

## CHAPTER 02

# ĐỆ QUY – QUAY LUI

Design by Minh An

Email: anvanminh.hau@gmail.com

### Nội dung

- ❖ Bài toán liệt kê
- ❖ Một số kiến thức về đại số tổ hợp
- ❖ Phương pháp sinh
- ❖ Đệ quy
- ❖ **Quay lui**
- ❖ Một số bài tập

Design by Minh An

## Quay lui

- **Khái niệm**
- **Phương pháp**
- **Thuật toán quay lui**
- **Một số bài toán**
  - Liệt kê hoán vị
  - Mã đi tuần
  - Tám hậu

Design by Minh An

## Khái niệm

- Người đầu tiên đưa thuật ngữ này là nhà toán học người Mỹ D. H. Lehmer vào những năm 1950s.
- Quay lui – Backtracking: là chiến lược tìm kiếm lời giải cho các bài toán thỏa mãn ràng buộc.
- Nghiệm của bài toán là một hoặc một tập cấu hình thỏa mãn đồng thời 2 tính chất P và Q.
  - P: Cách xác định một cấu hình
  - Q: Tính dừng của bài toán
- Cấu hình là tập  $x = (x_1, x_2, \dots, x_n)$  với  $x_i$  thuộc tập S cho trước.

Design by Minh An

## Phương pháp

- Giả sử  $x = (x_1, x_2, \dots, x_n)$  là cấu hình cần tìm.
- Hiện tại đã tìm được  $k - 1$  phần tử của  $x$ , là  $x_1, x_2, \dots, x_{k-1}$
- Tìm phần tử thứ  $k$  bằng cách duyệt hết tất cả các khả năng  $i \in S$  có thể có của  $x_k$ .
- Với mỗi khả năng  $i$ , kiểm tra xem  $i$  có chấp nhận được không (thỏa mãn  $P$ ). Có hai khả năng:
  - Nếu  $i$  chấp nhận được ( $v_k = i$  thỏa  $P$ ), thì kiểm tra  $Q$ .
    - Nếu thỏa  $Q$  (đk dừng) thì dừng tìm kiếm và xuất kết quả.
    - Ngược lại thì tiếp tục tìm  $v_{k+1}$ .
  - Nếu không có  $i$  nào mà  $v_k = i$  thỏa  $P$  (ngõ cụt) thì quay lui lại bước xác định  $v_{k-1}$ .

Design by Minh An

## Thuật toán quay lui

```
Try(k) {  
    for (i ∈ S) {  
        if (i chấp nhận được) {  
            x[k] = i;  
            if (Thành công) { // tìm được phần tử cuối của cấu hình  
                Thông báo kết quả;  
            }  
        }  
        else {  
            Ghi nhận việc chọn i cho x[k];  
            Try(k + 1);  
            Hủy ghi nhận việc chọn i cho x[k]  
            nếu cần  
        }  
    }  
}
```

Design by Minh An

## Bài toán: Liệt kê hoán vị

- $S = \{1, 2, \dots, n\}$  tập giá trị của  $n$  số tự nhiên  $1, 2, \dots, n$
- $x[1..n] = (x[1], x[2], \dots, x[n])$  là một hoán vị / cấu hình
- Try( $k$ ): Tìm thành phần thứ  $k$  của hoán vị là  $x[k]$
- $i \in S$  chấp nhận được nếu  $i$  chưa được chọn trước đó
- Thực hiện bước chọn, gán  $x[k] = i$ ; đánh dấu  $i$  đã được chọn cho  $x[k]$ .
- Thành công: Chọn được phần tử  $x[k]$  với  $k = n$
- Thông báo kết quả: Hiển thị cấu hình  $x[1..n]$
- Hủy chọn: Đánh dấu  $i$  chưa được chọn cho  $x[k]$

Design by Minh An

## Liệt kê hoán vị: Thuật toán quay lui

```
void Try(k) {
    for (int i = 1; i <= n; i++) {
        if (dd[i] == 0) { // i chưa được chọn trước đó
            x[k] = i;
            if (k == n) { // tìm được phần tử cuối
                show(x);
            }
            else {
                dd[i] = 1; // Ghi nhận việc chọn i cho x[k];
                Try(k + 1);
                dd[i] = 0; // Hủy ghi nhận việc chọn i cho x[k]
            }
        }
    }
}
```

Design by Minh An

## Bài toán: Liệt kê các dãy nhị phân độ dài n

- $S = \{0, 1\}$  tập giá trị gồm 2 mã nhị phân
- $x[1..n] = (x[1], x[2], \dots, x[n])$  là một cấu hình
- Try(k): Tìm thành phần thứ k của cấu hình là  $x[k]$
- Thực hiện bước chọn, gán  $x[k] = i$ ;
- Thành công: Chọn được phần tử  $x[k]$  với  $k = n$
- Thông báo kết quả: Hiển thị cấu hình  $x[1..n]$

Design by Minh An

## Liệt kê các dãy nhị phân độ dài n

```
void Try(k) {
    for (int i = 0; i <= 1; i++) {
        x[k] = i;
        if (k == n) { // tìm được phần tử cuối
            show(x);
        }
        else {
            Try(k + 1);
        }
    }
}
```

Design by Minh An

## Bài toán: Mã đi tuần

- **Bài toán**

- **Input:**

- Bàn cờ (vua)  $n \times n$  ô
    - Vị trí xuất phát của quân mã
    - Luật đi của quân mã trên bàn cờ


- **Output:**

- Tìm một đường đi để sau  $n^2 - 1$  bước đi thì quân mã thăm kín bàn cờ (không đi ô nào quá 1 lần).

Design by Minh An

## Mã đi tuần

Start				



Start	6	15	10	21
14	9	20	5	16
19	2	7	22	11
8	13	24	17	4
25	18	3	12	23

- **Ý tưởng**

- Vết càn mọi khả năng có thể đi của quân mã
  - Quay lui: lần ngược; thử sai.
  - Độ qui.

Design by Minh An

## Mã đi tuần

```
void Try(k) {
    Khởi động việc chọn nước đi;
    do{
        Chọn nước đi tiếp từ danh sách ứng cử;
        if (chấp nhận được) {
            Ghi nhớ nước đi;
            if (bàn cờ chưa kín){
                Try(k + 1);
                if (không thành)
                    Xóa ghi nhớ trước;
            }
        }
    }while ((không còn nước đi) OR (không còn
    nước đi trong danh sách ứng cử));
}
```

Design by Minh An

## Bài toán: Tám Hậu

- **Bài toán**

- **Input:**

- Bàn cờ (vua) 8 x 8 ô
    - 8 quân hậu

- **Output:**

- Đặt 8 quân hậu vào 8 ô khác nhau trên bàn cờ sao cho trên mỗi hàng, mỗi cột, mỗi đường chéo chỉ có 1 quân hậu (các quân hậu không ăn nhau).

Design by Minh An

## Tám Hậu

	1	2	3	4	5	6	7	8
1	X							
2							X	
3					X			
4								X
5		X						
6				X				
7						X		
8			X					

- Ý tưởng: Vét cạn, quay lui, đệ quy.

Design by Minh An

## Tám Hậu

```
void Try(int k) {  
    Khởi động chọn vị trí cho hậu thứ k;  
    do{  
        Thực hiện việc chọn vị trí cho hậu thứ k;  
        if (an toàn) {  
            Đặt hậu vào vị trí;  
            if (k < 8) {  
                Try(k+1);  
            }  
            if (không thành)  
                Cắt hậu;  
        }  
    }  
} while ((chưa thành công) &&(chưa hết chỗ  
đặt hậu));  
}
```

Design by Minh An