

## TỔNG QUAN & TÌM KIẾM

### 1.1. Tổng quan về trí tuệ nhân tạo (artificial intelligence)

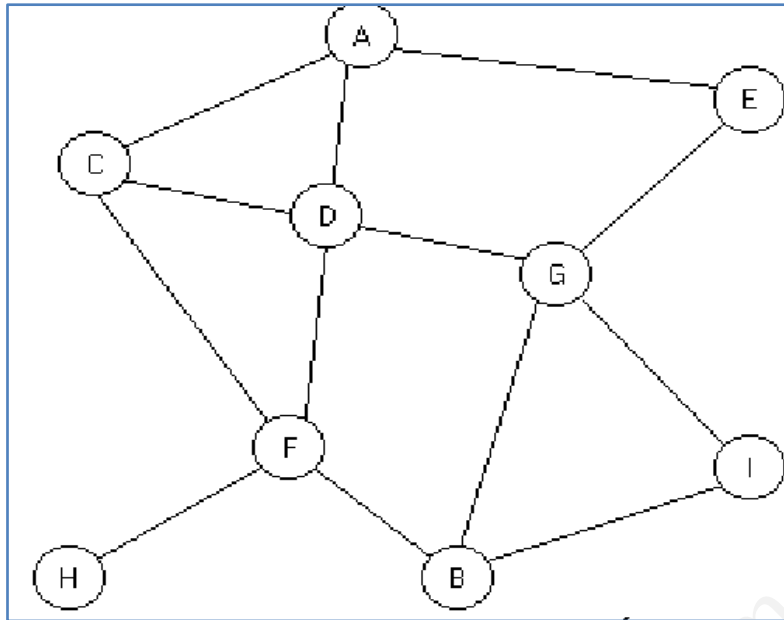
- Trí tuệ nhân tạo (TTNT) là một ngành khoa học máy tính, chuyên nghiên cứu các chương trình hoặc thiết bị thông minh, có khả năng suy luận, khả năng học,...
- Lĩnh vực nghiên cứu chính của TTNT là biểu diễn tri thức và suy luận.
- Áp dụng của TTNT rất đa dạng như là xây dựng hệ chuyên gia, hệ điều khiển tự động, robot, hệ dịch các ngôn ngữ tự nhiên, hệ nhận dạng, chương trình chơi cờ,...
- TTNT áp dụng kiến thức của nhiều lĩnh vực như là: toán học, kinh tế học, thần kinh học, tâm lý học, điều khiển học, triết học,...
- Vào năm 1956, TTNT được đưa ra hội thảo ở Dartmouth. Các nhà nghiên cứu đã thảo luận và đưa ra các hướng nghiên cứu và 1956 được xem là thời điểm ra đời của lĩnh vực nghiên cứu TTNT.

### 1.2. Vấn đề tìm kiếm

- Nhiều vấn đề có thể chuyển thành vấn đề tìm kiếm, có nghĩa là tìm kiếm trạng thái thỏa yêu cầu trong một tập hợp các trạng thái.

#### Ví dụ 1:

Có bản đồ giao thông như hình 1.1, khách du lịch đang ở thành phố A và muốn tìm đường đi tới thành phố B.



Hình 1.1: Tìm đường đi từ A đến B

Các thành phố gọi là các trạng thái. Thành phố A là trạng thái ban đầu, B là trạng thái kết thúc.

Các con đường nối các thành phố biểu diễn các phép biến đổi trạng thái (toán tử). Một phép biến đổi trạng thái biến đổi một trạng thái thành một trạng thái khác.

Vấn đề tìm đường đi trở thành vấn đề “tìm một chuỗi các phép biến đổi trạng thái” để biến đổi trạng thái ban đầu A tới trạng thái kết thúc B.

### **Ví dụ 2:**

Trong trò chơi cờ vua, mỗi cách bố trí các quân trên bàn cờ là một trạng thái.

Trạng thái ban đầu là sự sắp xếp các quân lúc bắt đầu chơi. Trạng thái kết thúc là trạng thái “chiếu bí”.

Mỗi nước đi hợp lệ là một phép biến đổi trạng thái, biến đổi một trạng thái bàn cờ thành một trạng thái khác.

Vấn đề chơi cờ vua trở thành vấn đề “tìm một chuỗi các phép biến đổi trạng thái” để biến đổi trạng thái ban đầu (trạng thái xếp quân bắt đầu chơi) tới trạng thái kết thúc (chiếu bí).

## **1.3 Không gian trạng thái, phép biến đổi trạng thái**

- Không gian trạng thái:** là tập hợp tất cả các trạng thái có thể đi đến từ trạng thái ban đầu bởi một dãy các phép biến đổi trạng thái.

- **Phép biến đổi trạng thái (toán tử):** là phép biến đổi một trạng thái sang trạng thái khác.
- **Biểu diễn Không gian trạng thái:** bằng đồ thị có hướng, trong đó:
  - đỉnh tương ứng với trạng thái.
  - cung (a,b) tương ứng với phép biến đổi trạng thái a sang trạng thái b.

#### 1.4 Giải quyết vấn đề bằng tìm kiếm

- **Muốn giải quyết vấn đề bằng tìm kiếm, ta cần xác định các yếu tố sau:**
  - ✓ Trạng thái của vấn đề
  - ✓ Không gian trạng thái (tập tất cả các trạng thái có thể có, tập này có thể rất lớn)
  - ✓ Trạng thái ban đầu
  - ✓ Các trạng thái kết thúc (hoặc tính chất của các trạng thái kết thúc).
  - ✓ Các phép biến đổi trạng thái.
- **Khi đó việc giải quyết vấn đề trở thành:**  
 Tìm một dãy các phép biến đổi trạng thái để biến đổi trạng thái ban đầu thành các trạng thái kết thúc, hoặc tìm đường đi từ trạng thái ban đầu đến trạng thái kết thúc trên đồ thị có hướng.

#### 🎮 Ví dụ 1: Trò chơi 8 số.

Cho bảng kích thước 3x3 và các số từ 1 đến 8 được xếp vào 8 ô một cách ngẫu nhiên, còn lại một ô trống. Có thể di chuyển số ở cạnh ô trống tới ô trống theo một trong bốn hướng (lên/xuống/trái/phải). Hãy tìm các bước di chuyển để biến đổi trạng thái ban đầu thành trạng thái kết thúc như hình 1.2.

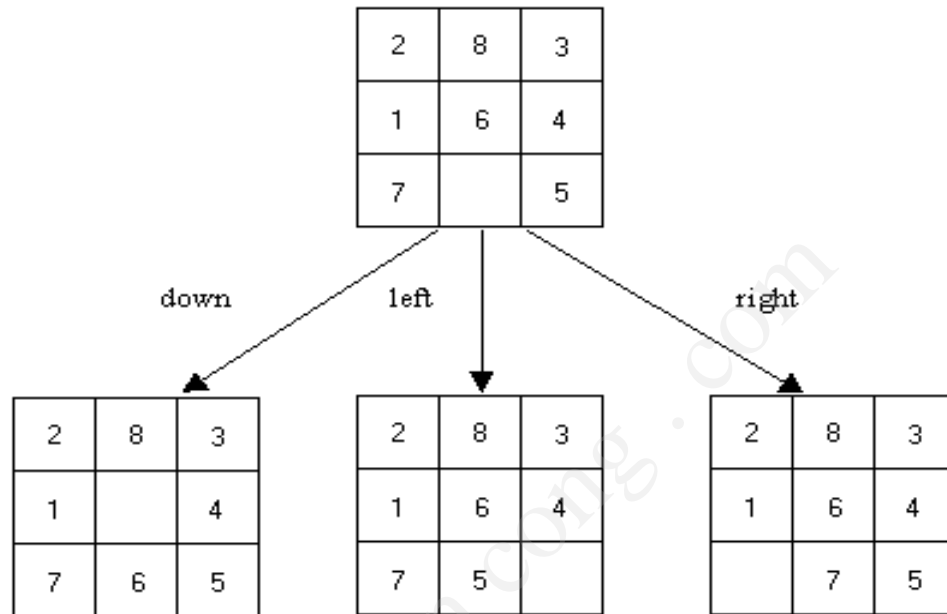
Ban đầu			Kết thúc		
2	8	3	1	2	3
1	6	4	8		4
7		5	7	6	5

Hình 1.2: trạng thái ban đầu và trạng thái kết thúc của trò chơi 8 số

- Trò chơi 8 số có thể chuyển thành bài toán tìm kiếm trong không gian trạng thái như sau:
  - ✓ Trạng thái: là một cách xếp 8 số.
  - ✓ Không gian trạng thái: tập hợp tất cả các cách xếp 8 số.
  - ✓ Trạng thái ban đầu: trạng thái ban đầu đã cho.

- ✓ Trạng thái kết thúc: trạng thái kết thúc mong muốn.
- ✓ Phép biến đổi trạng thái: di chuyển ô có số sang ô trống. Có bốn phép biến đổi là dịch trái/phải/ lên/xuống.

- Áp dụng các phép biến đổi ta có thể biến đổi trạng thái ban đầu (hình 1.2) thành các trạng thái khác như hình 1.3.



Hình 1.3: áp dụng các phép biến đổi trạng thái

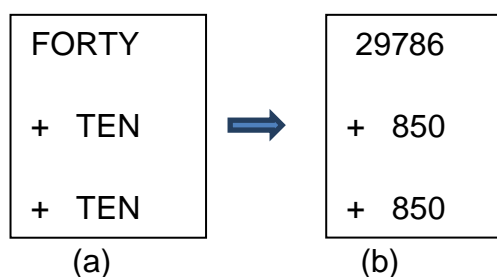
### 🚦 Ví dụ 2: Triệu phú và kẻ cướp.

Có  $n$  nhà triệu phú và  $n$  tên cướp ở bờ sông bên trái, và có một chiếc thuyền chở mỗi lần được một hoặc hai người (không tính người chèo thuyền). Hãy tìm cách đưa mọi người sang bờ sông bên phải sao cho không để lại ở một bên bờ sông kẻ cướp nhiều hơn triệu phú.

- Vấn đề triệu phú và kẻ cướp có thể chuyển thành bài toán tìm kiếm trong không gian trạng thái như sau:
  - ✓ Trạng thái:  $(T, C)$  với  $T$  và  $C$  lần lượt là số triệu phú và số tên cướp ở bờ sông bên trái.
  - ✓ Không gian trạng thái: tập hợp tất cả các cặp  $(T, C)$  với  $T \geq C$  ở cả hai bờ.
  - ✓ Trạng thái ban đầu:  $(n, n)$
  - ✓ Trạng thái kết thúc:  $(0, 0)$
  - ✓ Các phép biến đổi trạng thái: có 5 phép biến đổi tương ứng với việc chở qua sông 1 triệu phú, hoặc 1 kẻ cướp, hoặc 2 triệu phú, hoặc 2 kẻ cướp, hoặc 1 triệu phú và 1 kẻ cướp.

### 🚦 Ví dụ 3: Trò chơi “thay chữ bằng số”.

Hãy thay mỗi chữ trong phép toán cộng (hình 1.4a) bằng một số khác nhau sao cho thỏa phép toán cộng (hình 1.4b).



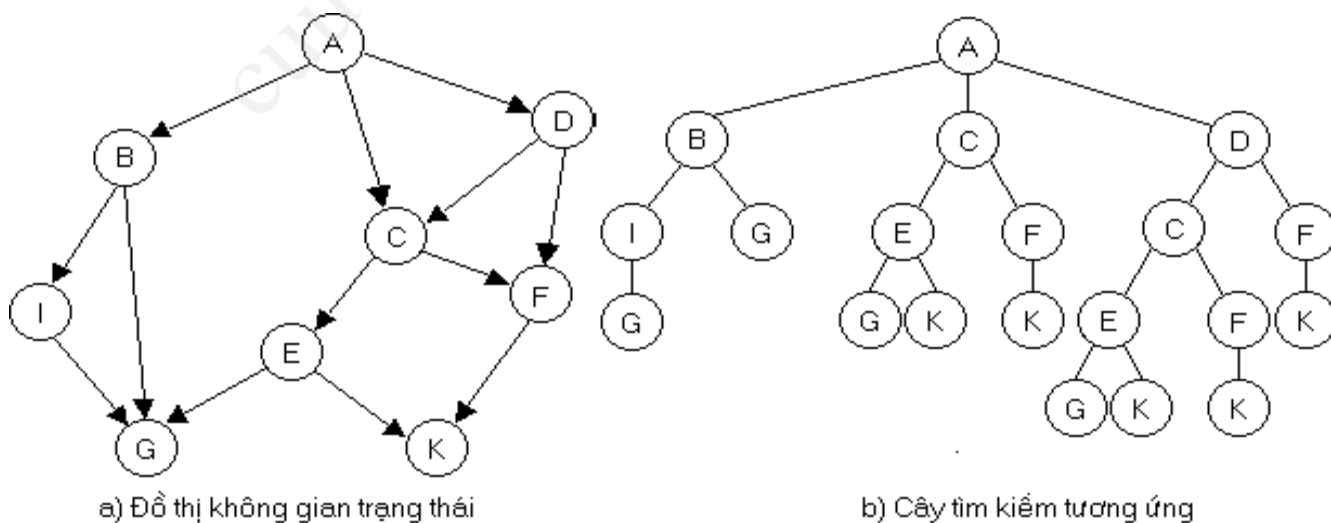
Hình 1.4: Một phép toán cộng dạng chữ (hình a) và một cách thay bằng số (hình b)

- Trò chơi “thay chữ bằng số” có thể chuyển thành bài toán tìm kiếm trong không gian trạng thái như sau:
  - ✓ Trạng thái: một phép toán cộng với không hoặc nhiều chữ đã được thay bằng số hợp lệ.
  - ✓ Không gian trạng thái: tất cả trạng thái dẫn xuất từ trạng thái ban đầu.
  - ✓ Trạng thái ban đầu: phép toán cộng tất cả là chữ
  - ✓ Trạng thái kết thúc: phép toán cộng tất cả là số
  - ✓ Các phép biến đổi trạng thái: thay một chữ bằng một số hợp lệ và không trùng với những số đã có.

### 1.5 Cây tìm kiếm

- Cây tìm kiếm là cây được xây dựng từ không gian trạng thái, trong đó các đỉnh chứa các trạng thái của không gian trạng thái, gốc của cây tìm kiếm lưu trạng thái ban đầu.

Đỉnh con v của u là đỉnh chứa trạng thái được biến đổi từ trạng thái của đỉnh u qua một phép biến đổi trạng thái.



Hình 1.5: Hình (a) là đồ thị biểu diễn một không gian trạng thái với trạng thái ban đầu là A. Hình (b) là cây tìm kiếm tương ứng với không gian trạng thái.

- Cấu trúc một đỉnh  $u$  trong cây tìm kiếm cần có các thông tin sau:
  - Trạng thái mà  $u$  biểu diễn.
  - Đỉnh cha của đỉnh  $u$ .
  - Phép biến đổi để biến đổi đỉnh cha của đỉnh  $u$  thành  $u$ .
  - Chi phí thực sự để chuyển từ đỉnh ban đầu tới đỉnh  $u$ , kí hiệu  $g(u)$ .
  - Chi phí dự kiến để chuyển đỉnh  $u$  đến đỉnh chứa trạng thái kết thúc, kí hiệu  $h(u)$ .

Lưu ý: hai đỉnh khác nhau có thể biểu diễn cùng trạng thái.

## 1.6 Thuật toán tìm kiếm

- Mỗi thuật toán tìm kiếm sẽ xây dựng một cây tìm kiếm.
- Thuật toán xây dựng cây bắt đầu từ cây chỉ có một đỉnh là đỉnh bắt đầu (đỉnh chứa trạng thái ban đầu) và đỉnh này chưa được phát triển, sau đó phát triển đỉnh này được các đỉnh con chưa được phát triển.
- Bước tiếp theo chọn một đỉnh con để phát triển, việc chọn đỉnh con nào để phát triển thì phụ thuộc vào thuật toán tìm kiếm.
- Khi phát triển đỉnh  $u$ , cây tìm kiếm được mở rộng bằng cách thêm vào các đỉnh con của đỉnh  $u$ . Lặp lại việc phát triển cho tới khi tìm được đỉnh kết thúc (đỉnh chứa trạng thái kết thúc) thì ngừng.

**Lưu ý:**

- Mỗi đỉnh cần có biến `father` để lưu lại cha của đỉnh đó. Dựa vào `father` ta sẽ tìm ra đường đi từ đỉnh đầu đến đỉnh kết thúc.
- Trong quá trình tìm kiếm cần tránh phát sinh ra các trạng thái đã phát triển hoặc chờ phát triển bằng cách sử dụng hai danh sách `open` và `close`.
- Danh sách `open`: lưu các đỉnh chờ phát triển. Ban đầu `open` chỉ có một đỉnh đó là đỉnh bắt đầu.

- Danh sách close: lưu các đỉnh đã phát triển.

Như vậy trạng thái v lần đầu được sinh ra nếu v không có trong close và không có trong open.

### 1.7 Các nhóm thuật toán tìm kiếm

- **Nhóm tìm kiếm không có thông tin (tìm kiếm mù):** không có thông tin hướng dẫn việc tìm kiếm. Nếu không gian trạng thái lớn, nhóm thuật toán này thường chậm hoặc rất chậm đến mức không thể sử dụng.
- **Nhóm tìm kiếm có thông tin (tìm kiếm heuristic):** dựa vào thông tin đã biết hoặc dựa vào kinh nghiệm để việc tìm kiếm nhanh hơn. Nhóm thuật toán này thường áp dụng cho không gian trạng thái lớn.
- **Các tiêu chuẩn đánh giá một thuật toán tìm kiếm**
  - ✓ Tính đầy đủ (Completeness): luôn tìm ra giải pháp nếu có.
  - ✓ Tính tối ưu (Optimality): tìm ra giải pháp tốt nhất
  - ✓ Tính tiết kiệm thời gian/không gian: tìm ra giải pháp nhanh, sử dụng bộ nhớ ít.

- Hết -