幂乘算法及应用

幂乘问题

输入： a为给定实数，n为自然数

输出： a^n

传统算法： 顺序相乘

$$a^n = (\ldots(((a\ a)a)a)\ldots)a$$

乘法次数： $\Theta(n)$

分治算法——划分

n为偶数 $\underbrace{a\,.\,.\,.\,.\,.\,.\,.\,a}_{n/2\text{个}} \mid \underbrace{a\,.\,.\,.\,.\,.\,.\,.\,a}_{n/2\text{个}}$

n为奇数 $\underbrace{a\,.\,.\,.\,.\,.\,.\,.\,a}_{(n\text{-}1)/2\text{个}} \mid \underbrace{a\,.\,.\,.\,.\,.\,.\,.\,a}_{(n\text{-}1)/2\text{个}} \mid a$

$$a^n = \begin{cases} a^{n/2} \times a^{n/2} & n \text{ 为偶数} \\ a^{(n-1)/2} \times a^{(n-1)/2} \times a & n \text{ 为奇数} \end{cases}$$

分治算法分析

以乘法作为基本运算

- 子问题规模：不超过$n/2$
- 两个规模近似 $n/2$的子问题完全一样，只要计算1次

$$W(n) = W(n/2) + \Theta(1)$$

$$W(n) = \Theta(\log n)$$

幂乘算法的应用

Fibonacci数列： 1, 1, 2, 3, 5, 8, 13, 21, …

增加 F_0=0，得到数列

0, 1, 1, 2, 3, 5, 8, 13, 21, …

问题：已知 $F_0=0, F_1=1$, 给定n, 计算 F_n.

通常算法：从 $F_0, F_1, \ldots$ 开始，根据递推公式

$$\boxed{F_n = F_{n-1} + F_{n-2}}$$

陆续相加可得 F_n，时间复杂度为$\Theta(n)$

Fibonacci数的性质

定理1 设 $\{F_n\}$为 Fibonacci 数构成的数列，那么

$$\boxed{\begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n}$$

归纳证明

$n=1$，左边= $\begin{bmatrix} F_2 & F_1 \\ F_1 & F_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ =右边

Fibonacci数的性质(续)

假设对任意正整数 n，命题成立，即

$$\begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n$$

那么 $$\begin{bmatrix} F_{n+2} & F_{n+1} \\ F_{n+1} & F_n \end{bmatrix} = \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n+1}$$

归纳假设代入

算法

令矩阵 $M=\begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix}$, 用乘幂算法计算$M^n$

时间复杂度：

- 矩阵乘法次数　$T(n)=\Theta(\log n)$
- 每次矩阵乘法需要做 8 次元素相乘
- 总计元素相乘次数为 $\Theta(\log n)$

小结

- 分治算法的例子——幂乘算法
- 幂乘算法的应用

  计算Fibonacci数

  通常算法$O(n)$，分治算法为$O(\log n)$