**THUẬT TOÁN TÌM KIẾM**

**SEARCH FOR SOLUTION**

**( Artificial Intelligence )**

**\*Chú ý : ( tìm kiếm trong không gian trang máy )**

1. **Định nghĩa :**

Tìm kiếm : là 1 quy trình từng bước giải quyết 1 vấn đề trong 1 ko gian nhất định. Có 3 yếu tố chính :

* Không gian tìm kiếm : tập hợp các giải pháp khả thi mà 1 hệ thống có thể có
* Trạng thái bắt đầu : trạng thái mà tácn nhân bắt đầu tìm kiếm
* Kiểm tra mục tiêu : hàm quan sát trạng thái hiện tại và trả về kết quả trạng thái mục tiêu có đạt được không

**2.Đặc điểm**

Để giải quyết bài toán tìm kiếm => xây dựng agent

Làm sao biết vấn đề nào cần dùng thuật toán tìm kiếm?

Các thành phần của 1 search problem :

* Không gian trạng thái : các giải pháp mà mình có thể đưa ra để giải quyết
* Tập các các hành động và chi phí
* Trạng thái bắt đầu / kết thúc
* Trạng thái mục tiêu : cần thoả mãn cái gì ?
* Có các thanh phần này thì có thể dùng thuật toán tìm kiém

Ví dụ :

**Không gian trạng thái ( không gian tìm kiếm ) :** hiển thị các sắp xếp vị trí các ô

**Các hành động và chi phí**

* Hành động : di chuyển ô trống trái, phải ,lên, xuống
* Chi phí : mỗi hành động có giá thành = 1

**Trạng thái xuất phát ?**

Trạng thái bên trái

**Trạng thái mục tiêu :**

Trạng thái bên phải

Bài tập luyện : máy hút bụi

* Không gian trạng thái : vị trí có bụi và vị trí của máy
* Các hành động : di chuyển lên, xuống, trái, phái
* Trạng thái xuất phát : vị trí xuất phát của máy hút bụi
* Trạng thái mục tiêu : làm cho sàn nhà không còn vị trí có bụi
* Chi phí : 1 ( mỗi hành động), 0( không làm gì )

Bài tập luyện : con hậu trên bàn cờ vua

* Không gian trạng thái : vị trí các ô trên bàn cờ
* Các hành động và chi phí :
  + Hành động : di chuyển lên, xuống, trái, phải, đi chéo theo ô
  + Chi phí : mỗi lần đặt là 1, không làm gì là 0
* Trạng thái xuất phát : bàn cờ trống không, chưa có con hậu
* Trạng thái mục tiêu : đặt xong 8 con hậu mà không ăn nhau

Tự nhóm tạo lập : truy tìm kho báu trong mê cung :

* Không gian trạng thái : mê cung
* Các hành động và chi phí :
  + Hành động : di chuyển lên, xuống, trái, phải, nhảy
  + Chi phí : mỗi lần hành động là 1, không làm gì là 0
* Trạng thái xuất phát : nhân vật ở vị trí đầu mê cung
* Trạng thái mục tiêu : tìm được kho báu trong mê cung

1. Ta có trạng thái ban đầu, để mở rộng trạng thái 🡪 thực hiện hành động
2. Mở rộng cho đến khi đạt trạng thái mục tiêu hoặc đến khi đã mở hết trạng thái mà vẫn chưa đạt trạng thái mục tiêu.

Một số cấu trúc trong tìm kiếm :

Tạo 1 nút tìm kiếm lưu trữ trạng thái đã đạt được, cách đạt được và chi phí bao nhiêu

\*tính chi phí từ nút gốc đến nút con

Các thuộc tính 1 nút tìm kiếm n:

n.state : trạng thái liên kết với nút này

n.parent : tìm kiếm nút đã tạo ra nút này ( không có cho nút gốc )

n.action hành động dẫn từ n.parent ….

….

\*tìm kiếm không nên dùng list

\*Nên sử dụng queue hoặc stack

Tìm hiểu danh sách đóng ( close list ) cô sẽ hỏi.

Thuật toán tìm kiếm tổng quát

\*Nguyên lý chung :

Start state 🡪 KGTT 🡪Goal state

**Đánh giá thuật toán tìm kiếm :**

* Tính hoàn chỉnh ( đảm bảo tìm được giải pháp )
* Tính tối ưu ( tốc độ nhanh )
* Độ phức tạp về thời gian
* Độ phức tạp về không gian

**Các thuật toán tìm kiếm :**

* Breath-first Search
* Depth-first Search
* Iterative Deepening
* Uniform Cost Search

1. **Breath-first Search ( duyệt theo chiều rộng )**

Tìm kiếm theo chiều rộng mở rộng các nút theo thứ tự tạo ( FIFO -first in, first out)

Ví dụ : danh sách mở dưới dạng danh sách liên kết hoặc deque

Độ phức tạp thuật toán không gian và thời gian : b mũ d ( b là số nhánh của cây, d là độ sâu của cây )

Thuật toán này chưa tối ưu, nó chỉ tối ưu khi tất cả có giá trị đồng nhất.

Duyệt trong 🡪 cấu trúc dữ liệu phải là queue

Các yếu tố cần lưu ý đối với thuật toán :

* Cấu trúc lưu trữ trạng thái mở rộng
* Có tối ưu chưa, nếu chưa thì tối ưu khi nào?
* Độ phức tạp là bao nhiêu ?

1. **Uniform Cost Search ( tìm kiếm với chi phí đồng nhất )**

**Các bước tương tự nhưng sử dụng hàng đợi ưu tiên ( queue )**