New Problem (DFT) Given $\langle a_0, \ldots, a_{N-1} \rangle$,

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

$$\hat{a}_k = \sum_{j=0}^{N-1} a_j e^{-\frac{2\pi i k}{N} j}$$

Alg'm for DFT:

idea - binary D & C.

recursively compute DFT of $\langle a_0, a_2, a_4, \ldots, a_{N-2} \rangle$
& DFT of $\langle a_1, a_3, a_5, \ldots, a_{N-1} \rangle$.

called FFT

combine...

Straightforward: for $k = 0, \ldots, N-1$,

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

$$= \underbrace{\sum_j a_{2j} e^{-\frac{2\pi i k j}{N/2}}}_{\text{already computed from recursion}} + \underbrace{\left(\sum_j a_{2j+1} e^{-\frac{2\pi i k}{N/2} j} \right)}_{\text{already computed from recursion}} e^{-\frac{2\pi i k}{N}}$$

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

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

$\Rightarrow \quad$ Convolution in $\boxed{O(n \log n)}$ time

---

Appl'n 1: Multiplying large integers
n-bit

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

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

Set $x=2$ $\Rightarrow$ $O(\ldots)$

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

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

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

Harvey-van der Hoeven '21: $O(n \log n)$ bit ops)

Appl'n 2: 3SUM for Bounded Integers

Given $A, B, C \subseteq \{0, \ldots, U-1\} = [U]$

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

Let $f_a = \begin{cases} 1 & \text{if } a \in A \\ 0 & \text{else} \end{cases}$ $\quad g_b = \begin{cases} 1 & \text{if } b \in B \\ 0 & \text{else} \end{cases}$

For each $c \in C$,

check iff $\exists\, a$ s.t. $a \in A$ & $c-a \in B$

iff $\exists\, a$ s.t. $f_a = 1$ & $g_{c-a} = 1$.

iff $\quad h_c = \underbrace{\sum_{a=0}^{U-1} f_a g_{c-a}}_{\text{Convolution!}} > 0$

$\Rightarrow$ $\boxed{O(U \log U)}$ time (good if $U \ll n^2$)

(alternative: multiply polynomial $\sum_{a \in A} x^a$ and $\sum_{b \in B} x^b$)

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

Given "pattern" string $p_1 p_2 \cdots p_m \in (\Sigma \cup \{'?'\})^*$

"text" string $t_1 t_2 \cdots t_n \in (\Sigma \cup \{'?'\})^*$ $(m < n)$

decide if pattern occurs in text

decide if pattern occurs in text

i.e. $\exists i,\ \forall j,\ p_j = t_{i+j}$ or $p_j = \text{'?'}$ or $t_{i+j} = \text{'?'}$

e.g. text: "algorithmisfun"

pattern: "th??s"

trivial: $O(mn)$ time

without "don't care": $O(n)$ time by standard string matching

(Knuth-Morris-Pratt, Rabin-Karp, ...)

with "don't care":

Fischer-Paterson '74: $O(n \log n \log |\Sigma|)$

Indyk '98: $O(n \log n)$ rand.

Kalai '02: $O(n \log n)$ rand.

Cole-Hariharan '02:

Simple Deterministic Alg'm by Clifford & Clifford '07:

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

match at position i

$$\iff \sum_{j=1}^{m} \alpha_j \beta_{i+j} (p_j - t_{i+j})^2 = 0$$

$$\iff \sum_{j=1}^{m} \underbrace{\alpha_j p_j^2}_{A_j} \underbrace{\beta_{i+j}}_{B_{i+j}} - 2 \sum_{j=1}^{m} \overbrace{\alpha_j p_j}^{A_j'} \overbrace{\beta_{i+j} t_{i+j}}^{B_{i+j}'}$$

$C_i =$

$$C_i = \underbrace{\sum_{j=1} A_j B_{i+j}}_{\text{Convolution!}} + \sum_{j=1}^{m} \underbrace{\alpha_j}_{A_j''} \underbrace{\beta_{i+j} t_{i+j}^2}_{B_{i+j}''} = 0$$

3 convolutions!

$$\Rightarrow \boxed{O(n \log n)} \text{ time}$$

(improved to $O(n \log m)$)