

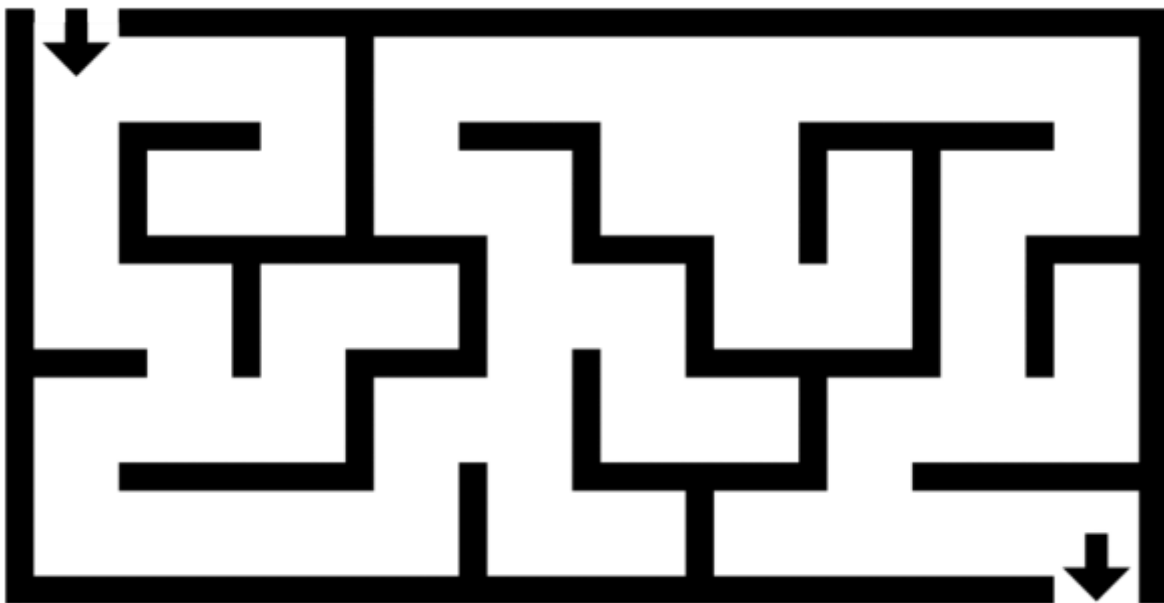
1. Backtracking là gì ?

- Backtracking là một tư tưởng thiết kế thuật toán dựa trên đệ quy để tìm ra tất cả (hoặc 1 số) giải pháp cho một bài toán. Đặc biệt là bài toán liên quan tới sự ràng buộc (Constraint satisfaction problem).
- Backtracking phụ thuộc vào yêu cầu (ràng buộc) và việc các “Ứng viên” (1 phần giải pháp) phát triển thành giải pháp hoàn chỉnh

2. Backtracking hoạt động như thế nào ?

***Ý tưởng:** Giống như cách mà con người chúng ta dùng để giải một mê cung

Mỗi khi vào một ngã rẽ có nhiều lựa chọn hoặc đường để đi liệu xem nó có dẫn đến đường ra hay không, nếu không thì quay lại và chọn con đường khả dụng tiếp theo và liệu xem nó có dẫn đến đường ra hay không.



Vì vậy chỉ cần lặp lại quá trình này mỗi khi phải đưa ra lựa chọn. Đó là lý do tại sao Backtracking là một thuật toán đệ quy.

***Cây không gian trạng thái:**

Để tiện diễn tả giải thuật quay lui, ta xây dựng cấu trúc cây ghi những lựa chọn đã được thực hiện. Cấu trúc cây này được gọi là cây không gian trạng thái (state space tree) hoặc cây tìm kiếm (search tree).

- Root node của cây diễn tả trạng thái đầu tiên trước khi quá trình tìm kiếm lời giải bắt đầu.
- Các node ở mức đầu tiên trong cây diễn tả những ứng viên được làm ứng với thành phần đầu tiên của lời giải.
- Các node ở mức thứ hai trong cây diễn tả những ứng viên được làm ứng với thành phần thứ hai của lời giải và các mức kế tiếp tương tự như thế.

Một node trên cây KGTT được gọi là triển vọng (promising) nếu nó tương ứng với lời giải bộ phận mà sẽ có thể dẫn đến lời giải đầy đủ; trái lại, nó được gọi là một lời giải không triển vọng (non-promising).

Các node lá diễn tả những trường hợp bế tắc (dead end) hay những lời giải đầy đủ.

****Mã giả:***

```
Backtracking(k) {  
    for([Mỗi phương án chọn i(thuộc tập D)]) {  
        if ([Chấp nhận i]) {  
            [Chọn i cho X[k]];  
            if ([Thành công]) {  
                [Đưa ra kết quả];  
            } else {  
                Backtracking(k+1);  
                [Bỏ chọn i cho X[k]];  
            }  
        }  
    }  
}
```

3. Khi nào sử dụng Backtracking?

****ỨNG DỤNG VÀO PROBLEM***

- Đối với những bài toán yêu cầu ràng buộc điều kiện thì backtracking là một sự lựa chọn tốt.
- Thường thường là các vấn đề giải đố (puzzle problems) như sudoku, crossword ,verbal arithmetic , hamiltonian cycle ...
- Ở đây nhóm mình sẽ sử dụng bài toán kinh điển để phân tích mô hình (tư tưởng) backtracking, đó là bài toán N-queens và Knight's Tour.

****ỨNG DỤNG VÀO THỰC TẾ***

- Lập trình game, giải các bài toán liên quan tới bản đồ (tô màu đồ thị, tìm đường đi ngắn nhất ,...)
- Vấn đề về phân công công việc. vd: giáo viên nào dạy lớp nào
- Vấn đề về phân bố thời gian biểu
- Cấu hình phần cứng
- Lập kế hoạch vận chuyển

4. Minh họa bài toán N-queens và Knight's Tour

a. N-queens

Mô tả bài toán: Bài toán N ($N \geq 4$) quân hậu là bài toán đặt N quân hậu trên bàn cờ vua kích thước $N \times N$ sao cho không có quân hậu nào có thể "ăn" được quân hậu khác theo quy tắc cờ vua. Màu của các quân hậu không có ý nghĩa trong bài toán này. Như vậy, lời giải của bài toán là một cách xếp tám quân hậu trên bàn cờ sao cho không có hai quân nào đứng trên cùng hàng, hoặc cùng cột hoặc cùng đường chéo.

Minh họa N-queen problem



b. Knight's Tour

Mã đi tuần hay hành trình của quân mã là bài toán về việc di chuyển một quân mã trên bàn cờ vua. Quân mã được đặt ở một ô trên một bàn cờ trống, nó phải di chuyển theo quy tắc của cờ vua để đi qua mỗi ô trên bàn cờ đúng một lần.

