

- 教材讨论
 - TC第30章

问题1：多项式的表示

- 什么是coefficient representation?
什么是point-value representation?
- 一个cr可以对应几个pvr?
一个pvr可以对应几个cr?
为什么?

问题1：多项式的表示 (续)

- 基于这两种表示
 - 以下运算的时间复杂度是多少？
 - 你能基于此对比它们的优劣吗？

	coefficient representation	point-value representation
加法的时间		
乘法的时间		
求值的时间		

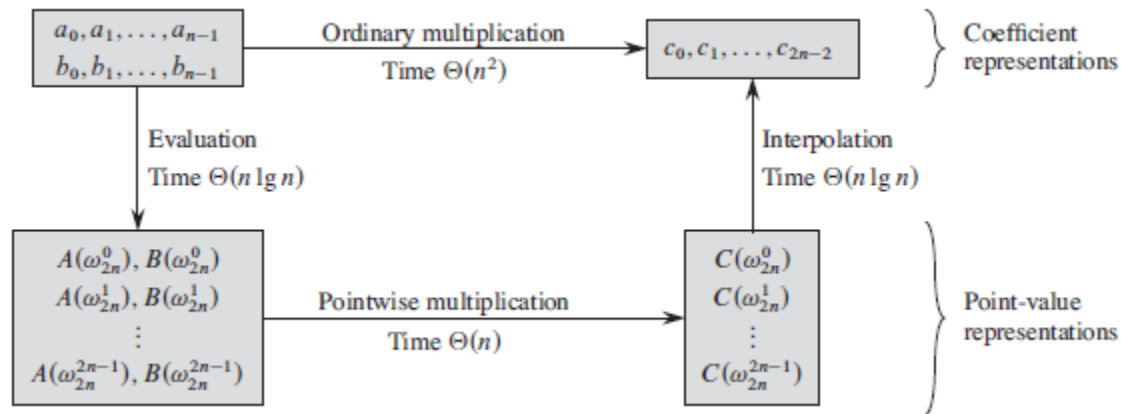
问题1：多项式的表示 (续)

- 基于这两种表示
 - 以下运算的时间复杂度是多少？
 - 你能基于此对比它们的优劣吗？

	coefficient representation	point-value representation
加法的时间	$\Theta(n)$	$\Theta(n)$
乘法的时间	$\Theta(n^2)$	$\Theta(n)$
求值的时间	$\Theta(n)$?

问题2：表示的转换

- 你理解这个流程了吗？
 - 目的是什么？
 - 手段是什么？



问题2：表示的转换 (续)

- DFT和FFT分别是什么意思？
它们之间是什么关系？
- 你能阐述FFT的基本思路吗？
 - 目标是求什么？
 - 用什么策略来求？
 - halving lemma在这里起了什么作用？

问题2: 表示的转换 (续)

- DFT和FFT分别是什么意思?
它们之间是什么关系?

$$\begin{aligned}y_k &= A(\omega_n^k) \\ &= \sum_{j=0}^{n-1} a_j \omega_n^{kj}\end{aligned}$$

- 你能阐述FFT的基本思路吗?

– 目标是求什么?

$$A(x) = \sum_{j=0}^{n-1} a_j x^j \quad \omega_n^0, \omega_n^1, \omega_n^2, \dots, \omega_n^{n-1}$$

– 用什么策略来求?

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

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

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

– halving lemma在这里起了什么作用?

问题2：表示的转换 (续)

- 递归FFT的运行时间如何递归表示？
- 你还记得怎么解这种递归式吗？

```
RECURSIVE-FFT(a)
1  n = a.length           // n is a power of 2
2  if n == 1
3      return a
4   $\omega_n = e^{2\pi i/n}$ 
5   $\omega = 1$ 
6   $a^{[0]} = (a_0, a_2, \dots, a_{n-2})$ 
7   $a^{[1]} = (a_1, a_3, \dots, a_{n-1})$ 
8   $y^{[0]} = \text{RECURSIVE-FFT}(a^{[0]})$ 
9   $y^{[1]} = \text{RECURSIVE-FFT}(a^{[1]})$ 
10 for k = 0 to n/2 - 1
11      $y_k = y_k^{[0]} + \omega y_k^{[1]}$ 
12      $y_{k+(n/2)} = y_k^{[0]} - \omega y_k^{[1]}$ 
13      $\omega = \omega \omega_n$ 
14 return y                 // y is assumed to be a column vector
```


问题2：表示的转换 (续)

- 递归FFT的运行时间如何递归表示？
- 你还记得怎么解这种递归式吗？

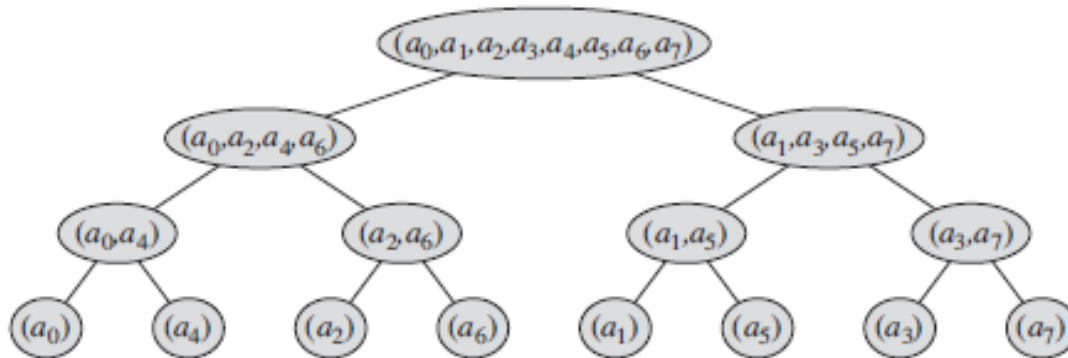
```
RECURSIVE-FFT(a)
1  n = a.length           // n is a power of 2
2  if n == 1
3      return a
4   $\omega_n = e^{2\pi i/n}$ 
5   $\omega = 1$ 
6   $a^{[0]} = (a_0, a_2, \dots, a_{n-2})$ 
7   $a^{[1]} = (a_1, a_3, \dots, a_{n-1})$ 
8   $y^{[0]} = \text{RECURSIVE-FFT}(a^{[0]})$ 
9   $y^{[1]} = \text{RECURSIVE-FFT}(a^{[1]})$ 
10 for k = 0 to n/2 - 1
11      $y_k = y_k^{[0]} + \omega y_k^{[1]}$ 
12      $y_{k+(n/2)} = y_k^{[0]} - \omega y_k^{[1]}$ 
13      $\omega = \omega \omega_n$ 
14 return y                 // y is assumed to be a column vector
```

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

问题2: 表示的转换 (续)

- 迭代FFT的基本思路是什么?
- 叶子的顺序是如何确定的?
你能给出直观解释吗?

```
BIT-REVERSE-COPY (a, A)  
1  n = a.length  
2  for k = 0 to n - 1  
3      A[rev(k)] = ak
```



问题2: 表示的转换 (续)

- 你理解interpolation的高效解法了吗?

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

参照 $y_k = A(\omega_n^k)$

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