快速傅立叶变换:FFT算法

多项式求值算法

给定多项式：

$$A(x)=a_0+a_1x+\ldots+a_{n-1}x^{n-1}$$

设 x 为1 的 $2n$ 次方根，对所有的 x 计算 $A(x)$ 的值.

算法1：对每个 x 做下述运算：

依次计算每个项 $a_i\,x^i$，$i=1,\ldots,n\text{-}1$

对 $a_i\,x^i$ $(i=0,1,\ldots,n\text{-}1)$求和.

蛮力算法的时间复杂度

$$T_1(n)=O(n^3)$$

改进的求值算法

算法2： 依次对 每个 x 做下述计算

$A_1(x)=a_{n-1}$

$A_2(x) = a_{n-2}+xA_1(x) = a_{n-2}+a_{n-1}x$

$A_3(x) = a_{n-3}+xA_2(x) = a_{n-3}+a_{n-2}x+a_{n-1}x^2$

…

$A_n(x) = a_0+ x A_{n-1}(x) = A(x)$

时间复杂度

$$T_2(n) = O(n^2)$$

偶系数与奇系数多项式

$n=4$

$$A(x)=a_0+a_1x+a_2x^2+a_3x^3$$

$$A_{even}(x) = a_0+a_2x$$

$$A_{odd}(x) = a_1+a_3x$$

$$A(x) = A_{even}(x^2) + xA_{odd}(x^2)$$

$$= a_0+a_2x^2 + x(a_1+a_3x^2)$$

分治多项式求值算法

一般公式（n为偶数）

$$A(x)= a_0+a_1x+a_2x^2+\ldots+a_{n-1}x^{n-1}$$

$$A_{\text{even}}(x)=a_0+a_2x+a_4x^2+\ldots+a_{n-2}x^{(n-2)/2}$$

$$A_{\text{odd}}(x)=a_1+a_3x+a_5x^2+\ldots+a_{n-1}x^{(n-2)/2}$$

$$A(x) = A_{\text{even}}(x^2) + xA_{\text{odd}}(x^2)$$

- x^2 也是1 的 $2n$ 次根
- 偶次数与奇次数多项式计算作为 $n/2$ 规模的子问题，然后利用子问题的解 $A_{\text{even}}(x^2)$ 与 $A_{\text{odd}}(x^2)$ 得到 $A(x)$

分治求值算法设计

算法 3：

1. 计算 1 的所有的 $2n$ 次根

   $1, \omega_1, \omega_2, \ldots, \omega_{2n-1}$

2. 分别计算$A_{even}(x^2)$与$A_{odd}(x^2)$

3. 利用步2 的结果计算 $A(x)$

   $A(x) = A_{even}(x^2) + x A_{odd}(x^2)$

注意：x^2不需要重新计算，所有根在单位圆间隔分布，隔一取一即可.

分治求值算法分析

$$T(n) = T_1(n) + f(n)$$

$f(n)=O(n)$ 是步 1 计算$2n$次根的时间

递归过程 $T_1(n) = 2T_1(n/2) + g(n)$

$T_1(1) = O(1)$，

$g(n) = O(n)$ 是对所有$2n$次根在步3组合解的时间

$$T_1(n)=O(n\log n)$$

$$T(n)=O(n\log n)+O(n)=O(n\log n)$$

FFT算法伪码

1. 求值$A(\omega_j)$和$B(\omega_j)$，$j=0,1,\ldots,2n-1$
2. 计算 $C(\omega_j)$，　　$j=0,1,\ldots,2n-1$
3. 构造多项式

$$D(x)=C(\omega_0)+C(\omega_1)x+\ldots+C(\omega_{2n-1})x^{2n-1}$$

4. 计算所有的 $D(\omega_j)$，$j=0,1,\ldots,2n-1$
5. 利用下式计算$C(x)$的系数 c_j，

$$D(\omega_j)=2nc_{2n-j}$$

$$j=0,1,\ldots,2n-1$$

FFT算法分析

步1: 求值 $A(\omega_j)$ 和 $B(\omega_j)$ $O(n\log n)$

步2: 计算所有的 $C(\omega_j)$ $O(n)$

步3:

步4: 计算所有的 $D(\omega_j)$ $O(n\log n)$

步5: 计算所有的 c_j $O(n)$

算法时间为 $O(n\log n)$

小结

- 多项式求值算法

  蛮力算法：$O(n^3)$

  改进的求值算法：$O(n^2)$

  FFT算法：$O(n\log n)$

- FFT算法的设计与分析