

# The Fast Fourier Transform

Darrell Whitley

(with thanks to Wim Bohm)

Colorado State AI Lab

Colorado State University

Consider a transform

$$S1 \xrightarrow{f} S1$$

$S1$  is some representation of some object (E.G. Let  $S1$  represent polynomials by coefficients)

$f$  is some function mapping members of  $S1$  to  $S1$ . (So we multiply two polynomials and obtain a polynomial and all are in coefficient form.)

Now consider

$\Pi$

$S_1 \xrightarrow{\quad} S_2$

$\Pi$  transforms one representation into another (E.G. transform polynomials in coefficient form to point value form).

Sometimes  $\Pi^{-1}(f'(\Pi(x)))$  is cheaper than  $f(x)$

$f$

$S_1 \xrightarrow{\quad} S_1$

$\Pi \quad f' \quad \Pi^{-1}$

$S_1 \xrightarrow{\quad} S_2 \xrightarrow{\quad} S_2 \xrightarrow{\quad} S_1$

Polynomials are normally expressed in coefficient form.

$$a(x) = x^3 + 7x^2 + 3x + 5$$

$$a(x) = \sum_{j=0}^{n-1} c_j x^j$$

A polynomial of degree  $n-1$  is also completely determined by its value at  $n$  different points.

So we might represent  $a(x)$  by

$$a(0) = 5, a(1) = 16, a(2) = 47, a(3) = 104$$

Or as a vector  $a' = [5, 16, 47, 104]$

This is a point-value representation.

Now consider

$$p(x) * q(x)$$

Multiplication in coefficient space has cost  $O(N^2)$  (Using Horner's Scheme).

Multiplying two polynomial is also known as "Convolution"

What about in point-value space? Multiplication has cost  $O(N)$ .

Just do pairwise multiplication of the vector components.

Great!

So can we convert the coefficient form to a point value form, do the multiplication, then convert back to coefficient form.

But at what cost?

Polynomials  $p$  and  $q$  have degree bound  $N$  while  $p * q$  has degree bound  $2N$

For this to work, we need to make the original Poly-coef forms of degree  $2N$ , so that the resulting vector is  $2N$ .

## STEPS

0. Double degree bound by  
adding high order zero coefficients

Pi

1. Poly-coef -----> point-value

multiply

2. point-value -----> point-value

pi-inverse

3. point-value -----> Poly-coef

What is the cost of the transform: Assume  $O(N \lg N)$

Step 0:  $N$

Step 1: 2 transforms, each  $O(2N \lg 2n)$

Step 2:  $2N$

Step 3: 1 transform, cost  $O(2N \lg 2n)$

TOTAL COST:  $3(2N \lg 2n) + 3N < 5(N \lg N) + 3N$

And  $6(N \lg N) + 3N < N^2$  for large  $N$

Great!

We need X's (points) with simple properties about  $x^2$  (e.g. 1,-1).

The complex roots of unity

$$\text{Imaginary numbers : } i = \sqrt{-1} \quad i^2 = 1$$

Complex Numbers: Part real, part imaginary,  $a + bi$

$$\text{Consider } (1 + i)^8 = 16$$

$$\text{Scale by division: } 16 = \sqrt{2}^8$$

$$\text{Yields a complex root of unity: } \frac{1}{\sqrt{2}} + \frac{i}{\sqrt{2}} = 1$$

$$\text{Note } (1 + i)^2 = 2i$$

$$\text{Note } (1 + i)^4 = -4$$

$$\text{Note } (1 + i)^8 = 16$$

A complex  $n^{\text{th}}$  root of unity  
is a complex number  $\omega$  such that  $\omega^n = 1$

There are  $n$  complex roots of unity

$e^{2\pi ik/n}$  for  $k = 0, 1, \dots, n - 1$

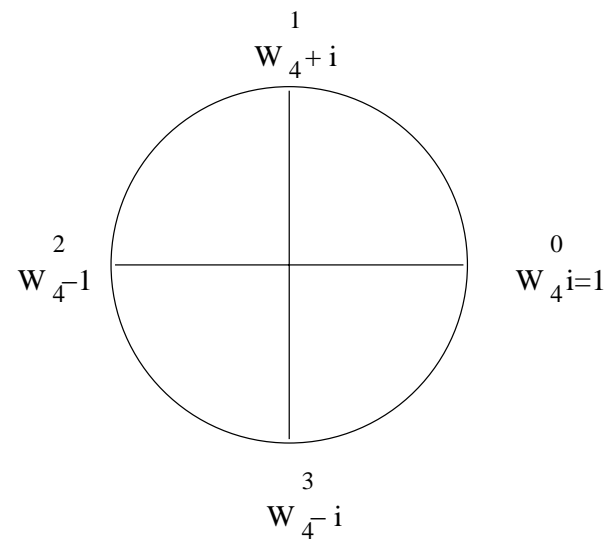
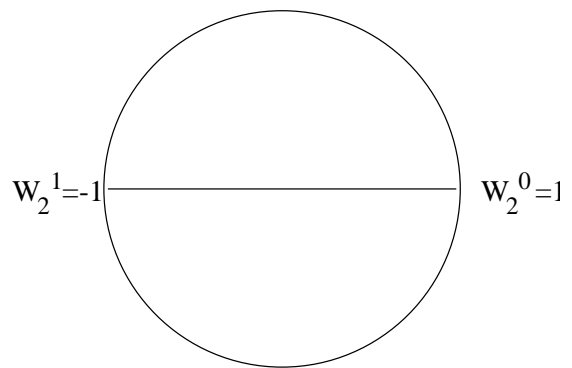
$$e^{iu} = \cos(u) + i \sin(u)$$

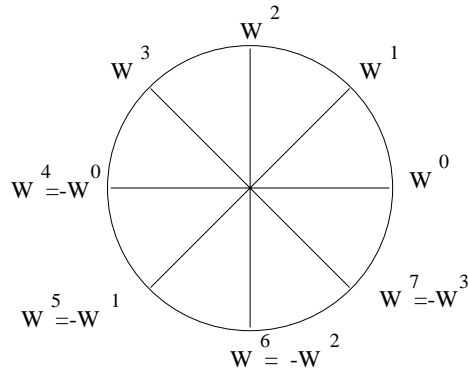
We denote the principle  $n^{\text{th}}$  root as

$$\omega_n = \omega_n^1 = e^{2\pi i/n}$$

All other roots are equally spaced around the circle of unit radius  
centered at the origin of the complex plane.

$n$  is a power of 2.





$$\omega_8^0 = 1$$

$$\omega_8^1 = \frac{1}{\sqrt{2}} + \frac{i}{\sqrt{2}}$$

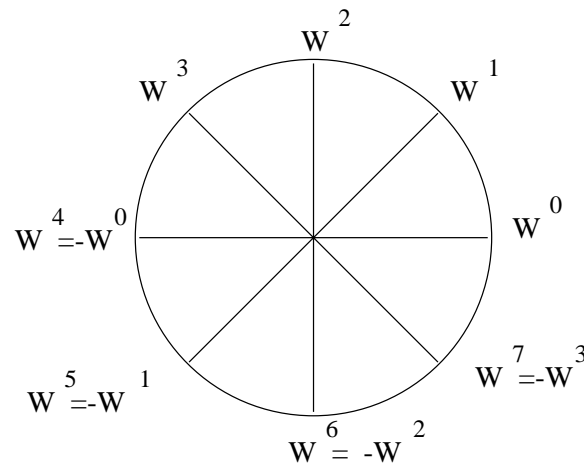
$$\omega_8^2 = \left(\frac{1}{\sqrt{2}} + \frac{i}{\sqrt{2}}\right)\left(\frac{1}{\sqrt{2}} + \frac{i}{\sqrt{2}}\right) = i = \omega_4^1$$

$$\omega_8^3 = \omega_8^2 \omega_8^1 = \omega_4^1 \omega_8^1 = i\left(\frac{1}{\sqrt{2}} + \frac{i}{\sqrt{2}}\right) = \frac{i}{\sqrt{2}} - \frac{1}{\sqrt{2}}$$

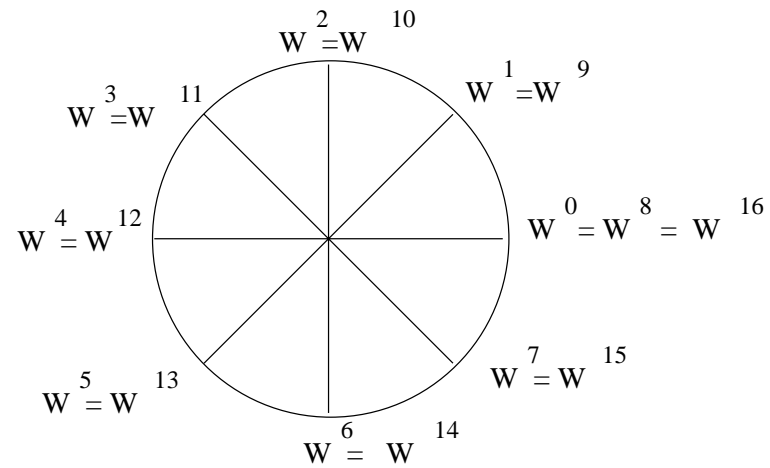
$$\omega_8^4 = \omega_8^2 \omega_8^2 = \omega_4^1 \omega_4^1 = ii = -1$$

## Properties of complex roots of unity

1. for non-negative integers  $n, k, d$        $\omega_{dn}^{dk} = \omega_n^k$
2.  $\omega_n^{n/2} = \omega_{n/2}^{n/2} \omega_2^1 = -1$
3.  $\omega_n^{k+n/2} = \omega_n^k \omega_n^{n/2} = \omega_n^k \omega_2^1 = -\omega_n^k$       And thus  $(\omega_n^k)^2 = (\omega_n^{k+n/2})^2$



$$4. \omega_n^m = \omega_n^m \omega_n^n = \omega_n^{m(\text{mod } n)}$$



$$5. \sum_{j=0}^{n-1} (\omega_n^k)^j = 0$$

7.

$$\omega^0 = \omega^0$$

$$\omega^{-1} = \omega^{n-1}$$

$$\omega^{-2} = \omega^{n-2}$$

$\vdots$

$$\omega^{-(n-1)} = \omega^1$$

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

index from 0  $k, j$  entry is  $\omega_n^{kj}$

Exponents form a multiplication table

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 2 & 4 & 6 \\ 0 & 3 & 6 & 9 \end{bmatrix}$$

Transform is a Vandemonde matrix.  $V_n$ , where  $y = aV_n$

Exponents

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 2 & 4 & 6 \\ 0 & 3 & 6 & 9 \end{bmatrix}$$

Exponents mod 4

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 2 & 0 & 2 \\ 0 & 3 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \omega_4^0 & \omega_4^0 & \omega_4^0 & \omega_4^0 \\ \omega_4^0 & \omega_4^1 & \omega_4^2 & \omega_4^3 \\ \omega_4^0 & \omega_4^2 & \omega_4^0 & \omega_4^2 \\ \omega_4^0 & \omega_4^3 & \omega_4^2 & \omega_4^1 \end{bmatrix}$$

$$\omega_4^0 = 1 \quad \omega_4^1 = i \quad \omega_4^2 = (i)^2 = -1 \quad \omega_4^3 = \omega_4^2 \omega_4^1 = -i$$

Thus

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

$$V = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

$$\bar{V} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}$$

$\bar{V}$  reverses the columns of  $V$  from 1 to  $n - 1$

$$\frac{1}{4}V\bar{V} = \frac{1}{4}\bar{V}V =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In general the inverse of  $V_n$  for  $j, k = 0, 1, \dots, n - 1$

$$V_n^{-1}(j, k) = \omega_n^{-kj} / n$$

$$a_j = \frac{1}{n} \sum_{k=0}^{n-1} y_k \omega_n^{-kj}$$

$$y_k = \sum_{j=0}^{n-1} a \omega_n^{kj}$$

Convolution  $a \otimes b = DFT_{2n}^{-1}(DFT_{2n}(a) * DFT_{2n}(b))$

where  $a$  and  $b$  are padded to length  $2n$ . We use pairwise multiplication of the vectors.

## Recursive FFT

Divide  $A(x)$  into odd/even components

$$A^{[0]}(x) = a_0 + a_2x + a_4x^2 + \dots + a_{n-2}x^{n/2-1}$$
$$A^{[1]}(x) = a_1 + a_3x + a_5x^2 + \dots + a_{n-1}x^{n/2-1}$$

Now  $A(x) = A^{[0]}(x^2) + xA^{[1]}(x^2)$

$$A(x) = 5 + 3x + 7x^2 + 1x^3$$

$$A^{[0]} = 5 + 7x$$

$$A^{[1]} = 3 + 1x$$

$$A(x) = A^{[0]}(x^2) + xA^{[1]}(x^2)$$

$$A(x) = [5(x^0)^2 + 7(x^1)^2] + x[3(x^0)^2 + 1(x^1)^2]$$

$$A(x) = 5 + 7x^2 + 3x + 1x^3$$

$$= 5 + 3x + 7x^2 + 1x^3$$

## Recursive-FFT(a)

1.  $n \leftarrow \text{length}(a)$
2. if  $n = 1$  then return  $a$
3.  $\omega_n \leftarrow e^{2\pi i/n}$        $/*\omega_n^1 */$
4.  $\omega \leftarrow 1$        $/*\omega_n^0 = 1*/$
5.  $a^{[0]} \leftarrow (a_0, a_2, \dots, a_{n-2})$
6.  $a^{[1]} \leftarrow (a_1, a_3, \dots, a_{n-1})$
7.  $y^{[0]} \leftarrow \text{Recursive-FFT}(a^{[0]})$
8.  $y^{[1]} \leftarrow \text{Recursive-FFT}(a^{[1]})$
9. for  $k \leftarrow 0$  to  $(n/2 - 1)$
10.       $y_k \leftarrow y_k^{[0]} + \omega y_k^{[1]}$
11.       $y_{k+(n/2)} \leftarrow y_k^{[0]} - \omega y_k^{[1]}$
12.       $\omega \leftarrow \omega \omega_n$
13. return  $y$

What was computed:

$$y^{[0]} = A^{[0]}(\omega_{n/2}^k)$$

$$y^{[1]} = A^{[1]}(\omega_{n/2}^k)$$

But since  $\omega_{n/2}^k = (\omega_{n/2}^k)^2 = \omega_n^{2k}$

$$y^{[0]} = A^{[0]}(\omega_n^{2k})$$

$$y^{[1]} = A^{[1]}(\omega_n^{2k})$$

Using this

$$y_k = y_k^{[0]} + \omega y_k^{[1]} \quad y_k = A^{[0]}(\omega_n^{2k}) + \omega_n^k A^{[1]}(\omega_n^{2k}) \quad y_k = A(\omega_n^k)$$

**Example**  $2x^3 + 5x^2 + 3x + 1$       $a = \langle 1 \ 3 \ 5 \ 2 \rangle$

Recursive-FFT( $a$ )

$\omega = 1, \omega_4 = i$

$a^{[0]} = \langle 1 \ 5 \rangle$

$a^{[1]} = \langle 3 \ 2 \rangle$

$y^{[0]} = \text{Recursive-FFT}(1, 5) = 6 \quad -4$

$y^{[1]} = \text{Recursive-FFT}(3, 2) = 5 \quad 1$

for  $k = 0$  to 1

$k=0 \quad \begin{bmatrix} y_0 \leftarrow 6 + \omega(5) \\ y_2 \leftarrow 6 - \omega(5) \end{bmatrix}$

$\omega = \omega\omega_n = 1\omega_n = i$

$k=1 \quad \begin{bmatrix} y_1 \leftarrow -4 + \omega_n(1) \\ y_3 \leftarrow -4 - \omega_n(1) \end{bmatrix}$

return  $y = \langle 11, \quad -4 + i, \quad 1, \quad -4 - i \rangle$

Recursive-FFT(1, 5)

$$a^{[0]} = 1$$

$$a^{[1]} = 5$$

$$y^{[0]} = 1$$

$$y^{[1]} = 5$$

$$y_0 = 1 + \omega 5$$

$$y_1 = 1 - \omega 5$$

return  $y = \langle 6, -4 \rangle$

Recursive-FFT(3, 2)

$$a^{[0]} = 3$$

$$a^{[1]} = 2$$

$$y^{[0]} = 3$$

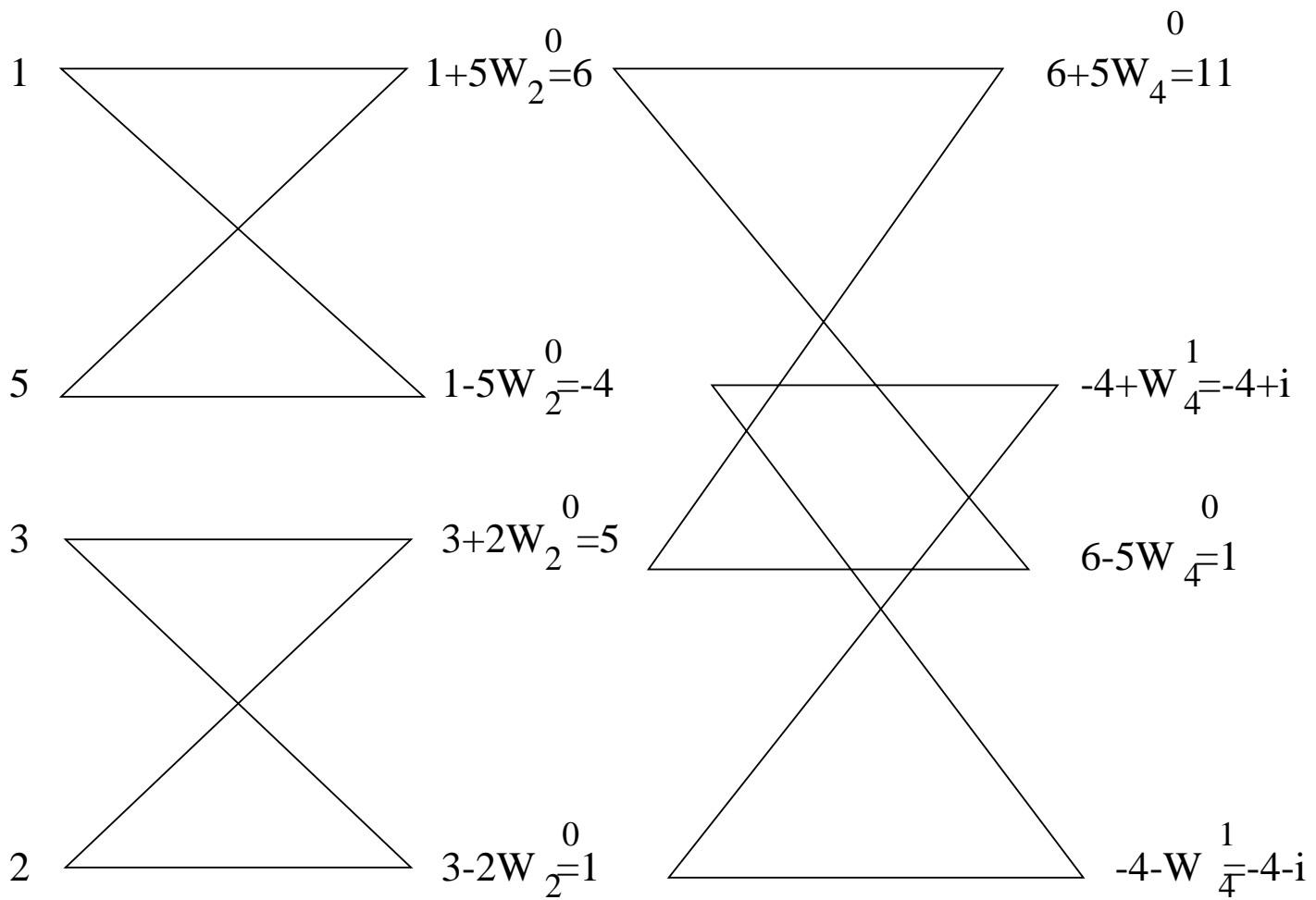
$$y^{[1]} = 2$$

$$y_0 = 3 + \omega 2$$

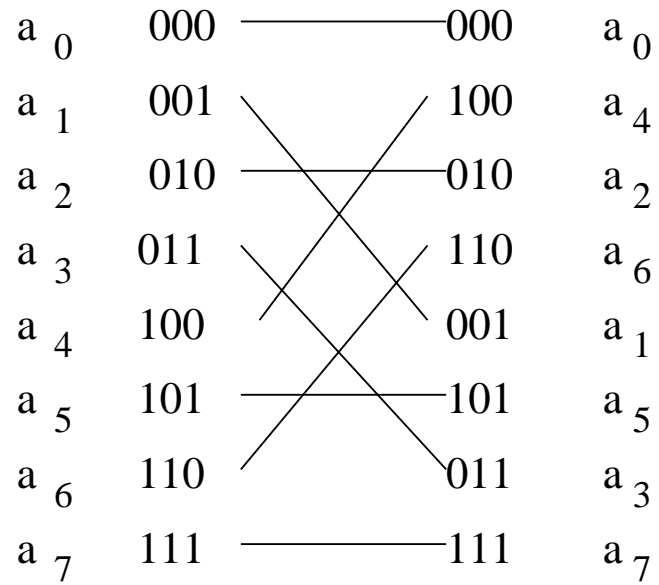
$$y_1 = 3 - \omega 2$$

return  $y = \langle 5, 1 \rangle$





# Bit Reverse



even :  $a_0 a_4 a_2 a_6$  odd:  $a_1 a_3 a_5 a_7$

Iterative-FFT1(a)

From Cormen 2001

1. Bit-Reverse-Copy(a,A)
2.  $n = \text{length}(a)$  /\* n is a power of 2 \*/
3. for  $s \leftarrow 1$  to  $\lg n$  /\* lg n loop \*/
4. {  $m \leftarrow 2^s$
5.  $\omega_m \leftarrow e^{2\pi i/m}$
6. for  $k \leftarrow 0$  to  $n - 1$  by  $m$
7. {  $\omega \leftarrow 1$
8. for  $j \leftarrow 0$  to  $(m/2 - 1)$
9. {  $t \leftarrow \omega A[k + j + m/2]$
10.  $u \leftarrow A[k + j]$
11.  $A[k + j] \leftarrow u + t$
12.  $A[k + j + m/2] \leftarrow u - t$
13.  $\omega \leftarrow \omega \omega_m$  } }

Iterative-FFT2(a)

From Cormen 1990

1. Bit-Reverse-Copy(a,A)
2.  $n = \text{length}(a)$  /\* n is a power of 2 \*/
3. for  $s \leftarrow 1$  to  $\lg n$  /\* lg n loop \*/
4. {  $m \leftarrow 2^s$
5.  $\omega_m \leftarrow e^{2\pi i/m}$
6.  $\omega \leftarrow 1$
7. for  $j \leftarrow 0$  to  $(m/2 - 1)$
8. { for  $k \leftarrow j$  to  $n - 1$  by  $m$
9. {  $t \leftarrow \omega A[k + m/2]$
10.  $u \leftarrow A[k]$
11.  $A[k] \leftarrow u + t$
12.  $A[k + m/2] \leftarrow u - t$
13.  $\omega \leftarrow \omega \omega_m$  }

## Example

$a_0$   $a_1$   $a_2$   $a_3$   $a_4$   $a_5$   $a_6$   $a_7$

bit reverse     $a_0$   $a_4$   $a_2$   $a_6$   $a_1$   $a_5$   $a_3$   $a_7$

for  $s = 1$  to  $3$  /\* lg 8 \*/

$s = 1$  /\* first loop \*/

$m = 2^s = 2$

$\omega_n$

$\omega = 1$

  for  $j \leftarrow 0$  to  $(m/2 - 1 = 0)$

    for  $k = 0$  to  $7$  by  $2$

In these case,  $j = 0$ ,  $m/2 = 1$

Therefore  $A[k] = A[k] + A[k+1]$

$$k=0 \quad A[0] = A[0] + \omega A[1]$$

$$A[1] = A[0] - \omega A[1]$$

$$k=2 \quad A[2] = A[2] + \omega A[3]$$

$$A[3] = A[2] - \omega A[3]$$

$$k=4 \quad A[4] = A[4] + \omega A[5]$$

$$A[5] = A[4] - \omega A[5]$$

$$k=6 \quad A[6] = A[6] + \omega A[7]$$

$$A[7] = A[6] - \omega A[7]$$

```

S = 2
M = 22 = 4
ωn reset
ω reset
for j ← 0 to (4/2 - 1 = 1)
    for k ← 0 to 7 by 4
        j=0   k=0   A[0] = A[0] + ω0A[2]
              A[2] = A[0] - ω0A[2]
        k=4   A[4] = A[4] + ω0A[6]
              A[6] = A[4] - ω0A[6]
        j=1   k=1   A[1] = A[1] + ω'nA[3]
              A[3] = A[1] - ω'nA[3]
        k=5   A[5] = A[5] + ω'nA[7]
              A[7] = A[5] - ω'nA[7]

```

```
S = 3
M = 8
wn reset
w reset

for j ← 0 to 3
  for k ← 0 to 7 by 8
```

$$j=0 \quad k=0 \quad A[0] = A[0] + \omega^0 A[4]$$

$$A[4] = A[0] - \omega^0 A[4]$$

$$j=1 \quad k=1 \quad A[1] = A[1] + \omega^1 A[5]$$

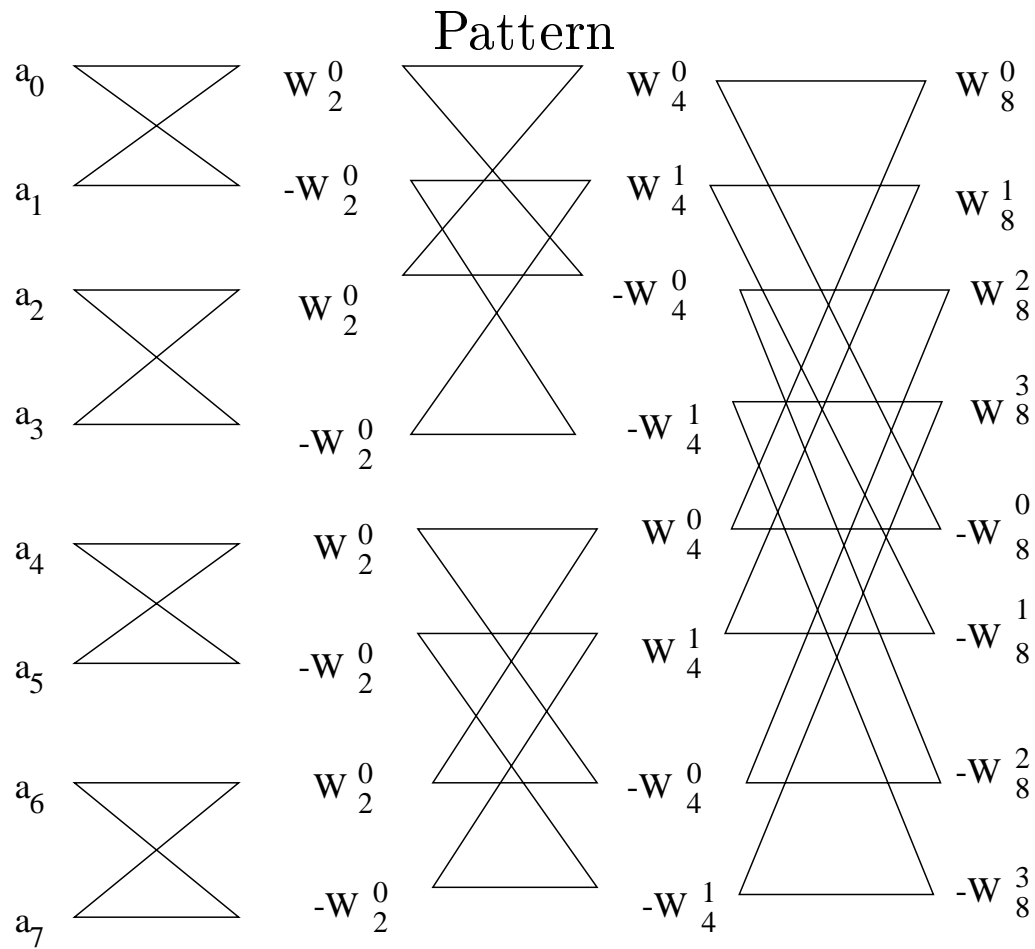
$$A[5] = A[1] - \omega^1 A[5]$$

$$j=2 \quad k=2 \quad A[2] = A[2] + \omega^2 A[6]$$

$$A[6] = A[2] - \omega^2 A[6]$$

$$j=3 \quad k=3 \quad A[3] = A[3] + \omega^3 A[7]$$

$$A[7] = A[3] - \omega^3 A[7]$$



$$\omega_4^1 = \omega_2^2 \quad \omega_4^1 = \omega_8^2$$

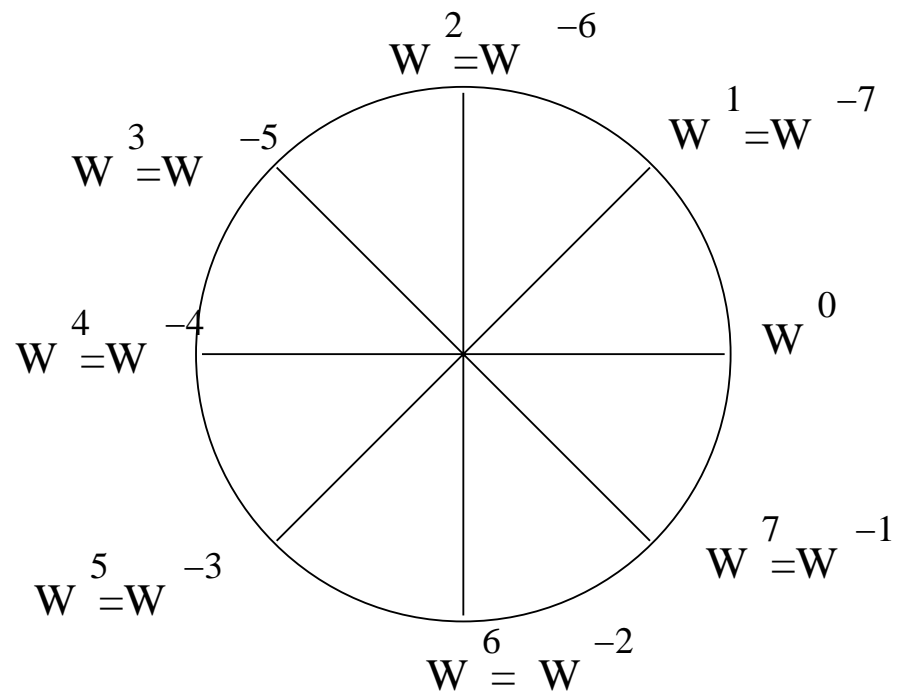
## Interpolation at the Roots of Unity The Inverse FFT

Given  $n$  pairs of point-values  $(\omega_n^0, v_0), (\omega_n^1, v_1), \dots, (\omega_n^{n-1}, v_{n-1})$  find the polynomial

$$r(x) = r_0 + r_1x + r_2x^2 + \dots + r_{n-1}x^{n-1}$$

Theorem

$$\begin{aligned} \omega_n^{inverse} &= \omega_n^0 \omega_n^{-1} \omega_n^{-2} \dots \omega_n^{-(n-1)} \\ &= \omega_n^0 \omega_n^{n-1} \omega_n^{n-2} \dots \omega_n^1 \end{aligned}$$



But we know the matrix  $V^{-1}$  is the same as  $V$ , but with column 1 to  $N - 1$  reversed.

So if we use  $V$  instead of  $V^{-1}$  to compute the inverse, we get the correct result with elements 1 to  $N - 1$  in reverse order.

E.G. using  $V$

$$\begin{bmatrix} 11 & 1 & 1 & 1 \\ -4+i & 1 & -i & -1 \\ 1 & 1 & -1 & 1 \\ -4-i & 1 & -i & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} = \begin{matrix} 11 - 4 + i + 1 - 4 - i = 4 \\ 11 - 4i - 1 - 1 - 4i - 1 = 8 \\ 11 + 4 - i + 1 + 4 + i = 20 \\ 11 + 4i + 1 - 1 - 4i + 1 = 12 \end{matrix}$$

Normalize  $1/4(4 \ 8 \ 20 \ 12) = 1 \ 2 \ 5 \ 3$

reverse 1 to N-1 = 1 3 5 2