MUA
Maratones uniandes

FAST FOURIER TRANSFORM (FFT)

ISIS 2801

Polynomial multiplication

polynomial of degree n-1

$$A(x) = a_0x^0 + a_1x^1 + \cdots + a_{n-1}x^{n-1}$$

$$B(x) = b_0x^0 + b_1x^1 + \cdots + b_{n-1}x^{n-1}$$

$$A(x)B(x) = \sum_{i=0}^{2n-2} c_i x^i$$ where $$c_i = \sum_{k=0}^{i} a_k b_{j-k}$$

Polynomial multiplication

polynomial of degree n-1

$$A(x) = a_0x^0 + a_1x^1 + \cdots + a_{n-1}x^{n-1}$$

$$B(x) = b_0x^0 + b_1x^1 + \cdots + b_{n-1}x^{n-1}$$

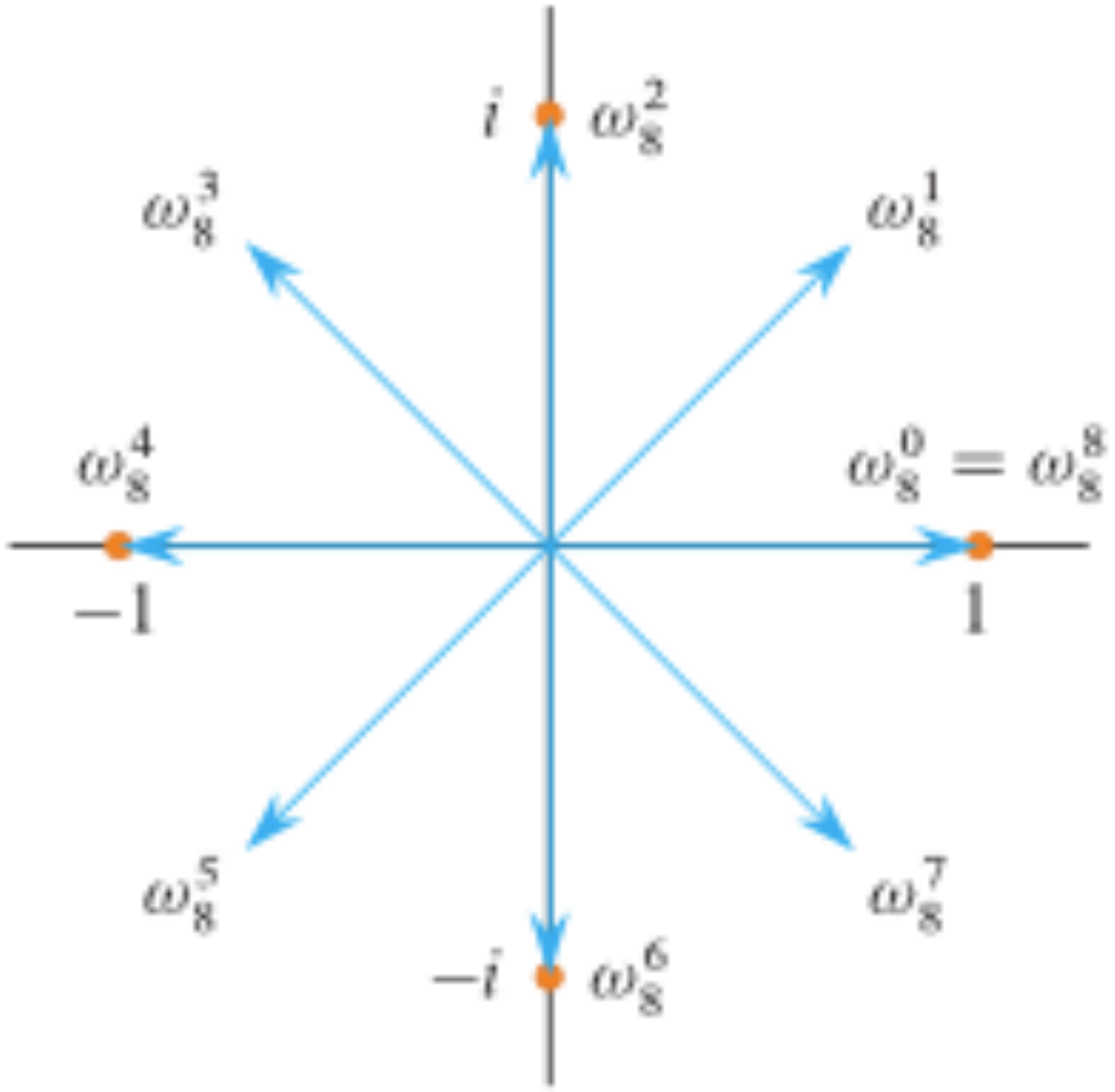
$$A(x)B(x) = \sum_{i=0}^{2n-2} c_i x^i \quad \text{where} \quad c_i = \sum_{k=0}^{i} a_k b_{j-k}$$

$O(n^2)$

Discrete Fourier Transform

Let $A(x) = a_0x^0 + a_1x^1 + \cdots + a_{n-1}x^{n-1}$ be a polynomial with **n** a power of **2**

Let $w_{n,k} = e^{\frac{2k\pi i}{n}}$ be the n roots of $x^n = 1$



Note that $w_{n,k} = (w_{n,1})^k$

Discrete Fourier Transform

The DFT of $A(x)$ is a vector of coefficients defined by the polynomials at $x = w_{n,k}$

$$
\begin{aligned}
DFT(a_0, a_1, \ldots, a_{n-1}) &= (y_0, y_1, \ldots, y_{n-1}) \\
&= (A(w_n^0), A(w_n^1), \ldots, A(w_n^{n-1}))
\end{aligned}
$$

Discrete Fourier Transform

The DFT of $A(x)$ is a vector of coefficients defined by the polynomials at $x = w_{n,k}$

$$
\begin{aligned}
DFT(a_0, a_1, \ldots, a_{n-1}) &= (y_0, y_1, \ldots, y_{n-1}) \\
&= (A(w_n^0), A(w_n^1), \ldots, A(w_n^{n-1}))
\end{aligned}
$$

$$InverseDFT(y_0, y_1, \ldots, y_{n-1}) = (a_0, a_1, \ldots, a_{n-1})$$

Polynomial multiplication

Given $A(x)$ and $B(x)$ polynomials of degree n, we want to compute $A(x)B(x)$

Polynomial multiplication

Given $A(x)$ and $B(x)$ polynomials of degree n, we want to compute $A(x)B(x)$

Compute $DFT(A)$ and $DFT(B)$

$$DFT(A) = (A(w_n^0), A(w_n^1), \ldots, A(w_n^{n-1}))$$

$$DFT(B) = (B(w_n^0), B(w_n^1), \ldots, B(w_n^{n-1}))$$

Polynomial multiplication

$$
\begin{aligned}
DFT(A)DFT(B) &= (DFT(A(w_n^0), A(w_n^1), \ldots, A(w_n^{n-1}))(DFT(B(w_n^0), B(w_n^1), \ldots, B(w_n^{n-1}))) \\
&= DFT(A(w_n^0)(B(w_n^0), A(w_n^1)B(w_n^1), \ldots, A(w_n^{n-1})B(w_n^{n-1})) \\
&= DFT(AB)
\end{aligned}
$$

Polynomial multiplication

$$
\begin{aligned}
DFT(A)DFT(B) &= (DFT(A(w_n^0), A(w_n^1), \ldots, A(w_n^{n-1}))(DFT(B(w_n^0), B(w_n^1), \ldots, B(w_n^{n-1}))\\
&= DFT(A(w_n^0)(B(w_n^0), A(w_n^1)B(w_n^1), \ldots, A(w_n^{n-1})B(w_n^{n-1}))\\
&= DFT(AB)
\end{aligned}
$$

$$AB = InverseDFT(DFT(A)DFT(B)$$

Fast Fourier Transform

Given $A(x)$ a polynomial of degree n with n a power of 2

Split $A(x)$ a in two polynomials:

$$A_0(x) = a_0x^0 + a_2x^1 + \cdots + a_{n-2}x^{\frac{n}{2}-1}$$

$$A_1(x) = a_1x^0 + a_3x^1 + \cdots + a_{n-1}x^{\frac{n}{2}-1}$$

such that $A(x) = A_0(x^2) + xA_1(x^2)$

Fast Fourier Transform

Given $A(x)$ a polynomial of degree n with n a power of 2

Split $A(x)$ a in two polynomials:

$$A_0(x) = a_0x^0 + a_2x^1 + \cdots + a_{n-2}x^{\frac{n}{2}-1}$$

$$A_1(x) = a_1x^0 + a_3x^1 + \cdots + a_{n-1}x^{\frac{n}{2}-1}$$

such that $A(x) = A_0(x^2) + xA_1(x^2)$

$O(nlog(n))$

Fast Fourier Transform

$$(y_k^0)_{k=0}^{n/2-1} = DFT(A_0)$$

$$(y_k^1)_{k=0}^{n/2-1} = DFT(A_1)$$

Given $A(x) = A_0(x^2) + xA(x^2)$, for the first $\dfrac{n}{2}$ values

$$y_k = y_k^0 + w_n^k y_k^1 \text{ for } k = 1,\ldots,\frac{n}{2} - 1$$

Fast Fourier Transform

For the second $\frac{n}{2}$ values

$$
\begin{aligned}
y_{k+n/2} &= A(w_n^{k+n/2}) \\
&= A_0(w_n^{2k+n} + w_n^{k+n/2}A_1(w_n^{2k+n}) \\
&= A_0(w_n^{2k}w_n^{n}) + w_n^{k}w_n^{n/2}A_1(w_n^{2k}w_n^{n}) \text{ for} \\
&= A_0(w_n^{2k}) - w_n^{k}A_1(w_n^{2k}) \\
&= y_k^0 - w_n^k y_k^1
\end{aligned}
$$

Fast Fourier Transform

$$y_k = y_k^0 + w_n^k y_k^1$$
$$y_{k+n/2} = y_k^0 - w_n^k y_k^1$$

Inverse DFT

Vandermonde matrix

$$\begin{pmatrix} w_n^0 & w_n^0 & w_n^0 & \cdots & w_n^0 \\ w_n^0 & w_n^1 & w_n^2 & \cdots & w_n^{n-1} \\ w_n^0 & w_n^2 & w_n^4 & \cdots & w_n^{2(n-1)} \\ w_n^0 & w_n^3 & w_n^6 & \cdots & w_n^{3(n-1)} \\ & \vdots & & \ddots & \vdots \\ w_n^0 & w_n^{n-1} & w_n^{2(n-1)} & \cdots & w_n^{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Inverse DFT

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} w_n^0 & w_n^0 & w_n^0 & \cdots & w_n^0 \\ w_n^0 & w_n^1 & w_n^2 & \cdots & w_n^{n-1} \\ w_n^0 & w_n^2 & w_n^4 & \cdots & w_n^{2(n-1)} \\ w_n^0 & w_n^3 & w_n^6 & \cdots & w_n^{3(n-1)} \\ & \vdots & & \ddots & \vdots \\ w_n^0 & w_n^{n-1} & w_n^{2(n-1)} & \cdots & w_n^{(n-1)(n-1)} \end{pmatrix}^{-1} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Inverse DFT

$$\begin{pmatrix} w_n^0 & w_n^0 & w_n^0 & \cdots & w_n^0 \\ w_n^0 & w_n^{-1} & w_n^{-2} & \cdots & w_n^{-(n-1)} \\ w_n^0 & w_n^{-2} & w_n^{-4} & \cdots & w_n^{-2(n-1)} \\ w_n^0 & w_n^{-3} & w_n^{-6} & \cdots & w_n^{-3(n-1)} \\ & \vdots & & \ddots & \vdots \\ w_n^0 & w_n^{-(n-1)} & w_n^{-2(n-1)} & \cdots & w_n^{-(n-1)(n-1)} \end{pmatrix}^{-1} = \frac{1}{n} \begin{pmatrix} w_n^0 & w_n^0 & w_n^0 & \cdots & w_n^0 \\ w_n^0 & w_n^{-1} & w_n^{-2} & \cdots & w_n^{-(n-1)} \\ w_n^0 & w_n^{-2} & w_n^{-4} & \cdots & w_n^{-2(n-1)} \\ w_n^0 & w_n^{-3} & w_n^{-6} & \cdots & w_n^{-3(n-1)} \\ & \vdots & & \ddots & \vdots \\ w_n^0 & w_n^{-(n-1)} & w_n^{-2(n-1)} & \cdots & w_n^{-(n-1)(n-1)} \end{pmatrix}$$

Inverse DFT

$$\begin{pmatrix} w_n^0 & w_n^0 & w_n^0 & \cdots & w_n^0 \\ w_n^0 & w_n^{-1} & w_n^{-2} & \cdots & w_n^{-(n-1)} \\ w_n^0 & w_n^{-2} & w_n^{-4} & \cdots & w_n^{-2(n-1)} \\ w_n^0 & w_n^{-3} & w_n^{-6} & \cdots & w_n^{-3(n-1)} \\ & \vdots & & \ddots & \vdots \\ w_n^0 & w_n^{-(n-1)} & w_n^{-2(n-1)} & \cdots & w_n^{-(n-1)(n-1)} \end{pmatrix}^{-1} = \frac{1}{n} \begin{pmatrix} w_n^0 & w_n^0 & w_n^0 & \cdots & w_n^0 \\ w_n^0 & w_n^{-1} & w_n^{-2} & \cdots & w_n^{-(n-1)} \\ w_n^0 & w_n^{-2} & w_n^{-4} & \cdots & w_n^{-2(n-1)} \\ w_n^0 & w_n^{-3} & w_n^{-6} & \cdots & w_n^{-3(n-1)} \\ & \vdots & & \ddots & \vdots \\ w_n^0 & w_n^{-(n-1)} & w_n^{-2(n-1)} & \cdots & w_n^{-(n-1)(n-1)} \end{pmatrix}$$

then $\quad a_k = \dfrac{1}{n} \displaystyle\sum_{j=0}^{n-1} y_j w_n^{-kj}$

FFT

```
using cd = complex<double>;
const double PI = acos(-1);

void fft(vector<cd> & a, bool invert) {
    int n = a.size();
    if (n == 1)
        return;

    vector<cd> a0(n / 2), a1(n / 2);
    for (int i = 0; 2 * i < n; i++) {
        a0[i] = a[2*i];
        a1[i] = a[2*i+1];
    }
    fft(a0, invert);
    fft(a1, invert);

    double ang = 2 * PI / n * (invert ? -1 : 1);
    cd w(1), wn(cos(ang), sin(ang));
    for (int i = 0; 2 * i < n; i++) {
        a[i] = a0[i] + w * a1[i];
        a[i + n/2] = a0[i] - w * a1[i];
        if (invert) {
            a[i] /= 2;
            a[i + n/2] /= 2;
        }
        w *= wn;
    }
}
```

Polynomial multiplication

```
vector<int> multiply(vector<int> const& a, vector<int> const& b) {
    vector<cd> fa(a.begin(), a.end()), fb(b.begin(), b.end());
    int n = 1;
    while (n < a.size() + b.size())
        n <<= 1;
    fa.resize(n);
    fb.resize(n);

    fft(fa, false);
    fft(fb, false);
    for (int i = 0; i < n; i++)
        fa[i] *= fb[i];
    fft(fa, true);

    vector<int> result(n);
    for (int i = 0; i < n; i++)
        result[i] = round(fa[i].real());
    return result;
}
```

Number multiplication

```
vector<int> multiply(vector<int> const& a, vector<int> const& b) {
    vector<cd> fa(a.begin(), a.end()), fb(b.begin(), b.end());
    int n = 1;
    while (n < a.size() + b.size())
        n <<= 1;
    fa.resize(n);
    fb.resize(n);

    fft(fa, false);
    fft(fb, false);
    for (int i = 0; i < n; i++)
        fa[i] *= fb[i];
    fft(fa, true);

    vector<int> result(n);
    for (int i = 0; i < n; i++)
        result[i] = round(fa[i].real());
     int carry = 0;
    for (int i = 0; i < n; i++)
        result[i] += carry;
        carry = result[i] / 10;
        result[i] %= 10;
    }
    return result;
}
```

Caveats

- This algorithm is only able to handle polynomials of size 10^5 or number multiplication of numbers of size 10^6
- A new (2021) mechanisms is able to multiply arbitrary integers in $O(nlog(n))$. Harvey, D and van Der Hoeven, J. https://hal.science/hal-02070778/file/nlogn.pdf