MTHSC 985 Fall 2001 Clemson University
Fast Fourier Transforms and
sparse linear systems over finite fields Instructor: Shuhong Gao
Lecture 2, August 30 & September 4 Scribe: Jira Limbupasiriporn

2. Chinese Remainder Theorem (CRT) and Interpolation

Let us start with an ancient Chinese problem:

> *There is a certain number. When divided by three this number has remainder two; when divided by five, it has remainder three; when divided by seven, it has remainder two. What is the number?*

This problem appears in *Sunzi Suanjing* (Sunzi's Computational Canon), a mathematical book written during the period from 280 to 483 AD. (Note that the Sunzi is the same person as in *Sunzi Bingfa* (Master Sun's Art of War).) In modern notation, the problem is to find an integer x such that

$$\begin{aligned} x &\equiv 2 \pmod 3,\\ x &\equiv 3 \pmod 5,\\ x &\equiv 2 \pmod 7. \end{aligned}$$

The answer is given in *Sunzi Suanjing*, and in 1592 Dawei Cheng put it as a poem:

> *Three septuagenarians walking together, 'tis rare!*
> *Five plum trees with twenty one branches in flower,*
> *Seven disciples gathering right by the half-moon,*
> *One hundred and five taken away, lo the result shall appear!*

Note that in Chinese calendar, each month has thirty days, so half-moon means fifteen. The poem also indicates that the gathering takes place at mid August, which is a traditional Chinese festival (called Mid Autumn Festival) for family gathering and for celebrating harvest. The result is

$$x \equiv 2 \cdot 70 + 3 \cdot 21 + 2 \cdot 15 \pmod{105}.$$

The solution also indicates that if one is given three integers r_1, r_2, r_3 and is asked to find a number x such that

$$\begin{aligned} x &\equiv r_1 \pmod 3,\\ x &\equiv r_2 \pmod 5,\\ x &\equiv r_3 \pmod 7. \end{aligned}$$

Then the answer is

$$x \equiv r_1 \cdot 70 + r_2 \cdot 21 + r_3 \cdot 15 \pmod{105}.$$

In the above problem the moduli $3, 5, 7$ are pairwise coprime. The ancient Chinese also solved the problem for an arbitrary set of moduli (not necessarily pairwise coprime). The rules (or procedures) for solving any system of simultaneous congruences are given explicitly by Jiushao Qin in 1247 AD in his book *Shushu Jiuzhang* (Mathematical Treatise in Nine Chapters). Some of the rules show how to compute modular inverses and lcm of several integers. More details can be found at
http://www.math.sfu.ca/histmath/China/13thCenturyAD/QinTa.html.

In the above solution, $105 = 3 \cdot 5 \cdot 7$, and $e_1 = 70$, $e_2 = 21$, $e_3 = 15$ satisfy

$$\begin{array}{lll} e_1 \equiv 1 \pmod 3, & e_1 \equiv 0 \pmod 5, & e_1 \equiv 0 \pmod 7, \\ e_2 \equiv 0 \pmod 3, & e_2 \equiv 1 \pmod 5, & e_2 \equiv 0 \pmod 7, \\ e_3 \equiv 0 \pmod 3, & e_3 \equiv 0 \pmod 5, & e_3 \equiv 1 \pmod 7, \end{array}$$

so

$$\begin{aligned} e_1 &= 5 \cdot 7 \cdot ((5 \cdot 7)^{-1} \bmod 3), \\ e_2 &= 3 \cdot 7 \cdot ((3 \cdot 7)^{-1} \bmod 5), \\ e_3 &= 3 \cdot 5 \cdot ((3 \cdot 5)^{-1} \bmod 7). \end{aligned}$$

This shows how to find the solution in general. Also note that

$$\begin{aligned} e_i^2 &\equiv e_i \pmod{105}, \qquad 1 \le i \le 3, \\ e_i \cdot e_j &\equiv 0 \pmod{105}, \qquad i \ne j. \end{aligned}$$

That is, e_1, e_2, e_3 are orthogonal idempotents in the ring $\mathbb{Z}/(105)$.

Theorem 2.1 (Chinese Remainder Theorem for integers)**.** *Let $m_1, \ldots, m_t$ be positive integers that are pairwise coprime. Let $m = m_1 m_2 \ldots m_t$. Then for any integers $r_1, \ldots, r_t$, there is a unique integer $x \pmod m$ such that*

$$x \equiv r_i \pmod{m_i}, \qquad 1 \le i \le t. \tag{13}$$

Furthermore, any such solution x is of the form

$$x \equiv r_1 e_1 + \cdots + r_t e_t \pmod m \tag{14}$$

where

$$e_i = \frac{m}{m_i} \cdot \left(\left(\frac{m}{m_i} \right)^{-1} \bmod m_i \right), \qquad 1 \le i \le t. \tag{15}$$

Note that $\frac{m}{m_i} = m_1 \cdots m_{i-1} m_{i+1} \cdots m_t$ is coprime to m_i, as $\gcd(m_i, m_j) = 1$ for $i \ne j$, hence the inverse in (15) exists. One can check that x in (14) is indeed a solution to (13). The formula (15) gives us an efficient algorithm for solving (13).

Algorithm 1: Chinese Remainder Theorem for integers.

Input: positive integers $m_1, \ldots, m_t$, pairwise coprime, and integers $r_1, \ldots, r_t$.

Output: an integer x such that

$$\begin{aligned} &x \equiv r_i \pmod{m_i}, \quad 1 \le i \le t, \quad \text{and} \\ &0 \le x < m \quad \text{where} \quad m = m_1 \cdots m_t. \end{aligned}$$

```
1. compute $m = m_1 \cdots m_t$ and set $x = 0$.
2. for $i = 1, 2, \ldots, t$ do
       compute $y_i = \frac{m}{m_i}$,
       apply extended Euclidean algorithm to find $s_i = y_i^{-1} \bmod m_i$,
       set $c_i = r_i \cdot s_i \bmod m_i$,
       set $x = x + c_i \cdot y_i \pmod{m}$.
3. return $x$.
```

One can check that the number x computed is exactly the formula in (14). The numbers in (15) are orthogonal idempotents in the ring $\mathbb{Z}/(m)$ where $m = m_1 \cdots m_t$, that is,

$$\begin{aligned} e_i^2 &\equiv e_i \pmod{m}, \qquad 1 \le i \le t, \\ e_i \cdot e_j &\equiv 0 \pmod{m}, \qquad i \ne j. \end{aligned}$$

The formula (14) also says that $R = \mathbb{Z}/(m)$ can be decomposed into a direct sum of rings

$$R = R_1 \oplus R_2 \oplus \cdots \oplus R_t$$

where

$$R_i = Re_i = \{a \cdot e_i \pmod{m} : a \in \mathbb{Z}\} \cong \mathbb{Z}/(m_i), \ 1 \le i \le t.$$

So the Chinese Remainder Theorem tells us that

$$\mathbb{Z}/(m) \cong \mathbb{Z}/(m_1) \times \cdots \times \mathbb{Z}/(m_t). \tag{16}$$

The formula (14) gives the isomorphism map from the right side to the left in (16).

The Chinese Remainder Theorem also holds for any Euclidean domain. We state below the analogous results for the ring of univariate polynomials over a field.

Theorem 2.2 (Chinese Remainder Theorem for polynomials)**.** *Let $\mathbb{F}$ be any field, $f_1, \ldots, f_t \in \mathbb{F}[x]$ nonconstant polynomials, pairwise coprime, and let $f = f_1 \cdots f_t$. Then for any polynomials $r_1, \ldots, r_t \in \mathbb{F}[x]$, there is a unique polynomial $g \in \mathbb{F}[x] \pmod{f}$ such that*

$$g \equiv r_i \pmod{f_i}, \qquad 1 \le i \le t. \tag{17}$$

Furthermore, any such solution g is of the form

$$g \equiv r_1 e_1 + \cdots + r_t e_t \pmod{f} \tag{18}$$

where

$$e_i = \frac{f}{f_i} \cdot \left(\left(\frac{f}{f_i} \right)^{-1} \bmod f_i \right), \quad 1 \le i \le t. \tag{19}$$

This is completely analogous to the integer case. The $e_1, \ldots, e_t$ in (19) are orthogonal idempotents in the ring $\mathbb{F}[x]/(f)$, and we have

$$\mathbb{F}[x]/(f) \cong \mathbb{F}[x]/(f_1) \times \cdots \times \mathbb{F}[x]/(f_t). \tag{20}$$

Slight modification of Algorithm 1 can be used to compute g.

Algorithm 2: Chinese Remainder Theorem for polynomials.

Input: $f_1, \ldots, f_t \in \mathbb{F}[x]$, pairwise coprime, and $r_1, \ldots, r_t \in \mathbb{F}[x]$.

Output: a polynomial g such that

$$\begin{aligned} &g \equiv r_i \pmod{f_i}, \quad 1 \le i \le t, \quad \text{and} \\ &\deg g < \deg f, \quad \text{where} \quad f = f_1 \cdots f_t. \end{aligned}$$

```
1. compute $f = f_1 \cdots f_t$ and set $g = 0$.
2. for $i = 1, 2, \ldots, t$ do
       compute $y_i = \frac{f}{f_i}$,
       apply extended Euclidean algorithm to find $s_i = y_i^{-1}$ mod $f_i$,
       set $c_i = r_i \cdot s_i$ mod $f_i$,
       set $g = g + c_i \cdot y_i \pmod{m}$.
3. return $g$.
```

Special case. Let $f_i = x - a_i$, and $r_i \in \mathbb{F}$, $1 \le i \le t$, where $a_1, \ldots, a_t$ are distinct. We observe that for any $v(x) \in \mathbb{F}[x]$ and $a \in \mathbb{F}$, we have $v(x)$ mod $(x - a)$ equals $v(a)$, i.e., evaluation of $v(x)$ at $x = a$. Then from (19), we obtain

$$e_i = \prod_{j \ne i} (x - a_j) \cdot \left(\prod_{j \ne i} (a_i - a_j) \right)^{-1}, \quad 1 \le i \le t.$$

So the solution is

$$g = \sum_{i=1}^{t} r_i \cdot \frac{\prod_{j \ne i}(x - a_j)}{\prod_{j \ne i}(a_i - a_j)}.$$

This is nothing but the Lagrange interpolation formula. This g is the unique polynomial in $\mathbb{F}[x]$ of degree $< t$ such that

$$g(a_i) = r_i, \quad 1 \le i \le t.$$

Both Algorithm 1 and 2 have quadratic running time and are practical. One disadvantage is that the algorithms need to know all the moduli in advance. We present next an 'online' algorithm which finds a solution incrementally for any sequence of moduli and update it whenever a new modulus is given.

We use mixed-radius representation for integers (and polynomials). Let $m_1, \ldots, m_t$ be integers greater than one (or nonconstant polynomials) with $m = m_1 \cdots m_t$. Then an integer $0 \le x < m$ can be represented uniquely as

$$x = x_0 + x_1 m_1 + x_2 m_1 m_2 + \cdots + x_{t-1} m_1 m_2 \cdots m_{t-1} \tag{21}$$

where $0 \le x_i < m_{i+1}$ for $i = 0, 1, \ldots, t-1$.

To find an integer $0 \leq x < m$ of the form (21) satisfying (13), we compute $x_0, x_1, \ldots, x_{t-1}$ iteratively. First, let $x_0 = r_1 \bmod m_1$ so that $x \equiv r_1 \pmod{m_1}$. Suppose $x_0, x_1, \ldots, x_{i-1}$ have been found for some $i \geq 1$ so that

$$x \equiv r_j \pmod{m_j}, \quad 1 \leq j \leq i.$$

We want to find x_i so that

$$x \equiv r_{i+1} \pmod{m_{i+1}},$$

i.e.,

$$x_0 + x_1 m_1 + \cdots + x_{i-1} m_1 \cdots m_{i-1} + x_i m_1 \cdots m_i \equiv r_{i+1} \pmod{m_{i+1}}.$$

It follows that

$$x_i = \frac{r_{i+1} - (x_0 + x_1 m_1 + \cdots + x_{i-1} m_1 \cdots m_{i-1})}{m_1 \cdots m_i} \pmod{m_{i+1}}.$$

Hence x_i can be computed from $x_0, x_1, \ldots, x_{i-1}$ whenever $(m_1 \ldots m_i)^{-1} \bmod m_{i+1}$ exists.

Algorithm 3: Garner's algorithm for CRT.

Input: positive integers $m_1, \ldots, m_t$, pairwise coprime, and integers $r_1, \ldots, r_t$.

Output: an integer g such that $g \equiv r_i \pmod{m_i}$, $1 \leq i \leq t$.

```
1. set $g = r_1 \bmod m_1$, and $m = 1$.
2. for $i = 2, 3, \ldots, t$ do
       set $m = m \cdot m_{i-1}$,
       set $u = (r_i - g) \cdot m^{-1} \bmod m_i$,
       set $g = g + u \cdot m$.
3. return $g$.
```

This algorithm also works for univariate polynomials: one just needs to switch integers to polynomials in the above algorithm. We emphasize a special case for polynomials, namely, when all moduli are linear.

Algorithm 4: Interpolation for univariate polynomials.

Input: $(a_i, b_i) \in \mathbb{F}^2$, $1 \leq i \leq t$ where $a_1, \ldots, a_t$ are distinct.

Output: a polynomial $g \in \mathbb{F}[x]$ of degree $< t$ such that $g(a_i) = b_i$, $1 \leq i \leq t$.

```
1. set $g = b_1$ and $m = 1$.
2. for $i = 2, 3, \ldots, t$ do
       set $m = m \cdot (x - a_{i-1})$,
       set $u = (b_i - g(a_i))/m(a_i)$,
       set $g = g + u \cdot m$.
3. return $g$.
```

In the next lecture, we'll discuss fast methods for evaluation of polynomials.