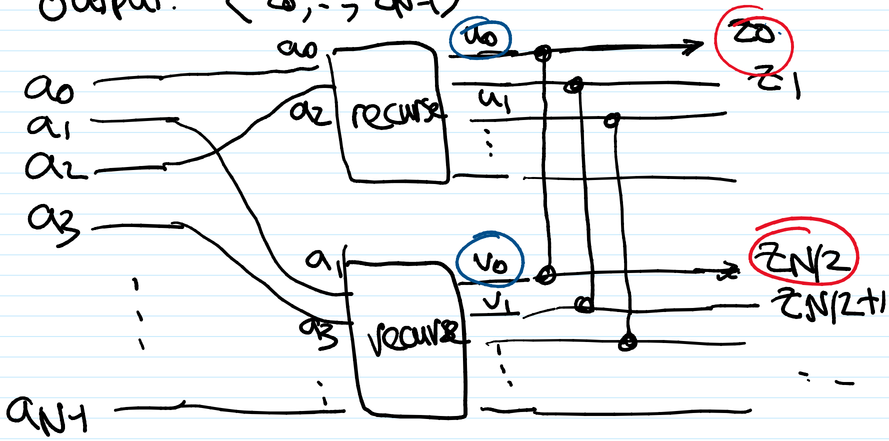
$$\Rightarrow \boxed{z_k = \begin{cases} u_k e^{-\frac{2\pi i k}{N}} + v_k & \text{for } k = 0, \ldots, \frac{N}{2}-1 \\ u_{k-\frac{N}{2}} e^{-\frac{2\pi i k}{N}} + v_{k-N/2} & \text{for } k = \frac{N}{2}, \ldots, N-1 \end{cases}}$$

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

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

input: $\langle a_0, \ldots, a_{N-1} \rangle$ $\qquad P(x) = \sum_{i=0}^{N-1} a_i x^i$

output: $\langle z_0, \ldots, z_{N-1} \rangle$



"butterfly network"

Rmk – above algm called Fast Fourier Transform (FFT)

Why? given $P(x) = \sum_{j=0}^{N-1} a_j x^j$,

compute $z_k = P\left(e^{-\frac{2\pi i k}{N}}\right) = \sum_{j=0}^{N-1} a_j e^{-\frac{2\pi i k j}{N}}$ $\forall k$.

(similar to Fourier transform

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

Alg'm for Problem B:

same idea, run backwards

in fact, inverse-FFT equiv. to FFT !!

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

$\Rightarrow$ orig problem of poly mult / convolution in $\boxed{O(n \log n)}$ time

$$\left(\widehat{f \circ g} = \hat{f}\,\hat{g} \right)$$

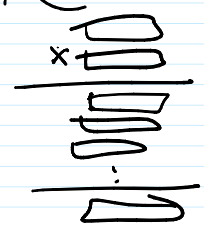
Rmk - N power of 2

- precision issues ...

⋮

Appl'n 1 - multiplying 2 large numbers (n-bit)

$$\left(\sum_{i=0}^{n-1} a_i 2^i\right)\left(\sum_{j=0}^{n-1} b_j 2^j\right)$$



elementary school method $O(n^2)$ time

FFT $O(n \log n)$ ops on $(\log n)$-bit numbers

Schönhage-Strassen '71 $O(n \log n \log\log n)$ bit ops.

Fürer '07 $O(n \log n\ 2^{O(\log^* n)})$ bit ops

Harvey, van der Hoeven '19 $O(n \log n)$ bit ops

Appl'n 2: pattern matching with "don't cares"

given 2 strings $a_1 a_2 \cdots a_m \in (\Sigma \cup \{?\})^*$ "pattern"

$b_1 b_2 \cdots b_n \in \Sigma^*$ "text" $(m \le n)$

decide whether $\exists k$, $a_1 a_2 \cdots a_m$ matches $b_{k+1} \cdots b_{k+m}$

i.e. $\forall i = 1, \ldots, m$ $a_i = b_{k+i}$ or $a_i = ?$

e.g. $\underline{\text{th??s}}$

algo$\underline{\text{rithmis}}$fun

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

without "don't care": $O(n)$ time (several known algms)

with "don't care":

Fischer–Paterson '74 $O(n \log n \log|\Sigma|)$ by convolution/FFT

Indyk '98
Kalai '02
Cole-Hariharan '02 } $O(n \log n)$ rand. or complicated

Alg'm by Clifford-Clifford '07:

let $z_j = \begin{cases} 0 & \text{if } a_j = \text{'?'} \\ 1 & \text{else} \end{cases}$

Compute $c_k = \sum_{j=1}^{m} z_j (a_j - b_{k+j})^2$

check $a_1 \cdots a_m = b_{k+1} \cdots b_{k+m}$?

there is match at k iff $c_k = 0$.

$$c_k = \underbrace{\sum_{j=1}^{m} z_j a_j^2}_{\text{precompute once}} - 2 \underbrace{\sum_{j=1}^{m} \overbrace{z_j a_j}^{A_j} \overbrace{b_{k+j}}^{B_{k+j}}}_{\text{convolution}} + \underbrace{\sum_{j=1}^{m} \overbrace{z_j}^{A_j'} \overbrace{b_{k+j}^2}^{B_{k+j}'}}_{\text{convolution}}$$

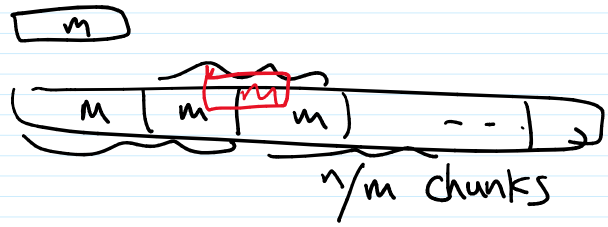
$$\sum_{j=1}^{m} A_j B_{k+j}$$

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

(can be reduced to $O(n \log m)$)

pattern

text



$$O\left(\frac{n}{m} \cdot (2m)\log(2m)\right) = O(n \log m)$$