CHIA ĐỂ TRỊ
DIVIDE AND CONQUER

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

Nội dung

- Kỹ thuật quan trọng, được áp dụng rộng rãi để thiết kế các giải thuật có hiệu quả.
- Để giải quyết một bài toán kích thước n, ta chia bài toán này thành một số bài toán con có kích thước nhỏ hơn. Giải các bài toán con này rồi tổng hợp kết quả lại để được lời giải ban đầu.
- Những bài toán con này cũng có thể được chia thành các bài toán có kích thước nhỏ hơn nữa để giải quyết. Quá trình này sẽ đưa đến những bài toán mà lời giải là hiển nhiên hay dễ dàng thực hiện. Ta gọi những bài toán này là **bài toán cơ sở**.

Một số bài toán tiêu biểu

- MergeSort và QuickSort
- Nhân số nguyên lớn
- Xếp lịch thi đấu thể thao
- Bài toán con cân bằng
- …

MergeSort và QuickSort

- Chia tập dữ liệu làm 2 tập con, quá trình chia đến khi chỉ còn 01 phần tử → dừng (bài toán cơ sở), tổng hợp 2 tập con bằng cách trộn có thứ tự → tập dữ liệu được sắp xếp.
- Giống MergeSort nhưng cần phần tử cầm canh đứng giữa để chia thành 2 tập con: một tập sẽ có các phần tử có giá trị nhỏ hơn hay bằng, tập còn lại sẽ có các phần tử có giá trị lớn hơn.

Bài toán nhân hai số nguyên lớn

- Trong các ngôn ngữ lập trình, kiểu dữ liệu số nguyên đều có miền giá trị hạn chế,
  ví dụ: Pascal, C số nguyên từ -32768 đến 32767
- Khi gặp ứng dụng cần số nguyên lớn hơn (hàng chục hay hàng trăm chữ số), chúng ta phải đi xây dựng cấu trúc dữ liệu số nguyên lớn. Các thao tác đi kèm là: cộng, trừ, nhân, …
- Chúng ta xem xét cách nhân 02 số nguyên lớn có n chữ số sao cho hiệu quả.

- Nếu chúng ta dùng cách nhân thông thường, nghĩa là từng chữ số nhân với nhau rồi cộng lại thì chi phí là $O(n^2)$.
- Áp dụng kỹ thuật chia để trị. Ta chia 02 số nguyên X, Y thành các số nguyên lớn có n/2 chữ số: $X=A10^{n/2}+B$ và $Y=C10^{n/2}+D$
  Ví dụ: A=1234 thì $A=12x10^2 +34$
  Khi đó $X.Y = AC10^n+(AD+BC)10^{n/2}+BD$. Giống như trên ta lại chia tiếp tục để có bài toán cơ sở dễ dàng thực hiện.

- Theo cách làm trên thì phải thực hiện 4 phép nhân các số nguyên lớn n/2 chữ số (AC, AD, BC, BD), sau đó dùng 3 phép cộng các số nguyên lớn n chữ số và 2 phép nhan với 10^n và $10^{n/2}$ để tổng hợp.
- Phép cộng số nguyên lớn cần O(n), phép nhân 10^n có thể thực hiện đơn giản bằng cách thêm n chữ số 0 → cũng cần O(n). Gọi T(n) là thời gian nhân hai số nguyên lớn, ta có phương trình đệ quy:

- Giải phương trình ta có T(n) =

  → không cải thiện!
- Viết lại:

$$X.Y = AC10^n+[(A-B)(D-C)+AC+BD]10^{n/2}+BD$$

Công thức này chỉ cần tính 3 phép nhân của các số nguyên lớn n/2 chữ số: AC,BD và (A-B)(D-C), 6 phép cộng trừ và 2 phép nhân với 10^n. Lập luận tương tự ta có phương trình đệ quy:

T(1)=1 → Nghiệm?

T(n)=

Nghiệm của phương trình T(n)=
→ cải thiện hơn.

Thuật giải thô:

```
longDigit multi2Integer(longDigit X, longDogit Y, int n){
    if(n=1) then return X*Y;
    A=left(X,n/2);
    B=right(X,n/2);
    C=left(Y,n/2);
    D=right(Y,n/2);
    m1=multi2Integer(A,C,n/2);
    m2=multi2Integer(A-B,D-C,n/2);
    m3=multi2Integer(B,D,n/2);
    return (m1*10^n +(m1+m2+m3)*10^(n/2) +m3);
}
```


Xếp lịch thi đấu thể thao

- Xét việc xếp lịch thi đấu vòng tròn một lượt cho n đội đá banh. Mỗi đội thi đấu với nhau, mỗi đội chỉ đấu nhiều nhất một trận một ngày. Làm sao ta xếp lịch thi đấu cho số ngày ít nhất.
- Ta có tổng số trận đấu của toàn giải là

  nếu n chẵn thì ta có thể sắp n/2 cặp thi đấu trong một ngày → cần ít nhất (n-1) ngày. Nếu n lẻ thì ta có thể sắp (n-1)/2 cặp thi đấu trong một ngày → cần ít nhất n ngày

- Lịch thi đấu là một bảng n dòng và n-1 cột và được đánh số tứ 1 trở đi, dòng i đại diện cho đội thứ i và cột j đại diện cho ngày thi đấu j, ô(i,j) ghi đội phải thi đấu với đội i trong ngày j.
- Dùng chiến lược chia để trị: để sắp lịch cho n đội, ta sắp cho n/2 đội, để sắp lịch cho n/2 đội ta sắp lịch cho n/4 đội, … → sắp lịch thi đấu cho 2 đội (bài toán cơ sở).
- Từ lịch thi đấu của 2 đội, chúng ta sắp lịch thi đấu cho 4 đội như sau:
  - Lịch thi đấu cho 4 đội là một bảng 4 dòng 3 cột.
  - Lịch thi đấu cho 2 đội 1 và 2 trong ngày 1 chính là lịch thi đấu của 2 đội (bài toán cơ sở). Vậy ô(1,1)=2, ô(2,1)=1. Tương tự cũng có lịch thi đấu cho 2 đội 3 và 4 trong ngày 1: ô(3,1)=4 và ô(4,1)=3. Ta có thể thấy ô(3,1)=ô(1,1)+2 và ô(4,1)=ô(2,1)+2.
  - Lịch thi đấu của 4 đội, ta lấy góc trên bên trái của bảng lắp vào cho góc dưới bên phải và lấy góc dưới bên trái lắp cho góc trên bên phải.

	Ngày 1
Đội 1	2
Đội 2	1

	Ngày 1	Ngày 2	Ngày 3
Đội 1	2		
Đội 2	1		
Đội 3	4		
Đội 4	3		

	Ngày 1	Ngày 2	Ngày 3
Đội 1	2		4
Đội 2	1		3
Đội 3	4		2
Đội 4	3		1

	Ngày 1	Ngày 2	Ngày 3
Đội 1	2	3	4
Đội 2	1	4	3
Đội 3	4	1	2
Đội 4	3	2	1

	Ngày 1	Ngày 2	Ngày 3	Ngày 4	Ngày 5	Ngày 6	Ngày 7
Đội 1	2	3	4	5	6	7	8
Đội 2	1	4	3	6	5	8	7
Đội 3	4	1	2	7	8	5	6
Đội 4	3	2	1	8	7	6	5
Đội 5	6	7	8	1	2	3	4
Đội 6	5	8	7	2	1	4	3
Đội 7	8	5	6	3	4	1	2
Đội 8	7	6	5	4	3	2	1

Bài tập: cài đặt chương trình

Bài toán con cân bằng

- Với kỹ thuật chia để trị, nếu bài toán ban đầu được thành các bài toán con có kích thước gần bằng nhau thì hiệu suất sẽ cao hơn.
- Ví dụ: MergeSort chia làm 2 tập con bằng nhau (n/2 phần tử - có thể sai khác 1) thì độ phức tạp là O . Đối với QuickSort, nếu phân hoạch không tốt thì độ phức tạp vẫn là $O(n^2)$, trường hợp xấu nhất.