05 FFT

Tuesday, January 21, 2025 13:18

Q4) There is no formula $x = \dots$

x is such that $1 = \sum_{i=1}^{k} a_i \frac{1}{b_i^x}$

$p \in (0,1)$

$p^x = p \iff x = 1$

$p^x < p \iff x > 1$ eg $\left(\frac{1}{2}\right)^2 = \frac{1}{4} < \frac{1}{2}$

$p^x > p \iff x < 1$ eg $\left(\frac{1}{4}\right)^{1/2} = \frac{1}{2} > \frac{1}{4}$

$f(x) = \sum_{i=0}^{n-1} x^i \cdot f_i$

$g(x) = \sum_{i=0}^{n-1} x^i \cdot g_i$

Compute: $(f \cdot g)(x) = \sum_{i=0}^{2(n-1)} \left(x^i \cdot \underline{\sum_{k=0}^{i} f_k \cdot g_{i-k}} \right)$

Naive: $O(n^2)$

Today: $O(n \log n)$

Representation: array of coefficients

$f = [f_0, f_1, \dots f_{n-1}]$ // length n array

$g = [g_0, g_1, g_2, \dots g_{n-1}]$

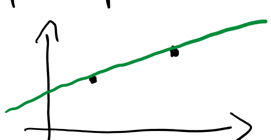
"coefficient representation"

$h = f \cdot g$

$\underline{h[i]} = \sum_{k=0}^{i} f[k]\, g[i-k]$ // $O(n)$ for one coefficient

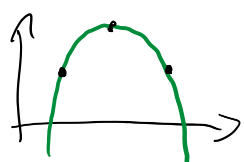
$\Rightarrow O(n^2)$ total

"point representation"

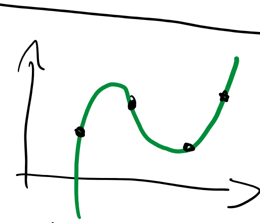


for 2 points

there is a unique degree 1 polynomial going through these points



3 points $\to$ degree 2



4 points $\to$ degree 3

If we are given n points, then there is a unique degree $n-1$ poly. going through these points

---

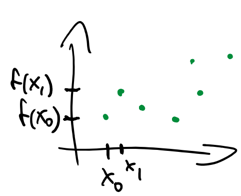
Multiplication in point representation

$\to$ Fix some $\underline{x_0\ x_1 \dots x_{n-1}}$ x-coordinates

$\to \bar{f} = [\underline{f(x_0), f(x_1), f(x_2), \dots} f(x_{n-1})]$

$\to \bar{g} = [g(x_0), g(x_1), g(x_2), \dots g(x_{n-1})]$

} Eval



n points

$\overline{f(x) \cdot g(x)} = [\, f(x_0) \cdot g(x_0),\ f(x_1) \cdot g(x_1), \dots \,]$ $\leftarrow$ for-loop

$\overline{f \cdot g}\,[i] = \bar{f}[i] \cdot \bar{g}[i]$ // $O(1)$ for one i

$\Rightarrow O(n)$ for all i

High level Idea: n = degree f +1
= length of coefficient array $\bar{f}[0 \ldots 2n-1]$ $\bar{g}[0 \ldots 2n-1]$

Mult($f[0 \ldots n-1]$, $g[0 \ldots n-1]$)

- transform f, g into point representation $\bar{f}, \bar{g}$ with $2n-1$ points via Eval
- $\bar{h}[i] = \bar{f}[i] \cdot \bar{g}[i]$ for $i = 0 \ldots 2(n-1)$
- transform point representation $\bar{h}$ into coefficient representation $h[0 \ldots 2(n-1)]$
- return h

$\deg(h) = 2(n-1)$
$\rightarrow 2(n-1) + 1$ points
$2n - 1$ points

$x_0^i = x_0 \cdot x_0^{i-1}$

$f(x_0) = \sum_{i=0}^{n} x_0^i \cdot f[i]$ // $O(n)$ time for one x_k

$\Rightarrow O(n^2)$ time for all $x_0, x_1, x_2 \ldots x_{2n-1}$

$z = 1$
for $i = 0 \ldots n$
  out += $z \cdot f[i]$
  $z = x_0 \cdot z$ // $z = x_0^{i+1}$
$O(n)$ time

$\bar{h}[i] = h(x_i) = f(x_i) \cdot g(x_i) = \bar{f}[i] \cdot \bar{g}[i]$

$h(z) = f(z) \cdot g(z)$

How to get point representation $\bar{f}$ in $O(n \log n)$?

$f(x) = 5x^5 + 2x^4 + 3x^3 - 2x^2 + 4x + 6$

$f_{even}(x) = 2x^2 - 2x + 6$

$f_{odd}(x) = 5x^2 + 3x + 4$

$f_{even}(x) = \sum_{i=0}^{d} x^i \cdot f[2i]$

$f_{odd}(x) = \sum_{i=0}^{d} x^i \cdot f[2i+1]$

$f_{even} = [6, -2, 2]$

$f(x) = f_{even}(x^2) + x \cdot f_{odd}(x^2)$

$f_{even}(x^2) = 2(x^2)^2 - 2(x^2) + 6$
$= 2x^4 - 2x^2 + 6$

Eval($f[0 \ldots d]$, x_0)
  $f_{even} = [f[2i] \mid$ for $i = 0 \ldots d/2]$
  $f_{odd} = [f[2i+1] \mid$ for $i = 0 \ldots d/2]$
  return Eval(f_{even}, x_0^2) + $x_0 \cdot$ Eval(f_{odd}, x_0^2)

$T(d) = 2 \cdot T(\frac{d}{2}) + O(1)$
$= O(d)$ for one x_0
but we need $x_0\, x_1 \ldots x_d$
$\Rightarrow O(d^2)$ total

$z = \{x_0\ x_1\ x_2 \ldots x_d\}$

Eval($f[0 \ldots d]$, z) // return $f(x_0)\, f(x_1) \ldots f(x_d)$

Eval(f_{even}, $\{z[i]^2 \mid$ for $i = 0 \ldots d\}$)
Eval(f_{odd}, $\{z[i]^2 \mid$ for $i = 0 \ldots d\}$)

f_{even} length $d/2$
f_{odd} length $d/2$
size of $\{z[i]^2 \mid$ for $i = 0 \ldots d\}$
does not decrease in general

1.1 11 $z = \{1\ -1\}$

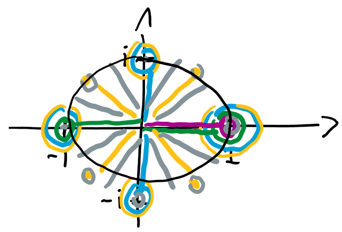
$z = \{1, 2, 3, -1, -2, -3\}$

size of $\{z[i] \mid \text{for } i = 0 \dots d\}$ does not decrease in general

Idea: If $z = \{1, -1\}$
$\{z[i] \mid i = 0,1\} = \{1\}$

$z = \{1, 2, 3, -1, -2, -3\}$
$\downarrow$ one recursion
$z = \{1, 4, 9\}$
$\downarrow$
$z = \{1, 16, 81\}$

$z = \{1, -1, i, -i\}$ $\quad i^2 = -1$, $(-i)^2 = -1$
$\downarrow$
$z = \{1, -1\}$
$\downarrow$
$z = \{1\}$



$\pm\sqrt{z[i]} \downarrow$ $\quad \uparrow z[i]^2$

$Z_0 = \{1\}$
$Z_1 = \{1, -1\}$
$Z_2 = \{1, -1, i, -i\}$
$\vdots$
$Z_k = \{\exp(2\pi i/k \cdot j) \mid \text{for } j = 0, 1, 2 \dots k-1\}$

Eval(f [0.. d-1])
  $Z_d = \{e^{2\pi i/d \cdot j} \mid j = 0 \dots d-1\}$ // d many x-coordinates
  $Y_{even} = \text{Eval}(f_{even})$ // return hash-map with $Y_{even}[z] = f_{even}(z)$ for all $z \in Z_{d/2}$
  $Y_{odd} = \text{Eval}(f_{odd})$
  for $z \in Z_d$
    $Y[z] = Y_{even}[z^2] + z \cdot Y_{odd}[z^2]$ // $z^2 \in Z_{d-1}$ for all $z \in Z_d$
  return Y

$Y[z] = f(z) = f_{even}(z^2) + z \cdot f_{odd}(z^2) = Y_{even}[z^2] + z \cdot Y[z^2]$
if $z \in Z_d$ then $z^2 \in Z_{d-1}$

Eval(f [0.. d-1], Z)
  if d == 1
    return f[0]
  $Y_{even} = \text{Eval}(f_{even}, \{z[i]^2 \mid \text{for } i = 0 \dots d-1\})$
  $Y_{odd} = \text{Eval}(f_{odd}, \{z[i]^2 \mid \text{for } i = 0 \dots d-1\})$
  for $z \in Z_d$
    $Y[z] = Y_{even}[z^2] + z \cdot Y_{odd}[z^2]$
  return Y

$Y[z] = f(z) = f_{even}(z^2) + z \cdot f_{odd}(z^2) = Y_{even}[z^2] + z \cdot Y[z^2]$

For general $Z = \{z_1 \dots z_n\}$
$$T(d, n) = 2 \cdot T(d/2, n) + O(dn)$$
For specific complex $Z = \{e^{2\pi i\, j/n} \mid j = 0 \dots n-1\}$
$$T(d, n) = 2 \cdot T(d/2, n/2) + O(n)$$
$$= O(n \log n)$$

$$T(d) = 2 \cdot T(d/2) + O(d)$$
$\Rightarrow O(d \log d)$

fine if $Z = \{e^{2\pi i/d \cdot j} \mid \text{for } j = 0 \dots d-1\}$

if both f and Z decrease by factor 2

f_{even} and f_{odd} are half size

$\Rightarrow O(d \log d)$ ← if both f and z decrease by factor 2

↑ fine because f_{even} and f_{odd} are half size

$$\boxed{\begin{aligned} z &= \{1, -1, i, -i\} \\ f &= 2x^4 + 3x^3 + 2x^2 + 4x + 2 \end{aligned}}$$

$$\boxed{\begin{aligned} f_{even} &= 2x^2 + 2x + 2 \\ z &= \{1, -1\} \end{aligned}} \qquad \boxed{\begin{aligned} f_{odd} &= 3x + 4 \\ z &= \{1, -1\} \end{aligned}}$$

$f_{even}(1) = 6$

$f_{even}(-1) = 2$

$f_{odd}(1) = 7$

$f_{odd}(-1) = 1$

$$f(x) = f_{even}(x^2) + x \cdot f_{odd}(x^2)$$

$$f(1) = f_{even}(1) + 1 \cdot f_{odd}(1) = 6 + 7$$

$$f(-1) = f_{even}(1) + (-1) \cdot f_{odd}(1) = 6 - 7$$

$$f(i) = f_{even}(-1) + i \cdot f_{odd}(-1) = 2 + i \cdot 1$$

$$f(-i) = f_{even}(\underbrace{-1}_{(-i)^2}) + (-i) \cdot f_{odd}(-1) = 2 - i \cdot 1$$

How to revert back from point to coefficient?

$$\underbrace{\begin{pmatrix} z^0 & z^1 & z^2 & z^3 \end{pmatrix}}_{M} \cdot \begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{pmatrix} = \sum_{i=0}^{3} z^i \cdot h_i = h(z)$$

↖ unknown

unknown

$$\begin{pmatrix} z_0^0 & z_0^1 & z_0^2 & z_0^3 \\ z_1^0 & z_1^1 & z_1^2 & z_1^3 \\ z_2^0 & z_2^1 & z_2^2 & z_2^3 \\ z_3^0 & z_3^1 & z_3^2 & z_3^3 \end{pmatrix} \begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{pmatrix} = \begin{pmatrix} h(z_0) \\ h(z_1) \\ h(z_2) \\ h(z_3) \end{pmatrix} = \begin{pmatrix} \bar{h}[0] \\ \bar{h}[1] \\ \bar{h}[2] \\ \bar{h}[3] \end{pmatrix}$$

known   unknown   known

Linear system: $M h = \bar{h}$

solve linear system $h = M^{-1} \bar{h}$

$M^{-1} \approx M$ up to some normalization

so to revert we can comple $M^{-1} \cdot \begin{pmatrix} h(z_0) \\ \vdots \\ h(z_3) \end{pmatrix}$ via same algo

$$\approx M \cdot \begin{pmatrix} h(z_0) \\ \vdots \\ h(z_3) \end{pmatrix}$$

by interpeting y-coordinates as the coefficients

FFT is transformation from coefficient representation

(Fast Fourier Transform)

representation

(Fast Fourier Transform) to point representation

$f, g \leftarrow$ coefficients $f = [f_0, f_1 \dots f_d]$

$\bar{f}, \bar{g}$ via FFT $\bar{f} = [f(z_0)\ f(z_1)\ f(z_2) \dots f(z_d)]$ // $O(d \log d)$

$\bar{h}[i] = \bar{f}[i]\,\bar{g}[i]$

$\bar{h} = [\,f(z_i) \cdot g(z_i) \mid \text{for } i = 0 \dots d\,] = [\,h(z_i) \mid \text{for } i = 1 \dots d\,]$ $\Big\}\ O(d)$

convert point rep. $\bar{h}$ to h coefficient rep. // $O(d \log d)$

return h

$\bar{h}$