

BÀI 02

CHIẾN LƯỢC ĐỆ QUY – QUAY LUI – CHIA ĐỂ TRỊ

Design by Minh An

Email: anvanminh.hau@gmail.com

1

Nội dung

- ❖ Phương pháp sinh
- ❖ Giải thuật đệ quy
- ❖ **Chiến lược quay lui**
- ❖ Chiến lược chia để trị
- ❖ Bài tập

Design by Minh An

2

3. Chiến lược quay lui

- Giới thiệu chiến lược quay lui
- Phương pháp quay lui
- Thuật toán quay lui tổng quát
- Giải quyết bài toán bằng chiến lược quay lui

Design by Minh An

3

3.1. Giới thiệu chiến lược quay lui

- 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

4

3.2. Phương pháp quay lui

- ❖ Giả sử $x = \{x_1, x_2, \dots, x_n\}$ là một cấu hình cần tìm, các phần tử của x được tìm trong tập S cho trước.
- ❖ Phương pháp quay lui cho phép lần lượt tìm được các phần tử của cấu hình x .
 - Giả sử đã tìm được các phần tử x_1, x_2, \dots, x_{k-1} của cấu hình x .
 - Tìm phần tử thứ k của x là x_k bằng cách duyệt mọi khả năng $i \in S$.
 - Với mỗi khả năng i , kiểm tra xem i có chấp nhận được cho x_k không:
 - Nếu chấp nhận được ($x_k = i$ thỏa P), thì kiểm tra Q :
 - Nếu thỏa Q thì dừng tìm kiếm và xuất kết quả.
 - Ngược lại thì tiếp tục tìm x_{k+1} .
 - Nếu không có khả năng $i \in S$ nào chấp nhận được cho x_k thì quay lui lại bước tìm x_{k-1} .

Design by Minh An

5

3.3. Thuật toán quay lui tổng quát

```
Try(k) {  
  for ( $i \in S$ ) {  
    if ( $i$  chấp nhận được) {  
       $x[k] = i$ ;  
      if (Thành công){ //tìm được phần tử cuối của cấu hình  
        Xuất 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

6

3.4. Giải quyết bài toán bằng chiến lược quay lui

❖ 3.4.1. 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

7

Bài toán 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

8

3.4.2. 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

9

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

10

3.4.3. 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

11

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.
 - Đệ qui.

Design by Minh An

12

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

13

3.4.4. 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

14

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

15

Tám Hậu

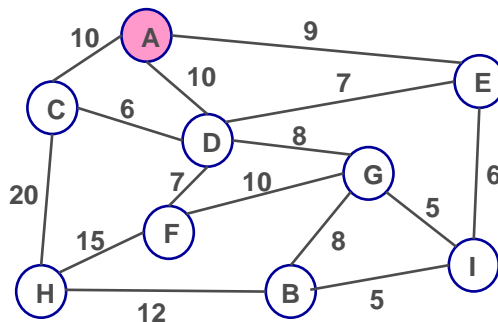
```
void Try(int k) {
    Khởi động chọn vị trí cho hậu thứ k;
    do{
        Thực hiện việc chọn vị trí cho hậu thứ k;
        if (an toàn) {
            Đặt hậu vào vị trí;
            if (k < 8) {
                Try(k+1);
                if (không thành)
                    Cắt hậu;
           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

Design by Minh An

16

3.4.5. Bài toán người du lịch – Traveling Sales Man

Cho một bản đồ du lịch như hình vẽ bên dưới



Xuất phát từ điểm du lịch A, 1 người muốn đi thăm tất cả các thành phố như trong bản đồ, mỗi thành phố 1 lần và quay về thành phố xuất phát A sao cho chi phí là thấp nhất.

Hãy tìm một lộ trình thỏa mãn yêu cầu.

Design by Minh An