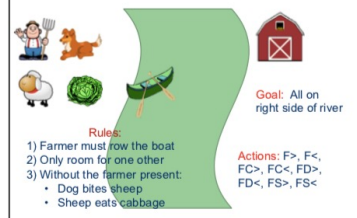


AI – Giải quyết vấn đề bằng tìm kiếm

Search Example: River Crossing Problem



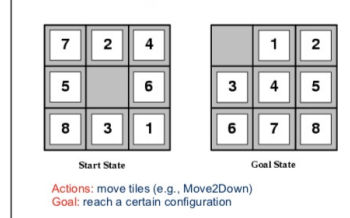
Goal: All on right side of river

Rules:

- 1) Farmer must row the boat
- 2) Only room for one other
- 3) Without the farmer present:
 - Dog bites sheep
 - Sheep eats cabbage

Actions: F>, F<, FC>, FC<, FD>, FD<, FS>, FS<

Search Example: 8-Puzzle



Start State

7	2	4
5		6
8	3	1


Goal State

	1	2
3	4	5
6	7	8

Actions: move tiles (e.g., Move2Down)
Goal: reach a certain configuration


Search Example: Water Jugs Problem

Given 4-liter and 3-liter pitchers, how do you get exactly 2 liters into the 4-liter pitcher?



4 **3**

Search Example: Robot Motion Planning



Actions: translate and rotate joints
Goal: fastest? most energy efficient? safest?

13

Biểu diễn bài toán trên KGTT

- Trò chơi Tic-tac-toe:
- Xác định các thành phần của bài toán:
 - Trạng thái của bài toán?
 - Trạng thái đầu
 - Trạng thái đích
 - Các thao tác



14

14

Ví dụ: Trò chơi Tic-tac-toe

■ Phân tích:

- Các trạng thái: mỗi lượt đi sẽ tạo nên 1 trạng thái mới
- Các hành động: đi X hoặc O
- Mục tiêu: dãy 3 ký hiệu giống nhau (ngang, dọc hoặc chéo)
- Chi phí: 1 (mỗi lượt đi)

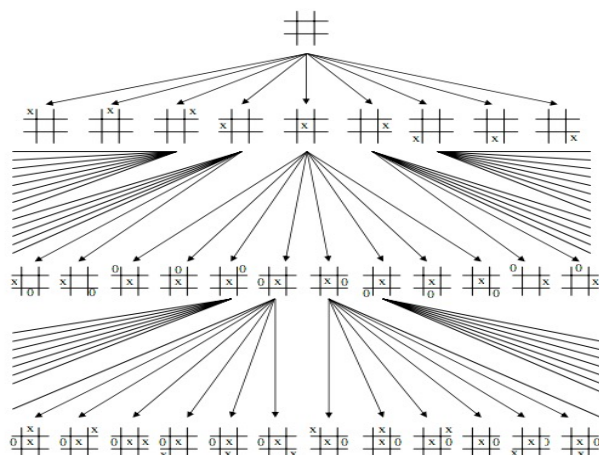


15

15

Biểu diễn bài toán trên KGTT

■ Trò chơi Tic-tac-toe:



16

16

Ví dụ: Trò chơi 8 ô số (8-Puzzle)

■ Cần xác định:

- Các trạng thái?
- Các hành động?
- Mục tiêu?
- Chi phí đường đi?

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

17

17

Ví dụ: Trò chơi 8 ô số (8-Puzzle)

■ Phân tích:

- Các trạng thái: vị trí khác nhau của các ô số
- Các hành động: di chuyển ô trống sang trái phải, lên, xuống
- Mục tiêu: Goal state
- Chi phí đường đi: 1 (mỗi lần di chuyển)

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

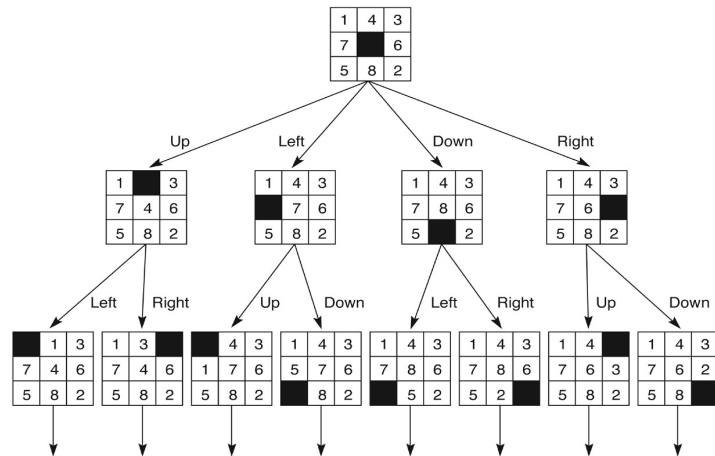
Goal State

18

18

Biểu diễn bài toán trên KGTT

■ Trò chơi 8-Puzzle

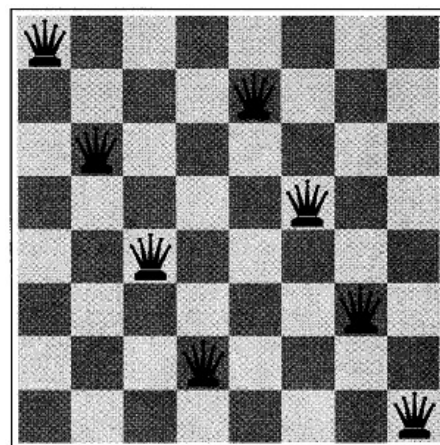


19

19

Ví dụ: Bài toán 8 quân hậu

- Đặt 8 quân hậu lên bàn cờ vua sao cho không có quân hậu nào bị khống chế.
- Xác định trạng thái, phép toán biến đổi trạng thái



20

Ví dụ: Bài toán 8 quân hậu

Phương pháp 1:

- Trạng thái:
 - Bàn cờ và vị trí của các quân hậu (có thể có từ 0 đến 8 quân hậu).
- Phép toán:
 - Đặt thêm một quân hậu vào một vị trí nào đó trên bàn cờ
- Hàm kiểm tra mục tiêu:
 - Bàn cờ đủ 8 quân hậu và không có quân nào bị khổng chế không
- 64^8 đường đi cần phải xem xét

21

Ví dụ: Bài toán 8 quân hậu

Phương pháp 2:

- Trạng thái:
 - Bàn cờ và vị trí của các quân hậu trên bàn cờ (có thể có từ 0 đến 8 quân hậu trên bàn cờ), không có quân nào bị khổng chế.
- Phép toán:
 - Đặt thêm một quân hậu vào một vị trí nào đó trên cột trống đầu tiên của bàn cờ sao cho quân hậu vừa mới đặt không bị khổng chế
- Hàm kiểm tra mục tiêu:
 - Có đủ 8 quân hậu chưa
- Số lượng đường đi sẽ giảm đi rất đáng kể

22

Ví dụ: Bài toán 8 quân hậu

Phương pháp 3

- Trạng thái:
 - Bàn cờ và vị trí của cả 8 quân hậu, mỗi quân trên một cột.
- Phép toán:
 - Di chuyển 1 quân hậu bị khống chế qua một ô khác trong cùng một cột.
- Hàm kiểm tra mục tiêu:
 - Có quân hậu nào còn bị khống chế không

23

Giải quyết bài toán bằng tìm kiếm

- Khi bài toán đã được biểu diễn bằng không gian trạng thái => **Tìm kiếm một đường đi lời giải/ một trạng thái đích mong ước**
- Đường đi lời giải (Solution path)
 - Là đường đi trong đồ thị/cây trạng thái dẫn từ trạng thái bắt đầu đến trạng thái thỏa mãn mục tiêu/trạng thái đích
- **Một trạng thái đích mong ước:** là trạng thái cuối cùng thỏa mãn mục tiêu yêu cầu

24

24

Giải pháp cho các bài toán

- Biểu diễn bài toán trên KGTT để thực hiện tìm kiếm
- Các bước xây dựng KGTT:
 - Bắt đầu từ một trạng thái đã cho (trạng thái ban đầu)
 - Kiểm tra xem có phải là trạng thái đích
 - Mở rộng 1 trong các trạng thái
 - Nếu có nhiều khả năng, chọn 1 khả năng nào đó
 - Quy trình: chọn, kiểm tra và mở rộng đến khi tìm ra giải pháp hoặc không thể tiếp tục mở rộng => quy trình xây dựng cây tìm kiếm
- Tập các trạng thái được mở rộng sẽ có nhiều lựa chọn, việc chọn trạng thái nào để mở rộng được gọi là chiến lược tìm kiếm

25

25

Giải quyết bài toán bằng tìm kiếm

Chiến lược tìm kiếm:

- Từ một trạng thái ban đầu hay trạng thái bất kỳ chưa phải là trạng thái đích, dựa vào các hành động để sinh ra các trạng thái mới (mở rộng không gian trạng thái)
- Để tìm được lời giải, quá trình mở rộng KGTT được thực hiện cho đến khi tìm được trạng thái đích hay không thể mở rộng được KGTT
- Tập các trạng thái được mở rộng sẽ có nhiều lựa chọn, việc chọn trạng thái nào để mở rộng được gọi là chiến lược tìm kiếm

26

26

Các kỹ thuật tìm kiếm

- **Tìm kiếm mù (uninformed/blind search):**
 - *Trạng thái được chọn để phát triển chỉ dựa theo cấu trúc của KGTT mà không dùng thêm thông tin hỗ trợ.*
- **Tìm kiếm dựa trên kinh nghiệm (informed/ heuristic search):**
 - *Dựa vào kinh nghiệm và sự hiểu biết để xây dựng hàm đánh giá hướng dẫn tìm kiếm*

27

27

Đánh giá phương pháp tìm kiếm

- *Phương pháp tìm kiếm xác định thứ tự triển khai của các đỉnh trong cây tìm kiếm*
- *Các giải thuật tìm kiếm thường được đánh giá dựa trên 4 tiêu chí sau:*
 - **Tính trọn vẹn:** Liệu nó luôn tìm ra nghiệm không nếu bài toán tồn tại nghiệm.
 - **Độ phức tạp thời gian:** giải thuật có mất nhiều thời gian không?
 - **Độ phức tạp không gian:** yêu cầu bộ nhớ lưu trữ?
 - **Tính tối ưu:** Giải thuật có đảm bảo tìm ra nghiệm với hàm chi phí ít nhất không?

28

28

TÌM KIẾM MÙ (UNINFORMED BLIND SEARCH)



29

Tìm kiếm mù

- Tìm kiếm rộng (breath-first search)
- Tìm kiếm sâu (depth-first search)
- Tìm kiếm theo độ sâu có giới hạn (depth-limited search)
- Tìm kiếm theo chiều sâu lặp (iterative deepening depth-first search)
- Tìm kiếm giá thành đồng nhất

30

30

Tìm kiếm rộng (breath-first search - BFS)

```

Procedure breadth-first-search;
  Begin                               % khởi đầu
    Open:= [start];                   Closed:= [ ];
    While open ≠ [ ] do               % còn các trạng thái chưa khảo sát
      Begin
        Loại bỏ trạng thái ngoài cùng bên trái khỏi open, gọi nó là X;
        If X là một đích then trả lời kết quả (thành công) % tìm thấy đích
      else begin
        Phát sinh các con của X;
        Đưa X vào closed;
        Loại các con của X đã tồn tại trong open hoặc closed; % kiểm tra vòng lặp
        Đưa các con còn lại của X vào đầu bên phải của open; % hàng đợi
      end;
    End;
    Trả lời kết quả (thất bại);       % không còn trạng thái nào
  End;

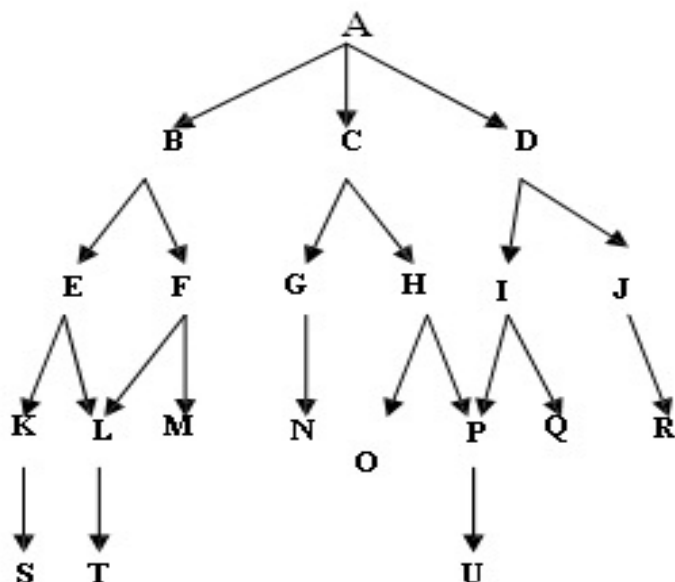
```

31

31

Tìm kiếm mù

■ Xét KGTT sau:



32

32

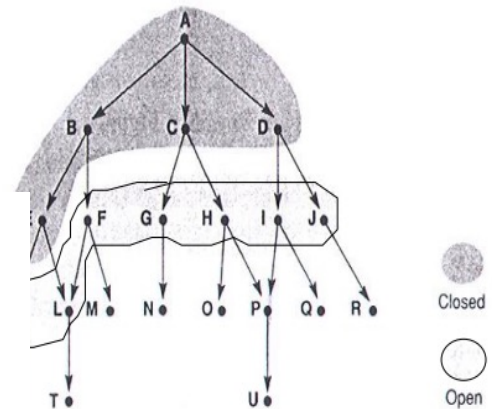
Tìm kiếm rộng (breath-first search - BFS)

- Tìm kiếm rộng:
- Các bước thực hiện tìm kiếm rộng:
 1. $Open = [A]; closed = []$
 2. $Open = [B, C, D]; closed = [A]$
 - 3.

Procedure breadth-first-search;

```

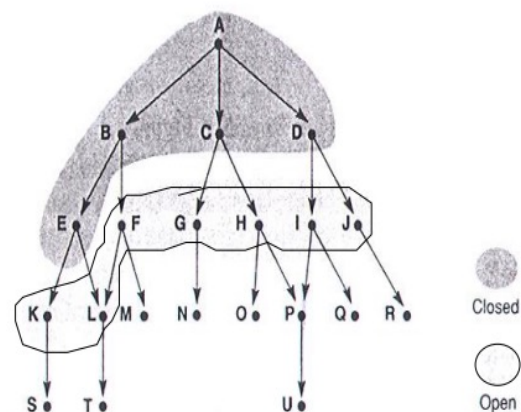
Begin           % khởi đầu
  Open := [start];           Closed := [];
  While open ≠ [] do        % còn các trạng thái chưa khảo sát
    Begin
      Loại bỏ trạng thái ngoài cùng bên trái khỏi open, gọi nó là X;
      If X là một đích then trả lời kết quả (thành công) % tìm thấy đích
    else begin
      Phát sinh các con của X;
      Đưa X vào closed;
      Loại các con của X đã tồn tại trong open hoặc closed; % kiểm tra vòng lặp
      Đưa các con còn lại của X vào đầu bên phải của open; % hàng đợi
    end;
  End;
  Trả lời kết quả (thất bại); % không còn trạng thái nào
End;
```



33

Tìm kiếm rộng (breath-first search - BFS)

- Tìm kiếm rộng:
- Các bước thực hiện tìm kiếm rộng:
 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. $Open = [G, H, I, J, K, L, M];$
 (vì L đã có trong open);
 $closed = [F, E, D, C, B, A]$
 8. ...

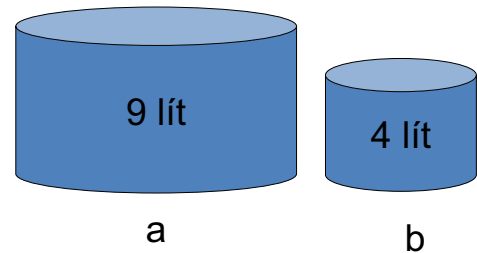


34

34

Ví dụ: Bài toán đong nước

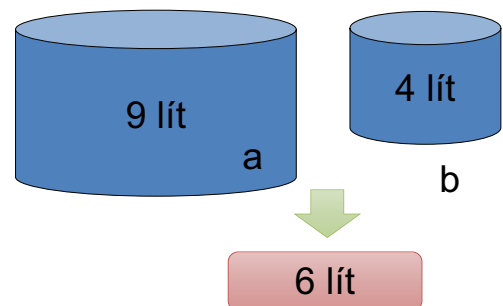
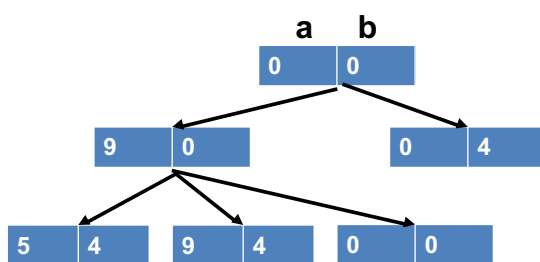
	a	b
Start	0	0
1		



35

35

Ví dụ: Bài toán đong nước



- Tìm kiếm rộng:
- Các bước thực hiện tìm kiếm rộng:
 1. $Open = [A: 0-0]; closed = []$
 2. $Open = [B: 9-0, C: 0-4]; closed = [A: 0-0]$
 3. $Open = [C: 0-4, D: 5-4, E: 9-4, F: 0-0]; closed = [B, A]$
 4.

36

36