## 2-7：

> *设 $P(x)=a_0+a_1x+...+a_dx^d$ 是一个 $d$ 次多项式。假设已有一算法能在 $O(i)$ 时间内计算一个 $i$ 次多项式与一个 1 次多项式的乘积，以及一个算法能在 $O(ilogi)$ 时间内计算两个 $i$ 次多项式的乘积。对于任意给定的 $d$ 个整数 $n_1,n_2,...,n_d$ ，用分治法设计一个有效算法，计算出满足 $P(n_1)=P(n_2)=...=P(n_d)=0$ 且最高次项系数为 1 的 $d$ 次多项式 $P(x)$ ，并分析算法的效率。*

## 题目分析：

若多项式 $P(x)$ 满足 $P(n_1)=P(n_2)=...=P(n_d)=0$，则可设多项式为：

$$P(x)=(x-n_1)(x-n_2)...(x-n_d)$$

由于我们已知算法可在 $O(i)$ 时间内计算一个 $i$ 次多项式与一个 1 次多项式的乘积，以及一个算法能在 $O(ilogi)$ 时间内计算两个 $i$ 次多项式的乘积。我们只需要展开此多项式即可。为减小时间复杂度，采用分治法，将原式不断对半划分，直到变为两个多项式后进行相乘，然后依次自底向上乘，最后化为一个 $d$ 次多项式，即可得到对应的系数。

> *注：由于本题题设中假设了算法，故不进行编程实现，仅进行理论分析。*

## 算法分析：

本算法是递归式，其复杂度为：

$$T(d)=2T(\frac{d}{2})+O(\frac{d}{2}log\frac{d}{2})$$

由主定理，$f(n)=O(\frac{d}{2}log\frac{d}{2})=\Theta(n^{log_2 2}logn)$可得：

本算法的时间复杂度为：$T(d)=O(dlog^2d)$