ĐỆ QUY VÀ ĐÁNH GIÁ

Phạm Thế Bảo
Khoa Toán – Tin học
Trường Đại học Khoa học Tự nhiên Tp.HCM

Thuật toán đệ quy

- Là mở rộng cơ bản nhất của khái niệm thuật toán.
- Tư tưởng giải bài toán bằng đệ quy là đưa bài toán hiện tại về một bài toán *cùng loại, cùng tính chất* (đồng dạng) nhưng ở *cấp độ thấp hơn*, quá trình này tiếp tục cho đến khi bài toán được đưa về một cấp độ mà tại đó có thể giải được. Từ cấp độ này ta *lần ngược* để giải các bài toán ở cấp độ cao hơn cho đến khi giải xong bài toán ban đầu.
- Ví dụ:
  - định nghĩa giai thừa: n!=n*(n-1)! với 0!=1
  - Dãy Fibonacci: $f_0=1$, $f_1=1$ và $f_n = f_{n-1}+f_{n-2}$ $\forall n>1$
  - Danh sách liên kết.
  - ...

- Mọi thuật toán đệ quy gồm 02 phần:
  - Phần cơ sở:

    Là các trường hợp không cần thực hiện lại thuật toán (không yêu cầu gọi đệ quy). Nếu thuật toán đệ quy không có phần này thì sẽ bị lặp vô hạn và sinh lỗi khi thực hiện. Đôi lúc gọi là *trường hợp dừng*.
  - Phần đệ quy:

    Là phần trong thuật toán có yêu cầu gọi đệ quy, yêu cầu thực hiện thuật toán ở một cấp độ thấp hơn.

Các loại đệ quy

Có 03 loại đệ quy:

1. Đệ quy đuôi:

   Là loại đệ quy mà trong một cấp đệ quy chỉ có duy nhất một lời gọi đệ quy xuống cấp thấp.

   Ví dụ:

   i. Tính giai thừa

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

ii. Tìm kiếm nhị phân

```
int searchBinary(int left,int right, intx){
    if(left<right){
        int mid=(left+right)/2;
        if(x==A[i])return i;
        if(x<A[i])return searchBinary(left,mid-1,x);
        return searchBinary(mid+1,right,x);
    }
    return -1;
}
```

iii. Phân tích một số nguyên ra thừa số nguyên tố (Bài tập)

2. Đệ quy nhánh

   Là dạng đệ quy mà trong quá trình đệ quy, lời gọi được thực hiện nhiều lần.

   Ví dụ:

   i. Tháp Hà nội.
   ii. Liệt kê tất cả hoán vị của n phần tử khác nhau.

      Thuật toán:

      Xét tất cả các phần tử a_i với i=1..n
      Bỏ phần tử a_i ra khỏi dãy số
      Ghi nhận đã lấy ra phần tử a_i
      Hoán vị (Dãy số)
      Đưa phần tử a_i vào lại dãy số
      Nếu (Dãy số) rỗng thì thứ tự các phần tử được lấy ra chính là một hoán vị

   iii. Bài toán tô màu (floodfill)

3. Đệ quy hỗ tương

   Là dạng đệ quy mà trong đó việc gọi có xoay vòng, như A gọi B, B gọi C, và C gọi A. Đây là trường hợp rất phức tạp.

   Ví dụ:

   i. Đường Hilbert
   ii. Đường Sierpinski

Các phương pháp khử đệ quy

1. Vòng lặp
2. Bằng stack

Thành lập phương trình đệ quy

- Phương trình đệ quy là một phương trình biểu diễn mối quan hệ giữa T(n) và T(k). Với T(n) là "thời gian" thực hiện chương trình với kích thước dữ liệu nhập là n, T(k) là "thời gian" thực hiện chương trình với kích thước dữ liệu nhập là k, k<n. Dựa trên chương trình đệ quy ta sẽ thành lập phương trình đệ quy.
- Dạng tổng quát của phương trình đệ quy:

$$T(n) = \begin{cases} C(n) \\ F(T(k)) + d(n) \end{cases}$$

- C(n) "thời gian" thực hiện chương trình ứng với trường hợp đệ quy dừng.
- F(T(k)) hàm xác định thời gian theo T(k).
- d(n) thừa số hằng

Ví dụ: phương trình đệ quy của bài toán giai thừa.

- Gọi T(n) là "thời gian" tính n giai thừa thì T(n-1) là "thời gian" tính n-1 giai thừa.
- Trong trường hợp n=0 thì chỉ có 01 lệnh gán nên tốn O(1) → T(1)=C_1.
- Trong trường hợp n>0, phải gọi đệ quy giaiThua(n-1) nên tốn T(n-1), sau khi có kết quả phải nhân kết quả với n và gán lại vào giaiThua. Thời gian để thực hiện phép nhân và gán là hằng C_2.

Vậy ta có

$$T(n) = \begin{cases} C_1 & \text{nếu n=0} \\ T(n-1) + C_2 & \text{nếu n>0} \end{cases}$$

Ví dụ: Phương pháp MergeSort

Chia dãy ban đầu thành 2 dãy gần bằng nhau.

Chia đến khi nào chỉ còn một phần tử thì dừng chia.

Trộn các dãy lại thành dãy hoàn chỉnh được sắp xếp.

Lý luận tương tự ta có:

$$T(n) = \begin{cases} C_1 & \text{nếu n=1} \\ 2T(\frac{n}{2}) + nC_2 & \text{nếu n>1} \end{cases}$$

Giải phương trình đệ quy

1. Phương pháp truy hồi
2. Đoán nghiệm
3. Lời giải tổng quát của một lớp các phương trình đệ quy
4. Phương pháp hàm sinh

Phương pháp truy hồi

- Thay thế các giá trị trong phương trình để suy ra T(n).

Ví dụ: giải phương trình

$$T(n) = \begin{cases} C_1 & \text{nếu } n=0 \\ T(n-1) + C_2 & \text{nếu } n>0 \end{cases}$$

Ta có

$T(n) = T(n-1) + C_2$

$= [T(n-2) + C_2] + C_2 = T(n-2) + 2C_2$

$= [T(n-3) + C_2] + 2C_2 = T(n-3) + 3C_2$

...

$T(n) = T(n-i) + iC_2$

Quá trình kết thúc khi n-i=0 hay i=n. Khi đó

$T(n) = T(0) + nC_2 = C_1 + nC_2 = O(n)$

Ví dụ: giải phương trình

$$T(n)=\begin{cases} C_1 & \text{nếu n=1} \\ 2T(\frac{n}{2})+nC_2 & \text{nếu n>1} \end{cases}$$

Có

$$\begin{aligned} T(n) &= 2T\left(\frac{n}{2}\right)+nC_2 \\ &= 2\left[2T\left(\frac{n}{4}\right)+\frac{n}{2}C_2\right]+nC_2 = 4T\left(\frac{n}{4}\right)+2nC_2 \\ &= 4\left[2T\left(\frac{n}{8}\right)+\frac{n}{4}C_2\right]+2nC_2 = 8T\left(\frac{n}{8}\right)+3nC_2 \\ &\ldots \\ T(n) &= 2^iT\left(\frac{n}{2^i}\right)+inC_2 \end{aligned}$$

quá trình dừng khi $\frac{\text{n}}{2^{\text{i}}}=1$ hay i=logn

$\Rightarrow$ T(n)=nT(1)+nC_2logn

=$\text{nC}_1 + nC_2$logn

=O(nlogn)

Bài tập

Giải các phương trình đệ quy sau với T(1)=1:

1. T(n)=3T(n/2)+n
2. T(n)=4T(n/3)+n
3. T(n)=T(n/2)+1
4. T(n)=2T(n/2)+logn
5. T(n)=2T(n/2)+n