问题求解（三）第9周作业

黄奕诚161220049

October 31, 2017

30.1-2

Proof. 首先，$A(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + \cdots + a_{n-1}x^{n-1}$
有$A(x_0) = r$，即$A(x_0) = a_0 + a_1x_0^1 + a_2x_0^2 + a_3x_0^3 + \cdots + a_{n-1}x_0^{n-1} = r$
则$A(x) = q(x)(x - x_0) + r = (r - q(0)x_0) + (q(0) - q(1)x_0)x + (q(1) - q(2)x_0)x^2 + \cdots + q(n-2)x$
可进行如下构造:
设$n \times n$的矩阵A为:

$$A = \begin{bmatrix} 1 & -x_0 & 0 & \cdots & 0 \\ 0 & 1 & -x_0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$q = (r, q(0), q(1), \cdots, q(n-2))^T$$

$$a = (a(0), a(1), \cdots, a(n-1))^T$$

只要满足：$Aq = a$.因为矩阵A已经是一个上三角矩阵，且稀疏，这个方程通过向后替代的方法可以在线性时间内解出向量q，也即求出r及$q(0), q(1), \cdots, q(n-2)$ □

30.1-4

Proof. 假设$n-1$个点值对可以唯一确定一个阶数为n的多项式P_1.可以在原有的$n-1$个点值对的基础上添加一个点值对(x_{n-1}, y_{n-1})，于是根据定理30.1可知这共n个点值对可以唯一确定一个n阶多项式P_2.不妨在原$n-1$个点值对的基础上添加另一个点值对(x'_{n-1}, y'_{n-1})，它不等同于(x_{n-1}, y_{n-1})，则可以唯一确定一个n阶多项式P_3.由于两个点值对的构成不同，则有$P_2 \neq P_3$.而在添加新的点值对之前，这两个多项式应当是相同的，此时与"$n-1$个点值对可以唯一确定一个阶数为n的多项式P_1"矛盾.因此，需要n个点值对才可以唯一确定一个阶数为n的多项式. □

30.1-5

计算矩阵A的系数的过程可分为如下步骤:
(1)计算$\prod_{j \neq k}^{x - x_j} \cdot (x - x_k)$，因为共计算了$n$次，所以只需要$O(n)$的时间;
(2)我们得到了$\prod_j (x - x_j)$的系数表示，此时需要证明对于每一个$k \prod_{j-k}(x - x_j)$，都只需要$O(n)$的时间，这点由题30.1-2可直接证得.
(3)此时问题就转化为$\sum_k y_k \frac{f_k(x)}{f_k(x_k)}$.计算每一个$f_k(x)$只需要$\Theta(n)$的时间，故所有的$f_k(x)$需要$\Theta(n^2)$的时间.除以$f_k(x_k)$再乘以$y_k$需要$O(n)$的时间，因此这个过程共需要$\Theta(n^2)$的时间.
综上所述，整个计算过程的运行时间为$\Theta(n^2)$.

30.2-1

Proof.

$$\omega_n^{n/2} = (e^{2\pi i/n})^{n/2} = e^{\pi i} = -1 = e^{2\pi i/2} = \omega_2$$

□

30.2-4

见Algorithm 1

Algorithm 1 RECURSIVE-FFT-INV(a)

```
n = a.length
if n == 1 then
    return a
end if
ω_n = e^{2πi/n}
ω = 1/n
a^[0] = (a_0, a_2, ···, a_{n-2})
a^[1] = (a_1, a_3, ···, a_{n-1})
y^[0] = RECURSIVE-FFT-INV(a^[0])
y^[1] = RECURSIVE-FFT-INV(a^[1])
for k = 0 to n/2 - 1 do
    y_k = y_k^[0] + ω y_k^[1]
    y_{k+(n/2)} = y_k^[0] - ω y_k^[1]
    ω = ω ω_n
end for
return y
```

30.2-5

首先，$(\omega_n^{k+n/3})^3 = \omega_n^{3k+n} = (\omega_n^k)^3 \cdot \omega_n^n = (\omega_n^k)^3$.于是设$A^{[i]} = \sum_{j=0}^{n/3-1} a_{i+3j}x^j$，有$A(x) = A^{[0]}(x^3) + xA^{[1]}(x^3) + x^2A^{[2]}(x^3)$。运行时间即为$T(n) = 3T(n/3) + \Theta(n) = \Theta(n \lg n)$.

30.2-7

设当$i = 1, 2, \cdots, n$时，$P_{i,0}(x) = (x - z_{i-1})$，对于每一个小于等于$\frac{n}{2^k}$的正整数$i$，计算$P_{i,k} = P_{i,k-1} \cdot P_{2i,k-1}$.最终要求解的即为$P_{1,[lg(n)]+1}$，此时得到了一个多项式，$n$表示当给定$n$个为零的点时所需要的计算时间.于是

$$T(n) = 2T(n/2) + \Theta(nlgn)$$

利用主定理的方法可知运行时间满足$O(nlg^2n)$.

30.3-2

在题30.2-4，已经写出了RECURSIVE-FFT-INV的算法执行过程，这里只需要仿照FFT改进为ITERATIVE-FFT的过程，将RECURSIVE-FFT-INV的算法改进为ITERATIVE-FFT-INV即可，只需要将ITERATIVE-FFT的第7行改写为$\omega = \frac{1}{n}$即可.

30-1

a.

因为$(a+b)(c+d) = ac+ad+bc+bd$，所以$(ax+b)(cx+d) = acx^2 + (ad+bc)x + bd = acx^2 + ((a+b)(c+d) - ac - bd)x + bd$，此时只需要计算$ac, bd, (a+b)(c+d)$即可，符合题意.

b.

不妨设n是2的幂(否则将其归为最近的2的幂，这不影响平均运行时间)，设两个多项式分别为$A_1(x) = \sum_{j=0}^{n-1} a_{j,1}x^j$和$A_2(x) = \sum_{j=0}^{n-1} a_{j,2}x^j$

方法1

设$L_1(x) = \sum_{j=0}^{n/2-1} a_{j,1}x^j, H_1(x) = \sum_{j=n/2}^{n-1} a_{j,1}x^j$，第二个多项式也如此设.显然有$A_1(x) = H_1(x) \cdot x^{n/2} + L_1(x)$，于是

$$A_1(x) \cdot A_2(x) = (H_1(x)x^{n/2} + L_1(x))(H_2(x)x^{n/2} + L_2(x))$$

由第a小问的方法可知，只需要计算三种式子即可完成这一乘法，有

$$f(n) = 3f(n/2) + \Theta(n)$$

由主定理可知运行时间为$\Theta(n^{lg3})$.

方法2

构造方法为:设$O_i(x) = \sum_{j=0}^{n/2-1} a_{2j+1,i}x^j, E_1(x) = \sum_{j=0}^{n/2-1} a_{2j,i}x^j$.有$A_i(x) = xO_i(x^2) + E_i(x^2), i = 1, 2$.同理将非$x^i$式子作为参数，由$a$中方法可知运行时间为$\Theta(n^{lg3})$.

c.

对于整数乘法来说，只要把它构造为多项式乘法，再利用前几题的结论便可证明运行时间为$\Theta(n^{lg3})$.设这两个整数为$A_i = \sum_{k=0}^{lg(A_i)} a_{k,i}2^k$，则可设多项式为$f(x, i) = \sum_{k=0}^{lg(A_i)} a_{k,i}x^i$，有$f_i(2) = A_i$，于是便由上两题证得结论.