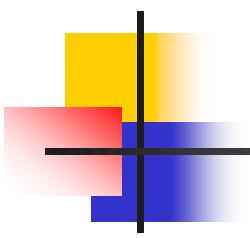


FFT(Fast Fourier Transform)



■ FFT

$$A(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1}$$

$$B(x) = b_0 + b_1x + \dots + b_{n-1}x^{n-1}$$

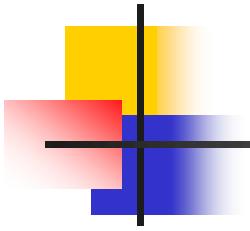
$$C(x) = A(x)B(x)$$

$$= c_0 + c_1x + c_2x^2 + \dots + c_{2n-3}x^{2n-3} + c_{2n-2}x^{2n-2}$$

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

•**Coefficient representation:**

How to evaluate $A(x_0)$?



- Horner's rule:

$$A(x_0) = a_0 + x_0(a_1 + x_0(a_2 + \dots + x_0(a_{n-2} + x_0 a_{n-1})\dots))$$

$\theta(n)$

- Point-value representation:

A point-value representation of a polynomial $A(x)$ of degree-bound n is a set of n point-value pairs $\{(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})\}$, where $y_k = A(x_k)$

$$A(x): \{(x_0, y_0), (x_1, y_1), \dots, (x_{2n-1}, y_{2n-1})\}$$

$$B(x): \{(x_0, y'_0), (x_1, y'_1), \dots, (x_{2n-1}, y'_{2n-1})\}$$

$$C(x) = A(x)B(x)$$

$$C(x): \{(x_0, y_0 y'_0), (x_1, y_1 y'_1), \dots, (x_{2n-1}, y_{2n-1} y'_{2n-1})\}$$

■ Thm1

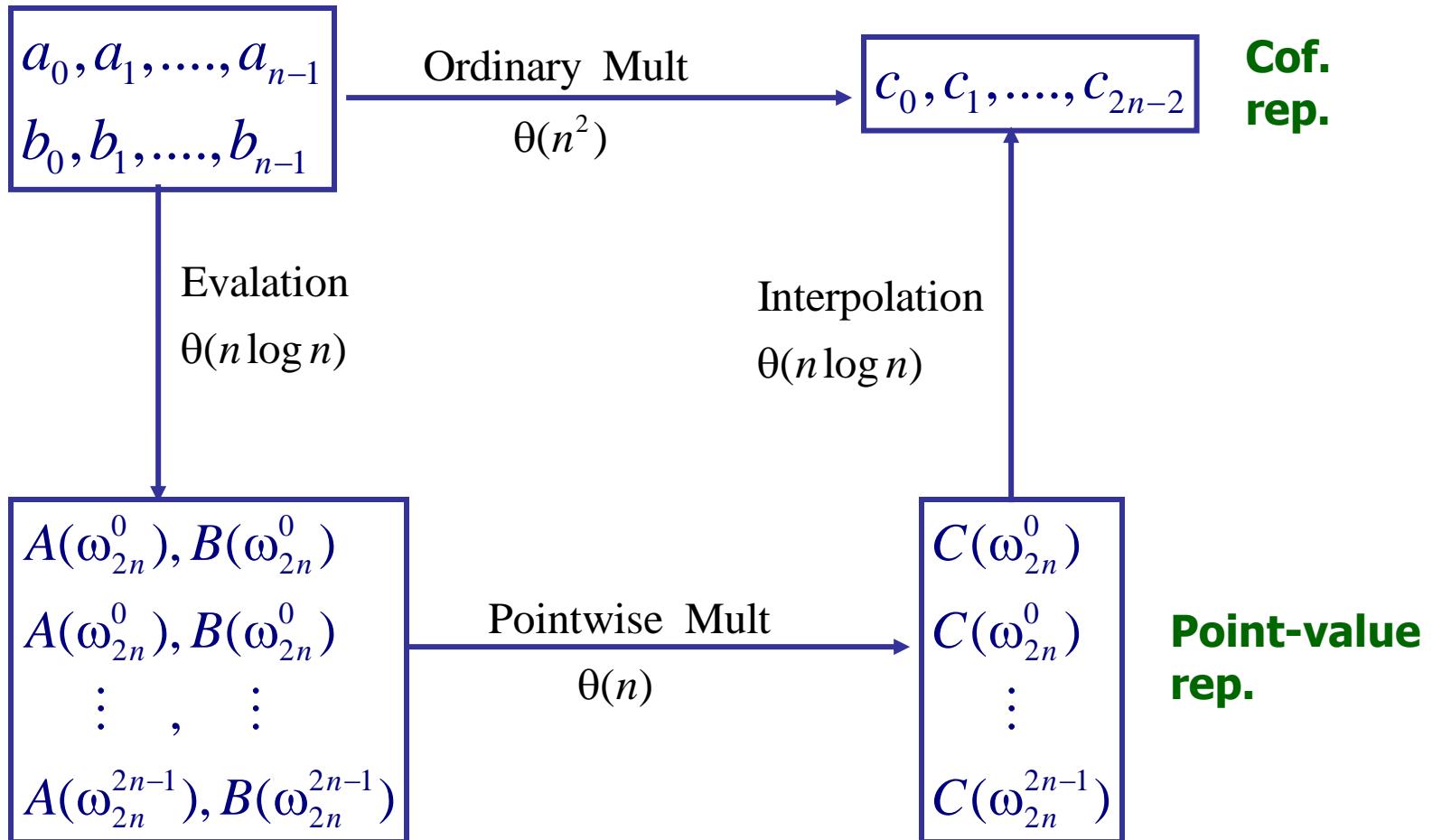
For any set $\{(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})\}$ of n point-value pairs, there is a unique poly $A(x)$ of degree $\leq n-1$, such that $y_k = A(x_k)$ for $k=0, 1, \dots, n-1$

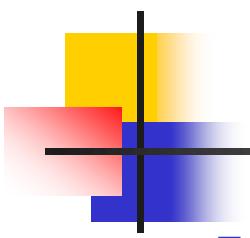
Pf:

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Vandermonde matrix has determinant:

$$\prod_{0 \leq j < k \leq n-1} (x_k - x_j)$$





■ Thm2

The product of 2 polynomials of deg-bound n can be computed in time $\theta(n \log n)$, with both the input and output representation in coefficient form

Complex roots of unity:

$$\omega^n = 1, \quad e^{2\pi i k/n} \quad \text{for } k=0, 1, \dots, n-1 \quad e^{iu} = \cos u + i \sin u$$

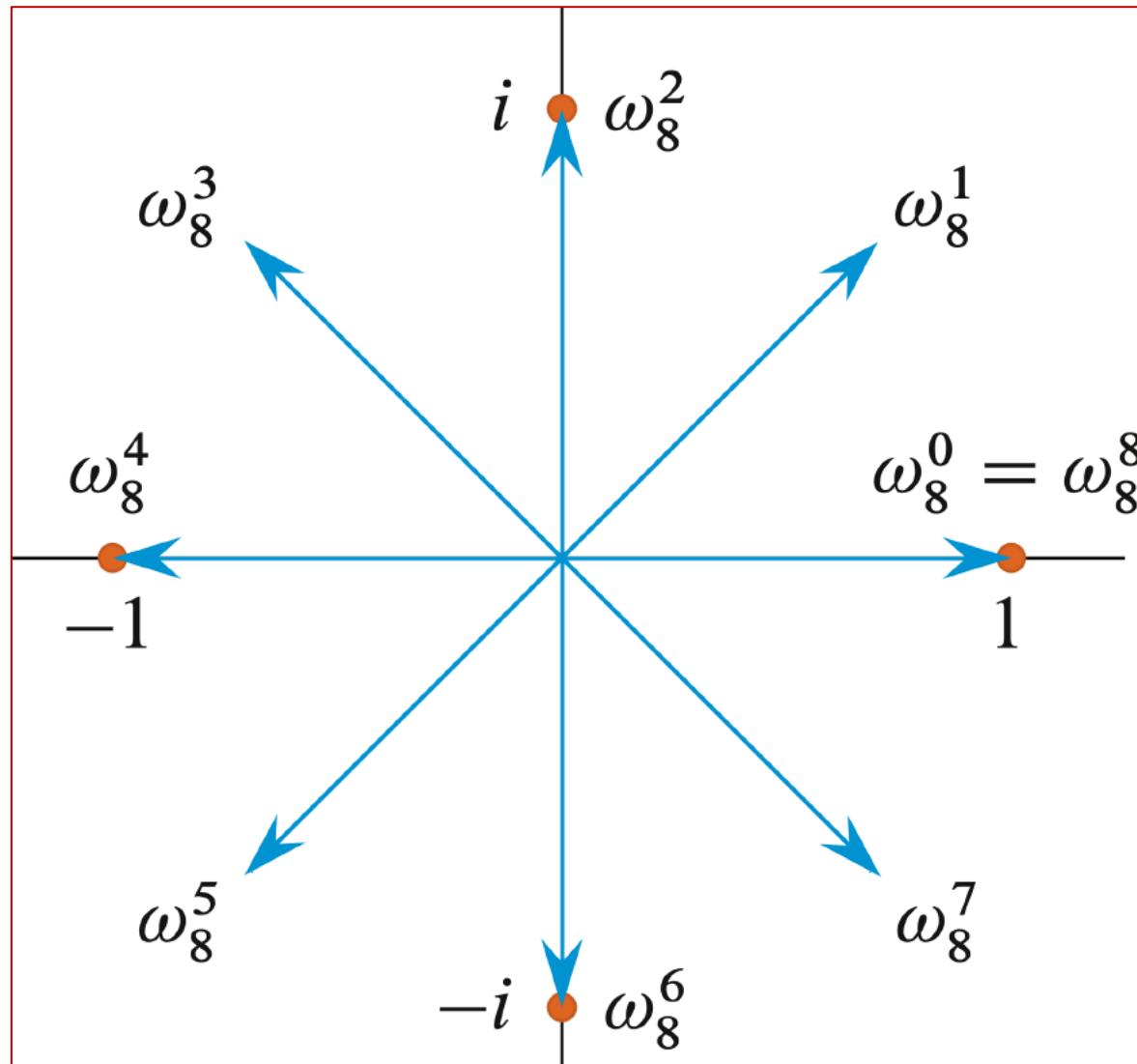
$\omega_n = e^{2\pi i/n}$, the principal n-th root of unity

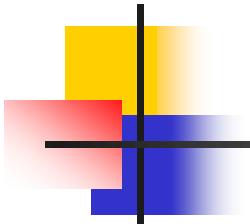
$$\omega_n^0, \omega_n^1, \omega_n^2, \dots, \omega_n^{n-1}$$

$$\omega_n^n = ?, \quad \omega_n^0 = 1$$

$$\omega_n^j \omega_n^k = \omega_n^{(j+k) \bmod n}$$

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





- Lemma 3 (Cancellation Lemma)

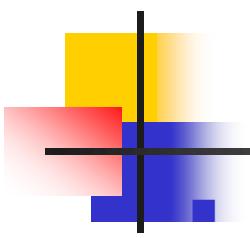
n, k, d : non-negative integers, $\omega_{dn}^{dk} = \omega_n^k$

Pf:

$$\omega_{dn}^{dk} = (e^{2\pi i / dn})^{dk} = (e^{2\pi i / n})^k = \omega_n^k$$

- Cor. 4 n : even positive integer

$$\omega_n^{n/2} = \omega_2 = -1$$



■ Lemma 5 (Halving lemma)

n: even positive integer

The squares of the n complex n-th roots of unity are n/2 complex (n/2)th roots of unity.

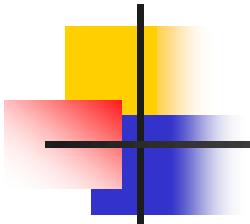
Pf: $(\omega_n^k)^2 = \omega_n^{2k} = \omega_{n/2}^k, k \in \mathbb{Z}^+ \cup \{0\}$
 $(\omega_n^{k+n/2})^2 = \omega_n^{2k+n} = \omega_{n/2}^k$
 $\Rightarrow \omega_n^k$ and $\omega_n^{k+n/2}$ have the same square

■ Lemma 6 (Summation lemma)

$n \in \mathbb{Z}^+, k \in \mathbb{Z}^+ \cup \{0\}, n \nmid k, \sum_{j=0}^{n-1} (\omega_n^k)^j = 0$

Pf:

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



■ DFT

Evaluate $A(x) = \sum_{j=0}^{n-1} a_j x^j$ at $\omega_n^0, \omega_n^1, \dots, \omega_n^{n-1}$,

Assume n is a power of 2

Let $a = \langle a_0, a_1, \dots, a_{n-1} \rangle$, and $y_k = A(\omega_n^k) = \sum_{j=0}^{n-1} a_j \omega_n^{kj}$

$y = \langle y_0, y_1, \dots, y_{n-1} \rangle$ is the DFT of the coefficient

vector $a = \langle a_0, a_1, \dots, a_{n-1} \rangle$,

$y = \text{DFT}_n(a)$

- Interpolation at the complex roots of unity:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

$$y = V_n a, \quad (V_n)_{k,j} = \omega_n^{kj}$$

$$a = V_n^{-1} y$$

■ Thm 7

$$j, k = 0, 1, \dots, n-1, \quad (V_n^{-1})_{j,k} = \omega_n^{-kj} / n$$

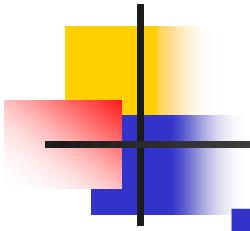
Pf:

$$\begin{aligned} V_n^{-1} V_n &= I_n, \quad (V_n^{-1} V_n)_{j,j'} = \sum_{k=0}^{n-1} (V_n^{-1})_{j,k} (V_n)_{k,j'} \\ &= \sum_{k=0}^{n-1} \frac{1}{n} \omega_n^{-kj} \omega_n^{kj'} \\ &= \frac{1}{n} \sum_{k=0}^{n-1} \omega_n^{k(j'-j)} \quad \text{-----(*)} \end{aligned}$$

if $j=j'$, then $(*)=1$, if $j \neq j'$, then by lemma 6, $(*)=0$

$\because -(n-1) < j'-j < n-1$, and $n \nmid (j'-j)$

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



FFT

$$A^{[0]}(x) = a_0 + a_2 x + a_4 x^2 + \dots + a_{n-2} x^{n/2-1}$$

$$A^{[1]}(x) = a_1 + a_3 x + a_5 x^2 + \dots + a_{n-1} x^{n/2-1}$$

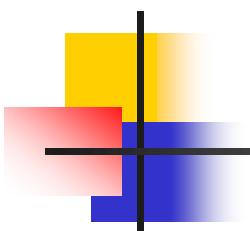
$$(*) \quad A(x) = A^{[0]}(x^2) + x A^{[1]}(x^2)$$

Thus evaluating $A(x)$ at $\omega_n^0, \omega_n^1, \dots, \omega_n^{n-1}$ reduce to

1. evaluating $A^{[0]}(x)$ and $A^{[1]}(x)$ at

$$(\omega_n^0)^2, (\omega_n^1)^2, \dots, (\omega_n^{n-1})^2$$

2. combining the results according to $(*)$



Let $\begin{cases} y_k^{[0]} = A^{[0]}(\omega_{n/2}^k) \\ y_k^{[1]} = A^{[1]}(\omega_{n/2}^k) \end{cases}$

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

$$\begin{aligned} y_k &= A(\omega_n^k) = A^{[0]}(\omega_n^{2k}) + \omega_n^k A^{[1]}(\omega_n^{2k}) \\ &= A^{[0]}(\omega_{n/2}^k) + \omega_n^k A^{[1]}(\omega_{n/2}^k) \\ &= y_k^{[0]} + \omega_n^k y_k^{[1]} \end{aligned}$$

$$\begin{aligned} y_{k+n/2} &= A(\omega_n^{k+n/2}) = A^{[0]}(\omega_n^{2k+n}) + \omega_n^{k+n/2} A^{[1]}(\omega_n^{2k+n}) \\ &= A^{[0]}(\omega_{n/2}^k) + \omega_n^{k+n/2} A^{[1]}(\omega_{n/2}^k) \\ &= y_k^{[0]} + \omega_n^{k+n/2} y_k^{[1]} = y_k^{[0]} - \omega_n^k y_k^{[1]} \end{aligned}$$

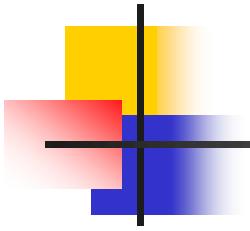
Recursive-FFT(a)

```
{ n=length[a]; /* n: power of 2 */  
if n=1 the return a;  
 $\omega_n = e^{2\pi i / n};$   
 $\omega=1$   
 $a^{[0]} = (a_0, a_2, \dots, a_{n-2});$   
 $a^{[1]} = (a_1, a_3, \dots, a_{n-1});$   
 $y^{[0]} = \text{Recursive-FFT}(a^{[0]});$   
 $y^{[1]} = \text{Recursive-FFT}(a^{[1]});$   
for k=0 to (n/2 - 1) do  
{  
     $y_k = y_k^{[0]} + \omega y_k^{[1]};$   
     $y_{k+n/2} = y_k^{[0]} - \omega y_k^{[1]};$   
     $\omega=\omega\omega_n;$   
}  
}
```

$$y_k = y_k^{[0]} + \omega_n^k y_k^{[1]}$$
$$y_{k+n/2} = y_k^{[0]} - \omega_n^k y_k^{[1]}$$

$$T(n) = 2T(n/2) + \Theta(n)$$

$$= \Theta(n \log n)$$



■ Thm 8 (Convolution thm)

n : power of 2

a, b : vectors of length n

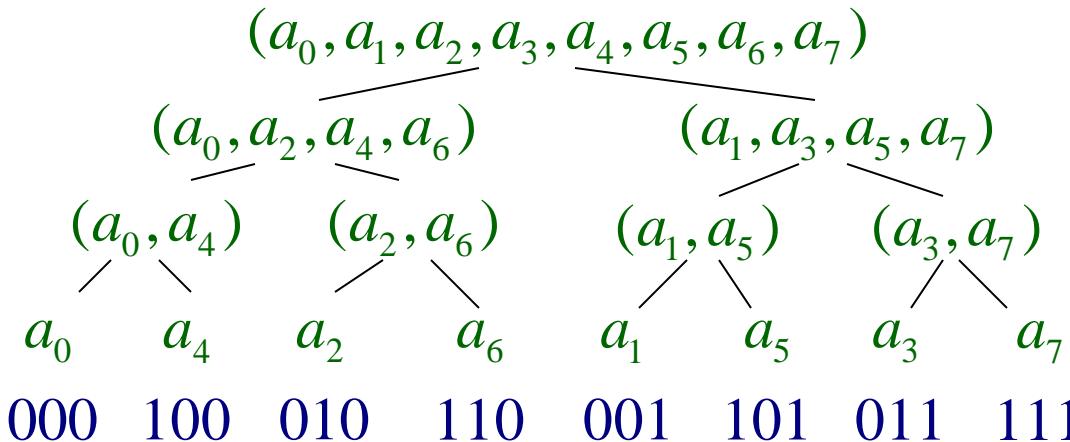
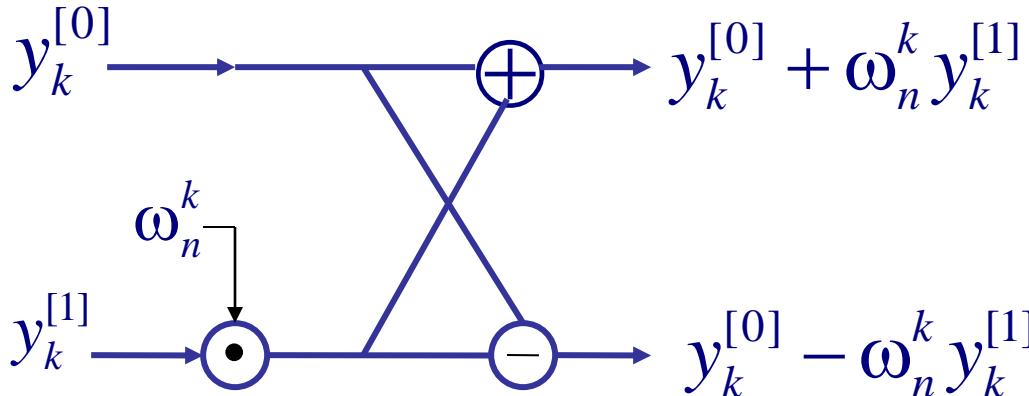
$$a \otimes b = \text{DFT}_{2n}^{-1}(\text{DFT}_{2n}(a) \bullet \text{DFT}_{2n}(b))$$

 Componentwise product

$$a \otimes b = < a_0 b_0, a_0 b_1 + a_1 b_0, a_0 b_2 + a_1 b_1 + a_2 b_0, \dots >$$

$$\text{j-th elt: } \sum_{k=0}^j a_k b_{j-k}$$

■ Efficient FFT implement



Idea:

for s=1 to $\lg n$ do

 for k=0 to $n-1$ by 2^s

 combine the two 2^{s-1} – element DFT's in

$y^{[0]} \rightarrow A[k..k+2^{s-1}-1]$ and $A[k+2^{s-1}..k+2^s-1]$

 into one 2^s – element DFT in $A[k..k+2^s-1]$

$y^{[1]}$

FFT-Base(a)

{ n = length[a];

for s=1 to (lg n) do

{ m=2^s;

$\omega_m = e^{2\pi i / m};$

for k=0 to n-1 by m do

{ $\omega=1$;

for j=0 to (m/2 - 1) do

{ t= $\omega A[k + j + m / 2]$;

$u=A[k+j]$;

$A[k+j]=u+t$;

$A[k+j+m/2]=u-t$;

$\omega = \omega \omega_m;$ }

}

Iterative-FFT(a)

```
{ Bit-Reverse-Copy(a, A);
  n=length[a];
  for s=1 to (lg n) do
    { m=2s;  $\omega_m = e^{2\pi i / m}$ ;  $\omega = 1$ ;
      for j=0 to (m/2 -1) do
        { for k=j to n-1 by m do
          { t =  $\omega A[k + m / 2]$ ;
            u = A[k];
            A[k]=u+t;
            A[k+m/2]=u-t;    }
           $\omega = \omega \omega_m$ 
        }
      }
    }
```

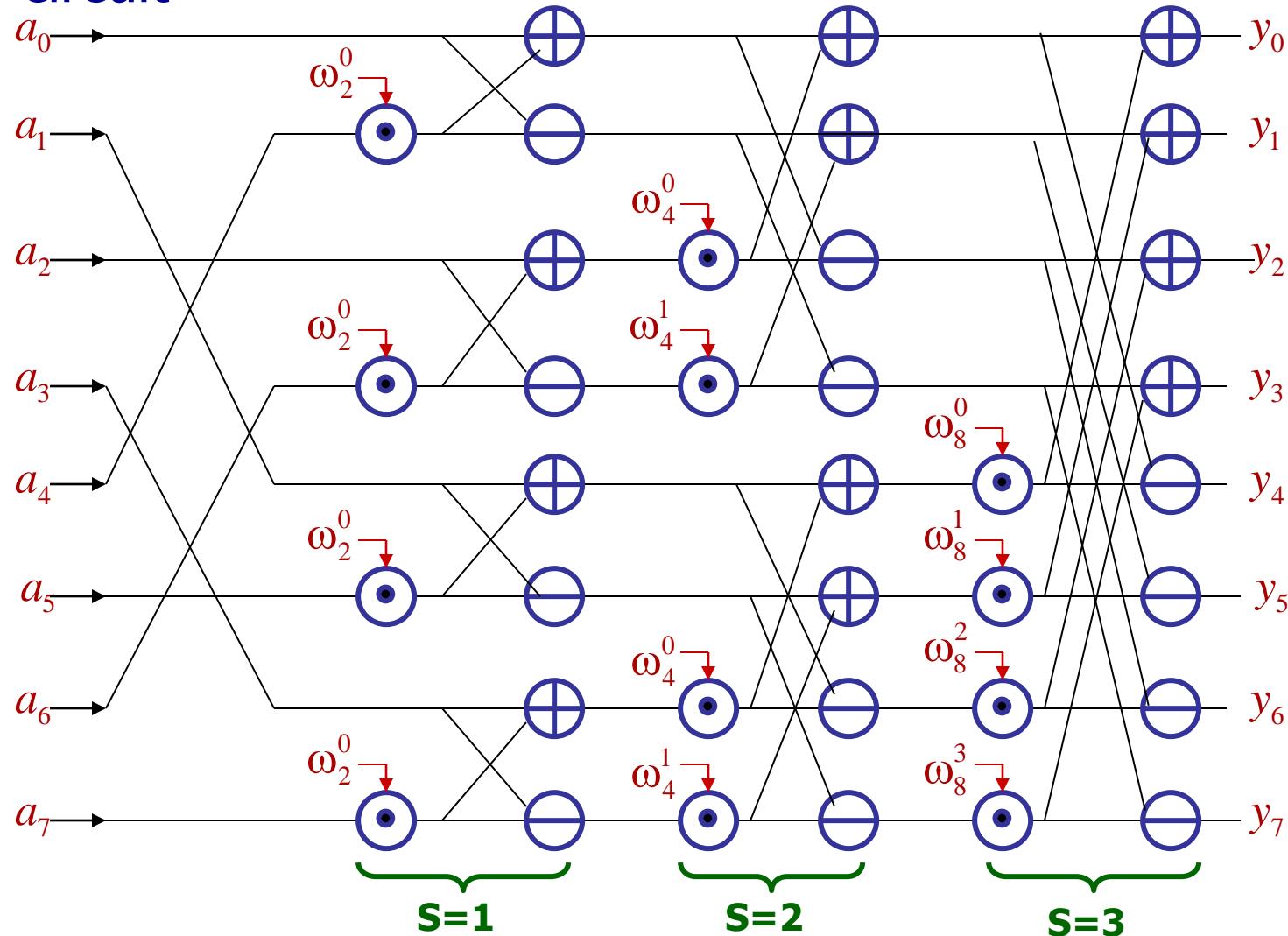
Bit-Reverse-Copy(a,A)

```
{ n=length[a];
  for k=0 to n-1 do
    A[rev(k)] = ak
}
```

eg

rev(011)=110, rev(001)=100

FFT circuit



$$n = 8$$

In general, depth: $\theta(\lg n)$

size : $\theta(n \lg n)$