3.) Other important FFTs

a.) Good-Thomas or prime-factor FFT

$n = km$, $\gcd(k, m) = 1$

$$DFT_{km} = P_{k,m} (DFT_k \otimes DFT_m) Q_{k,m}$$
$$= P_{k,m} (DFT_k \otimes I_m)(I_k \otimes DFT_m) Q_{k,m}$$

$Q_{k,m}: i \mapsto \left(m \left\lfloor \frac{i}{m} \right\rfloor \bmod k + ki \bmod m \right) \bmod n$

$P_{k,m}: i \mapsto \left(m\alpha \left\lfloor \frac{i}{m} \right\rfloor \bmod k + k\beta i \bmod m \right) \bmod k$

$\alpha = m^{-1} \bmod k$, $\beta = k^{-1} \bmod m$ > only exists since $\gcd(k, m) = 1$

good: no twiddles = less ops

bad: - extra permutation
- more costly permutations

b.) Rader FFT

handles $n = p$ prime

observation: $DFT_p = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & W_{p-1} \end{pmatrix}$

W_p is circulant up to permutations!

i.e. $W_{p-1} = \underbrace{P^T}_{\text{perm}} \cdot C_{p-1} \underbrace{P}_{\text{perm}}$ (C_{p-1}: circulant)

so $C_{p-1} = DFT_{p-1}^{-1} \cdot D_{p-1} \cdot DFT_{p-1}$

$$DFT_p = R_p^T (I_1 \oplus DFT_{p-1}^{-1})\left(\begin{pmatrix} 1 & 1 \\ 1 & -\frac{1}{p-1} \end{pmatrix} \oplus D\right)(I_1 \oplus DFT_{p-1}) R_p$$

$$R_p: i \mapsto \begin{cases} 0 & i = 0 \\ 2^{i-1} \bmod p & \text{else} \end{cases}$$

2 is such that $\{2^0, 2^1, \ldots, 2^{p-2}\}$ are mod p

$= \{1, 2, \ldots, p-1\}$

(in math: 2 is a generator of $\mathbb{Z}/p\mathbb{Z}$)

- Cooley-Tukey + Rader
  $\Rightarrow$ DFT$_n$ is $O(n \log(n))$ for all n

c.) Bluestein

trick: $kl = -\frac{1}{2}\left[(k-l)^2 - k^2 - l^2\right]$

$= -\frac{1}{2}(k-l)^2 + \frac{1}{2}k^2 + \frac{1}{2}l^2$

$$\text{DFT}_n = [\omega_n^{kl}] = [\omega_n^{-\frac{1}{2}(k-l)^2} \cdot \omega_{2n}^{k^2} \cdot \omega_{2n}^{l^2}]$$

$$= D \underbrace{[\omega_n^{-\frac{1}{2}(k-l)^2}]}_{\text{Toeplitz}_2} D \,,$$

with $D = \operatorname{diag}_{k=0}^{n-1}(\omega_{2n}^{k^2})$

↳ can be multiplied with using 2 DFTs of $\geq$ twice the size

d.) other algorithms
- Winograd (linear, $O(n)$, number of constant mults with constant $\notin \mathbb{Q}$)
- DFT via DCTs, DHT etc.
- split radix (Cooley-Tukey variant)

e.) Complexity

complexity measure L_c: $2 \leq c \leq \infty$
- complex add/sub counts 1
- complex constant mult $y = ax$ with $|a| \leq c$ counts 1

L_2 is the strictest, L_∞ the loosest (and most natural)

- $n = 2^k$: $L_2(\text{DFT}_n) \leq \frac{3}{2} n \log_2(n)$
- general n: $L_2(\text{DFT}_n) \leq 8n \log_2(n)$
  (Clausen/Baum, proof needs Bluestein)

lower bound:

Theorem (Nagenstern): $L_c(\Pi) \geq \log_c(\#\text{leaves}(\Pi))$, $c < \infty$

$\text{deg}(DFT_n) = n^{n/2} \Rightarrow L_c(DFT_n) \geq n \log_c(n)$

$\Rightarrow$ in the measure L_c, $c < \infty$, DFT_n is $\Omega(n \log(n))$

f.) Some properties of DFT_n:

$DFT_n = DFT_n^T$ (symmetric)

$(DFT_n)^{-1} = \frac{1}{n} \overline{DFT_n}$, $\overline{DFT_n}^{*} = [(\omega_n^{-1})^{ke}]_{k,e}$

$= [(\overline{\omega_n})^{ke}]_{k,e}$

$\Rightarrow \overline{DFT_n}^{*}$ algorithms are obtained from DFT_n algorithms by complex conjugation of all constants

$\cos(..) + \sin(..) \longrightarrow \cos(..) - \sin(..)$

$\overline{DFT_n} = DFT_n \cdot \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$

$DFT_n^2 = n \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$

4.) Complex arithmetic in transforms

$y = \Pi x$, Π a complex transform

$x, y \in \mathbb{C}^n$

$x = (x_0, \dots, x_{n-1})^T$, $y = (y_0, \dots, y_{n-1})^T$

- The computer provides only real arithmetic. Convert x, y to real vectors. Two formats:

split format: $\bar{x} = (re(x_0), \dots, re(x_{n-1}), im(x_0), \dots, im(x_{n-1}))$

interleaved format: $\bar{x} = (re(x_0), im(x_0), \dots, re(x_{n-1}), im(x_{n-1}))$

↑ More popular since local data access.

- Different formats imply different code.
- After format is chosen, convert complex ops into real ops.

Since $y = \Pi x$ is linear, every algorithm has only

- ~~constant mults~~: adds: $u = v + w$
- constant mults: $u = cv$, $c \in \mathbb{C}$ constant

Conversion: $u = u_1 + ju_2$, $v = v_1 + jv_2$ etc.

complex		real
$u = v + w$	$\longleftrightarrow$	$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$
1 add		2 adds
$u = cv$	$\longleftrightarrow$	$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} c_1 & -c_2 \\ c_2 & c_1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$
1 mult		4 mults
		2 adds
		6 ops

- complex mult: $c = c_1 + jc_2$

$$\begin{pmatrix} c_1 & -c_2 \\ c_2 & c_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} c_1+c_2 & & \\ & c_2 & \\ & & c_1-c_2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$$

if c is constant can be precomputed

$\Rightarrow$ 3 adds
3 mults
6 ops

[Note: if c is not constant we have 5 adds + 3 mults] — 1

- If c is a root-of-unity, $c = \cancel{\sin\alpha + j}\cos\alpha + j\sin\alpha$ is a 2×2 rotation

$$\Rightarrow \begin{pmatrix} c_1 & -c_2 \\ c_2 & c_1 \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$$

and $$\begin{pmatrix} c_1 & -c_2 \\ c_2 & c_1 \end{pmatrix} = \begin{pmatrix} 1 & \frac{c_1-1}{c_2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -c_2 & 1 \end{pmatrix} \begin{pmatrix} 1 & \frac{c_1-1}{c_2} \\ 0 & 1 \end{pmatrix}$$

again, for constant c, $(c_1-1)/c_2$ is precomputed

$\Rightarrow$ 3 adds
3 mults
6 ops

[1] Note: $\begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ y & 1 \end{pmatrix}$ are called "lifting steps" — 1

Special cases:

- $c = \pm j = \omega_4^{\mp 1}$: $\begin{pmatrix} c_1 & -c_2 \\ c_2 & c_1 \end{pmatrix} = \begin{pmatrix} 0 & \mp 1 \\ \pm 1 & 0 \end{pmatrix}$ no ops
- $c = \frac{1}{\sqrt{2}} \pm j\frac{1}{\sqrt{2}} = \omega_8^{\mp 1}$: $\begin{pmatrix} c_1 & -c_2 \\ c_2 & c_1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & \mp 1 \\ \pm 1 & 1 \end{pmatrix}$ 2 adds, 2 mults
4 ops