Lecture 4. FFT Dec. 18. 2020

1. Polynomial Multiplication

Input: $A(x) = a_0 + a_1 x + \cdots + a_{d-1} x^{d-1}$

$B(x) = b_0 + b_1 x + \cdots + b_{d-1} x^{d-1}$

Goal: compute the vector of coefficients of $C = A \cdot B$

$C(x) = c_0 + c_1 x + \cdots + c_{2d-2} x^{2d-2}$ def: $N = 2d-1$

$c_0 = a_0 b_0$ $c_1 = a_0 b_1 + a_1 b_0$ $c_k = \sum_{j=0}^{k} a_j b_{k-j}$

Ex. Integer mult.

1074 $A(x) = 4 + 7x + 0 \cdot x^2 + 1 \cdot x^3$ $(\alpha = A(10))$

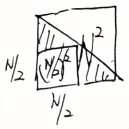
X 2351 $B(x) = 1 + 5x + 3 \cdot x^2 + 2 \cdot x^3$ $(\beta = B(10))$

$\alpha \cdot \beta = (A \times B)(10)$

Alg1. (poly mult).

Two nested for-loops $c_k = \sum_{j=0}^{k} a_j b_{k-j}$ $O(n^2)$

also $\Omega(n^2)$ since computing each of $c_{\frac{N}{2}} \cdots c_{N-1}$ requires $\geq \frac{N}{2}$ flops each $\Rightarrow \geq \frac{N^2}{2^2}$ flops



$\Omega((N/2)^2)$, $O(N^2) \Rightarrow \Theta(N)$

Alg2. karatsuba

$A(x) = \underbrace{a_0 + a_1 x + \cdots}_{A_l(x)} + \underbrace{\cdots + a_{N-1} x^{N-1}}_{A_h(x)\, x^{N/2}}$

$B(x) = \underbrace{b_0 + b_1 x + \cdots}_{B_l(x)} + \underbrace{\cdots + b_{N-1} x^{N-1}}_{B_h(x)\, x^{N/2}}$

$(A \cdot B)(x) = A_l \times B_l + (A_l B_h + A_h B_l) x^{N/2} + A_h B_h x^N$

$T(n) \leq 3T(n) + c \cdot n$ (illegible)

$T(n) = O(n^{\log_2 3})$

Interpolation: any degree $< N$ poly is determined by its evaluation on N distinct points.

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{N-1} \\ \vdots & & & & \\ 1 & x_{N-1} & x_{N-1}^2 & \cdots & x_{N-1}^{N-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_{N-1} \end{pmatrix}$$

① use $A(x)$, $B(x)$ calculate y. (n^2)

② $C = V^{-1}Y$

Vandermonde

$$\det(V) = \prod_{i<j} (x_i - x_j)$$

eval (p, x): ans $\leftarrow 0$ cpow $\leftarrow 1$

↓ value of p at points x

for $i = 0$ to $N-1$ $\quad O(n)$

ans $\leftarrow$ ans $+ p_i \cdot$ cpow

cpow $\leftarrow$ cpow $\cdot x$

return ans

$C(x) = (A \times B)(x)$

$= A(x)B(x) = Y$

Total of interpolation: $O(n^2)$: each eval (p, x) $O(n)$

n points totally $(x_0 \sim x_{N-1})$

After calculate $\dfrac{\text{eval}(C, x)}{y}$, $C = V^{-1}Y$. worse than Alg 1.

Fast Fourier Transform (FFT) is an algorithm

Discrete Fourier Transform (DFT) is a matrix

DFT: $F_{ij} = (\omega^i)^j$, $\omega = e^{-2\pi i/N}$ $(i = \sqrt{-1})$

$$\begin{aligned} p(z) &= p_0 + p_1 z + p_2 z^2 + \cdots + p_{N-1} z^{N-1} \\ &= (p_0 + p_2 z^2 + p_4 z^4 + \cdots) + (p_1 + p_3 z^2 + \cdots) z \\ &= p_{even}(z^2) + p_{odd}(z^2) \cdot z \end{aligned}$$

Goal: $p(\omega^0)$ $p(\omega^1) \ldots p(\omega^{N-1})$

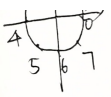
require: $p_{even}(z^2)$ $p_{odd}(z^2)$

$p_{even}(\omega^0)$ $p_{even}(\omega^2)$ $p_{even}(\omega^{2N-2})$

$p_{odd}(\omega^0)$ $p_{odd}(\omega^2)$ $p_{odd}(\omega^{2N-2})$

p_{even}, p_{odd}: $N/2$ Points: still N $(0 \sim 2N-2)$

N points are doubled, only need to



calculate $N/2$ points.

$P(x) = P_{odd}(x^2) \cdot x + P_{even}(x^2)$

$T(1) = 1$

$T(N) = 2T(\frac{N}{2}) + \Theta(N)$ — 2: even and odd; $N/2$: only need calculate $N/2$ points (N points totally, but a period)

P_{odd}, P_{even}

$G(N)$: $P_{odd}(x^2) \cdot x$ addition

$T(N) = \Theta(N \log N)$

Game Plan: (given A, B)

$F = \begin{pmatrix} 1 & w_0 & \cdots & w_0^{N-1} \\ \vdots & & \ddots & \vdots \\ 1 & w_{N-1} & \cdots & w_{N-1}^{N-1} \end{pmatrix}$

step1. compute $F \cdot \vec{a}$ to get $\hat{a}$ (FFT)

step2. compute $F \cdot \vec{b}$ to get $\hat{b}$ (FFT)

step3. compute $\hat{c}_j = \hat{a}_j \cdot \hat{b}_j \quad j = 0 \cdots N-1$

step4. return $F^{-1} \cdot \hat{c} = \frac{1}{N} \overline{F} \cdot \hat{c} = \frac{1}{N} \overline{F \cdot \overline{\hat{c}}}$

$\vec{c} \to \overline{\vec{c}}$

$F \cdot \overline{\vec{c}}$ (FFT)

$\overline{F \cdot \overline{\vec{c}}}$

Claim $F^{-1} = \frac{1}{N} \cdot \overline{F}$ $\overline{F}$: complex conjugate

Claim $\overline{Av} = \overline{A}\,\overline{v}$

cross-correlation (successive dot products)

$X = [x_0 x_1 \cdots x_{n-1}]$

$Y = [y_0 y_1 \cdots y_{m-1}]$

$\sum_{i=0}^{m-1} x_i y_i$

$\sum_{i=0}^{m-1} x_i y_{i+1}$

$\sum_{i=0}^{m-1} x_i y_{i+2}$

$\vdots$

} list of $n-m+1$ numbers.

$\Theta((n-m)m)$

each number: $\Theta(m)$

totally $(n-m+1)$ numbers

$\Rightarrow \Theta((n-m)m)$

Faster: $Y(z) = y_0 + y_1 z + \cdots + y_{m-1} z^{m-1}$

$X(z) = x_{n-1} + x_{n-2} z + \cdots + x_0 z^{n-1}$ $\quad > Q = Y \cdot X$

$Q(z) = (Y \cdot X) z = q_0 + q_1 z + \cdots + q_{n+m-2} z^{n+m-2}$

$q_0 = y_0 x_{n-1}$ $\quad q_{m-1} = x_{n-1} y_{m-1} + x_{n-2} y_{m-2}$

$+ \cdots + x_0 y_0$ (the first of list of)

numbers /

$q_m = x_{m+1} y_m + x_{m-2} y_{m+1} + \cdots + x_0 y_1$ (the second)

return $q_{m-1}, q_m, q_{m+1}, \ldots, q_{n-1}$

$T(M) = \Theta(M \log M)$