September 2, 2020 10:48 AM

Convolution Problem Given $a_0, \ldots, a_{n-1}$, $b_0, \ldots, b_{n-1}$,

compute $c_0, \ldots, c_{2n-1}$ where

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

Last Time:

naive $O(n^2)$

Karatsuba $O(n^{1.59})$

Toom-Cook $O(n^{1+\delta})$ for any const $\delta > 0$

Cooley-Tukey's Alg'm via FFT

New Problem: DFT Given $a_0, \ldots, a_{N-1}$,

compute DFT $\hat{a}_0, \ldots, \hat{a}_{N-1}$ where

$$\hat{a}_k = \sum_j a_j e^{-\frac{2\pi i}{N} kj}$$

Fast Fourier Transform (FFT)

Solve this problem by binary D & C:

compute DFT of $a_0, a_2, a_4, \ldots, a_{N-2}$ recursively

& DFT of $a_1, a_3, a_5, \ldots a_{N-1}$ "

Combine ...

$\Rightarrow T(N) = 2T(\frac{N}{2}) + O(N)$

$\Rightarrow \boxed{O(N \log N)}$

$$\hat{a}_k = \sum_j a_{2j} e^{-\frac{2\pi i k}{N} 2j} + \sum_j a_{2j+1} e^{-\frac{2\pi i k}{N}(2j+1)}$$

$(N' = N/2)$

$$= \sum_j a_{2j} e^{-\frac{2\pi i k j}{N'}} + e^{-\frac{2\pi i k}{N}} \sum a_{2j+1} e^{-\frac{2\pi i k j}{N'}}$$

$(N' = N/2)$ $\quad - \underbrace{\sum_j a_{2j} e^{\frac{}{N'}}}_{\text{known}} + e^{\frac{}{N}} \underbrace{\sum_j a_{2j+1} e^{\frac{}{N'}}}_{\text{known}}$

(for $k > N'$: $e^{\frac{-2\pi i k j}{N'}} = e^{\frac{-2\pi i (k-N') j}{N'}}$ because $e^{-2\pi i} = 1$)

Note - precision issues ($O(\log n)$-bit suffices)
- or can work w. finite fields ...

Appl'0: other ops on polynomials like division ...

Appl 1: multiplying large integers

(n-bit)

$$\left(\sum a_j x^j\right)\left(\sum b_j x^j\right) = \sum c_j x^j \qquad (c_j: O(\log n)\text{-bit})$$

set $x = 2$: $\Rightarrow$ $O(n \log n)$ ops on $(\log n)$-bit #s

$\Rightarrow$ $O(n \log^2 n)$ bit ops

Schönhage-Strassen '71: $O(n \log n \log\log n)$

Fürer '07: $O(n \log n \cdot c^{\log^* n})$ ($\log^*$ = iterated log)

last yr: $O(n \log n)$

Appl 2: 3SUM for bounded integers

Given sets A, B, C of n numbers with $A, B \subseteq [u]$, $C \subseteq [2u]$

decide $\exists\, a \in A, b \in B, c \in C$ s.t. ~~$a + b + c = 0$~~

decide $\exists\, a \in A, b \in B, c \in C$ s.t. ~~$a+b+c=0$~~ $a+b=c$

$C \subseteq [2u]$ $\{0,\dots,u\}$ $(u \geq n)$

Soln: for each $c \in C$,
decide $\exists a$ s.t. $a \in A$, and $c-a \in B$

i.e. $\bigvee_{a=0}^{u} [a \in A] \wedge [c-a \in B]$

$[I] = \begin{cases} 1 & \text{if } I \text{ true} \\ 0 & \text{else} \end{cases}$

evaluate $\sum_{a=0}^{u} f_a \cdot g_{c-a}$

for all $c = 0 \dots 2u$ CONVOLUTION!

$\Rightarrow$ $O(u \log u)$ time
which is better when $u \ll n^2$.

[alternatively:
multiply $\sum_{a \in A} x^a$ & $\sum_{b \in B} x^b$
& check for coeffs of x^c]

Appl 3: String matching with "don't cares"

Given 2 strings
"pattern" $p_1 \cdots p_m \in (\Sigma \cup \{?\})^*$
"text" $t_1 \cdots t_n \in \Sigma^*$ $(m \leq n)$

decide if $\exists i$, $p_1 \cdots p_m$ "matches" $t_{i+1} \cdots t_{i+m}$

$t_{i+1} \cdots t_{i+m}$

$\forall j$, $p_j = t_{i+j}$ or $p_j = ?$
or $t_{i+j} = ?$

e.g. text: "algorithmisfun"
pattern: "thmis"
"h m ? s ? u"

naive alg'm: $O(mn)$ time

without "don't care": $O(n)$ time
standard string matching:
Knuth-Morris-Pratt, Rabin-Karp, ...

with "don't care":

Fischer-Paterson '74	$O(n \log m \log \lvert\Sigma\rvert)$
Indyk '98, Kalai '02, Cole-Hariharan '02	$O(n \log n)$ rand.

Simple (Det.) Alg'm by Clifford-Clifford '07:

let $\alpha_i = \begin{cases} 0 & \text{if } p_i = ? \\ 1 & \text{else} \end{cases}$ $\beta_i = \begin{cases} 0 & \text{if } t_i = ? \\ 1 & \text{else} \end{cases}$

Compute $c_i = \sum_{j=1}^{m} \alpha_j \beta_{i+j} (p_j - t_{i+j})^2$ for all i

Want match at position $i \iff c_i = 0$.

$c_i = \sum_{j=1}^{m} \alpha_j p_j^2 \beta_{i+j}$ (A_j, B_{i+j})

B_i !

$$c_i = \sum_{j=1}^{m} \alpha_j p_j \beta_{i+j} - 2\sum_{j=1}^{m} \alpha_j p_j \beta_{i+j} t_{i+j} + \sum_{j=1}^{m} \alpha_j \beta_{i+j} t_{i+j}^2$$

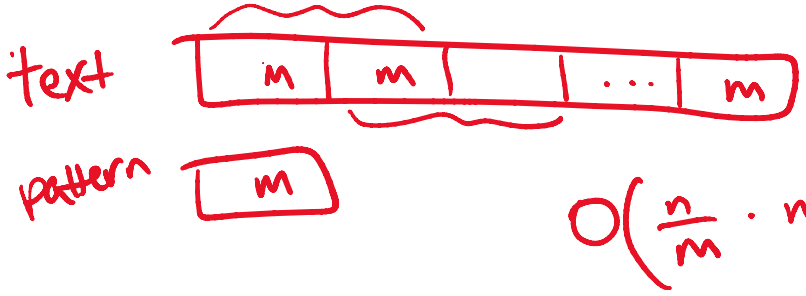
$A_j' = \alpha_j p_j$, $B_{i+j}' = \beta_{i+j} t_{i+j}$, $A_j'' = \alpha_j$, $B_{i+j}'' = \beta_{i+j} t_{i+j}^2$

$C_i = \sum_j A_j B_{i+j}$ is a CONVOLUTION!

reverse B

3 convolutions $\Rightarrow$ $O(n \log n)$ time

Rmk - can be improved to $O(n \log m)$



$O\left(\frac{n}{m} \cdot m \log m\right)$

Appl 4: String matching with mismatches

Given 2 strings $p_1 \cdots p_m \in \Sigma^*$
$t_1 \cdots t_n \in \Sigma^*$ & k,

decide $\exists i$ s.t. Hamming dist between $p_1 \cdots p_m$ & $t_{i+1} \cdots t_{i+m}$ is $\leq k$.

i.e. $|\{j : p_j \neq t_{i+j}\}| \leq k$.

i.e. $|\{j : P_j \neq t_{i+j}\}| \leq k$.

e.g. text: algorithmsfun
pattern: muffin $k = 3$

naive alg'm: $O(mn)$ time

faster?

next time: $O(|\Sigma| n \log n)$ time

What if $|\Sigma|$ large?

$\tilde{O}(n\sqrt{m})$ time

TO BE CONTINUED...