August 28, 2020 10:41 AM

# Basic Alg'mic Tools

## Convolution Problem

Given 2 polynomials of deg $n-1$

$$A(x) = \sum_{j=0}^{n-1} a_j x^j, \qquad B(x) = \sum_{j=0}^{n-1} b_j x^j$$

compute their product $C(x) = A(x)B(x)$

$$= \sum_{j=0}^{2n-2} c_j x^j$$

e.g. $(x^2+3x+1)\cdot(2x^2+x+4)$

$$= 2x^4 + \underbrace{(1\cdot 1+3\cdot 2)}_{7}x^3 + \underbrace{(1\cdot 4+3\cdot 1+1\cdot 2)}_{9}x^2$$

$$+ \underbrace{(3\cdot 4+1\cdot 1)}_{13}x + 1\cdot 4$$

i.e. Given $a_0, \ldots, a_{n-1}$, $b_0, \ldots, b_{n-1}$

compute $\boxed{c_j = \sum_{k=0}^{j} a_k b_{j-k}}$

for $j = 0, \ldots, 2n-2$

Naive Alg'm: for each $j$, $O(n)$ time

$\Rightarrow$ $\boxed{O(n^2)}$ time total

## Karatsuba's Alg'm ('60)

first consider $n=2$:

given $a_0, a_1$, $b_0, b_1$

compute $c_0 = a_0 b_0$

$c_1 = a_0 b_1 + a_1 b_0$

$c_2 = a_1 b_1$

naive: 4 mults.

better sol'n:

rewrite $c_1 = (a_1+a_0)\cdot(b_1+b_0)$

better soln:

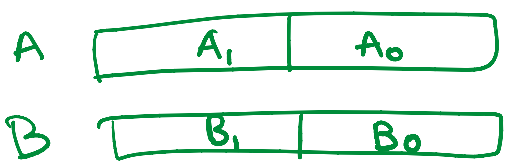
rewrite $c_1 = (a_1 + a_0)\cdot(b_1 + b_0) - a_1 b_1 - a_0 b_0$

3 mults!

power of subtraction!

general $n$?

idea - binary D & C




write $A(x) = A_1(x) \cdot x^{n/2} + A_0(x)$

$B(x) = B_1(x)\ x^{n/2} + B_0(x)$

where $A_1, A_0, B_1, B_0$ have deg $\frac{n}{2} - 1$

Then $A(x)B(x) = A_1(x)\,B_1(x)\ x^n + (A_1(x)\,B_0(x) + A_0(x)B_1(x))\ x^{n/2} + A_0(x)\,B_0(x)$

4 recursive calls

$\Rightarrow\ T(n) = 4\,T(\frac{n}{2}) + O(n)$

$\Rightarrow\ O(n^2)$ bad!

by above, 3 recursive calls

$\Rightarrow\ T(n) = 3\,T(\frac{n}{2}) + O(n)$

$\Rightarrow\ O(n^{\log_2 3}) = \boxed{O(n^{1.59})}$

Toom-Cook's Alg'm ('63)

$n = 3$: given $a_0, a_1, a_2, b_0, b_1, b_2$

Compute $c_0 = a_0 b_0$

$c_1 = a_0 b_1 + a_1 b_0$

$n=3$: given $a_0, a_1, a_2, b_0, b_1, b_2$

Compute
$$
\begin{aligned}
c_0 &= a_0 b_0 \\
c_1 &= a_0 b_1 + a_1 b_0 \\
c_2 &= a_0 b_2 + a_1 b_1 + a_2 b_0 \\
c_3 &= a_1 b_2 + a_2 b_1 \\
c_4 &= a_2 b_2
\end{aligned}
$$

naive: 9 mults.

idea:
$$
\begin{cases}
d_0 = a_0 b_0 \\
d_1 = (a_2 + a_1 + a_0)(b_2 + b_1 + b_0) \\
d_2 = (4a_2 + 2a_1 + a_0)(4b_2 + 2b_1 + b_0) \\
d_3 = (9a_2 + 3a_1 + a_0)(9b_2 + 3b_1 + b_0) \\
d_4 = (16a_2 + 4a_1 + a_0)(16b_2 + 4b_1 + b_0)
\end{cases}
$$

$\{0,1,2,3,4\}$ (alternate: $\{0,1,2,-1,-2\}$)

i.e. $d_k = (\underline{a_2 k^2 + a_1 k + a_0})(\underline{b_2 k^2 + b_1 k + b_0})$ $(k = 0, \ldots, 4)$

$= c_4 k^4 + c_3 k^3 + c_2 k^2 + c_1 k + c_0$

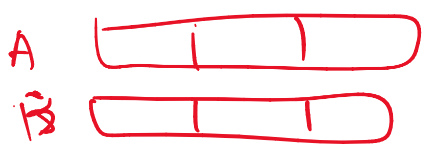
5 linear eq'ns in 5 vars

alternatively: interpolating 5 pts w. deg-5 polynomial

$\Rightarrow$ can recover $c_0, \ldots, c_4$ from $d_0, \ldots, d_4$

$\Rightarrow$ 5 mults.

for general $n$, 3-way D&C




$\Rightarrow \quad T(n) = \underline{5}\, T\left(\frac{n}{3}\right) + O(n)$

$\Rightarrow \quad O(n^{\log_3 5}) = \boxed{O(n^{1.41})}$

r-way D&C

$T(n) = (2r-1)\, T\left(\frac{n}{r}\right) + O(n)$

$$T(n) = \underline{(2r-1)}\, T\left(\frac{n}{r}\right) + O(n)$$

$$\Rightarrow O\left(n^{\log_r (2r-1)}\right)$$

$$\leq O\left(n^{\frac{\log 2r}{\log r}}\right)$$

$$= O\left(n^{1+\frac{1}{\log r}}\right)$$

$$= \boxed{O\left(n^{1+\delta}\right)}$$ for any const $\delta > 0$.

$\log n = o(n^{\delta})$

## Cooley-Tukey's Alg'm ('65)

forget about D&C (for now)

compute $d_k = \left(\sum_{j=0}^{N-1} a_j \lambda_k^j\right)\left(\sum_{j=0}^{N-1} b_j \lambda_k^j\right)$ for $k=0,\dots,N-1$ with $N = 2n-1$.

for some careful chosen values $\lambda_0, \dots, \lambda_{N-1}$

Know $d_k = \sum_{j=0}^{N-1} c_j \lambda_k^j$

Recover $c_0, \dots, c_{N-1}$ from $d_0, \dots, d_{N-1}$.

idea - choose $\boxed{\lambda_k = e^{-\frac{2\pi i}{N} k}}$

where $i = \sqrt{-1}$ complex numbers

(called $N^{th}$ roots of unity because $(\lambda_k)^N = (e^{-2\pi i})^k = 1^k = 1$)

$z^N = 1$.

Sol'n: compute $\boxed{\hat{a}_k = \sum_j a_j e^{-\frac{2\pi i}{N} kj}}$ ← $O(n \log n)$

called discrete Fourier transform

$\hat{b}_k = \sum_j b_j e^{-\frac{2\pi i}{N} kj}$ ← $O(n \log n)$

Called discrete Fourier transform (DFT)

$$\hat{b}_k = \sum_j b_j e^{-\frac{2\pi i}{N}kj}$$ ← $O(n\log n)$

$$d_k = \hat{a}_k \hat{b}_k$$ ← $O(n)$

inverse Fourier transform →

$$c_j = \sum_k d_k e^{\frac{2\pi i}{N}jk}$$ ← $O(n\log n)$

[ similar to continuous Fourier transform:

$$\hat{f}(t) = \int_{x=-\infty}^{\infty} f(x) e^{-2\pi i t x} dx$$

$$\widehat{f \circ g} = \hat{f} \cdot \hat{g}$$ ←

inverse transform:

$$f(x) = \int_{t=-\infty}^{\infty} \hat{f}(t) e^{2\pi i t x} dt$$ ]