PART II

ALGORITHMS

1. THE FAST FOURIER TRANSFORM

Overview

- The *Fast Fourier Transform* (FFT) is method for fast computation of the *Discrete Fourier Transform* (DFT)
- DFT is multipoint evaluation algorithm at roots of unity
- Polynomial *evaluation* and *interpolation* are each other's inverse operations for converting between dense coefficient and multi-value representations for polynomials
- Multi-value representation is important because it makes polynomial multiplication easy
- The same ideas make integer multiplication fast (Schönhage-Strassen)

Ring essentials

In this chapter: rings R commutative with 1.

An element $u \in R$ is a *unit* if it is invertible, so there exists $v \in R$ with $u \cdot v = 1$.

An element $z \in R$ is a zero-divisor if $z \neq 0$ and there exists non-zero $w \in R$ with $z \cdot w = 0$.

Lemma *An element in R cannot be simultaneously a unit and a zero-divisor.*

An *integral domain* is a ring without zero-divisors. A *field* is a domain in which every non-zero element is a unit. For every domain R we can construct a *field of fractions* F that contains R as a subring.

If there exists a positive integer n such that $n \cdot 1 = 0$ in R then the smallest such n is the *characteristic of* R; if it does not exist, R has characteristic 0. A field has characteristic 0, or p for some prime number p.

Polynomial essentials

If $f \in R[x]$ then $z \in R$ is a *root* of f if $f(z) = 0$. Furthermore, z is a root of *multiplicity* k if $f = (x - z)^k \cdot g$ for some $g \in R[x]$.

Theorem *Let R be an integral domain. Then the number of roots in R of $f \in R[x]$ counted with multiplicities is at most* $\deg f$.

This uses Euclidean division (see, next chapter).

A *monic* polynomial has leading coefficient 1.

Interpolation

Theorem *Suppose that* $x_0, x_1, \ldots, x_n$ *(distinct) and* $y_0, y_1, \ldots, y_n$ *are elements from a domain* R *with field of fractions* F*. Then there exists a unique polynomial of degree at most* n*, say* f*, in* $F[x]$ *with* $f(x_i) = y_i$ *for* $i = 0, 1, \ldots, n$*.*

Constructive proof: **Lagrange interpolation**. Let $f = f_0 + f_1 + \ldots + f_n$, where

$$f_i = y_i \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}.$$

Then clearly $f_i(x_i) = y_i$ and $f_i(x_j) = 0$ for $j \neq i$, so $f(x_i) = y_i$ for all i. Any g that is also of degree at most n with the same values at x_i leads to a difference $f - g$ of degree at most n with at least $n+1$ roots in F: a contradiction.

Evaluation and interpolation

A polynomial $f = a_0 + a_1 x + \cdots + a_n x^n \in F[x]$ of degree $\leq n$ can now either be represented by $n+1$ coefficients $a_0, a_1, \ldots, a_n \in F$, or by $n+1$ values $y_0, y_1, \ldots, y_n$ at prescribed points $x_0, x_1, \ldots, x_n$.

The conversion one way is that of evaluation

$$E : (a_0, a_1, \ldots, a_n)^{\mathsf{T}} \mapsto (y_0, y_1, \ldots, y_n),$$

which is given by matrix multiplication

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

The inverse map also exists (and is interpolation) according to the above; so the Vandermonde matrix must be invertible if all x_i are distinct!

Instead of over a field F we may also work again over a domain R, provided that all $x_i - x_j$ are units.

The importance of the value representation: since $(f+g)(x_i) = f(x_i)+g(x_i)$ and $(f \cdot g)(x_i) = f(x_i) \cdot g(x_i)$ addition and multiplication on polynomials in value representation can be done *componentwise*! The main bottleneck for its use then lies in the conversion from and to (standard) coefficient representation.

The Fourier transforms that we will define below can be seen as fast way to convert between the representation of f by coefficients and a representation by value vectors, using the freedom of choice for the points $x_0, x_1, \ldots, x_n$.

Roots of unity

An element $\zeta \in R$ is an *n-th root of unity* for a positive integer n, if $\zeta^n = 1$. It is a *primitive n-th root of unity* if moreover n is a unit in R and $\zeta^{n/p} - 1$ is neither zero nor a zero-divisor in R, for any prime divisor p of n.

It follows easily that $\zeta^\ell - 1$ cannot be a zero-divisor (or zero) for any ℓ with $1 < \ell < n$. Also $1 + \zeta + \zeta^2 + \cdots + \zeta^{n-1} = 0$, and the same is true for ζ^ℓ with $1 < \ell < n$.

Examples: $\zeta_n \in \mathbb{C}$ and $g^{\frac{p-1}{n}}$ in $\mathbb{F}_p^*$ when $n|p-1$. Another important example is in the ring $R = \mathbb{Z}/n\mathbb{Z}$, when n is of the particular form $n = 2^{s \cdot 2^{t-1}} + 1$. Then $\zeta = 2^s$ is a primitive 2^t-th root of unity in R, since $\zeta^{2^{t-1}} = -1 \in R$.

Discrete Fourier Transform

Suppose that $\zeta \in R$ is a primitive n-th root of unity. The *discrete Fourier transform* is the evaluation map $D : R^n \to R^n$ at $\zeta_n^0, \zeta_n^1, \ldots, \zeta_n^{n-1}$.

The discrete Fourier transform is given by multiplication by the Vandermonde matrix

$$V_{\zeta_n} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \zeta_n & \cdots & \zeta_n^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta_n^{n-1} & \cdots & \zeta_n^{(n-1)(n-1)} \end{pmatrix}.$$

If n is a unit in R then D^{-1} is

$$\frac{1}{n} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \zeta_n^{-1} & \cdots & \zeta_n^{-(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta_n^{-(n-1)} & \cdots & \zeta_n^{-(n-1)(n-1)} \end{pmatrix} = \frac{1}{n} V_{\zeta_n^{-1}}.$$

Proof $V_{\zeta_n} \cdot V_{\zeta_n^{-1}} = nI.$

Divide and conquer

If $n = 2k$ is even then $\zeta_n^k = -1$, so $\zeta_n^{k+i} = -\zeta_n^i$, for $i = 0, 1, \ldots, k-1$.

To evaluate f at ζ_n^j, for $j = 0, 1, \ldots, n$, write

$$f = f_0(x^2) + x f_1(x^2),$$

where

$$f_0 = a_0 + a_2 x + \cdots a_{n-2} x^{k-1}$$

and

$$f_1 = a_1 + a_3 x + \cdots a_{n-1} x^{k-1}$$

are the *even* and *odd parts* of f. Then

$$f(\zeta_n^i) = f_0(\zeta_n^{2i}) + \zeta_n^i f_1(\zeta_n^{2i}),$$

and

$$f(\zeta_n^{k+i}) = f_0(\zeta_n^{2k+2i}) + \zeta_n^{k+i} f_1(\zeta_n^{2k+2i}) =$$

$$= f_0(\zeta_n^{2i}) - \zeta_n^i f_1(\zeta_n^{2i}),$$

for $i = 0, 1, \ldots, k-1$.

Fast Fourier transform

This requires the evaluation of two polynomials of half the degree of f, as well as k squarings to compute ζ_n^{2i}, k multiplications for $\zeta_n^i f_1(\zeta_n^{2i})$ and k additions and subtractions in R. Using this idea recursively we find the following.

Theorem *Let $n = 2^t$, and $\zeta_n \in R$. Then we can evaluate $f \in R[x]$ of degree $n-1$ at the points ζ_n^j for $j = 0, \ldots, n-1$ in $O(n \log n)$ arithmetic operations in R.*

Proof Show by induction that the number of arithmetic steps $A(t)$ satisfies $A(t) \leq t2^{t+1}$.

Corollary *Let $n = 2^t$, and $\zeta_n \in R$. Then we can interpolate $f \in R[x]$ of degree $n-1$ at the points ζ_n^j for $j = 0, \ldots, n-1$ in $O(n \log n)$ arithmetic operations in R.*

Example

The fast Fourier transform in a simple example. Let $R = \mathbb{F}_{17}$. Since $2^4 = 16 \equiv -1 \bmod 17$, the element 2 is a primitive 8-th root of unity in $\mathbb{F}_{17}$. That means that we can multiply polynomials using the fast Fourier transform as long as the product has degree at most 7.

To multiply, for example, $f = 1 + 7x + 3x^2$ and $g = -2 + 3x^2 + 5x^4$, using the value representation we need to evaluate f and g at the powers of 2, then multiply componentwise and transform back. It turns out that $(1, 7, 3, 0, 0, 0, 0, 0)$ transforms to

$$(11, 10, 9, 11, 14, 16, 4, 1),$$

and that $(-2, 0, 3, 0, 5, 0, 0, 0)$ transforms to

$$(6, 5, 0, 15, 6, 5, 0, 15).$$

Their product is

$$(15, 16, 0, 12, 16, 12, 0, 15),$$

and for the inverse transform

$$8^{-1}(1, 7, 10, 15, 10, 8, 1, 0)$$

we then get

$$(15, 3, 14, 4, 14, 1, 15, 0).$$

So the product of f and g is

$$15 + 3x + 14x^2 + 4x^3 + 14x^4 + x^5 + 15x^6.$$

Convolution

Since we sometimes identify polynomials with coefficient vectors it is natural to define a product on vectors (of equal length) that corresponds to multiplication of the corresponding polynomials. For a vector $a \in R^n$ we let a_i denote its i-th coefficient, for $i = 0, \ldots, n-1$. For $i < 0$ and $i > n-1$ we define $a_i = 0$. We also define the coordinatewise product on R^m by $a * b = (a_0 b_0, a_1 b_1, \ldots, a_{n-1} b_{n-1})$.

Definition The *convolution product* $\odot$ of two elements $a, n \in R^n$ is the element $c = a \odot b \in R^{2n}$ defined by

$$c_i = \sum_{j=0}^{n-1} a_j b_{i-j} = \sum_{\substack{j,k=0 \\ j+k=i}}^{n-1} a_j b_k.$$

Convolution Theorem *Suppose that* $\zeta_n \in R$. *If* $a, b \in R^{2n}$ *with* $a_i = b_i = 0$ *for* $i \geq n-1$, *then*

$$a \odot b = D^{-1}(D(a) * D(b)).$$

Thus the convolution theorem is one way of stating that polynomial multiplication can be done by Fourier transformation, n multiplications in R^n, followed by inverse Fourier transformation.

The problem with this is that we had to artificially blow up the size of the vectors by a factor 2 (by padding with n zeroes, because the product of two polynomials of degree n has degree $2n$ in general. One way to overcome that is by using a wrapped convolution product.

Wrapped convolution

Definition The *positive wrapped convolution product* $\oplus$ of two elements $a, n \in R^n$ is the element $c = a \oplus b \in R^n$ defined by

$$c_i = \sum_{j=0}^{i} a_j b_{i-j} + \sum_{j=i+1}^{n} a_j b_{n+i-j}.$$

The *negative wrapped convolution product* $\ominus$ is the element $c = a \ominus b \in R^n$ defined by

$$c_i = \sum_{j=0}^{i} a_j b_{i-j} - \sum_{j=i+1}^{n} a_j b_{n+i-j}.$$

If $\zeta_{2n} \in R$, we let $\widehat{\ }$ denote the transformation on R^n given by $x \mapsto \widehat{x} = (x_0, \zeta_{2n} x_1, \ldots, \zeta_{2n}^{n-1} x_{n-1})$.

Wrapped Convolution Theorem Suppose that $\zeta_{2n} \in R$ and let $\zeta_n = \zeta_{2n}^2$. If $a, b \in R^n$, then

(i) $a \oplus b = D^{-1}(< D(a), D(b) >)$;
(ii) if $d = a \ominus b$, then $\hat{d} = D^{-1}(< D(\hat{a}), D(\hat{b}) >)$.

We now turn to fast Fourier transforms in $\mathbb{Z}/m\mathbb{Z}$ for certain special moduli m. Namely, let $m = 2^{s2^{t-1}} + 1$; then $\zeta = 2^s$ is a primitive 2^t-th root of unity in $\mathbb{Z}/m\mathbb{Z}$. In this ring we can prove the following complexity result — in bit operations.

Theorem *Let $a \in (\mathbb{Z}/m\mathbb{Z})^n$, with $m = 2^{s2^{t-1}} + 1$ and $n = 2^t$. Then $D(a)$ and $D^{-1}(a)$ can be computed in time $O(n^2 \log n \log s^s)$.*

The only operations required in $\mathbb{Z}/m\mathbb{Z}$ are additions and multiplications by a power of ζ, which is just a shift, followed by reduction modulo m. The size of m is $2^{t-1} \log 2^s + 1 = m \log 2^s + 1$ bits. Shifting and additions and reduction modulo m can be done in linear time, and the number of them is $O(n \log n)$.

Integer multiplication

As an application we describe the Schönhage-Strassen fast multiplication method for large integers. The idea is to write integers u and v of $m = 2^t$ bits in base 2^l, requiring b digits, where $bl = m$. To be precise we let $l = 2^{t/2} = b$ if t is even and $l = 2^{(t+1)/2} = 2b$ if t is odd. One then applies the Fourier transform to the coefficient vectors, does the inner product multiplication on the b coefficients, (with a recursive call if necessary) and transforms back, to obtain a wrapped convolution product.

Theorem *Multiplication of integers of length n can be done in $\mathcal{O}(n \log n \log\log n)$ word operations.*

Continuous Fourier Transform

For 2π periodic real complex-valued functions f:

$$\widehat{f}(k) = \int_0^{2\pi} f(t) e^{-\sqrt{-1}kt} dt,$$

for $k \in \mathbb{Z}$.

Then

$$f(t) = \frac{1}{2\pi} \sum_{k \in \mathbb{Z}} \widehat{f}(k) e^{\sqrt{i}kt}.$$