

AI

GIẢI QUYẾT CÁC VẤN ĐỀ BẰNG CÁCH TÌM KIẾM

Nguyễn Ngọc Thảo - Nguyễn Hải Minh
{nnthao, nhminh}@fit.hcmus.edu.vn

Đề cương

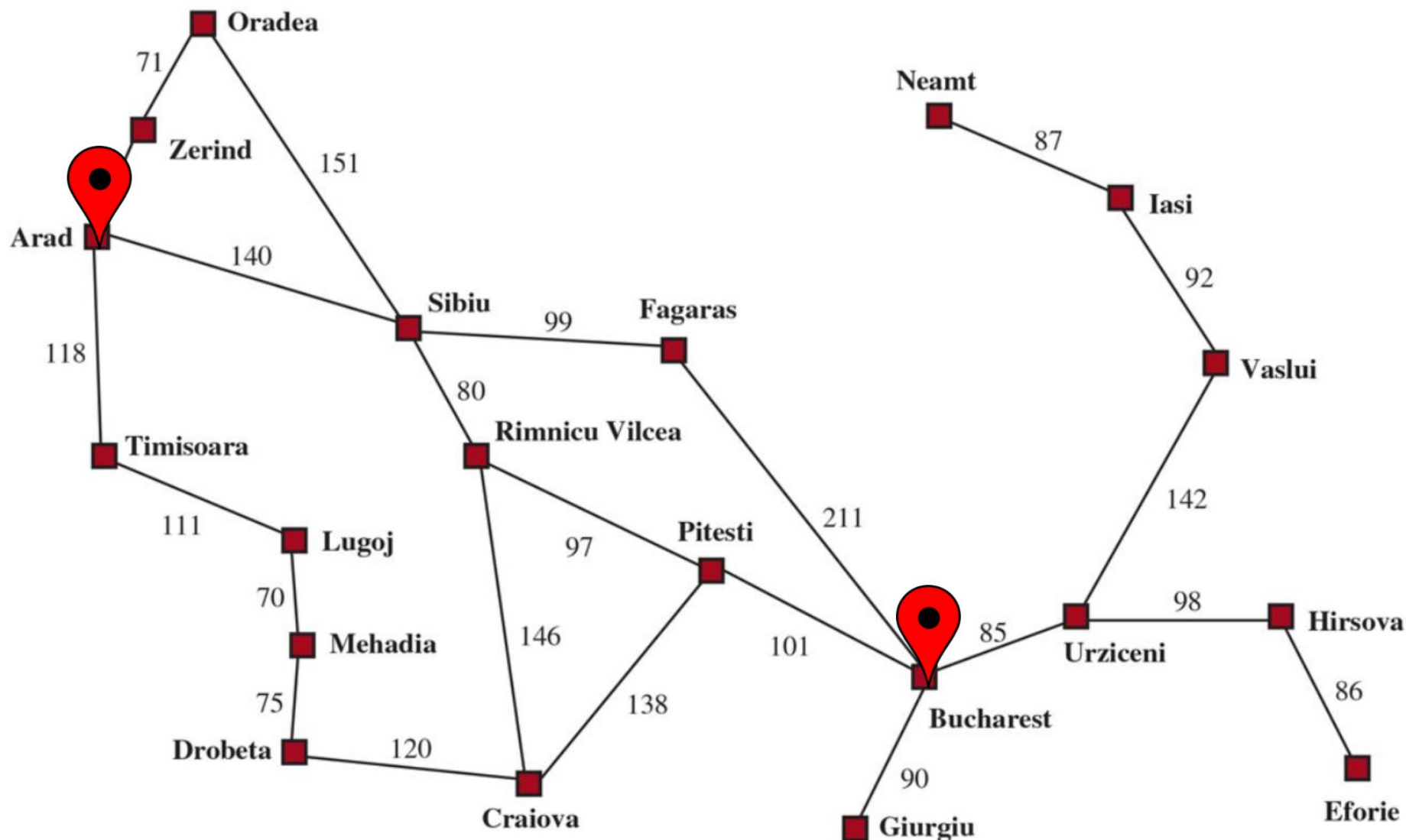
- Tác nhân giải quyết vấn đề
- Các bài toán ví dụ
- Tìm kiếm giải pháp

Tác nhân giải quyết vấn đề



- Các vấn đề và giải pháp được xác định rõ ràng
- Xây dựng bài toán

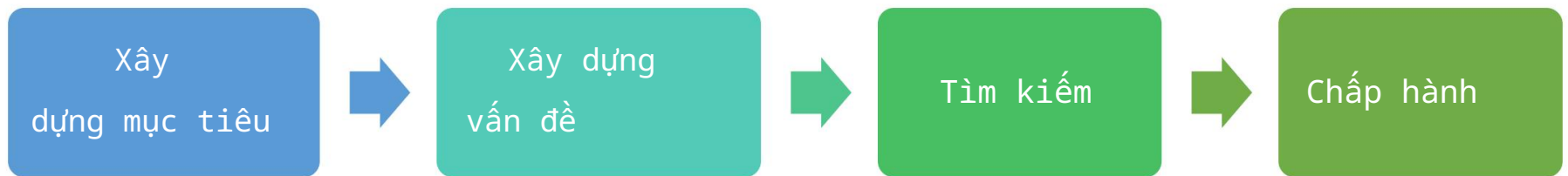
Một kỳ nghỉ du lịch ở Romania



Bản đồ đường đi đơn giản hóa một phần của Romania, với khoảng cách đường bộ tính bằng dặm.

Quá trình giải quyết vấn đề

- Giả sử tác nhân luôn có quyền truy cập thông tin về thế giới bằng cách sử dụng bản đồ.
- Sau đó, thực hiện theo quy trình giải quyết vấn đề gồm bốn giai đoạn



Quá trình giải quyết vấn đề

- **Xây dựng mục tiêu:** Mục tiêu tổ chức hành vi bằng cách hạn chế mục tiêu và do đó các hành động cần được xem xét.
 - Ví dụ, trong ví dụ về chuyến du lịch trong kỳ nghỉ, mục tiêu là đến Bucharest.



- **Hình thành vấn đề:** Tác nhân nghĩ ra một mô tả về trạng thái và hành động cần thiết để đạt được mục tiêu.
 - Ví dụ, trong ví dụ trên, một mô hình tốt là xem xét hành động di chuyển từ thành phố này sang thành phố lân cận.

Quá trình giải quyết vấn đề

- **Tìm kiếm:** Tác nhân mô phỏng chuỗi hành động trong mô hình của nó, tìm kiếm cho đến khi tìm thấy giải pháp.
 - Việc này được thực hiện trước khi thực hiện bất kỳ hành động nào trong thế giới thực.
 - Giải pháp là một chuỗi các hành động nhằm đạt được mục tiêu, ví dụ như đi từ Arad đến Sibiu đến Faragas đến Bucharest.
- **Thực thi:** Bây giờ tác nhân có thể thực hiện các hành động trong giải pháp, từng cái một.

Vòng lặp mở và vòng kín

Môi trường hoàn toàn có thể quan sát được, xác định, được biết đến

- Giải pháp cho mọi vấn đề là một chuỗi hành động cố định.
- Hệ thống vòng lặp mở: bỏ qua các nhận thức sẽ phá vỡ vòng lặp giữa đại lý và môi trường

Môi trường có thể quan sát được một phần hoặc không xác định

- Giải pháp sẽ là chiến lược phân nhánh đề xuất các hành động khác nhau trong tương lai tùy thuộc vào nhận thức nào sẽ đến.
- Hệ thống khép kín: giám sát nhận thức

Tìm kiếm vấn đề và giải pháp

- Một bài toán tìm kiếm có thể được định nghĩa một cách hình thức như sau.
- Một tập hợp các trạng thái có thể xảy ra của môi trường, gọi là không gian trạng thái
 - Ví dụ: các thành phố, {Arad, Zerind,...}, trong ví dụ về kỳ nghỉ du lịch
- Trạng thái ban đầu mà tác nhân bắt đầu
 - Ví dụ, trong ví dụ trên, đại lý bắt đầu tại Arad.
- Tập hợp một hoặc nhiều trạng thái mục tiêu
 - Ví dụ, trong ví dụ trên, đại lý muốn đến Bucharest.
 - Có thể có một hoặc nhiều trạng thái mục tiêu, hoặc mục tiêu được xác định bởi một thuộc tính áp dụng cho nhiều trạng thái (có thể là vô hạn).

Tìm kiếm vấn đề và giải pháp

- Một **tập hợp hữu hạn các hành động** có thể thực thi được ở trạng thái , ()

• Ví dụ, () = { , , }

- Một **mô hình chuyển tiếp**, (,), trả về trạng thái kết quả thực hiện hành động ở trạng thái .

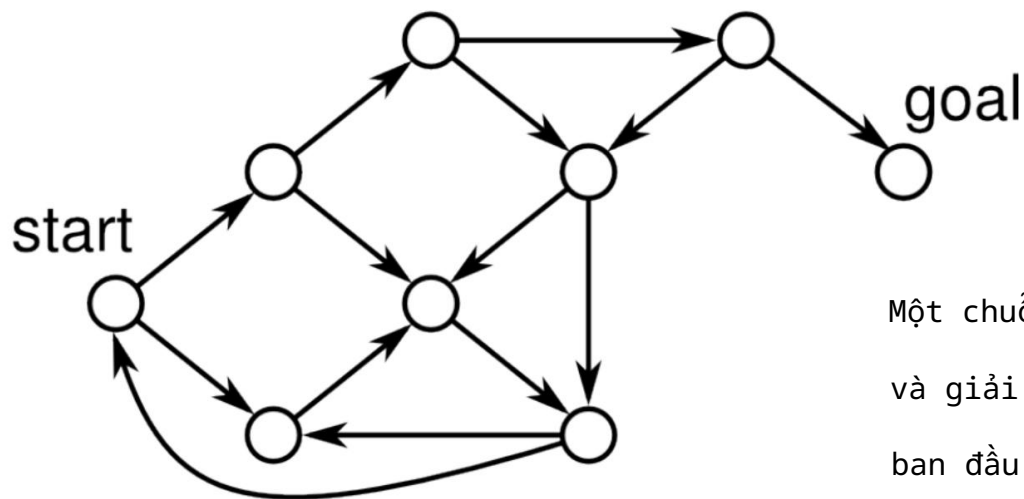
• Ví dụ, (,) =

Tìm kiếm vấn đề và giải pháp

- Hàm **chi phí hành động**, $f(s, a, s')$, đưa ra chi phí bằng số của việc áp dụng hành động ở trạng thái để đạt đến trạng thái s' .
 - Hàm chi phí phải phản ánh thước đo hiệu suất của tác nhân.
 - Ví dụ, đối với các đại lý tìm đường, chi phí có thể là chiều dài tính bằng dặm hoặc thời gian để hoàn thành hành động.
- Chi phí hành động được giả định là **dương và cộng dồn**.
- Tổng chi phí của một đường đi là tổng chi phí của từng hành động.
- **Giải pháp tối ưu** có **chi phí thấp nhất** trong số các giải pháp.

Không gian trạng thái

- Không gian trạng thái có thể được biểu diễn dưới dạng đồ thị.
- Các đỉnh là các trạng thái và các cạnh có hướng giữa chúng là những hành động.



Một chuỗi hành động tạo thành một đường dẫn và giải pháp là một đường dẫn từ trạng thái ban đầu đến trạng thái mục tiêu.

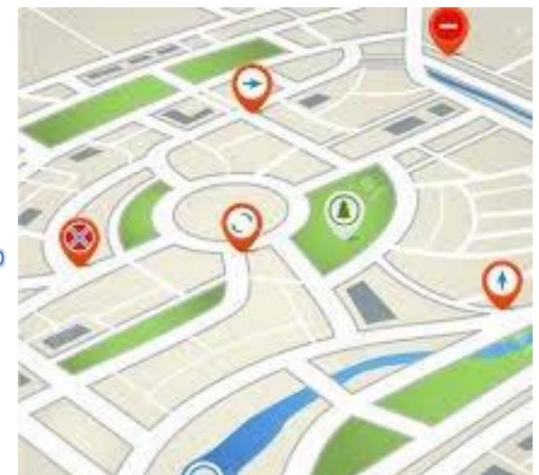
Xây dựng vấn đề

- Việc trình bày vấn đề là một ~~mô hình~~—một sự trừu tượng mô tả toán học—và không phải là sự thật.

Bạn giải quyết vấn đề
điều hướng như thế nào?



VS.



So sánh mô tả trạng thái nguyên tử đơn giản với một chuyến đi xuyên quốc gia thực tế, nơi trạng thái của thế giới bao gồm rất nhiều thứ: những người bạn đồng hành, tình trạng đường sá, thời tiết, giao thông, v.v.

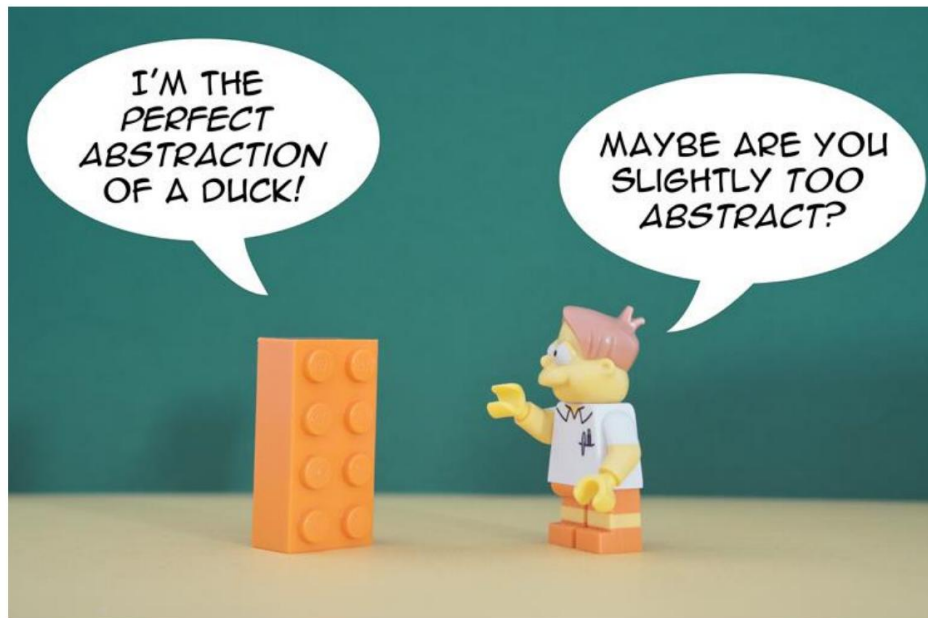
Mô hình chỉ xử lý sự thay đổi về vị trí trong khi bỏ qua tất cả các cân nhắc khác.

Xây dựng vấn đề

- Tính trừu tượng liên quan đến việc loại bỏ chi tiết khỏi biểu diễn.
- Nó tạo ra một mô hình gần đúng và đơn giản hóa về thế giới, điều này rất quan trọng để giải quyết vấn đề tự động.
- Một sự trừu tượng hóa hợp lệ cho phép bất kỳ giải pháp trừu tượng nào được biến thành một giải pháp trong thế giới chi tiết hơn.
- Sự trừu tượng hóa rất hữu ích nếu thực hiện từng hành động trong giải pháp dễ dàng hơn vấn đề ban đầu.

Xây dựng vấn đề

- Sự trừu tượng hóa tốt sẽ loại bỏ được càng nhiều chi tiết càng tốt.
- Trong khi đó, nó vẫn giữ được giá trị và đảm bảo rằng các hành động trừu tượng được thực hiện dễ dàng.



Vấn đề ví dụ

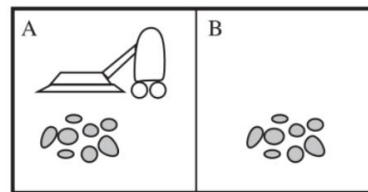
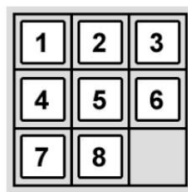
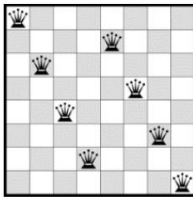
- Các bài toán tiêu chuẩn hóa
- Các vấn đề trong thế giới thực



Vấn đề ví dụ

Vấn đề tiêu chuẩn hóa

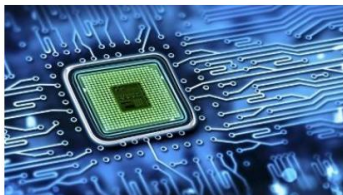
- Minh họa hoặc thực hiện các phương pháp giải quyết vấn đề khác nhau.
- Mô tả ngắn gọn và chính xác. • Thích hợp làm chuẩn mực để so sánh hiệu suất của các thuật toán.



$$\begin{array}{r} \text{TWO} \\ + \text{TWO} \\ \hline \text{FOUR} \end{array}$$

Các vấn đề trong thế giới thực

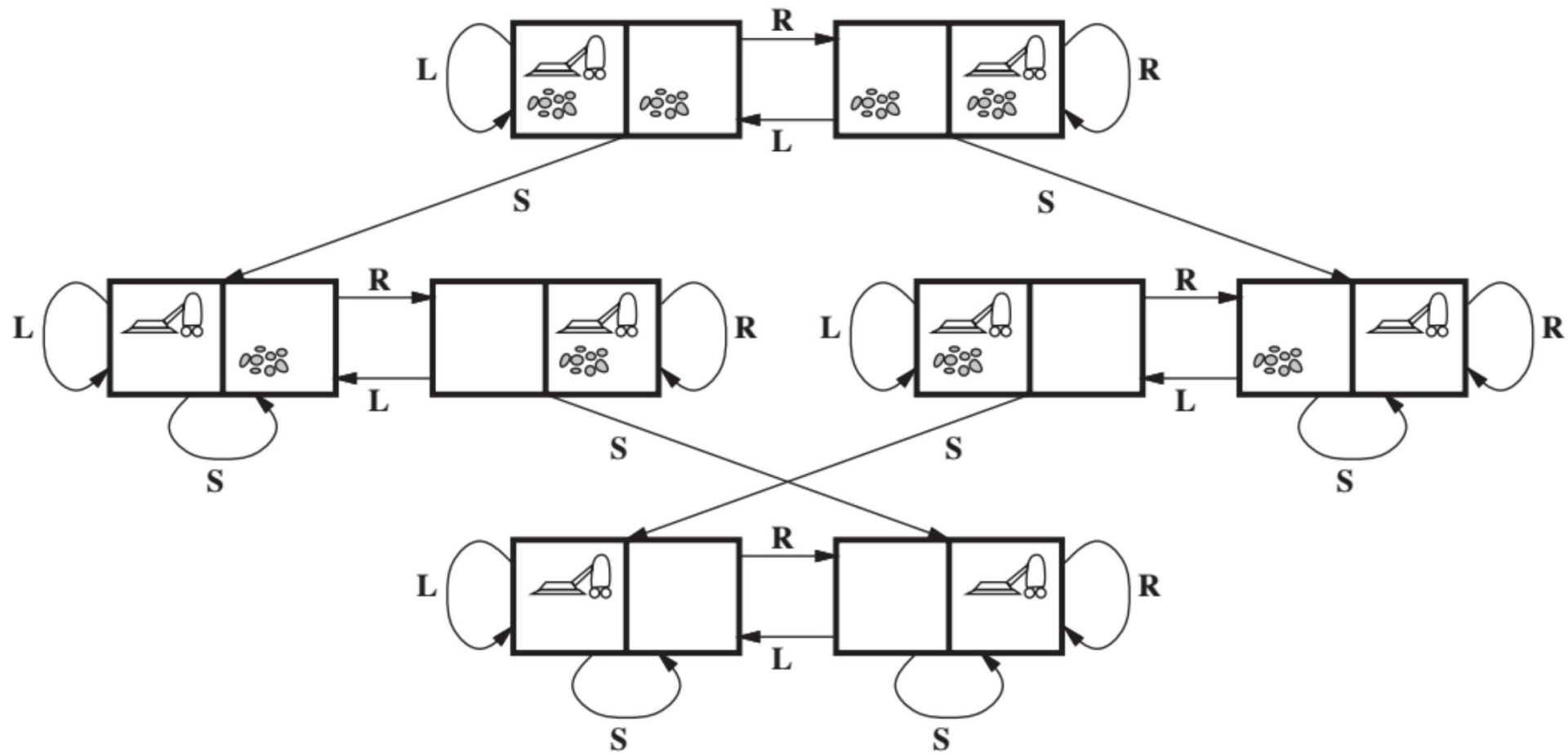
- Mang lại giải pháp cho các vấn đề thực tiễn
- Công thức có đặc tính riêng và không được tiêu chuẩn hóa.



Thế giới máy hút bụi

- **Kỳ:** được xác định bởi vị trí đại lý và tình trạng sẵn có của chất bẩn
 $2 \cdot 2 \times 2 = 8$ trạng thái thế giới có thể có ($\times 2$ đối với môi trường có ô)
- **Trạng thái ban đầu:** Bất kỳ trạng thái nào cũng có thể được chỉ định làm trạng thái ban đầu.
- **Hành động:** Trái, Phải và Hút
 - Các mô hình lớn hơn có thể bao gồm Hư ớng lên và Hư ớng xuống.
- **Mô hình chuyển tiếp:** Các hành động có tác dụng như mong đợi.
 - Lưu ý di chuyển Trái ở ô vuông ngoài cùng bên trái, di chuyển Phải ở ô vuông ngoài cùng bên phải và Hút trong ô vuông sạch sẽ không có tác dụng.
- **Trạng thái mục tiêu:** Trạng thái trong đó mọi ô đều sạch.
- **Chi phí hành động:** Mỗi bước tốn 1.

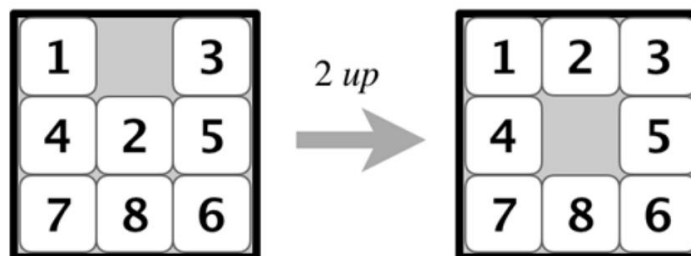
Thế giới máy hút bụi



Không gian trạng thái cho thế giới chân không
Các liên kết biểu thị các hành động: L = Left, R = Right, S = Suck.

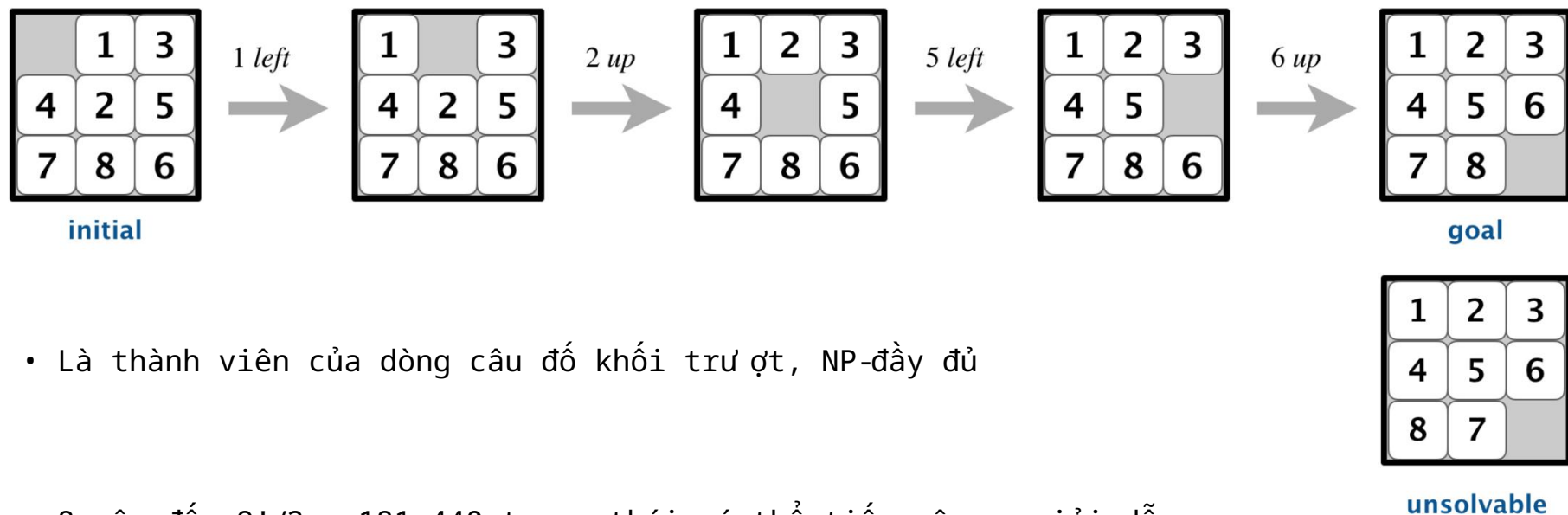
Bài toán 8 câu đố

- **Trạng thái:** được chỉ định bởi vị trí của mỗi ô trong số tám ô.
- **Trạng thái ban đầu:** Bất kỳ trạng thái nào cũng có thể được chỉ định làm trạng thái ban đầu.
- **Hành động:** Chuyển động của khoảng trống—Trái, Phải, Lên, Xuống
- **Mô hình chuyển tiếp:** Ánh xạ trạng thái và hành động tới trạng thái kết quả.



- **Trạng thái mục tiêu:** Bất kỳ trạng thái nào cũng có thể là mục tiêu
- **Chi phí hành động:** Mỗi bước tốn 1.

Bài toán 8 câu đố



- Là thành viên của dòng câu đố khối trượt, NP-đầy đủ

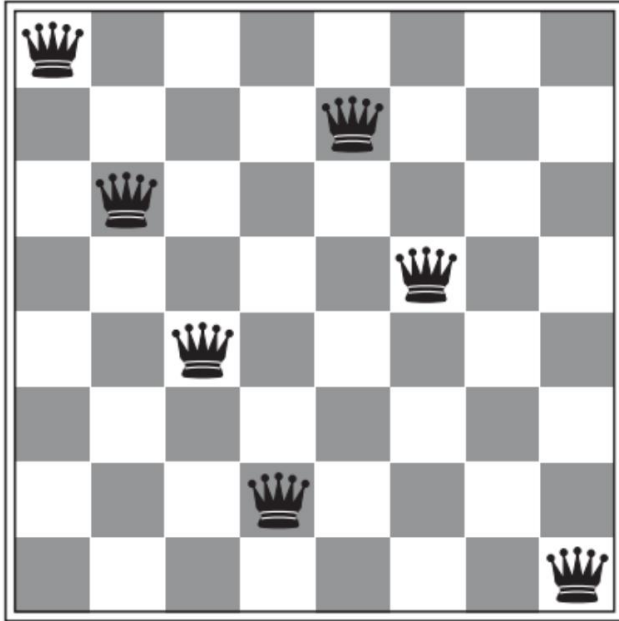
• 8 câu đố: $9!/2 = 181.440$ trạng thái có thể tiếp cận giải dễ

dàng. • 15 câu đố: 1,3 nghìn tỷ (10^{12}) trạng thái giải quyết tối ưu trong vài

mili giây • 24 câu đố: khoảng 10^{25} trạng thái giải quyết tối ưu trong vài giờ

- Lưu ý thuộc tính chẵn lẻ phân chia không gian trạng thái–bất kỳ mục tiêu nào có thể đạt được từ chính xác một nửa số trạng thái ban đầu có thể có.

Bài toán 8 quân hậu



- Công thức tăng dần: thêm từng bước một nữ hoàng vào trạng thái ban đầu trống
- Công thức trạng thái hoàn chỉnh: bắt đầu với tất cả 8 quân hậu trên bàn cờ và di chuyển chúng xung quanh
- Chi phí hành động/đường đi là không đáng kể vì chỉ tính trạng thái cuối cùng.

8 quân hậu: Công thức tăng dần

- **Kỳ:** Bất kỳ sự sắp xếp nào từ 0 đến 8 quân hậu trên bàn cờ.
 - **Trạng thái ban đầu:** Không có quân hậu trên bàn cờ
 - **Hành động:** Thêm quân hậu vào bất kỳ ô trống nào.
 - **Mô hình chuyển tiếp:** Trả lại bàn cờ có quân hậu mới được thêm vào ô đã chỉ định
 - **Mục tiêu nêu rõ:** Có 8 quân hậu trên bàn cờ, không có quân hậu nào bị tấn công.
- $64 \cdot 63 \cdot 57 \approx 1,8 \times 10^{14}$ trình tự có thể khảo sát

8 quân hậu: Công thức tăng dần

- Một công thức tốt hơn sẽ cấm đặt quân hậu vào bất kỳ ô nào đó đã bị tấn công rồi.
- **Trạng thái:** Tất cả các cách sắp xếp có thể có của quân hậu ($0 \dots 8$), mỗi quân hậu ở cột ngoài cùng bên trái, không có quân hậu nào tấn công quân hậu khác.
- **Hành động:** Thêm một quân hậu vào bất kỳ ô nào ở cột trống ngoài cùng bên trái để nó không bị các quân hậu khác tấn công
- 8 nữ hoàng: từ $1,8 \times 10^{14}$ tiểu bang xuống chỉ còn 2.057 tiểu bang
- 100 nữ hoàng: từ 10^{400} tiểu bang đến khoảng 10^{52} tiểu bang.
- Một **cải tiến lớn như ng chưa đủ** để giải quyết vấn đề,

Bài toán thứ 4 của Donald Knuth (1964)

- Bắt đầu từ số 4, một chuỗi các phép tính giai thừa, căn bậc hai và tăng sẽ đạt đến bất kỳ số nguyên dư ơng mong muốn nào.

$$\left[\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{(4!)!}}}}} \right] = 5$$

620.448.401.733.239.439.360.000 bư ớc

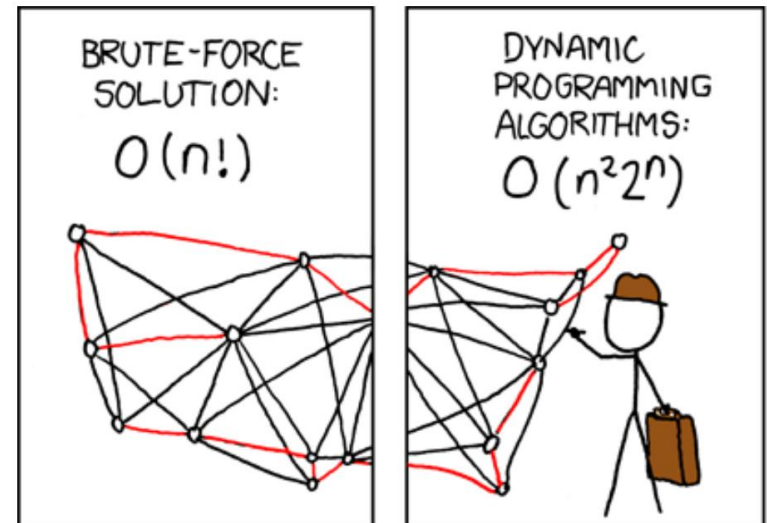
- **Trạng thái:** số dư ơng không gian trạng thái vô hạn
- **Trạng thái ban đầu:** 4
- **Hành động:** Áp dụng phép tính giai thừa (chỉ dành cho số nguyên), căn bậc hai và phép toán sà
- **Mô hình chuyển tiếp:** đư ợc đư a ra bởi định nghĩa toán học của các phép tính
- **Trạng thái mục tiêu:** Số nguyên dư ơng mong muốn.
- **Chi phí hành động:** Mỗi hành động tốn 1.
- Không gian trạng thái vô hạn phát sinh trong các nhiệm vụ như biểu thức toán học tạo, mạch, bằng chứng, chương trình, v.v.

Bài toán tìm đường

- Hãy xem xét các vấn đề về du lịch hàng không được giải quyết bằng một trang Web lập kế hoạch du lịch.
- **Tiểu bang:** Mỗi tiểu bang bao gồm một địa điểm (ví dụ: sân bay) và thời gian hiện tại.
 - Thông tin bổ sung về các khía cạnh “lịch sử”, ví dụ: các chặng trước, cơ sở giá vé, trạng thái là trong nước hoặc quốc tế, là cần thiết.
- **Trạng thái ban đầu:** Sân bay quê hương của người dùng.
- **Hành động:** Thực hiện bất kỳ chuyến bay nào từ địa điểm hiện tại, ở bất kỳ hạng ghế nào, khởi hành sau thời gian hiện tại, chứa đủ thời gian để di chuyển trong sân bay.
- **Mô hình chuyển tiếp:** Trạng thái khi thực hiện chuyến bay sẽ có điểm đến của chuyến bay là địa điểm mới và thời gian đến của chuyến bay là thời gian mới.
- **Mục tiêu nêu rõ:** Một thành phố đích đến.
- **Chi phí hành động:** Phụ thuộc vào các yếu tố khác nhau của thứ tự đo hiệu quả hoạt động
 - Sự kết hợp giữa chi phí tiền tệ, thời gian chờ đợi, thời gian bay, thủ tục hải quan và nhập cảnh, chất lượng chỗ ngồi, thời gian trong ngày, loại máy bay, v.v.

Các vấn đề về du lịch

- Các vấn đề về du lịch mô tả một tập hợp các địa điểm phải được ghé thăm, thay vì một điểm đến mục tiêu duy nhất.
- Bài toán nhân viên bán hàng du lịch (TSP): Mỗi thành phố phải được ghé thăm đúng một lần và chuyến tham quan phải ngắn nhất (nếu được yêu cầu).
 - Ứng dụng: Đặt máy móc trên sàn nhà xưởng, lập kế hoạch di chuyển của máy khoan bảng mạch tự động, v.v.
 - Đây là một trường hợp bài toán NP-khó.



Robot điều hướng

- Robot điều hướng khái quát hóa bài toán tìm đường.
- Không gian tìm kiếm của robot hình tròn chuyển động trên mặt phẳng là về cơ bản là hai chiều.
- Robot có tay và chân cũng phải được điều khiển không gian trở nên đa chiều, một chiều cho mỗi góc khớp.
 - Cần có các kỹ thuật tiên tiến chỉ để thực hiện liên tục về cơ bản không gian tìm kiếm hữu hạn.
- Robot thực cũng phải xử lý các lỗi trong kết quả đọc cảm biến và điều khiển động cơ với khả năng quan sát một phần và với các tác nhân khác.

Câu đố 01: Tháp Hà Nội

• **Trạng thái:** Bất kỳ sự sắp xếp nào của các đĩa thành ba chốt sao cho bất kỳ đĩa nào cũng nằm trên đầu của một đĩa lớn hơn khác. ba đĩa.

• **Trạng thái ban đầu:**

• **Thao tác:** Lấy đĩa phía trên từ một trong các thanh và trượt nó lên một thanh khác, lên trên các đĩa khác có thể đã có trên thanh đó mà không vi phạm quy tắc về kích thước đĩa.

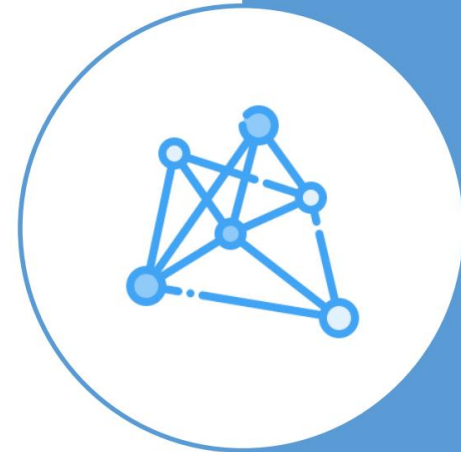
• **Mô hình chuyển tiếp:** Trả về một trạng thái khi có một trạng thái và một hành động

• **Trạng thái mục tiêu:**

• **Chi phí hành động:** Mỗi bước tốn 1

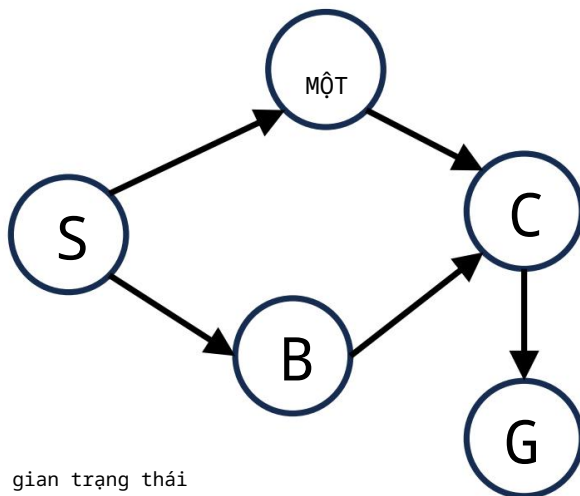
Thuật toán tìm kiếm

- Tìm kiếm đầu tiên tốt nhất
- Tìm kiếm cấu trúc dữ liệu

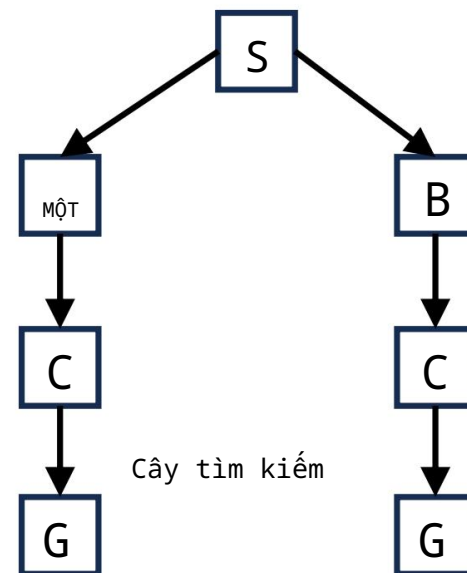


Thuật toán tìm kiếm

- Thuật toán tìm kiếm lấy vấn đề tìm kiếm làm đầu vào và trả về một giải pháp hoặc một dấu hiệu thất bại.
- Cây tìm kiếm hiển thị các đường dẫn khác nhau từ trạng thái ban đầu, cố gắng tìm đường dẫn đến trạng thái mục tiêu.
 - Nút = một trạng thái trong không gian trạng thái. Cạnh = hành động. Gốc = trạng thái ban đầu.



Không gian trạng thái

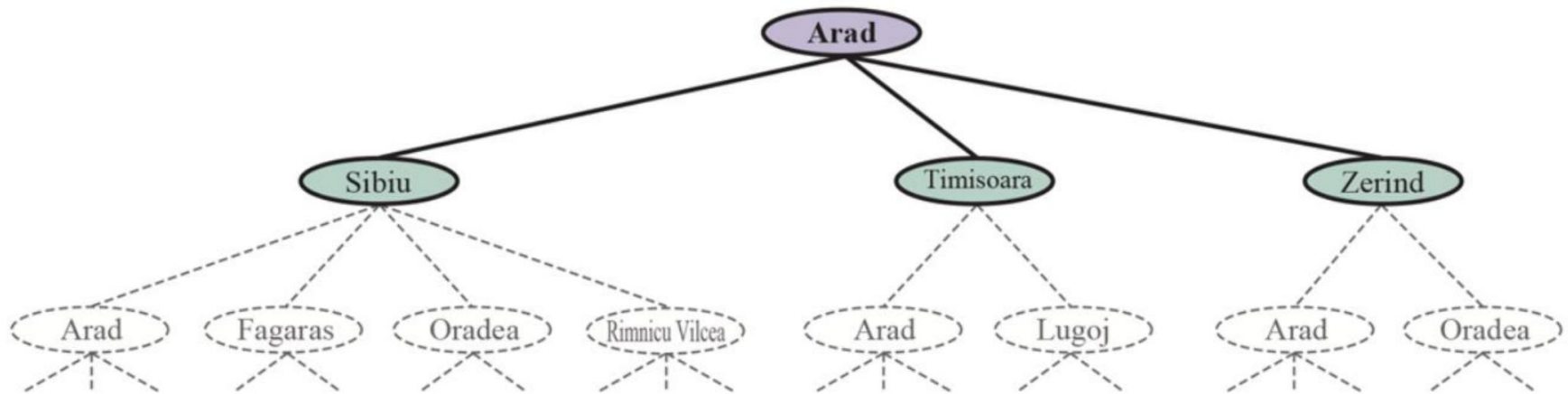


Cây tìm kiếm

Không gian trạng thái so với cây tìm kiếm

- Không gian trạng thái mô tả tập hợp các trạng thái trên thế giới và các hành động cho phép chuyển đổi từ trạng thái này sang trạng thái khác.
- Cây tìm kiếm mô tả đường đi giữa các trạng thái này để đạt được mục tiêu.
 - Có thể có nhiều đường dẫn ở bất kỳ trạng thái nào, nhưng mỗi nút trong cây có một đường đi duy nhất trở về gốc.

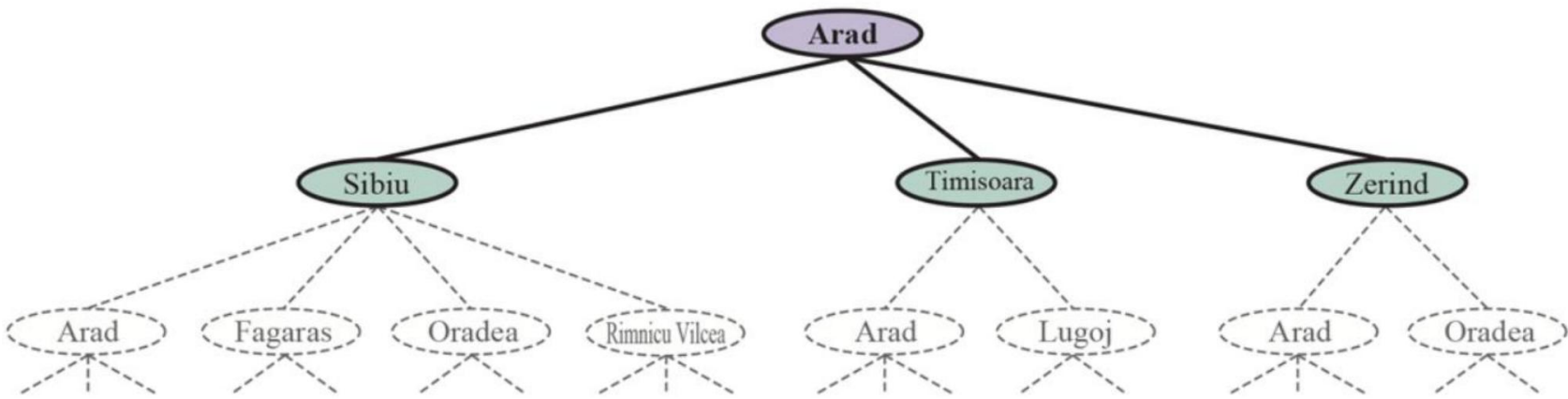
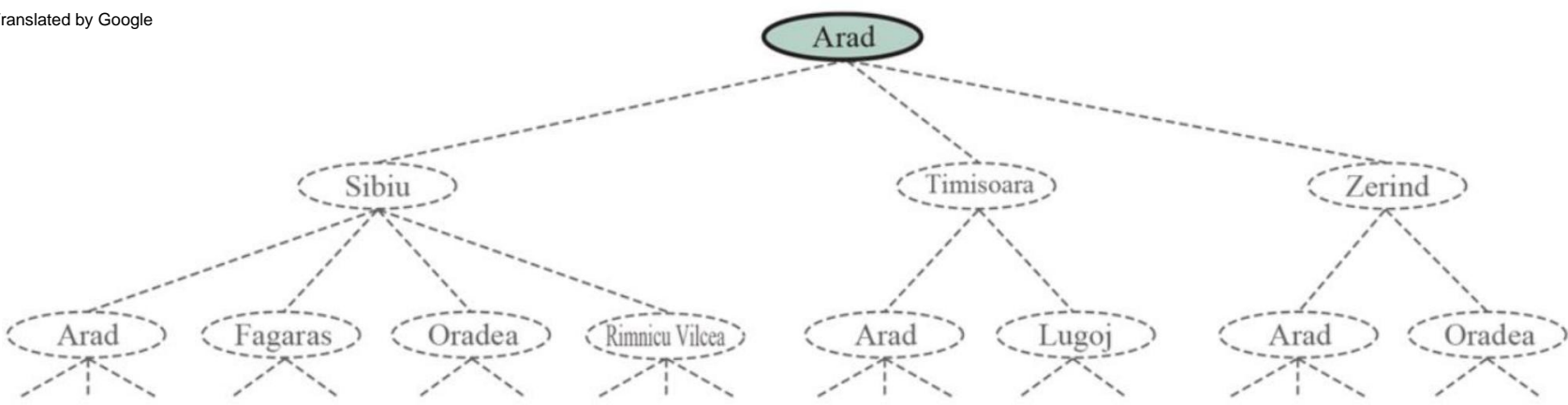
Cây tìm kiếm: Một ví dụ cho bản đồ Romania



Cây tìm kiếm một phần để tìm tuyến đường từ Arad đến Bucharest. Các nút đã được mở rộng là hoa oải hương với chữ in đậm; các nút trên đường biên đã được tạo nhưng chưa được mở rộng có màu xanh lục; tập hợp các trạng thái tương ứng với hai loại nút này được cho là đã đạt được. Các nút có thể được tạo tiếp theo được hiển thị bằng các đường đứt nét mờ.

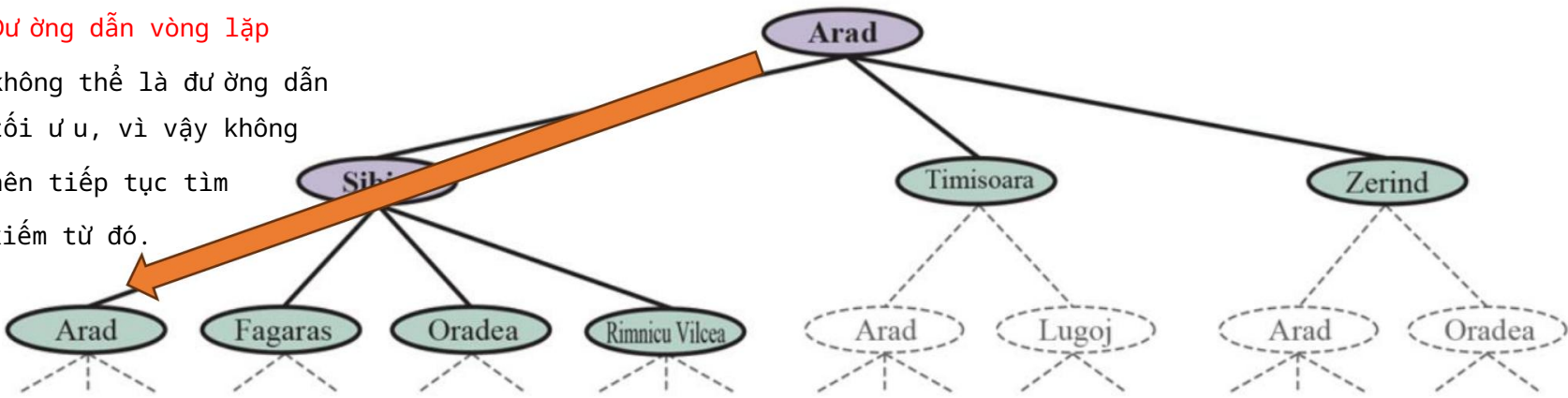
Cây tìm kiếm: Thuật ngữ

- Một nút đư ợc gọi là **mở rộng** nếu chúng ta áp dụng các hành động có sẵn vào trạng thái đó, dẫn đến **việc tạo ra** các **nút kế thừa**.
- Những ngư ời kế thừa đó đư ợc thêm vào **biên giới**, có sẵn để mở **rộng** tại bất kỳ thời điểm nào.
- Bất kỳ trạng thái nào có nút đư ợc tạo cho nó đều đã đư ợc **đạt**.
 - Các nút đã tiếp cận = Các nút đư ợc mở rộng + Các nút ở biên giới.

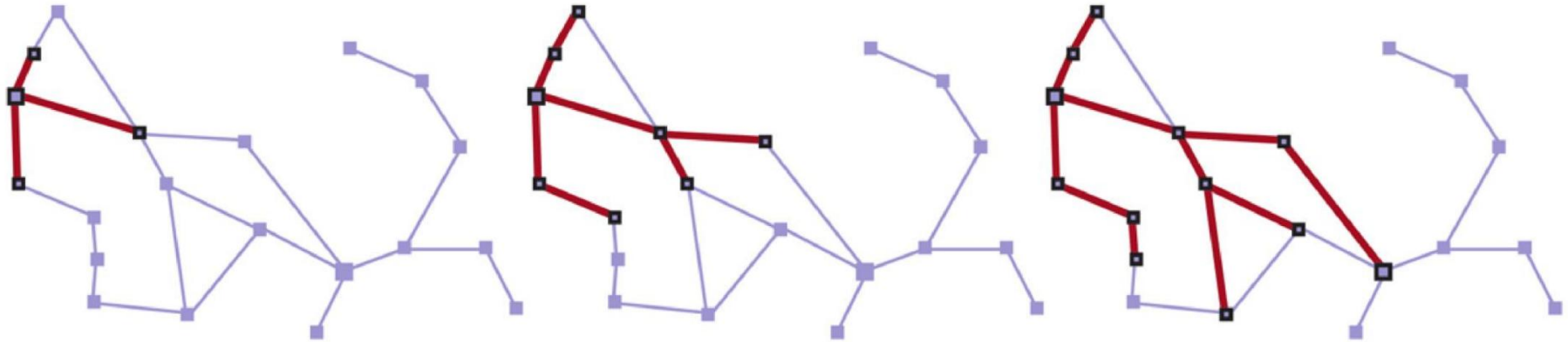


Đường dẫn vòng lặp

không thể là đường dẫn
tối ưu, vì vậy không
nên tiếp tục tìm
kiếm từ đó.



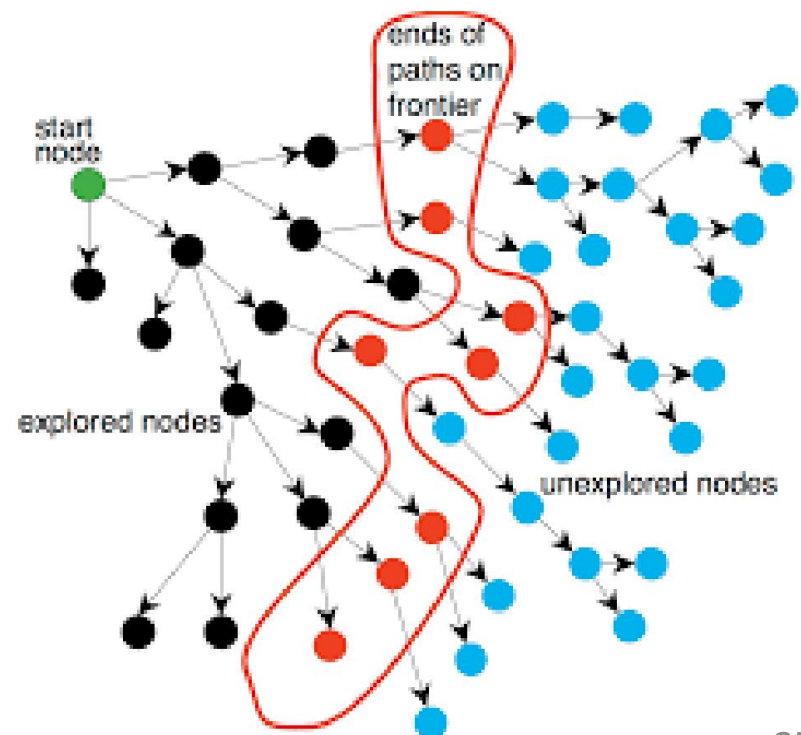
Cây tìm kiếm: Một ví dụ cho bản đồ Romania



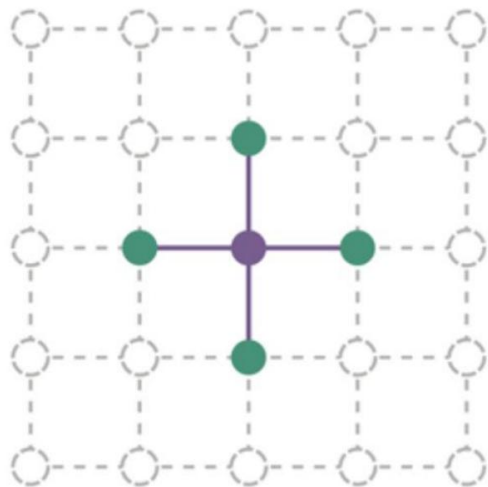
Một chuỗi các cây tìm kiếm được tạo ra bởi một tìm kiếm đồ thị về bài toán Romania. Ở mỗi giai đoạn, chúng tôi đã mở rộng mọi nút trên biên giới, mở rộng mọi đường dẫn với tất cả các hành động có thể áp dụng mà không dẫn đến trạng thái đã đạt được.

Thuộc tính phân tách tìm kiếm đồ thị

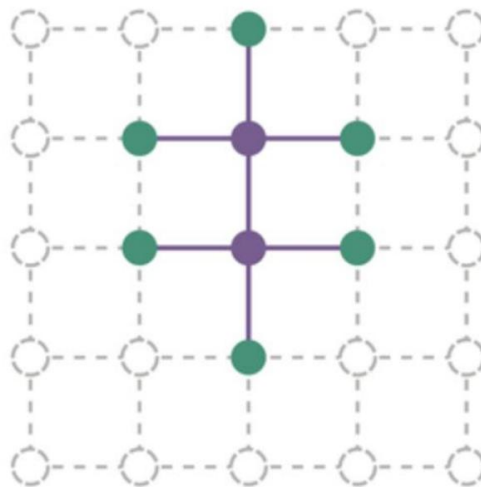
- Đứng biên **ngăn cách hai vùng** của đồ thị không gian trạng thái.
- **Vùng bên trong** là nơi mọi trạng thái đã được mở rộng và **vùng bên ngoài** dành cho các trạng thái chưa được mở rộng.
- Thuật toán kiểm tra các trạng thái một cách có hệ thống cho đến khi tìm ra giải pháp.



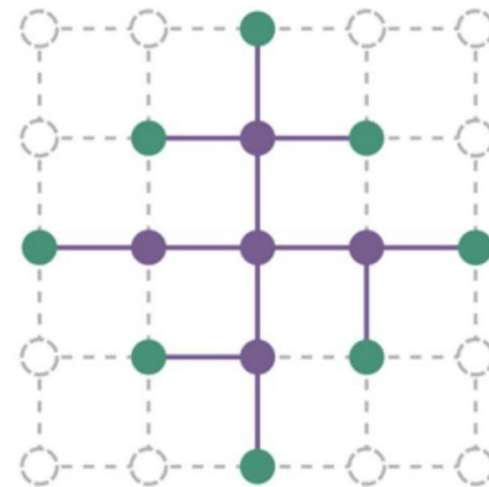
Thuộc tính phân tách tìm kiếm đồ thị



(a)



(b)



(c)

Thuộc tính tách biệt của tìm kiếm đồ thị, được minh họa bằng bài toán lưới hình chữ nhật. Đờng viền (màu xanh lá cây) ngăn cách bên trong (màu hoa oải hương) với bên ngoài (nét đứt nét). Biên giới là tập hợp các nút (và trạng thái tương ứng) đã đạt được nhưng chưa được mở rộng; phần bên trong là tập hợp các nút (và trạng thái tương ứng) đã được mở rộng; và bên ngoài là tập hợp các trạng thái chưa đạt tới. Trong (a), chỉ có gốc đã được mở rộng. Trong (b), nút biên trên cùng được mở rộng. Trong (c), các phần tử kế thừa còn lại của gốc được mở rộng theo chiều kim đồng hồ.

Tìm kiếm đầu tiên tốt nhất

- **Tìm kiếm tốt nhất đầu tiên** là một cách tiếp cận tổng quát chọn một nút có giá trị tối thiểu của một **hàm đánh giá nào đó**, ().
- Sự lựa chọn xác định chiến lược tìm kiếm cụ thể.

hàm TÌM KIẾM TỐT NHẤT(vấn đề, f) trả về nút giải pháp hoặc lỗi

nút NODE(STATE=problem.INITIAL) Frontier

hàng đợi ưu tiên được sắp xếp theo f, với nút là một phần tử đạt

được một bảng tra cứu, với một mục nhập có khóa problem.INITIAL và nút giá trị trong khi không phải IS -EMPTY(frontier) nút

POP(frontier) nếu

problem.IS-GOAL(node.STATE) thì trả về nút cho mỗi phần

tử con trong EXPAND(problem, node) thực hiện s

child.STATE

nếu s không đạt được hoặc con.PATH-COST < đã đạt[s].PATH-COST thì đạt[s]

con thêm con vào

biên giới thất bại

hàm EXPAND(vấn đề, nút) mang lại các nút

s node.STATE

cho mỗi hành động trong vấn đề.ACTIONS(s) do

s' problem.RESULT(s, action)

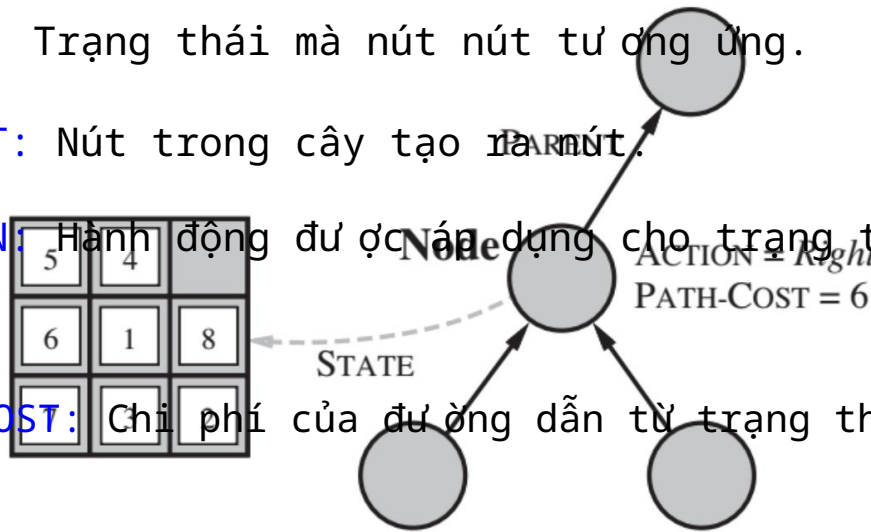
cost s.PATH-COST + problem.ACTION-COST(s, action, s') mang

lại NODE(STATE=s' , PARENT=nút, ACTION=hành động, PATH-COST=chi phí)

Tìm kiếm cấu trúc dữ liệu

- Chúng ta cần một số cấu trúc dữ liệu để theo dõi cây tìm kiếm.
- Một nút trong cây được biểu diễn bằng bốn thành phần.

- **node.STATE**: Trạng thái mà nút nút tương ứng.
- **node.PARENT**: Nút trong cây tạo ra nút.
- **node.ACTION**: Hành động được áp dụng cho trạng thái của nút cha để tạo ra nút.
- **node.PATH-COST**: Chi phí của đường dẫn từ trạng thái ban đầu đến nút.



- **Giải pháp** được hình thành bằng cách làm theo con trỏ PARENT quay trở lại từ nút mục tiêu đến nút ban đầu.

Tìm kiếm cấu trúc dữ liệu

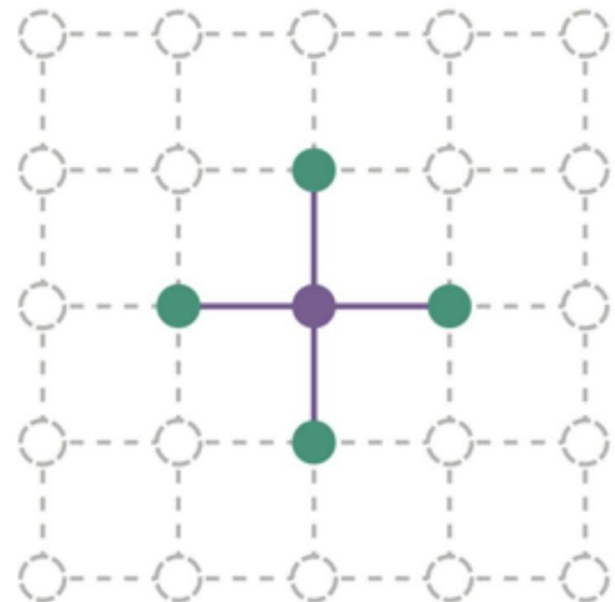
- Biên giới có thể được triển khai bằng hàng đợi ưu tiên (UCS và A^*), hàng đợi FIFO (BFS) hoặc hàng đợi LIFO (DFS).
- Các trạng thái đạt được có thể được lưu trữ trong bảng tra cứu (ví dụ: bảng băm).
 - Mỗi phần tử có trạng thái là khóa và nút cho trạng thái đó là giá trị.

Đường dẫn dự phòng

- Đường dẫn dự phòng có thể là đường vòng hoặc đường dẫn tối ưu, cả hai đều không thể tránh khỏi.
- Đi theo những con đường này có thể biến một vấn đề có thể giải quyết được thành một vấn đề một thứ khó chữa .

Mỗi bang có bốn bang kế nhiệm.

Cây tìm kiếm có độ sâu: 4 lá với các trạng thái lặp lại, khoảng 2^4 trạng thái riêng biệt trong các bước của bất kỳ trạng thái nhất định nào Với
= 20: một nghìn tỷ nút nhưng chỉ có khoảng 800 trạng thái riêng biệt.



Phương pháp tiếp cận các đường dẫn dư thừa

- **Thuật toán tìm kiếm cây:** Không kiểm tra các đường dẫn dư thừa.
 - Có một số cách giải bài toán hiểm hoặc không thể thực hiện được để hai đường đi đến cùng một trạng thái.
- **Thuật toán tìm kiếm đồ thị:** Ghi nhớ tất cả những gì đã đạt được trước đó các trạng thái và chỉ giữ lại đường đi tốt nhất tới mỗi trạng thái.
 - Thích hợp cho các tình huống có nhiều đường dẫn dư thừa, và bảng trạng thái đạt được sẽ nằm gọn trong bộ nhớ.

“Tìm kiếm trên cây” đề cập đến thực tế là không gian trạng thái được xử lý như thể nó là một cái cây, chỉ có một đường dẫn từ mỗi nút trở về gốc. Không gian trạng thái vẫn là đồ thị giống nhau cho dù chúng ta tìm kiếm nó như thế nào.

Phương pháp tiếp cận các đường dẫn dư thừa

- Chúng ta có thể thỏa hiệp và kiểm tra các chu trình, nhưng không thể kiểm tra các đường dẫn dư thừa nói chung.
- Ý tưởng: Theo dõi chuỗi phụ huynh để xem tình hình ở trạng thái phần cuối của đường dẫn đã xuất hiện trước đó trong đường dẫn.
 - Điều này có thể được thực hiện mà không cần thêm bộ nhớ.
- Một số cách triển khai tuân theo chuỗi này cho đến tận khi loại bỏ tất cả các chu trình.
- Những người khác chỉ theo dõi một vài liên kết (ví dụ cha mẹ ông bà ông cố), và do đó chỉ phát hiện các chu kỳ ngắn.

Đo lường hiệu quả giải quyết vấn đề

- **Tính đầy đủ:** Nó có luôn tìm ra giải pháp nếu có không?
 - **Tính tối ưu:** Có phải luôn tìm được giải pháp có chi phí thấp nhất?
 - **Độ phức tạp về thời gian:** Mất bao lâu để tìm ra giải pháp?
 - **Độ phức tạp về không gian:** Cần bao nhiêu bộ nhớ để thực hiện tìm kiếm?
- Độ phức tạp về thời gian và không gian được đo lường một cách trừu tượng bằng số trạng thái và hành động được xem xét.
- : số lượng nút kế thừa của một nút cần được xem xét
 - : số hành động trong một giải pháp tối ưu
 - : số lượng hành động tối đa trong bất kỳ đường dẫn nào (có thể là ∞)

Câu đố 02: Tháp Hà Nội

- Vẽ hai cấp độ đầu tiên của cây tìm kiếm Tháp

Bài toán Hà Nội với ba chốt và hai đĩa (có thể bỏ qua các trạng thái lặp lại giống hệt nhau trên một đường dẫn).

