Chương 2 : Các Phương Pháp Giải Quyết Vấn Đề Cơ Bản

Chương này gồm có :

- Biểu diễn không gian trạng thái của bài tóan.
- Chiến lược tìm kiếm trên không gian trạng thái.
- Phương pháp tìm kiếm thăm dò.
- Phương pháp tìm kiếm Heuristics.

Biểu Diễn Không Gian Trạng Thái Của Bài

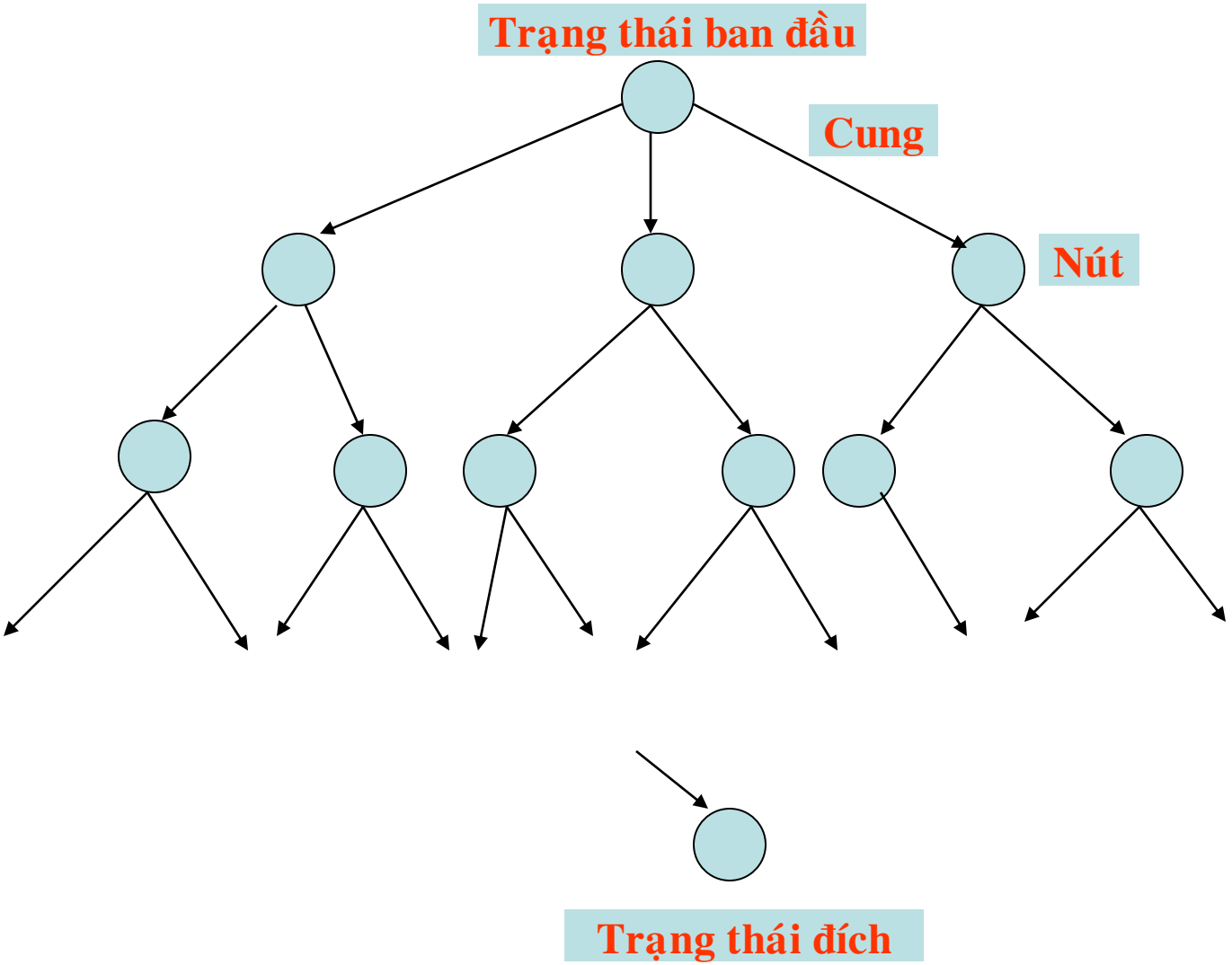
- Tri thức của bài tóan gồm có ba loại tri thức cơ bản đó là tri thức mô tả, tri thức thủ tục và tri thức điều khiển.
- Tri thức thủ tục mô tả tổng quát cách giải bài tóan thường được thể hiện dưới dạng luật

IF <Conditions> Then <Conclusion>

- Một bài tóan có nhiều tình huống cần phải được giải quyết.
- Để giải quyết mỗi tình huống, một luật If-Then hợp lệ cần phải được thiết lập.
- Tập tất cả các luật If-Then hợp lệ mô tả tổng quát cách giải quyết tất cả mọi tình huống của bài tóan đặt ra được gọi là tri thức thủ tục của bài tóan.

- Không gian trạng thái của bài tóan là ẩn chứa tất các trạng thái được phát sinh bởi tất cả các luật với vế trái hợp với trạng thái hiện hành và vế phải phát sinh ra trạng thái mới bằng cách quét thông qua tập các luật mô tả tổng quát cách giải bài tóan bắt đầu từ trạng thái ban đầu của bài tóan và kết thúc tại một hoặc nhiều trạng thái đó là trạng thái đích của bài tóan.

- Về mặt topo hình học, không gian trạng thái của bài tóan có thể được biểu diễn bằng đồ thị định hướng (V, E) như hình vẽ


Trạng thái ban đầu
Cung
Nút
Trạng thái đích

- Trong đó V là tập nút và E là tập cung tương ứng với tri thức mô tả của bài tóan.
- Nút : tương ứng với một trạng thái trong không gian trạng thái của bài tóan.
- Cung : tương ứng với một luật ứng dụng chuyển tiếp từ nút này đến nút khác.
- Luật ứng dụng là luật mà vế trái của nó hợp với trạng thái hiện hành và vế phải tạo ra một trạng thái mới.
- **Nút cha và nút con :** Một nút phát sinh ra một nút khác bằng luật ứng dụng, nút đó được gọi nút cha và nút khác được gọi là nút con.

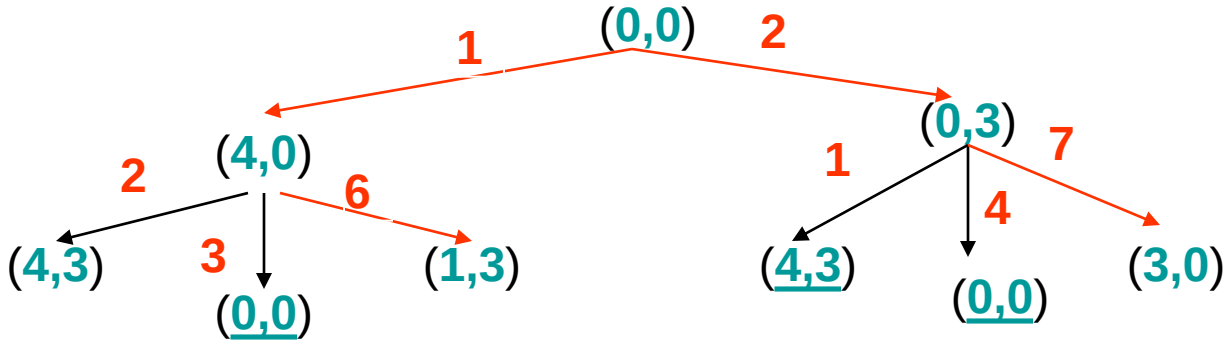
- **Nút tổ tiên và nút con cháu :** Nút cha hoặc cao hơn được gọi là các nút tổ tiên và các nút thừa kế của nút cha được gọi là các nút con cháu.
- **Nút mở rộng :** Nút phát sinh ra tất cả các con của nó bằng các luật ứng dụng, nút đó được gọi là nút mở rộng.
- **Đường lời giải** : là đường chứa một dãy nối tiếp của các trạng thái thông qua đồ thị không gian trạng thái bắt đầu từ trạng thái ban đầu đến trạng thái đích.
- **Không gian trạng thái tìm kiếm** : là quá trìm tìm kiếm lời giải thông qua không gian trạng thái bằng cách làm hiển thị một phần đồ thị của không gian trạng thái ẩn.
- **Giá chi phí của đường lời giải** : là tổng của các giá chi phí của các cung trên đường lời giải.

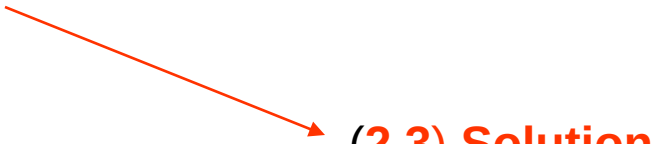
Ví dụ 1: Xét bài tóan bình đựng nước.

- Cho hai bình đựng nước, một bình có dung tích 4 lít và một bình khác có dung tích 3 lít, cả hai bình không có dấu dung tích.
- Trạng thái ban đầu của hai bình là rỗng, dùng một bơm nước làm đầy nước với hai bình. Làm cách nào để có chính xác 2 lít nước trong bình 4 lít ?
- Không gian trạng thái cho bài tóan bình đựng nước định nghĩa như sau :
  - Trạng thái bài tóan : được mô tả bằng tập của các cặp số nguyên (x, y), trong đó
    - x = 0, 1, 2, 3, hoặc 4 biểu diễn số lít nước trong bình 4,
    - y = 0, 1, 2, hoặc 3 biểu diễn số lít nước trong bình 3.

- Trạng thái ban đầu : hai bình rỗng với cặp số nguyên là (0, 0).
- Trạng thái đích : Cần chính xác 2 lít nước trong bình 4 lít (2,n), trong đó n là số không xác định.
- Trạng thái chuyển tiếp (luật ứng dụng) : thông qua tập các luật If-Then hợp lệ mô tả tổng quát cách giải bài tóan bình đựng nước được thiết lập là
  - Luật 1 : (x,y/ x < 4) → (4,y). // Vào x
  - Luật 2 : (x,y/ y < 3) → (x,3). // Vào y
  - Luật 3 : (x,y/ x > 0) → (0,y). // x ra
  - Luật 4 : (x,y/ y > 0) → (x,0). // y ra
  - Luật 5 : (x,y/ x + y >= 4 và y > 0) → (4,y – (4 – x)). // y qua x
  - Luật 6 : (x,y/ x + y >= 3 và x > 0) → (x – (3 –y),3). // x qua y
  - Luật 7 : (x,y/ x + y < 4 và y > 0) → (x + y,0). // y qua x
  - Luật 8 : (x,y/x + y < 3 và x > 0) → (0,x + y) . // x qua y

Không gian trạng thái của bài tóan bình đựng nước được biểu diễn bằng đồ thị như hình





Ví dụ 2 : Xét bài toán trò chơi 8 số.

- Bài toán trò chơi 8 số như một cái mâm hình vuông có ba hàng và ba cột gồm 9 ô, trong đó 8 ô chứa 8 viên ngói có đánh số từ 1 đến 8 và ô còn lại là ô trống với cấu hình như hình .

2	8	3
1	6	4
7		5

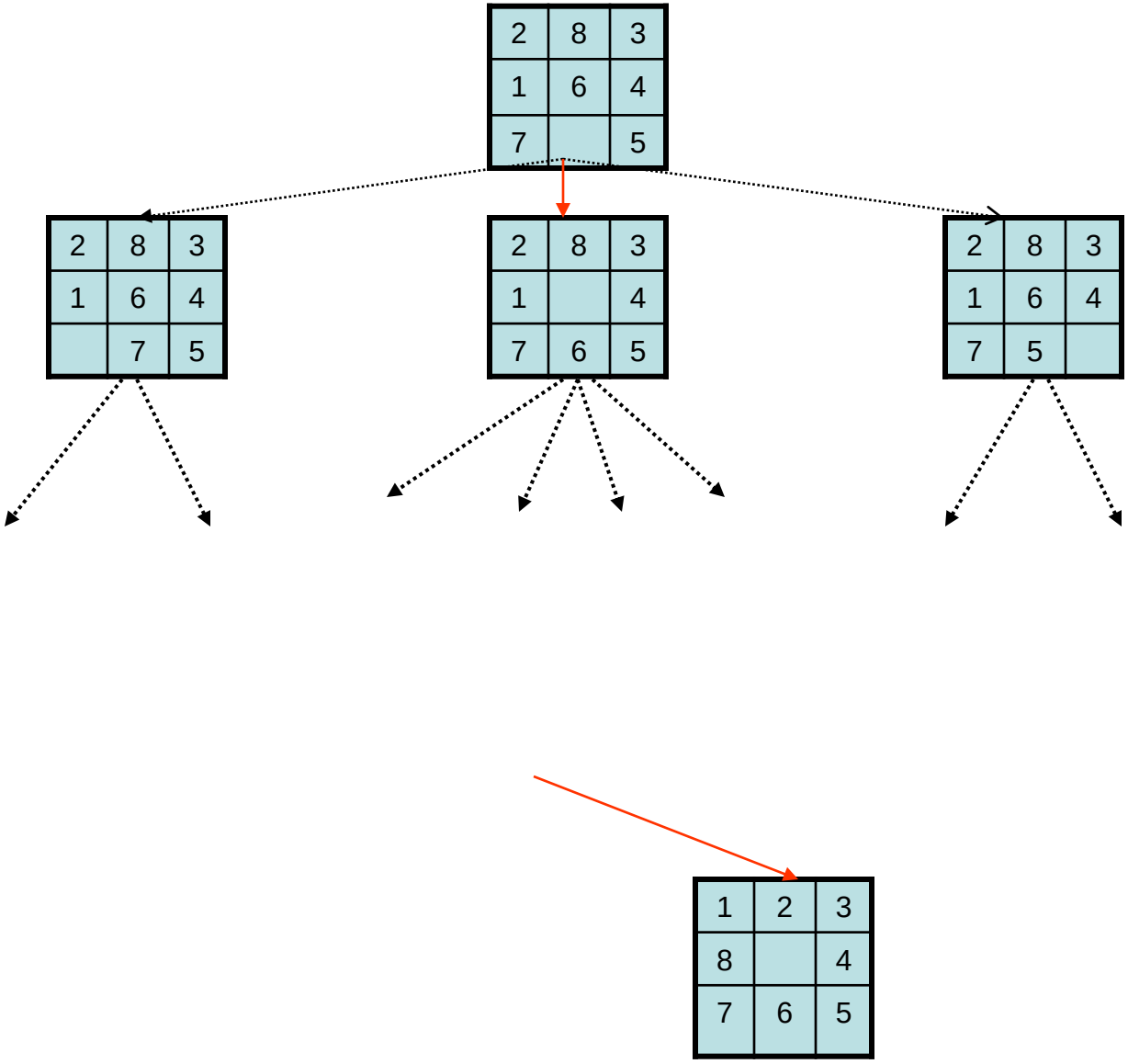
Initial state

1	2	3
8		4
7	6	5

Goal state

- Bài toán đặt ra là trượt các viên ngói đến ô trống kề nó, không được phép trượt theo đường chéo sao cho cấu hình trạng thái ban đầu đạt đến cấu hình trạng thái đích của bài toán.

- Không gian trạng thái của bài tóan này được định nghĩa như sau :
  - Trạng thái bài tóan : cấu hình mảng hai chiều 3×3 chứa các viên ngói mô tả các trạng thái trong không gian bài tóan.
  - Trạng thái ban đầu : cấu hình mảng hai chiều 3×3 chứa các viên ngói có đánh số cho trước.
  - Trạng thái đích : cấu hình mảng hai chiều 3×3 chứa các viên ngói có đánh số cho trước.
  - Trạng thái chuyển tiếp : thông qua các luật hợp lệ như trượt viên ngói đi lên↑, trượt viên ngói đi xuống ↓, trượt viên ngói sang trái ← hoặc trượt viên ngói sang phải →.
- Không gian trạng thái cho bài tóan trò chơi 8 số được biểu diễn bằng đồ thị như hình

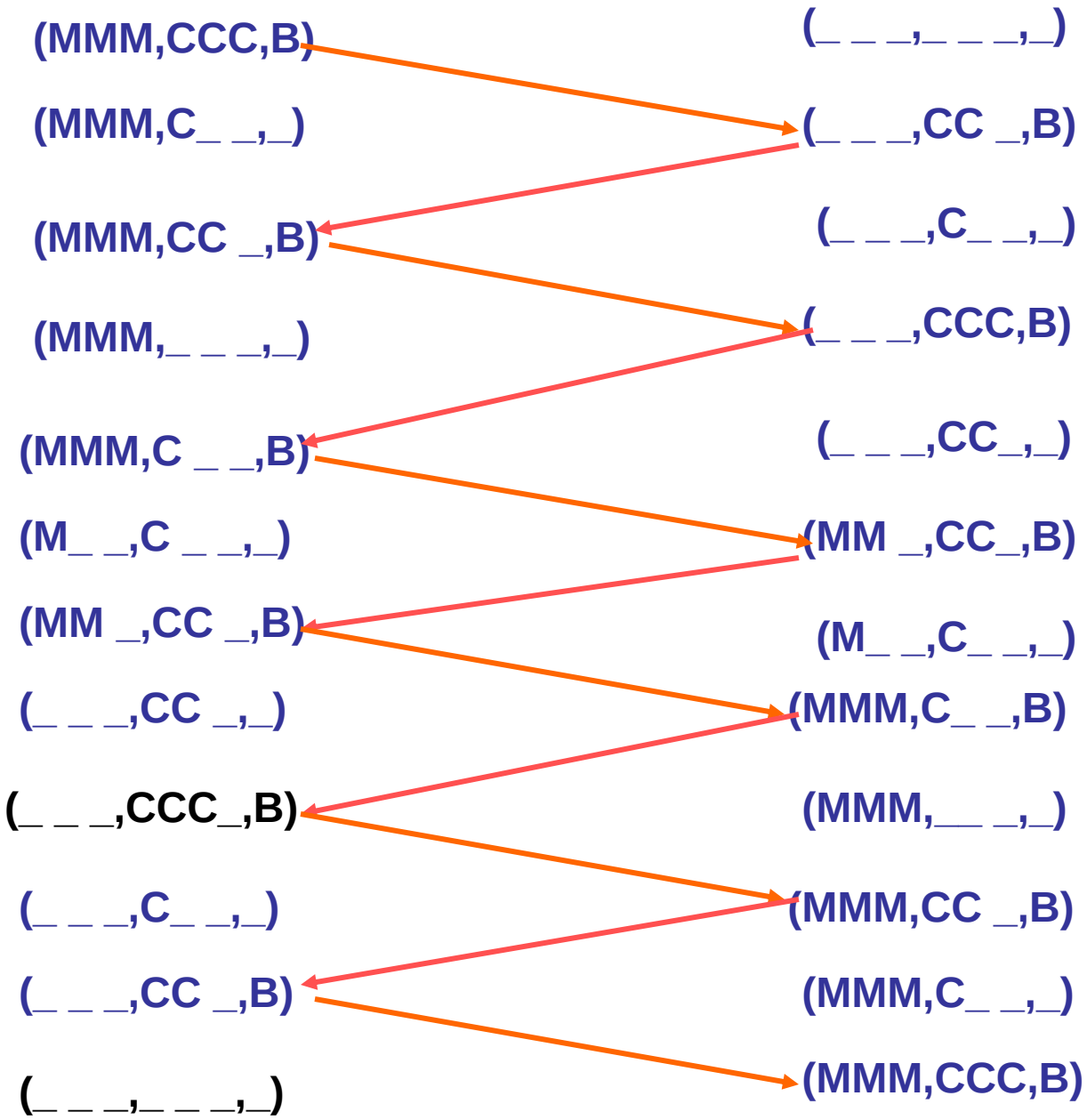

2 8 3
1 6 4
7 5
2 8 3
1 6 4
7 5
2 8 3
1 4
7 6 5
2 8 3
1 6 4
7 5
1 2 3
8 4
7 6 5

Ví dụ 3 : Xét bài tóan ba tu sĩ và ba kẻ ăn thịt người.

- Ba tu sĩ và ba kẻ ăn thịt người ở bên này sông muốn qua bên kia sông bằng một chiếc thuyền có sức chở tối đa là 2 thành viên.
- Bài tóan đặt ra là ở bất kỳ nơi nào bên này sông, trên thuyền hoặc bên kia sông, nếu số tu sĩ ít hơn số kẻ ăn thịt người thì số tu sĩ sẽ bị ăn thịt bởi số kẻ ăn thịt người.
- Hãy sắp xếp các chuyến thuyền qua lại sông sao cho đưa mọi người sang bên kia sông an tòan ?
- Không gian trạng thái của bài tóan này được định nghĩa như sau :

- Trạng thái bài tóan : Cấu hình số tu sĩ, số kẻ ăn thịt người ở bên này sông hoặc bên kia sông.
- Trạng thái ban đầu : Tất cả mọi người và thuyền ở bên này sông với cấu hình là (MMM, CCC, B), trong đó M là tu sĩ, C là kẻ ăn thịt người và B là thuyền.
- Trạng thái đích : Tất cả mọi người và thuyền đều được qua bên kia kia sông an tòan, vì thế cấu hình đích bên này sông là (_, _, _).
- Ràng buộc : Số tu sĩ phải là luôn luôn lớn hơn hoặc bằng số kẻ ăn thịt người ở bất cứ nơi nào bên này sông, trên thuyền hoặc bên kia sông.
- Trạng thái chuyển tiếp : dịch chuyển thuyền đưa một vài thành viên qua lại sông.

- Bài tóan được giải với các chuyến thuyền qua lại sông như hình


(MMM,CCC,B)
(MMM,C_ _,_)
(MMM,CC _,B)
(MMM,_ _ _,_)
(MMM,C _ _,B)
(M_ _,C _ _,_)
(MM _,CC _,B)
(_ _ _,CC _,_)
(_ _ _,CCC_,B)
(_ _ _,C_ _,_)
(_ _ _,CC _,B)
(_ _ _,_ _ _,_)
(_ _ _,_ _ _,_)
(_ _ _,CC _,B)
(_ _ _,C_ _,_)
(_ _ _,CCC,B)
(_ _ _,CC_,_)
(MM _,CC_,B)
(M_ _,C_ _,_)
(MMM,C_ _,B)
(MMM,___ _,_)
(MMM,CC _,B)
(MMM,C_ _,_)
(MMM,CCC,B)

Chiến Lược Tìm Kiếm Trên Không Gian Trạng Thái Của Bài Tóan

- Phương pháp giải bài tóan trong lĩnh vực trí tuệ nhân tạo là kỹ thuật tìm kiếm xây dựng đường lời giải thông qua không gian trạng thái của bài tóan.
- Có hai chiến lược tìm kiếm trên không gian trạng thái của bài tóan đó là chiến lược tìm kiếm suy diễn tiến và chiến lược tìm kiếm suy diễn lùi.

Chiến Lược Tìm Kiếm Suy Diễn Tiến :

- Chiến lược tìm kiếm suy tiến hay còn được gọi là lý giải suy diễn tiến.
- Với chiến lược này, hướng tìm kiếm bắt đầu từ dữ liệu ban đầu của bài tóan thông qua không gian trạng thái thái dẫn về đích của bài tóan.
- Thủ tục tìm kiếm với chiến lược này tiến hành thực hiện các bước như sau :

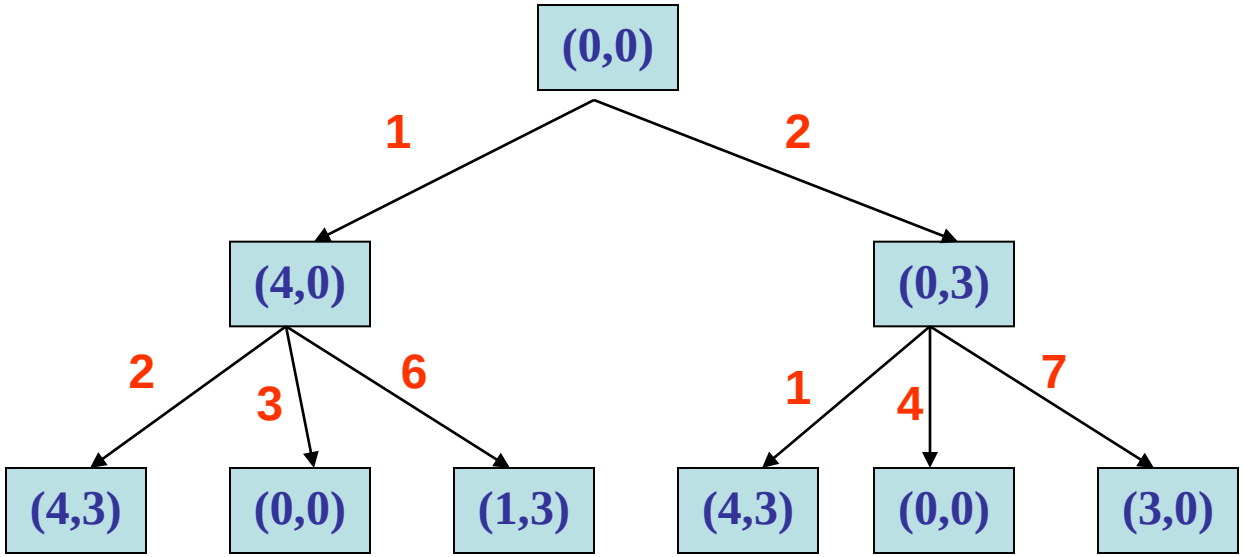
- Bắt đầu xây dựng một hình cây với trạng thái ban đầu của bài tóan là gốc của cây.
- Phát sinh mức kế theo của cây bằng cách ứng dụng tất cả các luật với các vế trái hợp với nút gốc và các vế phải phát sinh ra các trạng thái mới.
- Phát sinh mức kế theo bằng cách lấy mỗi nút đã phát sinh ở mức trước, áp dụng tất cả các luật với các vế trái hợp với nó và các vế phải tạo ra các trạng thái mới.
- Thủ tục này được lặp lại cho đến khi trạng thái đích được tìm thấy.

Ví dụ : Lý giải suy diễn tiến giải bài tóan bình đựng nước, thực hiện các theo sau :

- Xây dựng một hình cây với trạng thái ban đầu (0,0) là gốc của cây.

- Phát sinh mức kế theo sử dụng luật 1 và luật 2 với các vế trái hợp với trạng thái (0,0) và các vế phải phát sinh ra các trạng thái mới (4,0) và (0,3).
- Để phát sinh ra mức kế theo bằng cách lấy mỗi nút đã phát sinh ở mức trước như (4,0) và (0,3) ứng dụng, chọn các luật với các vế trái hợp với các nút này và các vế phải phát sinh ra các nút mới.
  - Tại nút (4, 0), chọn các luật 2, 3, và 6 với các vế trái hợp với trạng thái này và các vế phải phát sinh ra các nút mới như (4, 3), (0, 0) và (1,3).
  - Tại nút (0, 3), chọn các luật 1, 4, và 7 với các vế trái hợp với nút này và các vế phải phát sinh ra các nút mới như (4,3), (0,0) và (3,0).
- Thủ tục này được lặp lại cho các mức mới hơn cho đến khi nào trạng thái đích (2,n) xuất hiện thì dừng.

Lý giải suy diễn tiến giải bài tóan bình đựng nước được minh chứng như hình



(2,n) **Goal Stop**

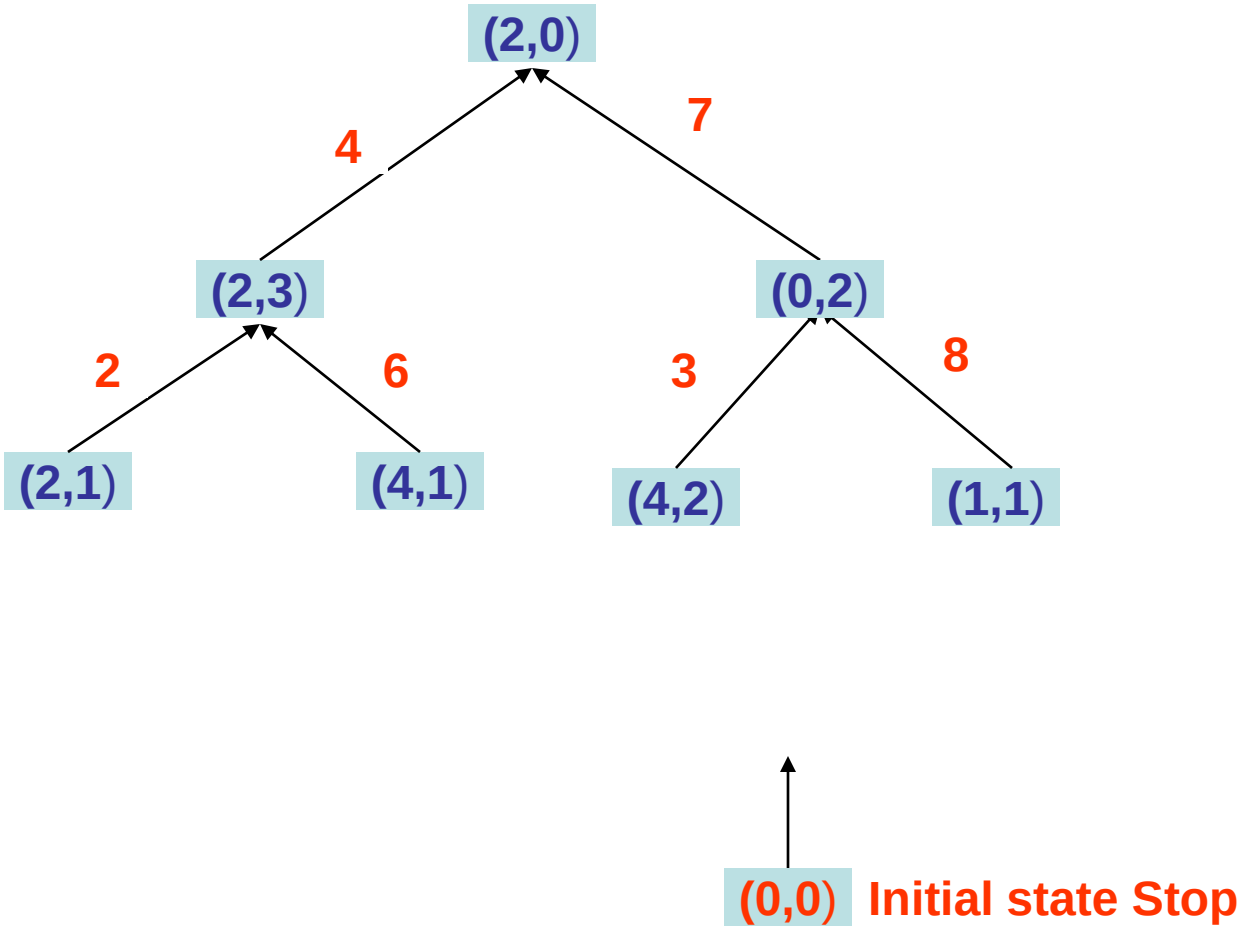
Chiến Lược Tìm Kiếm Suy Diễn Lùi :

- Chiến lược tìm kiếm suy diễn lùi còn được gọi là lý giải suy diễn lùi.
- Thủ tục tìm kiếm bắt đầu từ trạng thái đích của bài tóan tìm kiếm thông qua không gian trạng thái để đạt đến trạng thái ban đầu của bài tóan.
- Thủ tục tiến hành gồm các bước như sau :
  - Bắt đầu xây dựng một hình cây với trạng thái đích là gốc của cây.
  - Phát sinh mức kế theo của cây bằng cách tìm tất cả các luật mà các vế phải của chúng hợp với trạng thái đích và các vế trái phát sinh ra các nút mới.
  - Phát sinh mức kế theo của cây bằng cách lấy mỗi nút đã phát sinh ở mức trước, chọn tất cả các luật với các vế phải hợp với nút và các vế trái phát sinh ra các nút mới.
  - Tiếp tục cho đến khi trạng thái ban đầu được tìm thấy.

Ví dụ : Chiến lược tìm kiếm suy diễn lùi giải bài tóan bình đựng nước tiến hành các bước như sau :

- Xây dựng một hình cây với trạng thái đích (2,0) là gốc của cây.
- Phát sinh mức kế theo chọn các luật 4 và 7 với vế phải hớp với trạng thái đích (2,0) và vế các vế trái phát sinh các trạng thái mới (2,3) và (0,2).
- Phát sinh mức kế theo của cây bằng cách lấy mỗi nút đã phát sinh ở mức trước như (2,3) và (0,2), chọn tất cả các luật mà các vế phải của nó hợp với mỗi nút này và các vế trái phát sinh ra các nút mới.
  - Tại nút (2,3), các luật 2 và 6 với các vế phải hợp với nút này và các vế trái phát sinh ra các nút mới như (2,1) và (4,1).
  - Tại nút (0,2), các luật 3 và 8 với các vế phải hợp vơi nút này và các vế trái phát sinh ra các nút mới như (4,2) và (1,1).
- Tiếp tục cho đến khi trạng thái ban đầu (0,0) được phát sinh.

Chiến lược tìm kiếm suy diễn lùi giải bài tóan bình đựng nước được dẫn chứng bằng cây như hình

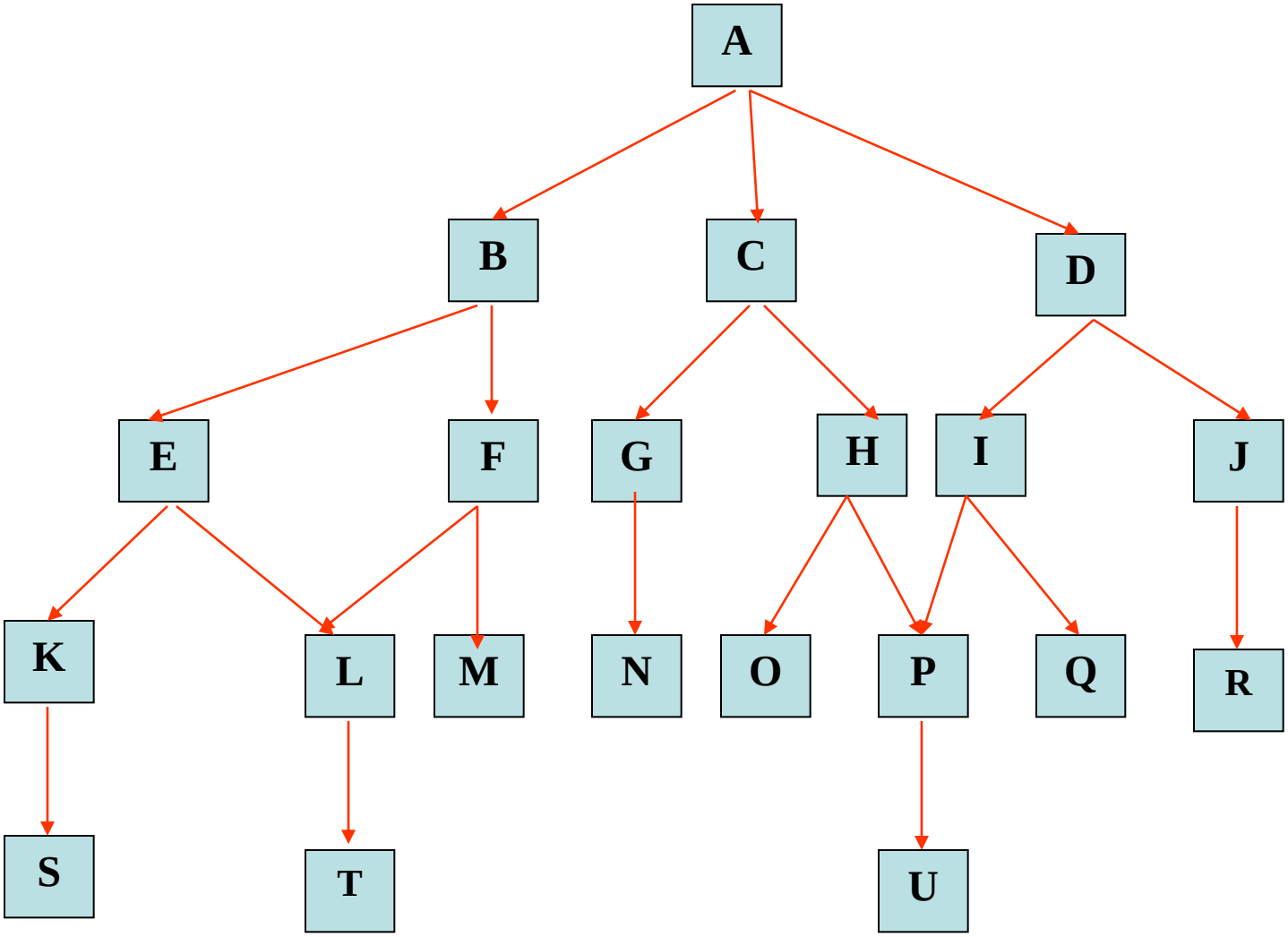


Phương Pháp Tìm Kiếm Thăm Dò

Giải Thuật Tìm Kiếm Theo Chiều Rộng :

- Giải thuật tìm lời giải của bài tóan bằng cách tìm kiếm thăm dò thông qua không gian trạng thái mức theo mức.
- Giải thuật bắt đầu quá trình tìm kiếm tại trạng thái ban đầu, mở rộng trạng thái này để phát sinh ra tất cả các nút con bằng cách ứng dụng tất cả các luật ứng dụng với nút này.
- Giải thuật tìm kiếm tất cả các nút ở mức vừa phát sinh.
- Giải thuật di chuyển đến mức kế
- Thủ tục này được lặp lại cho đến khi nó gặp nút trạng thái đích.

Giải thuật tìm kiếm theo chiều rộng tìm kiếm thăm dò thông qua không gian trạng thái mức theo mức như hình



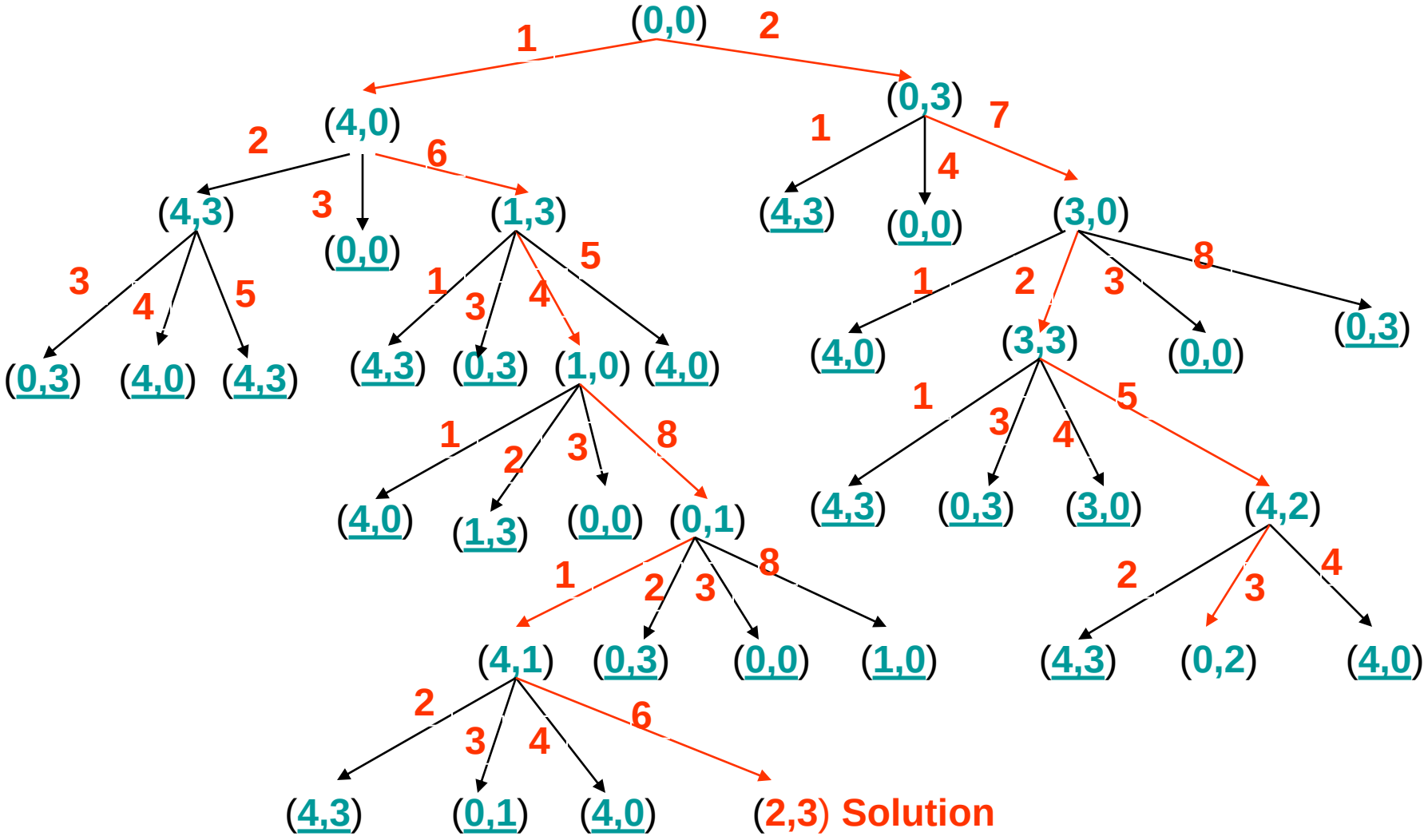
- Giải thuật tìm kiếm thông qua không gian trạng thái mức theo mức với các trạng thái theo thứ tự là A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S,T, U.
- Giải thuật sử dụng hai danh sách OPEN và CLOSED, trong đó
  - OPEN : chứa các nút đã phát sinh nhưng chưa được mở rộng (chưa được duyệt qua).
  - CLOSED : chứa các nút đã được mở rộng (đã được duyệt qua)
- Hai danh sách này giữ đường cho quá trình tìm kiếm thông qua không gian trạng thái của bài tóan.
- Giải thuật được mô tả chi tiết như sau :

```
Procedure breadth-first-search
Begin
    OPEN := [Start];
    CLOSED := [ ];
While OPEN ≠ [ ] do
Begin
    Lọai bỏ trạng thái đầu tiên từ danh sách OPEN, gọi nó là X;
    If X =  đích then return (success)
    Else begin
    Phát sinh tất cả các con của X sử dụng  các luật ứng dụng với X;
    Đặt X vào danh sách CLOSED;
    Lọai bỏ tất cả của X đã có mặt trên OPEN;
    Đặt các con còn lại vào cuối danh sách OPEN;
    End
End;
Return(failure)
End.
```

- Giải thuật tìm kiếm theo chiều rộng tìm lời giải U thăm dò thông qua không gian trạng thái cho trên với các vòng lặp là

1. OPEN = [A]; CLOSED = []
2. OPEN = [B, C, D]; CLOSED = [A]
3. OPEN = [C, D, E, F]; CLOSED = [B, A]
4. OPEN = [D, E, F, G, H]; CLOSED = [C,B, A]
5. OPEN = [E, F, G, H, I, J]; CLOSED = [D, C, B, A]
6. OPEN = [F, G, H, I, J, K, L]; CLOSED = [E, D, C, B, A]
7. OPEN = [G, H, I, J, K, L, M]; CLOSED = [F, E, D, C, B, A]
8. OPEN = [H, I, J, K, L, M, N]; CLOSED = [G, F, E, D, C, B, A]
9. and so on until either U is found or OPEN is empty.

Ví dụ :Giải thuật tìm kiếm theo chiều rộng giải bài tóan bình đựng nước bằng cây được xây dựng như hình



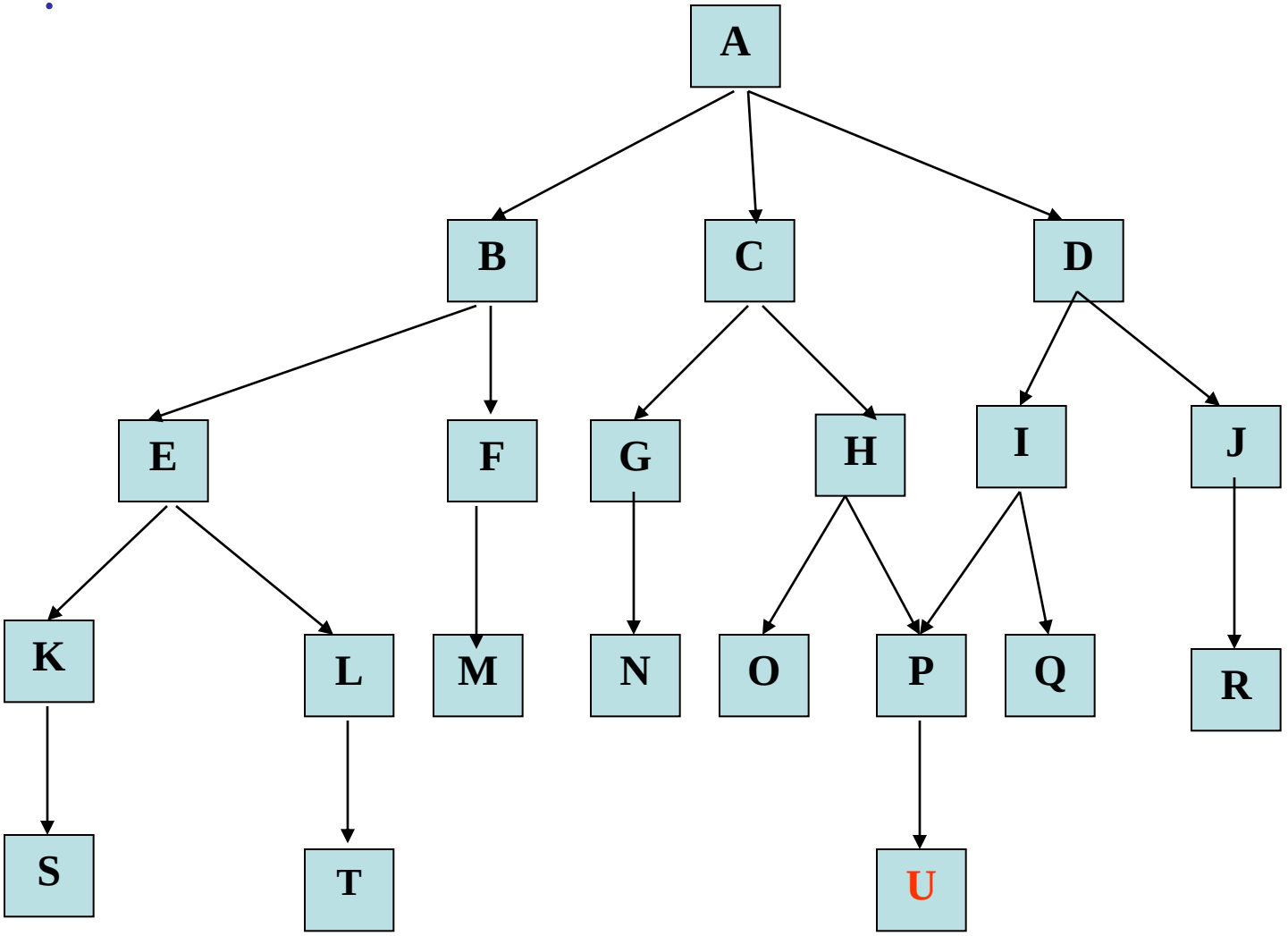
- Giải thuật giải bài tóan bình đựng nước với các vòng lặp cho là

Iteration	X	OPEN	CLOSED
0		[(0,0)]	[ ]
1	(0,0)	[(4,0),(0,3)]	[(0,0)]
2	(4,0)	[(0,3),(4,3),(1,3)]	[(4,0),(0,0)]
3	(0,3)	[(4,3),(1,3),(3,0)]	[(0,3),(4,0),(0,0)]
4	(4,3)	[(1,3),(3,0)]	[(4,3),(0,3),(4,0),(0,0)]
5	(1,3)	[(3,0),(1,0)]	[(1,3),(4,3),(0,3),(4,0),(0,0)]
6	(3,0)	[(1,0),(3,3)]	[(3,0),(1,3),(4,3),(0,3),(4,0),(0,0)]
7	(1,0)	[(3,3),(0,1)]	[(1,0),(3,0),(1,3),(4,3),(0,3),(4,0),(0,0)]
8	(3,3)	[(0,1),(4,2)]	[(3,3),(1,0),(3,0),(1,3),(4,3),(0,3),(4,0),(0,0)]
9	(0,1)	[(4,2),(4,1)]	[(0,1),(3,3),(1,0),(3,0),(1,3),(4,3),(0,3),(4,0),(0,0)]
10	(4,2)	[(4,1),(0,2)]	[(4,2),(0,1),(3,3),(1,0),(3,0),(1,3),(4,3),(0,3),(4,0),(0,0)]
11	(4,1)	[(0,2),(2,3)]	[(4,1),(4,2),(0,1),(3,3),(1,0),(3,0),(1,3),(4,3),(0,3),(4,0),(0,0)]
12	(0,2)	[(2,3),(2,0)]	[(0,2),(4,1),(4,2),(0,1),(3,3),(1,0),(3,0),(1,3),(4,3),(0,3),(4,0),(0,0)]
13	X=(2,3) solution Stop.		

Giải Thuật Tìm Kiếm Theo Chiều Sâu :

- Giải thuật tìm lời giải cho bài tóan bằng cách tìm kiếm thăm dò thông qua không gian trạng thái của bài tóan nhánh theo nhánh của cây.
- Giải thuật bắt đầu duyệt từ trạng thái gốc của cây, rồi thì nó tiếp tục duyệt đến các con, cháu, chắc trước khi nó dịch chuyển qua nhánh anh em khác của cây.
- Giải thuật tìm kiếm đi sâu dần vào không gian trạng thái cho đến khi nó còn có khả năng.
- Chỉ khi nào không còn con cháu, nó dịch chuyển sang nhánh anh em khác của cây.
- Thủ tục được lặp lại cho đến khi nó tìm thấy trạng thái đích.

Giải thuật tìm kiếm theo chiều sâu tìm kiếm thăm dò thông qua không gian trạng thái nhánh theo nhánh được mô tả như hình



- Giải thuật tìm kiếm thăm dò thông qua không gian trạng thái cho trên với các trạng thái theo thứ tự là A, B, E, K, S, L, T, F, M, C, G, N, H, O, P, U, D, I, Q, J, R.
- Giống như giải thuật tìm kiếm theo chiều rộng, giải thuật tìm kiếm theo chiều sâu sử dụng hai danh sách OPEN và CLOSED để giữ đường của quá trình tìm kiếm thông qua không gian trạng thái của bài tóan, trong đó
  - OPEN : liệt kê các trạng thái đã được phát sinh nhưng chưa được duyệt qua,
  - CLOSED : liệt kê các trạng thái đã được duyệt qua (đạ được mở rộng).
- Giải thuật được mô tả chi tiết như sau :

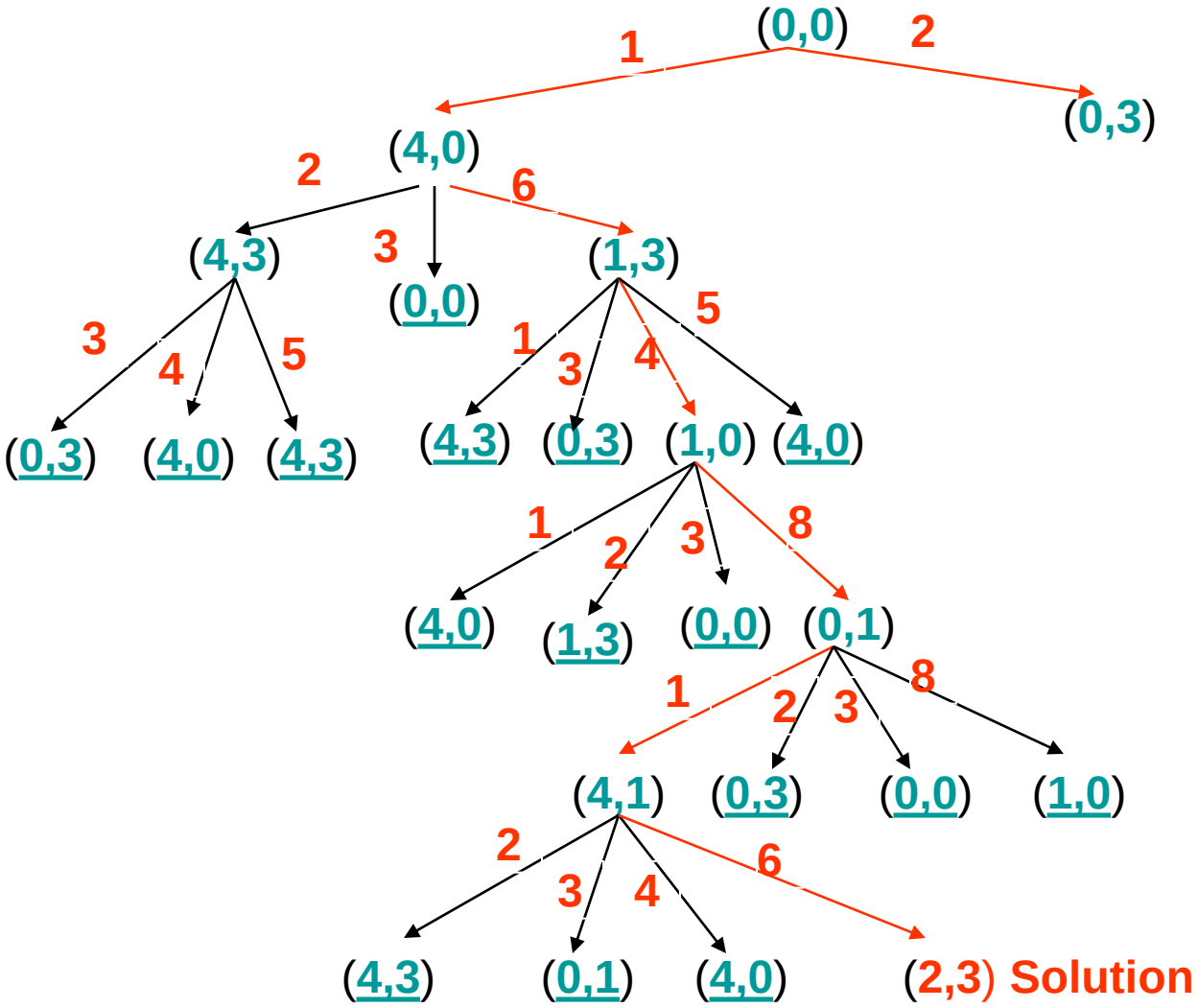
```
Procedure breadth-first-search
    Begin
        OPEN := [Start];
        CLOSED := [ ];
    While OPEN ≠ [ ] do
    Begin
        Lọai bỏ trạng thái đầu tiên từ danh sách OPEN, gọi nó là X;
        If X =  đích then return (success)
        Else begin
        Phát sinh tất cả các con của X sử dụng  các luật ứng dụng với X;
        Đặt X vào danh sách CLOSED;
        Lọai bỏ tất cả của X đã có mặt trên OPEN;
        Đặt các con còn lại vào đầu danh sách OPEN;
        End
    End;
    Return(failure)
    End.
```

- Giải thuật tìm kiếm theo chiều sâu tìm lời giải U thăm dò thông qua không gian trạng thái cho trên với các vòng lặp là

1. OPEN = [A]; CLOSED = [ ]
2. OPEN = [B, C, D]; CLOSED = [A]
3. OPEN = [E, F, C, D]; CLOSED = [B, A]
4. OPEN = [K, L, F, C, D]; CLOSED = [E, B, A]
5. OPEN = [S, L, F, C, D]; CLOSED = [K, E, B, A]
6. OPEN = [L, F, C, D]; CLOSED = [S, K, E, B, A]
7. OPEN = [T, F, C, D]; CLOSED = [L, S, K, E, B, A]
8. OPEN = [F, C, D]; CLOSED = [T, L, S, K, E, B, A]
9. OPEN = [M, C, D]; CLOSED = [F, T, L, S, K, E, B, A]
10. OPEN = [C, D]; CLOSED = [M, F, T, L, S, K, E, B, A]
11. OPEN = [G,H, D]; CLOSED = [C, M, F, T, L, S, K, E, B, A]

- And so on until either U is discovered or OPEN is empty.

Ví dụ : Giải thuật tìm kiếm theo chiều sâu giải bài tóan bình đựng nước được mô tả bằng cây như hình



Giải thuật giải bài tóan bình đựng nước với các vòng lặp được cho là

Iteration	X	OPEN	CLOSED
0		[(0,0)]	[ ]
1	(0,0)	[(4,0),(0,3)]	[(0,0)]
2	(4,0)	[(4,3),(1,3),(0,3)]	[(4,0),(0,0)]
3	(4,3)	[(1,3),(0,3)]	[(4,3),(4,0),(0,0)]
4	(1,3)	[(1,0),(0,3)]	[(1,3),(4,3),(4,0),(0,0)]
5	(1,0)	[(0,1),(0,3)]	[(1,0),(1,3),(4,3),(4,0),(0,0)]
6	(0,1)	[(4,1),(0,3)]	[(0,1),(1,0),(1,3),(4,3),(4,0),(0,0)]
7	(4,1)	[(2,3),(0,3)]	[(4,1),(0,1),(1,0),(1,3),(4,3),(4,0),(0,0)]
8	(2,3) Solution Stop.		

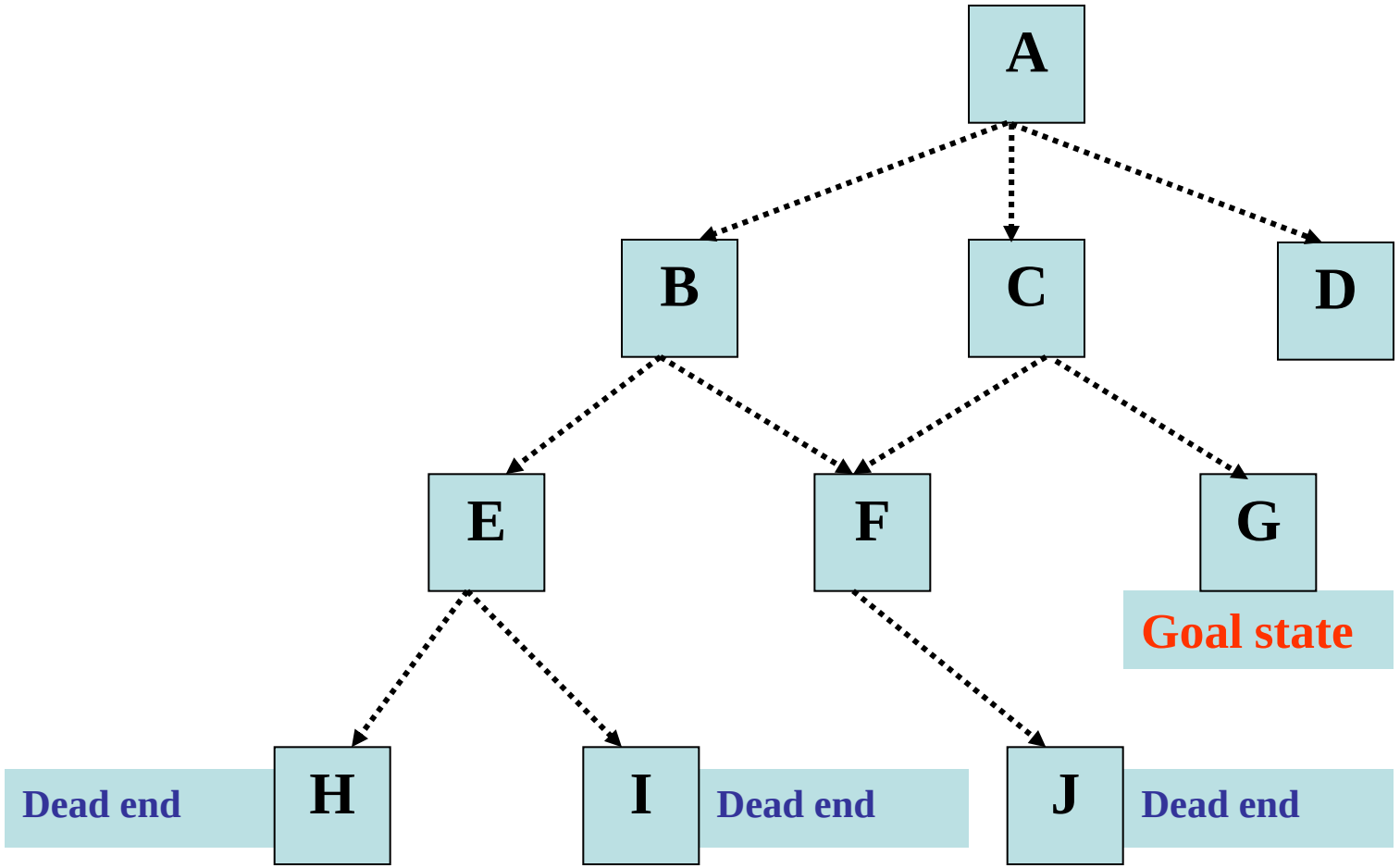
Gỉai Thuật Tìm Kiếm Truyền Lùi :

- Giải thuật tìm kiếm thăm dò lời giải cho bài tóan thông qua không gian trạng thái của bài tóan giống như cách tìm kiếm của giải thuật tìm kiếm theo chiều sâu.
- Giải thuật xây dựng dãy nối tiếp của tất cả các trạng thái trên đường lời giải bắt đầu từ trạng thái ban đầu thông qua không gian trạng thái đến trạng thái đích của bài tóan.
- Giải thuật sử dụng 3 danh sách S, N và D để giữ đường của quá trình tìm kiếm thông qua không gian trạng thái của bài tóan.
  - S : List kê các trạng thái trên đường tìm kiếm hiện hành. Nếu trạng thái đích được tìm thấy, S chứa dãy nối tiếp của tất cả các trạng thái trên đường lời giải bắt đầu từ trạng thái ban đầu thông qua không gian trạng thái đến trạng thái đích của bài tóan.
  - N : Liệt kê các trạng thái đã phát sinh nhưng chưa được mở rộng.
  - D : liệt kê tất cả các trạng thái trên các đường cụt (đường không có trạng thái dẫn đến đích).

- Giải thuật được mô tả chi tiết như sau :

```
Function backtrack;
    Begin
    S = [Start]; N = [Start]; D = [ ]; C = Start;
    While N ≠ [ ]
    Begin
    If C = Trạng thái đích  Then return(S)
    If C  không có con (không kể các nút trên D, S, or N)
    Begin
            While S ≠ [ ] và  C = phần tử đầu tiên của S
            Begin
                    Cộng C vào D ;
                    Loại bỏ phần tử đầu tiên từ S;
                    Loại bỏ phần tử đầu tiên từ N;
                    C := phần tử đầu tiên của N;
            End
    Cộng C vào đầu danh sách S;
    End
    Else begin
    Đặt tất cả các con của C (Ngoại trừ các con đã có mặt trên N, S, D) vào đầu danh sách N;
            C := Phần tử đầu tiên của N;
            Cộng C vào S;
    End
    End;
    Return fail;
    End.
```

Giải thuật tìm lời giải G thông qua không gian trạng của bài tóan với hình cây được xây dựng như hình.



Giải thuật tìm lời giải G thông qua không gian trạng thái của bài tóan với các vòng lặp cho là

Iteration	C	S	N	D
0	A	[A]	[A]	[ ]
1	B	[B,A]	[B,C,D,A]	[ ]
2	E	[E,B,A]	[E,F,B,C,D,A]	[ ]
3	H	[H,E,B,A]	[H,I,E,F,B,C,D,A]	[ ]
4	I	[I,E,B,A]	[I,E,F,B,C,D,A]	[H]
5	F	[F,B,A]	[F,B,C,D,A]	[E,I,H]
6	J	[J,F,B,A]	[F,B,C,D,A]	[E,I,H]
7	C	[C,A]	[C,D,A]	[B,F,J,E,I,H]
8	G	[G,C,A]	[G,C,D,A]	[B,F,J,E,I,H]

Phương Pháp Tìm KiếmHeuristic

Heuristic là gì ?

- Heuristic là kỹ thuật chọn lọc nút tốt nhất để tiếp tục quá trình tìm kiếm trong không gian trạng thái của bài tóan.
- Heuristic có thể được thể hiện dưới dạng luật hoặc dạng hàm số.
  - Nếu nó được thể hiện dưới dạng luật If-Then thì nó được gọi luật Heuristic.
  - Nếu nó được thể hiện dưới dạng hàm số thì nó được gọi là hàm heuristic.

- Cho h(n) là hàm heuristic đó là một ước lượng khỏang cách của nút n đến nút đích, h(n) phải thỏa mãn các tính chất sau :
  - h(n) >= 0 cho mọi nút n
  - h(n) = 0, suy ra rằng nút n là trạng thái đích
  - h(n) = vô cùng, suy ra rằng nút n là trạng thái cụt đó là trạng thái không dẫn đến đích.

Ví dụ :

- Heuristic của bài tóan trò chơi 8 số là số viên ngói đặt không đúng chổ so với trạng thái đích.
- Heuristic của bài tóan hành trình người bán hàng đó là khỏang cách giữa hai thành phố kề nhau.

- Tại mỗi trạng thái hiện hành, phương pháp tìm kiếm heuristic phát sinh ra các trạng thái con, ước lượng heuristic cho tất cả các trạng thái con và chọn lọc một trong những con có giá trị heuristic nhỏ nhất để tiếp tục quá trình tiếp kiếm trong không gian trạng thái của bài tóan.

Example : Xét bài tóan trò chơi 8 số với cấu hình đầu và đích cho như hình

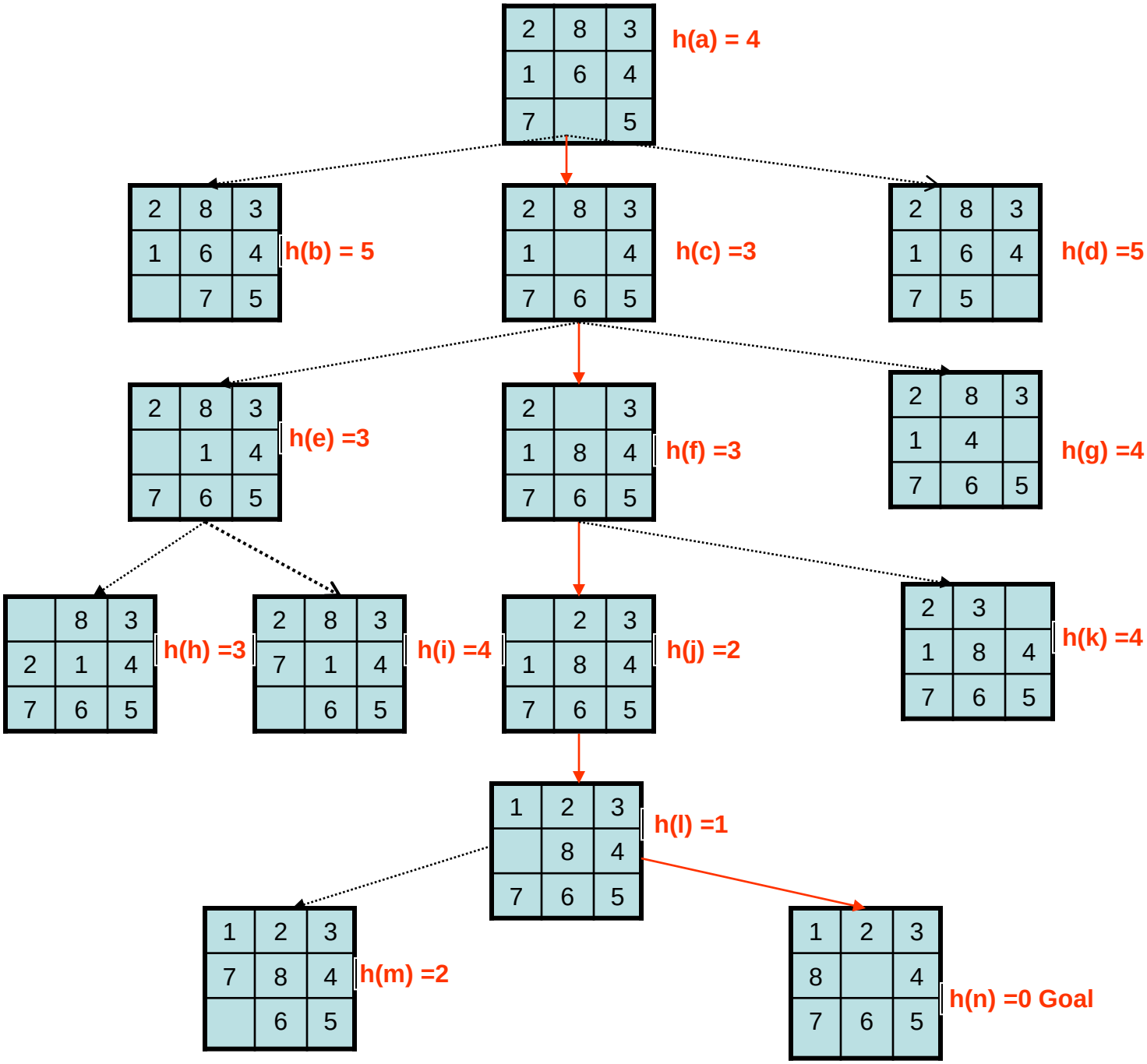
2	8	3
1	6	4
7		5

Initial state

1	2	3
8		4
7	6	5

Goal state

Phương pháp tìm kiếm heuristic giải bài tóan này sử dụng heuristic được mô tả bằng cây tìm kiếm như hình


2 8 3
1 6 4
7 5
h(a) = 4
2 8 3
1 6 4
7 5
h(b) = 5
2 8 3
1 4
7 6 5
h(c) =3
2 8 3
1 6 4
7 5
h(d) =5
2 8 3
1 4
7 6 5
h(e) =3
2 3
1 8 4
7 6 5
h(f) =3
2 8 3
1 4
7 6 5
h(g) =4
8 3
2 1 4
7 6 5
h(h) =3
2 8 3
7 1 4
6 5
h(i) =4
2 3
1 8 4
7 6 5
h(j) =2
2 3
1 8 4
7 6 5
h(k) =4
1 2 3
8 4
7 6 5
h(l) =1
1 2 3
7 8 4
6 5
h(m) =2
1 2 3
8 4
7 6 5
h(n) =0 Goal

Giải Thuật Tìm kiếm Best-First search

- Một trong các phương pháp tìm kiếm heuristic được trạng bị bằng heuristic để tìm lời giải hầu như tốt nhất với bài tóan đó là giải thuật Best-First-Search.
- Giống như giải thuật tìm kiếm theo chiều rộng, giải thuật sử dụng hai danh sách OPEN và CLOSED để giữ đường của quá trình tìm kiếm trong không gian trạng thái của bài tóan.
  - OPEN : Giữ đường tìm kiếm hiện hành (các trạng thái chưa được viếng thăm)
  - CLOSED : Ghi nhận các trạng thái đã được viếng thăm.

- Giải thuật được mô tả chi tiết như sau :

Procedure best-first-search

Begin

OPEN := [Start];

CLOSED = [ ];

While OPEN ≠ [ ]

Begin

-Loại bỏ trạng thái đầu tiên từ OPEN, gọi nó là X;

- If X = đích Then return đường lới giải từ trạng thái ban đầu đến X

Else begin

Phát sinh các con của X áp dụng các luật ứng dụng;

Cho mỗi con của X, thực hiện một trong các trường hợp sau :

Case : Con chưa có mặt trên **OPEN or CLOSED**

Begin

- Ước lượng heuristic cho con;
- Cộng con vào danh sách OPEN;;

End

Case : Con đã có mặt trên OPEN

Begin

-Nếu mới tốt hơn cũ, loại bỏ cũ từ danh sách OPEn, Đặt mới vào danh sách OPEN;

- Mặt khác, giữ cũ và loại bỏ mới;.

End

Case :Con đã có mặt trên CLOSED

Begin

-Nếu mới tốt hơn cũ, loại bỏ cũ từ danh sách CLOSED, đặt mới vào danh sách OPEN;

- Mặt khác giữ cũ và loại bỏ mới;

End

- Đặt X vào danh sách closed;
- Sắp xếp lại các trạng thái rong danh sách OPEN theo thứ từ từ đầu danh sách đến cuối tương ứng với trạng thái có heuristic tốt nhất đến trạng thái có heuristic xấu nhất;

End;

Return failure

End.

- Giải thuật Best-First-Search giải bài tóan trò chơi 8 số với các vòng lặp cho là

Iteration	OPEN	CLOSED
1	[a4]	[ ]
2	[c3,b5,d5]	[a4]
3	[e3,f3,g4,b5,d5]	[c3,a4]
4	[f3,h3,g4,i4,b5, d5]	[e3,c3,a4]
5	[j2,h3,g4,i4,k4,b5, d5]	[f3,e3,c3,a4]
6	[l1,h3,g4,l4,k4,b5, d5]	[j2,f3,e3,c3,a4]
7	[n0,m2,h3,g4,l4,k4,b5, d5]	[l1, j2,f3,e3,c3,a4]
8	success, n = goal	

Hàm Đánh Giá Heuristic :

- Giải thuật tìm kiếm best-first-search sử dụng hàm heuristic h(n) ước lượng khỏang cách từ trạng thái n đến trạng thái đích.
- Bây giờ, giả sử hai hoặc nhiều trạng thái xuất hiện có cùng giá trị heuristic, trạng thái nào gần gốc của cây nhất đó được xem như là trạng thái tốt nhất. Vì khả năng trạng thái này dẫn đến đích là đường đi tốt nhất.
- Với lý do này, một hàm đánh giá heuristic f(n) để đánh giá tại mỗi trạng thái n được đề xuất đó là tổng của hai thành phần được thiết lập là

$$f(n) = g(n) + h(n)$$

Trong đó,

  - g(n) là hàm chi phí đo giá chi phí từ trạng thái ban đầu đến trạng thái n,
  - h(n) là hàm heuristic ước lượng khỏang cách từ trạng thái n đến trạng thái đích.

- Giải thuật tìm kiếm heuristic được trang bị bằng hàm đánh giá heuristic được gọi là giải thuật A.