm)$
- **Tính tối ưu:** giải thuật tìm kiếm theo chiều sâu không cho lời giải tối ưu.

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.2. Tìm kiếm theo chiều sâu

Tìm kiếm theo chiều sâu có giới hạn

- Ưu điểm Giải thuật tìm kiếm theo chiều sâu: sinh ra lời giải nhanh chóng mà không tốn kém bộ nhớ của máy tính
- Nếu không gian trạng thái của bài toán là vô hạn ?
- Dẫn tới giới hạn độ sâu trong giải thuật

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.2. Tìm kiếm theo chiều sâu

Tìm kiếm sâu dần $l = 0$

Limit = 0



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.2. Tìm kiếm theo chiều sâu

Tìm kiếm sâu dần $l = 1$

Limit = 1

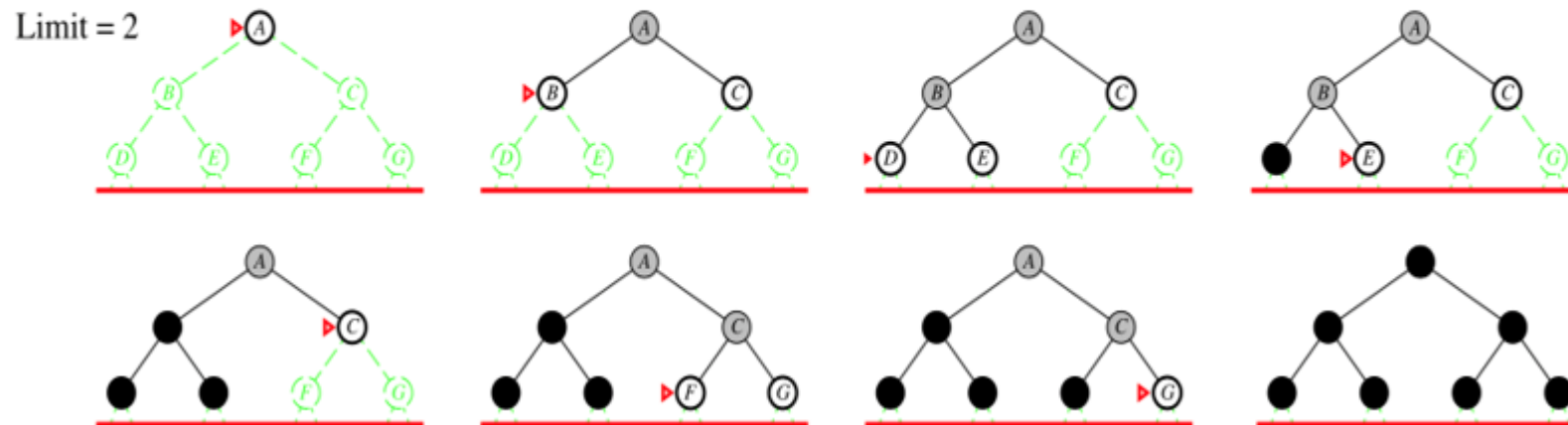


CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.2. Tìm kiếm theo chiều sâu

Tìm kiếm sâu dần $l = 2$



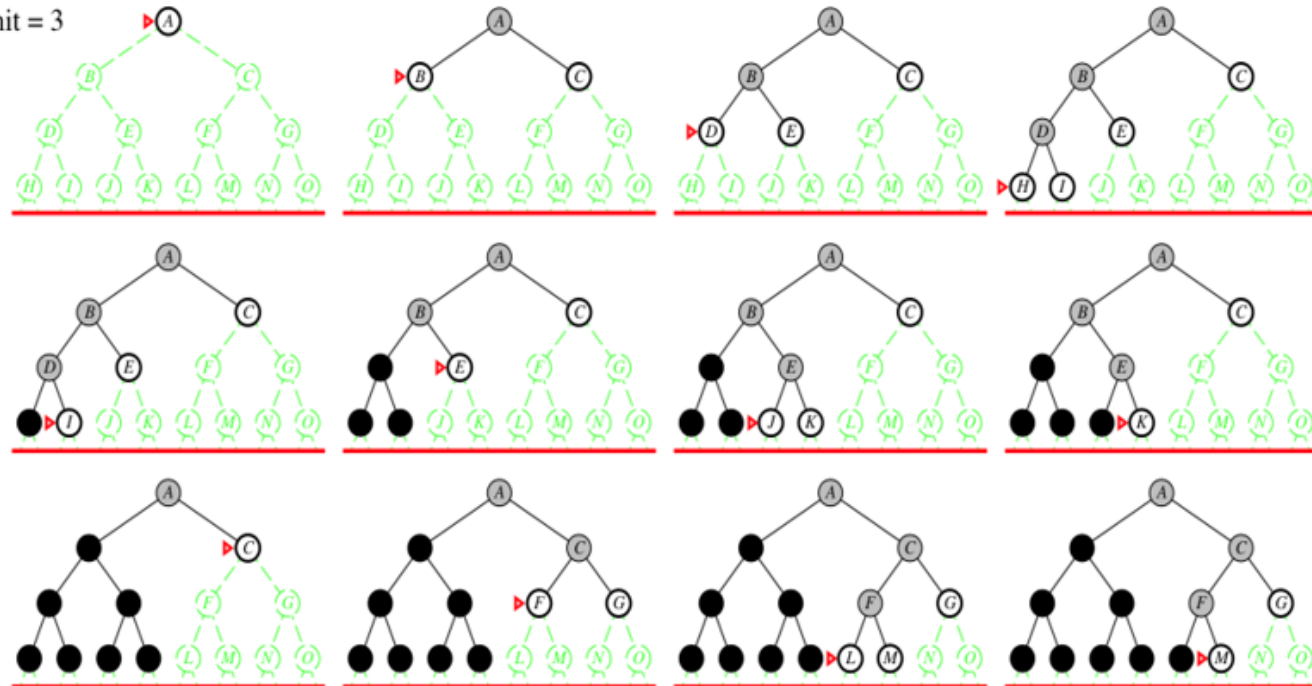
CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.2. Tìm kiếm theo chiều sâu

Tìm kiếm sâu dần $l = 3$

Limit = 3



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

2. Bài toán tìm đường đi

2.2. Tìm kiếm theo chiều sâu

- Tính đầy đủ: có
- Độ phức tạp thời gian: $O(b^d)$
- Độ phức tạp không gian: $O(bd)$
- Tính tối ưu: có cho lời giải tối ưu.

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

*Tổng kết thuật toán:

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening
Complete?	Yes*	Yes*	No	Yes, if $l \geq d$	Yes
Time	b^{d+1}	$b^{\lceil C^*/\epsilon \rceil}$	b^m	b^l	b^d
Space	b^{d+1}	$b^{\lceil C^*/\epsilon \rceil}$	bm	bl	bd
Optimal?	Yes*	Yes	No	No	Yes*

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

3. Bài toán 8 câu đố - 8 Puzzle Problem

+N - Puzzle hoặc câu đố trượt là một câu đố phổ biến bao gồm N ô . Trong đó, N Puzzle có thể là 8, 15, 24, v.v.

+Ví dụ: N = 8. Câu đố được chia thành:

$\sqrt{N+1}$ hàng và $\sqrt{N+1}$ cột.

+Ví dụ: 15 - Puzzle sẽ có 4 hàng và 4 cột;

8 - Puzzle sẽ có 3 hàng và 3 cột.

+Câu đố bao gồm N ô vuông và một ô trống để các ô có thể được di chuyển. Cấu hình Bắt đầu và Mục tiêu (là trạng thái) của câu đố được cung cấp. Câu đố có thể được giải bằng cách di chuyển từng ô một trong không gian trống duy nhất và đạt được cấu hình Mục tiêu.

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

3. Bài toán 8 câu đố - Puzzle Problem

+Quy tắc: Di chuyển các ô trong không gian trống, có thể hình dung việc di chuyển không gian trống thay cho ô, có thể hoán đổi ô với không gian trống. Không gian trống chỉ có thể di chuyển theo bốn hướng:

1) Lên

2) Xuống

3) Phải

4) Trái

Không gian trống *không thể* di chuyển theo đường chéo và chỉ có thể thực hiện từng bước một (tức là di chuyển không gian trống từng vị trí một).

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

3. Bài toán 8 câu đố - Puzzle

- +Đầu tiên, di chuyển không gian trống theo tất cả các hướng có thể ở trạng thái bắt đầu và tính điểm f (chi phí) cho mỗi trạng thái, được gọi là mở rộng trạng thái hiện tại.
- +Mở rộng trạng thái hiện tại, nó được đẩy vào danh sách đóng và các trạng thái mới tạo được đẩy vào danh sách mở.
- +Trạng thái có điểm f nhỏ nhất được chọn và mở rộng. Quá trình này tiếp tục cho đến khi trạng thái mục tiêu xảy ra như trạng thái hiện tại.
- +Thuật toán chọn hành động tốt nhất có thể và tiến hành theo con đường đó.
- +Giải quyết vấn đề tạo ra các trạng thái con dư thừa, vì thuật toán sẽ mở rộng nút có điểm f (chi phí) nhỏ nhất.

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

3. Bài toán 8 câu đố - Puzzle

+ $f(x)$: Hàm Chi phí lý tưởng cho bài toán 8 câu đố. Giả sử rằng việc di chuyển một ô theo bất kỳ hướng nào sẽ có chi phí đơn vị là 1. Xác định hàm chi phí cho thuật toán 8 câu đố như sau:

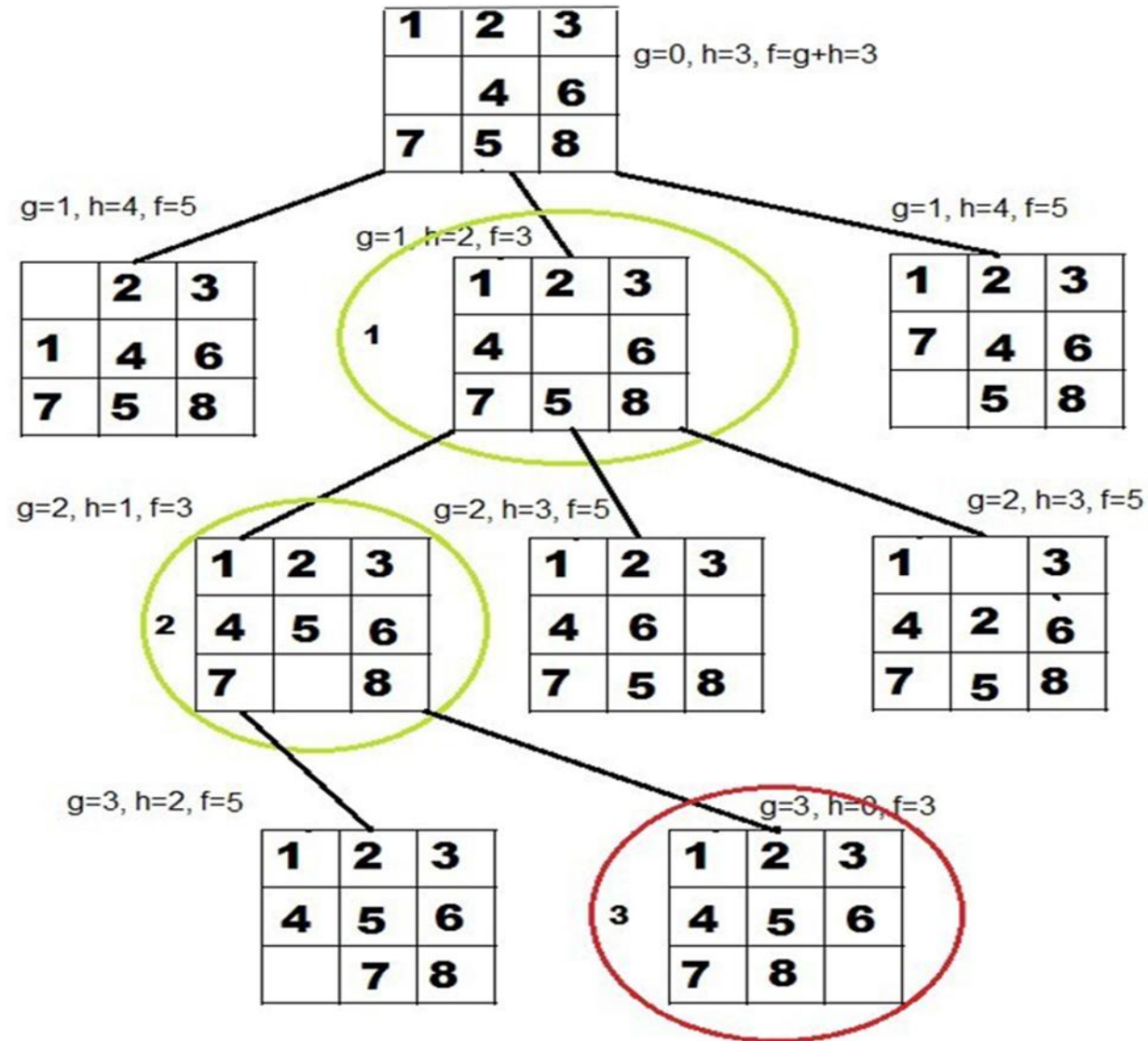
$$f(x) = g(x) + h(x) \text{ khi đó,}$$

$g(x)$: độ dài của đường dẫn từ gốc đến x - số lần di chuyển;

$h(x)$: số lượng gạch không trống không có trong vị trí mục tiêu. Có ít nhất $h(x)$ di chuyển để chuyển trạng thái x sang trạng thái mục tiêu;

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

3. Bài toán 8 câu đố - Puzzle



CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

3. Bài toán 8 câu đố - Puzzle

$$f(X) = g(X) + h(X),$$

Trong đó:

$g(X)$ = cost - chi phí tiếp cận nút hiện tại từ gốc
(mức 0); mức 1; mức 2; mức 3

$h(X)$ = cost - chi phí để đạt được nút trả lời từ X
(số ô bị đặt sai vị trí)

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

3. Bài toán 8 câu đố - Puzzle

#8 puzzle problem (8 câu đố)

- + Cho một bảng 3×3 có 8 ô (mỗi ô có một số từ 1 đến 8) và một ô trống.
- + Mục tiêu là đặt các số trên các ô để khớp với cấu hình cuối cùng bằng cách sử dụng khoảng trống.
- + Có thể trượt bốn ô liên kề (trái, phải, trên và dưới) vào khoảng trống

Initial
configuration

1	2	3
5	6	
7	8	4

Final
configuration

1	2	3
5	8	6
	7	4

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

3. Bài toán 8 câu đố - Puzzle

#out:

1	2	3
5	6	0
7	8	4

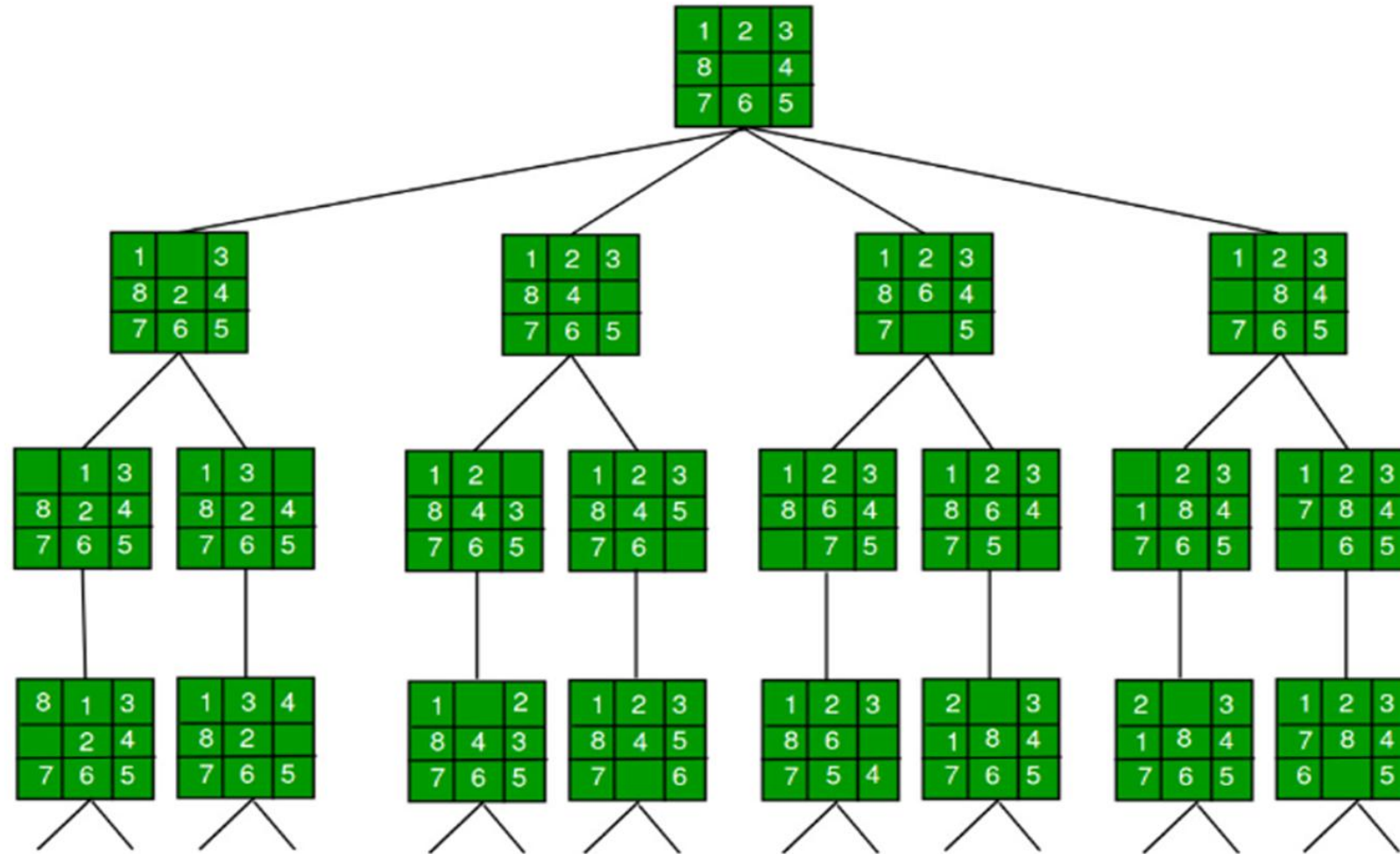
1	2	3
5	0	6
7	8	4

1	2	3
5	8	6
7	0	4

1	2	3
5	8	6
0	7	4

CHƯƠNG 1 - BÀI TOÁN TÌM KIẾM - SOLVING PROBLEMS BY SEARCHING

3. Bài toán 8 câu đố - Puzzle

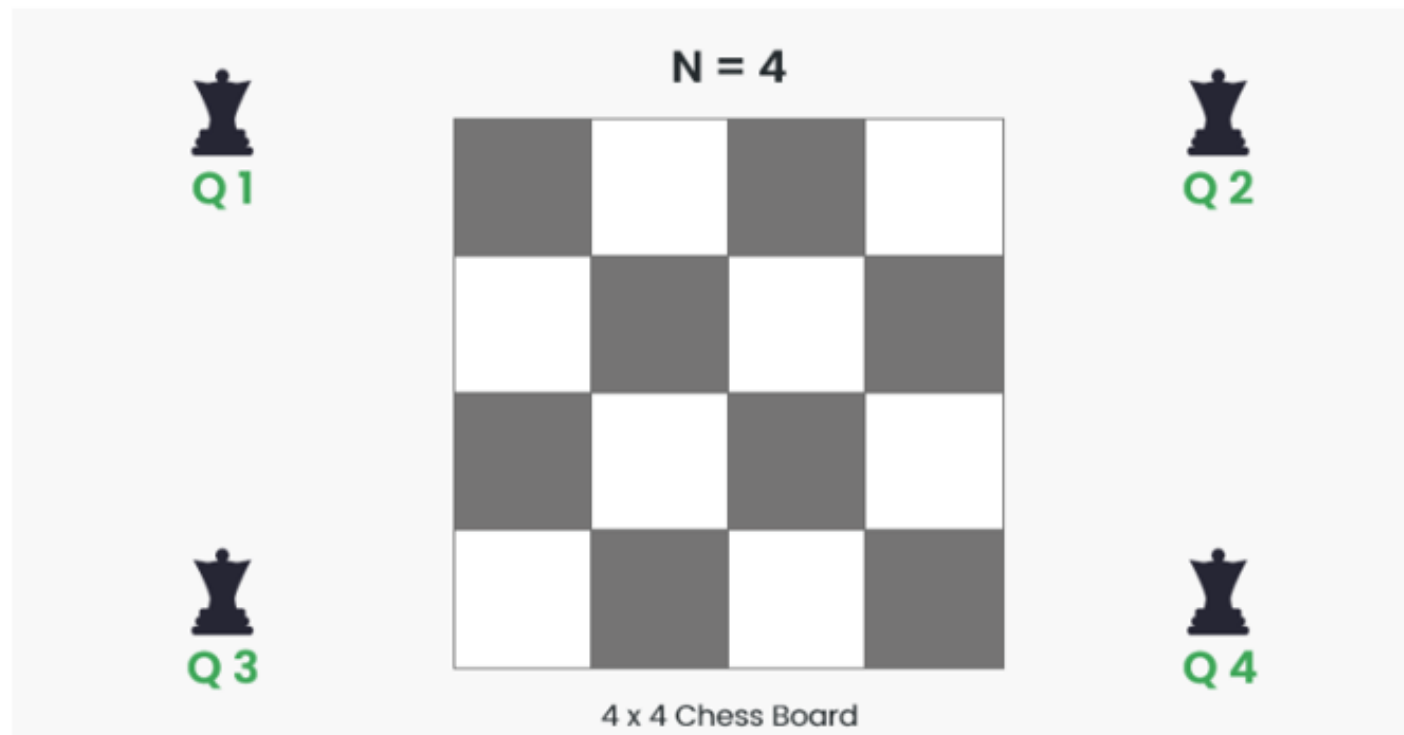


+Sơ đồ cho thấy để đạt được cấu hình cuối cùng từ cấu hình ban đầu đã cho của 8-Puzzle (8câu đố). Các nút có giá trị nhỏ nhất của hàm chi phí mới (mở rộng).

4. Bài toán tìm kiếm theo ràng buộc

+ Bài toán $n = 4$ Queen trên một bàn cờ vua có 4×4 .

+ Đặt ra không có hai Queen nào tấn công lẫn nhau. Ràng buộc điều kiện là không được phép đặt hai Queen trên cùng một hàng hoặc trên cùng một cột hoặc cùng một đường chéo.



4. Bài toán tìm kiếm theo ràng buộc

Step 1:

- +Đặt quân Queen Q1 vào ô (0,0).
- +‘X’ biểu thị các ô không an toàn tức là chúng đang bị Queen Q1 tấn công.
- +Sau đó di chuyển đến hàng tiếp theo.

	0	1	2	3
0	Q1	x	x	x
1	x	x		
2	x		x	
3	x			x

4 x 4 Chess Board

Step 2:

- +Đặt quân Queen Q2 tiếp theo vào ô (1, 2).
- +Di chuyển đến hàng tiếp theo.

	0	1	2	3
0	Q1	x	x	x
1	x	x	Q2	x
2	x	x	x	x
3	x		x	x

4 x 4 Chess Board

4. Bài toán tìm kiếm theo ràng buộc

Step 3:

- +Hàng 2 không có ô nào an toàn để đặt Queen Q3
- +Vì vậy, quay lại và loại bỏ Queen Q2 khỏi ô (1, 2)

Step 4:

- +Vẫn còn một ô an toàn ở hàng 1 tức là ô (1, 3).
- +Đặt Queen Q2 vào ô (1, 3).

	0	1	2	3
0	Q1	x	x	x
1	x	x	x	Q2
2	x		x	x
3	x	x		x

4 x 4 Chess Board

4. Bài toán tìm kiếm theo ràng buộc

Step 5:

+Đặt Queen Q3 vào ô (2, 1)

	0	1	2	3
0	Q1	x	x	x
1	x	x	x	Q2
2	x	Q3	x	x
3	x	x	x	x

4 x 4 Chess Board

Step 6:

+Không có ô nào để đặt Queen Q4 vào hàng 3.
+Quay lại và loại Queen Q3 khỏi hàng 2.
+Tiếp tục không có ô an toàn nào khác ở hàng 2, vì vậy quay lại và loại Queen Q2 khỏi hàng 1.
+Queen Q1 sẽ bị loại khỏi ô (0,0) và di chuyển đến ô an toàn tiếp theo tức là (0, 1).

Step 7:

+Đặt Queen Q1 vào ô (0, 1) và di chuyển đến hàng tiếp theo.

	0	1	2	3
0	x	Q1	x	x
1		x	x	
2		x		x
3		x		

4 x 4 Chess Board

4. Bài toán tìm kiếm theo ràng buộc

Step 8:

+Đặt Queen Q2 vào ô (1, 3) và di chuyển sang hàng tiếp theo.

	0	1	2	3
0	x	Q1	x	x
1	x	x	x	Q2
2		x	x	x
3		x		x

4 x 4 Chess Board

Step 9:

+Đặt Queen Q3 vào ô (2, 0) và di chuyển sang hàng tiếp theo.

	0	1	2	3
0	x	Q1	x	x
1	x	x	x	Q2
2	Q3	x	x	x
3	x	x		x

4 x 4 Chess Board

4. Bài toán tìm kiếm theo ràng buộc

Step 10:

+Đặt Queen Q4 vào ô (3, 2) và di chuyển đến hàng tiếp theo.

+Kết luận: đây là giải pháp được chấp nhận.

	0	1	2	3
0	x	Q1	x	x
1	x	x	x	Q2
2	Q3	x	x	x
3	x	x	Q4	x

4 x 4 Chess Board



TRƯỜNG ĐẠI HỌC
VĂN LANG

Đạo đức - Ý chí - Sáng tạo

KHOA CÔNG NGHỆ THÔNG TIN

