

Đệ Quy Quay Lui Nhánh Cận

Trainer: Thien Nguyen
16/09/2012





Tổng quan

Đệ quy (Recursion)

Quay Lui (Backtracking)

Nhánh Cận (Branch-and-Bound)



1. Độ quy

Độ quy là gì?

Cấu trúc

Chương trình con đệ quy



1. Định quy

Định quy là gì?

Một khái niệm X được định nghĩa theo định quy nếu trong định nghĩa X có sử dụng ngay chính khái niệm X.

VD:

+ Bố mẹ tôi là *tổ tiên* của tôi.

Bố mẹ của *tổ tiên* tôi cũng là *tổ tiên* của tôi.



1. **Đệ quy**

Cấu trúc:

Một khái niệm đệ quy căn bản gồm hai phần.

+ Phần cơ sở: Định nghĩa với trường hợp đơn giản nhất, không gọi lại chính nó.

+ Phần đệ quy: Định nghĩa các trường hợp còn lại, và gọi lại chính khái niệm đang định nghĩa.



1. Độ quy

Cấu trúc:

VD:

+ 0 là số tự nhiên.

+ n là số tự nhiên nếu $n-1$ là số tự nhiên.



1. **Đệ quy**

Chương trình con đệ quy:

Một chương trình con đệ quy căn bản gồm hai phần.

- + Phần cơ sở: thực hiện các thao tác với *đối số cơ bản* và không gọi lại chính nó.
- + Phần đệ quy: thực hiện các câu lệnh mà trong đó có ít nhất một lần gọi lại chính nó với *đối số đơn giản hơn*.

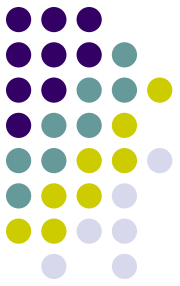


1. Đệ quy

Chương trình con đệ quy:

VD:

```
int giaithua(int n)
{
    if (n == 0)
        return 1;
    else
        return n * giaithua(n-1);
}
```

2. Quay Lui:

Khái niệm

Bản Chất

Phương pháp

Mã giả



2. Quay Lui:

Khái niệm:

Quay lui ([tiếng Anh: backtracking](#)) là một chiến lược tìm kiếm lời giải cho các [bài toán thỏa mãn ràng buộc](#).

Người đầu tiên đề ra thuật ngữ này (*backtrack*) là [nhà toán học](#) người [Mỹ D. H. Lehmer](#) vào những năm 1950.

-Wikipedia-



2. Quay Lui:

Khái niệm:

Quay lui là một chiến lược tìm kiếm lời giải cho các bài toán mà nghiệm của nó là một hay 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 một tập $v = (v_1, v_2, \dots, v_n)$, với v_i thuộc tập D cho trước.



2. Quay Lui:

Khái niệm:

VD:

Liệt kê tất cả các hoán vị của tập gồm n số tự nguyên dương đầu tiên theo thứ tự từ điển.

$N = 3$:

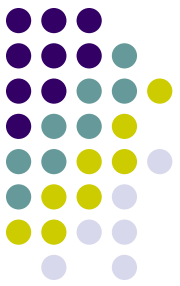
123, 132, 213, 231, 312, 321



2. Quay Lui:

Bản chất:

Bản chất của Quay lui là một quá trình tìm kiếm theo chiều sâu (Depth-First Search).



2. Quay Lui:

Phương pháp:

Giả sử $v = (v_1, v_2, \dots, v_n)$ là cấu hình cần tìm, hiện tại đã tìm được $k-1$ phần tử của v là v_1, v_2, \dots, v_{k-1} .

Ta tìm phần tử thứ k bằng cách duyệt hết tất cả các khả năng $i \in D$ có thể của v_k , với mỗi khả năng i kiểm tra xem có thể chấp nhận được không (thỏa mãn P). **Có 2 khả năng:**



2. Quay Lui:

Phương pháp (t.t):

... Kiểm tra v_k thỏa P. **Có 2 khả năng:**

+ Nếu v_k thỏa P, kiểm tra Q. Nếu thỏa Q (đk dừng) thì ta dừng tìm kiếm và xuất kết quả. Ngược lại tiếp tục tìm v_{k+1} .

+ Nếu $\nexists i \in D$ sao cho $v_{k+1} = i$ thỏa P (ngõ cụt), ta quay lại bước xác định v_{k-1} .



2. Quay Lui:

Mã giả:

```
Try(k){  
    For ([mỗi phương án chọn  $i \in D$  ])  
        If ([Chấp nhận i]){  
            [Chọn i cho  $v_k$ ];  
            If ([Thành công]) [Thông báo kq];  
            else Try(k+1);  
            [Hủy chọn i cho  $v_k$ ];  
        }  
    }  
}
```

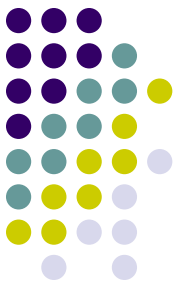



2. Quay lui

VD1: Liệt kê các hoán vị của các số tự nhiên $1..N$

Xây dựng các khái niệm trong giải thuật

- **Try(k):** Tìm thành phần thứ k của hoán vị
- Tập giá trị của từng phần tử: $D = \{1, 2, \dots, N\}$
- **Chấp nhận được i:** Khi i chưa được chọn trước đó.
- **Thực hiện bước chọn:** đánh dấu i đã chọn cho v_k .
- **Thành công:** khi chọn được thành phần thứ $k = N$
- **Thông báo kết quả:** Hiển thị N số của hoán vị
- **Hủy chọn:** đánh dấu i chưa được chọn.



2. Quay lui

VD2: Liệt kê các cách xếp N quân hậu lên bàn cờ $N \times N$ sao cho không có hai quân hậu nào ăn nhau.

Xây dựng các khái niệm trong giải thuật

- **Try(k)**: Tìm vị trí dòng đặt quân hậu ở cột thứ k
- Phương án chọn: $i = 1, \dots, N$
- **Chấp nhận được i** : Khi i được chọn trước vào ô (i, j) không cùng nằm trên một đường chéo với bất kì ô nào đã chọn trước đó.
- **Thực hiện bước chọn**: đánh dấu i đã chọn và cột, hàng, đường chéo chứa nó đã đặt quân hậu.
- **Thành công**: khi chọn được thành phần thứ $k = N$
- **Thông báo kết quả**: Hiển thị số dòng theo thứ tự cột tăng dần
- **Hủy chọn**: đánh dấu i chưa được chọn.



Nhánh Cận

Nhánh Cận là gì?

Phương Pháp

Một số ví dụ

Mã giả



Nhánh Cận

Nhánh Cận là gì?

Nhánh cận trong Quay lui:

- + là một kỹ thuật đánh giá việc tiếp tục đào sâu có tạo ra cấu hình tốt hơn cấu hình tốt nhất mà ta lưu trữ hay không.
- + Nhờ có Nhánh cận mà ta có thể đưa ra quyết định quay lui sớm hơn thuật toán backtracking cổ điển.



Nhánh Cận

Phương Pháp

Từ thuật toán backtracking cổ điển, khi xác định điều kiện P (điều kiện xác định cấu hình đề cử), ta sử dụng thêm một hàm đánh giá $f(v_1, v_2, \dots, v_{k-1})$ để xác định việc đi tiếp có hy vọng tìm ra lời giải hay không.



Nhánh Cận

VD: Bài toán người giao hàng

- Một người cần phải giao hàng tại N thành phố T_1, T_2, \dots, T_n
- C_{ij} : chi phí đi từ thành phố T_i đến thành phố T_j
($i=1,2,\dots,N$; $j = 1,2,\dots,N$)
- **Yêu cầu**: xác định hành trình thỏa mãn
 - + Đi qua tất cả các thành phố, mỗi thành phố qua đúng 1 lần, rồi quay trở lại thành phố xuất phát.
 - + Chi phí nhỏ nhất



Nhánh Cận

VD: Bài toán người giao hàng

Nhánh cận:

- Lưu 1 cấu hình BEST_CONFIG
- Đặt $C_{min} = \min\{C_{ij} : i, j = \{1, \dots, n\}\}$
- Giả sử đã đi đoạn đường $T_1 \rightarrow T_2 \rightarrow \dots \rightarrow T_i$ với chi phí:
$$S_i = C_{1,x_2} + C_{x_2,x_3} + \dots + C_{x_{i-1},x_i}$$
- Số thành phố chưa đi qua: $(n-i+1)$ thành phố.
- Như vậy, để đi tiếp ta sẽ tốn chi phí $C_{remain} > C_{min} * (n-i+1)$
- Hàm cận: $f(x_1=1, \dots, x_i) = S_i + (n-i+1)C_{min}$



Nhánh Cận

Mã giả

```
Try(k){  
  For ([mỗi phương án chọn  $i \in D$  ])  
    If ([Chấp nhận i]){  
      [Chọn i cho  $v_k$ ];  
      if (Còn hy vọng tìm ra c.hình tốt hơn BEST_CONFIG)  
      {  
        If ([Thành công]) [Thông báo kq];  
        else Try(k+1);  
        [Hủy chọn i cho  $v_k$ ];  
      }  
    }  
}
```


***Cám ơn các bạn đã
chú ý lắng nghe***

Trainer: Thien Nguyen

