Fast Polynomial Multiplication

- Say, f & g are polynomials in $R[x]$; of deg $\leq \ell$.

- We want to beat the "$O(\ell^2)$-many R-ops" algorithm; make it $O(\ell)$?

- New representation: Use evaluations & Gauss' trick.

- Suppose R has a primitive ℓ-th root of unity ω. [$\mathrm{Eg.}\ \sqrt[\ell]{1} \in \mathbb{C}$] [$X^{\ell}-1 = \prod_{i\in[\ell]}(X-\omega^i)$]

Idea: * 1) Evaluate f & g at $\{\omega^0 = 1, \omega, \omega^2, \ldots, \omega^{\ell-1}\}$.
2) Multiply $f(\omega^i)\cdot g(\omega^i)$ in R. $(0\le i<\ell)$.
*3) Interpolate to get $h := f(x)\cdot g(x)$.

- Let $f(x) =: \sum_{i=0}^{\ell-1} a_i x^i$, a_i's in R.

linear operator

- Discrete Fourier Transform $\mathrm{DFT}[\omega]: (a_0, \ldots, a_{\ell-1}) \mapsto (f(\omega^0), \ldots, f(\omega^{\ell-1}))$

where $\ell := 2^n$.

<u>Lemma 1</u>: $DFT[\omega^{-1}] \circ DFT[\omega] = \ell \cdot \underline{Id_\ell}$ $[\ell^{-1} \cdot DFT[\omega^{-1}]]$

<u>Pf</u>: • $DFT[\omega]$ is the following matrix action:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{\ell-1} \\ \vdots & & & \\ 1 & \omega^{\ell-1} & \cdots & \omega^{(\ell-1)(\ell-1)} \end{bmatrix} \cdot \begin{pmatrix} a_0 \\ \vdots \\ a_{\ell-1} \end{pmatrix} = \begin{pmatrix} f(1) \\ f(\omega) \\ \vdots \\ f(\omega^{\ell-1}) \end{pmatrix}$$

$\Rightarrow$ The action of $DFT[\omega^{-1}] \circ DFT[\omega]$ is:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \cdots & \omega^{-(\ell-1)} \\ \vdots & & & \vdots \\ 1 & \omega^{-(\ell-1)} & \cdots & \omega^{-(\ell-1)(\ell-1)} \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{\ell-1} \\ \vdots & & & \\ 1 & \omega^{\ell-1} & \cdots & \omega^{(\ell-1)(\ell-1)} \end{bmatrix} = \begin{bmatrix} \ell & 0 & \cdots & 0 \\ 0 & \ell & \cdots & 0 \\ & & \vdots & \\ 0 & \cdots & \cdots & \ell \end{bmatrix}$$

$$\Big[\ X^{\ell}-1 = (X-1)(X-\omega_\ell)\cdots(X-\omega_\ell^{\ell-1}) \quad (*)$$

$$\Rightarrow \operatorname{coef}(X^{\ell-1})(\cdot): \quad 0 = 1+\omega_\ell+\cdots+\omega_\ell^{\ell-1} \quad \checkmark$$

Defn. of primitive ω!

$$\sum_{i=0}^{\ell-1}\omega_\ell^{2i} = \sum_{i=0}^{\ell-1}\omega_{\ell/2}^{i} = 0. \quad \checkmark$$

$$\sum_{i=0}^{\ell-1}\omega_\ell^{3i} = \sum_{i=0}^{\ell-1}(\omega_\ell')^{i} = 0 \quad \checkmark$$

$$\vdots$$

& so on. $\Big]$

<u>Note</u>: Assume $\ell = 2^n \notin zd(R)$, i.e. $2 \nmid \operatorname{ch} R$ or $\operatorname{ch} R = 0$.

$[\text{odd}(\operatorname{ch} R) \Rightarrow \ell^{-1}\in R]$

– Naively, DFT[w] takes $O(\ell^2)$-time.
But, Gauss' had a better idea:

<u>Lemma 2</u>: DFT[w] can be computed in $O(\ell \cdot \lg \ell)$ R-operations.

<u>Pf</u>: • $f(x) =: f_0(x^2) + x \cdot f_1(x^2)$ & use divide-conquer paradigm:

1) Compute DFT$[w^2]$: $f_0 \longmapsto (e'_0, \ldots, e'_{\ell/2-1})$
& " : $f_1 \longmapsto (e''_0, \ldots, e''_{\ell/2-1})$.

2) Compute $\left.\begin{array}{l} e_i := e'_i + w^i \cdot e''_i \\ e_{i+\frac{\ell}{2}} := e'_i - w^i \, e''_i \end{array}\right\}$ $0 \le i < \ell/2$

3) Output $(e_0, e_1, \ldots, e_{\ell-1})$.

- Let it take $T(\ell)$ R-ops. for $DFT[w_\ell]$. Then, we've the recurrence:

$$T(\ell) \leqslant 2 \cdot T(\ell/2) + O(\ell)$$
$$\Rightarrow \quad T(\ell) = O(\ell \cdot \lg \ell). \qquad \Box$$

<u>Theorem:</u> $h = f \cdot g$ computable in $O(\ell \cdot \lg \ell)$ R-ops.

Pf:
$$\begin{array}{ccccc} f & \xmapsto{DFT[w]} & (f(1), \ldots, f(w^{\ell-1})) & \xmapsto{\text{Mult.}} & (h(1), \ldots, h(w^{\ell-1})) \\ g & & (g(1), \ldots, g(w^{\ell-1})) & & \Big\downarrow \ell^{-1} \cdot DFT[w^{-1}] \\ & & & & h(x) \end{array}$$
$\Box$

- What if R doesn't have ω? [eg. $R = \mathbb{Z}$ $= \mathbb{Q}$]
We'll create ω out of thin air!

- Consider $E := R[y]/\langle \underbrace{y^{k}+1}_{\text{is irreducible over } \mathbb{Q}} \rangle$ &
$\omega_{2k} := y$ in E

2-power

$\triangleright$ $\displaystyle\sum_{i=0}^{2k-1} y^{i} = \frac{y^{2k}-1}{y-1} = \left(\frac{y^{k}-1}{y-1}\right) \cdot (y^{k}+1) \equiv 0$ in E.

In fact, $\triangleright$ $\displaystyle\prod_{i=0}^{2k-1} (X-y^{i}) \equiv X^{2k}-1$ in $E[x]$.

- Rewrite the polynomials as:

$f =: \sum_{i=0}^{m-1} f_i \cdot x^{ki}$ | $g =: \sum_{i=0}^{m-1} g_i \cdot x^{ki}$

where, $k := 2^{\lfloor n/2 \rfloor}$ & $m := 2^{\lceil n/2 \rceil}$

▷ f_i, g_i's are polys. of deg $< k < m \leq 2k$.

<u>Idea</u>: $F(y,x) := \sum_{i=0}^{m-1} f_i(y) \cdot x^i$

$G(y,x) := \sum g_i(y) \cdot x^i$ $\quad [\, y = \omega_{2k} \in E \,]$

& multiply them.

<u>Fact</u>: Let $F(y,x) \cdot G(y,x) =: H(y,x)$ in $E[x]$.

Recover $h = f \cdot g$ as: $H(y = x, x = x^k) = h$.

Pf: • The $\deg_y(H) < k$.
$\Rightarrow$ Modulus $\langle y^k+1 \rangle$ has no information loss. $\square$

- Since, E has ω_{2k} (2-power $= 2k$-th primitive root of unity) & $ch(E) = $ (odd or zero):
Compute H using the DFT algorithm.

Lemma 1: DFT$[\omega]$ takes $O(\sqrt{\ell} \cdot \lg \ell)$ E-operations.
Hence, " " $\sqrt{\ell} \times O(\sqrt{\ell} \cdot \lg \ell)$ R-operations.
Pf sketch: Recursive algorithm mainly uses additions in E & multiplies by y^j's. $\square$

- Next, to multiply values of F & G, in E, we need m instances of multiplication each instance has $\deg < k$ (over R).
- Univariate fast multiplication gives us:

$$T(\ell) \le m \cdot T(k) + O(\ell \cdot \lg \ell) \text{ ---- (I)}$$

[# sq-roots $\le \lg\lg \ell$] [$T(\ell)/\ell \le T(k)/k + O(\lg \ell)$]

$$\Longrightarrow \quad T(\ell) \le O(\ell \cdot \lg \ell \cdot \lg\lg \ell).$$
$$= \tilde{O}(\ell) \quad \text{R-operations.}$$

→ If $\operatorname{ch} R = 2$ then use $\ell = 3^n$. Use ω_ℓ over R.
DFT works as well.
$E := R[y]/\langle \Phi_\ell(y) \rangle$

ℓ-th cyclotomic poly.

Theorem (Schönhage-Strassen '71): In all cases, $h = f \cdot g \in R[x]$ can be computed in $O(\ell \cdot \lg \ell \cdot \lg\lg \ell)$ R-operations.

not bit-operations.