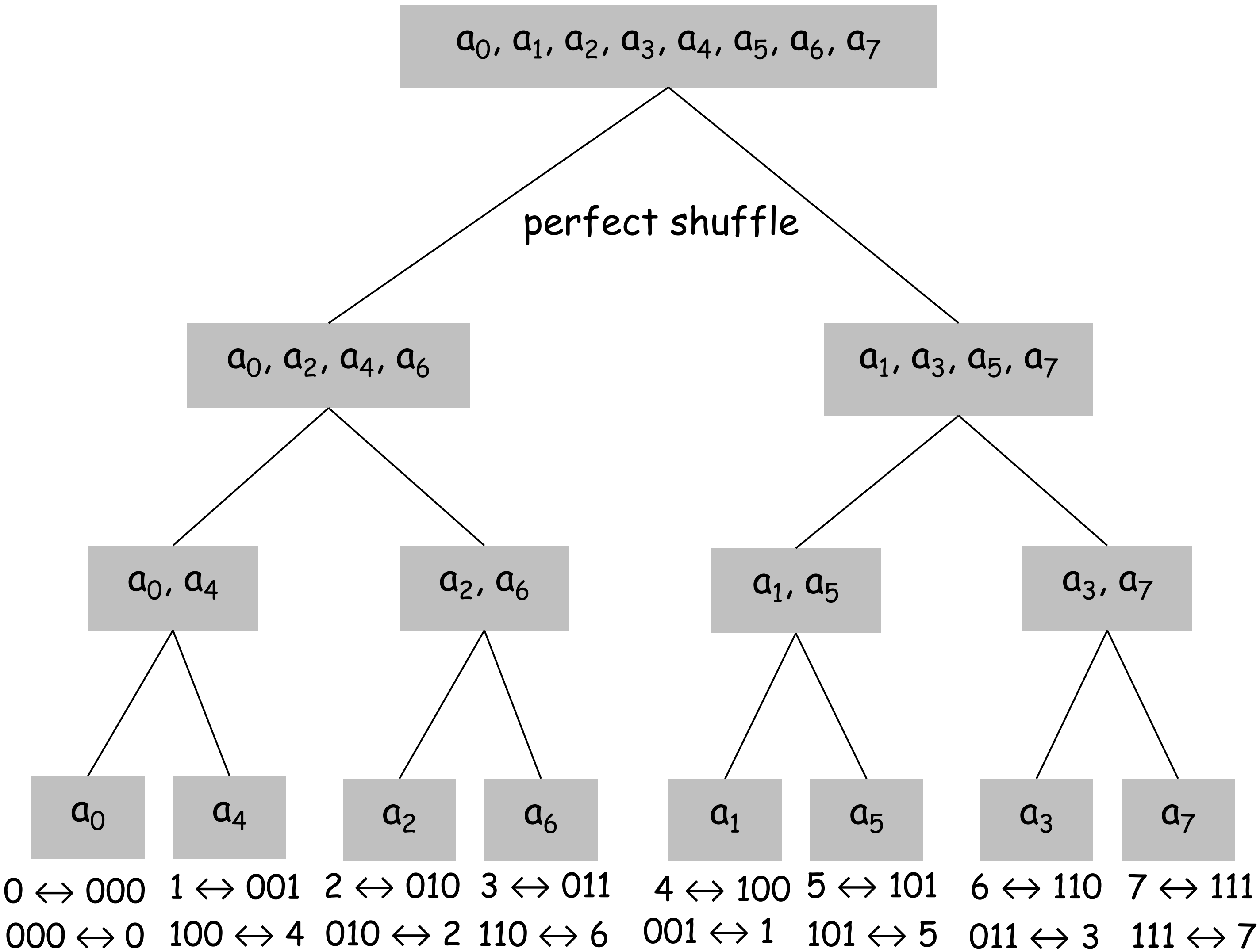
Fast Fourier Transform Part 2

Textbook Reading:

Chapter 7, Section 7.5,
pp. 302-313

Joseph Fourier

Recursion Tree



"bit-reversed" order

Exercise

For k = 7, i.e., $n = 2^k = 128$, consider the leaf nodes of the tree of recursive calls for *FFT*, which we will refer to from left to right as leaf node 0, 1, …, 127. What is the index i of a_i for leaf node 18?

18 = 16 + 2 so has binary representation 10010.

We need to add leading zeros to bring number of digits up to k = 7, i.e., **00**10010. We then reverse bits to get

0100100 = 32 + 4 = 36,

i.e., a_{36} ends up in leaf node 18.

PSN. For $k = 7$, i.e., $n = 2^k = 128$, what is the index i of a_i for leaf node 11?

FFT without recursive calls

procedure $FFT(a[0{:}n-1], n, \omega, b[0{:}n-1])$

Input: $a[0{:}n-1]$ (an array of coefficients of the polynomial $P(x) = a_{n-1}x^{n-1} + \ldots + a_1x + a_0$)

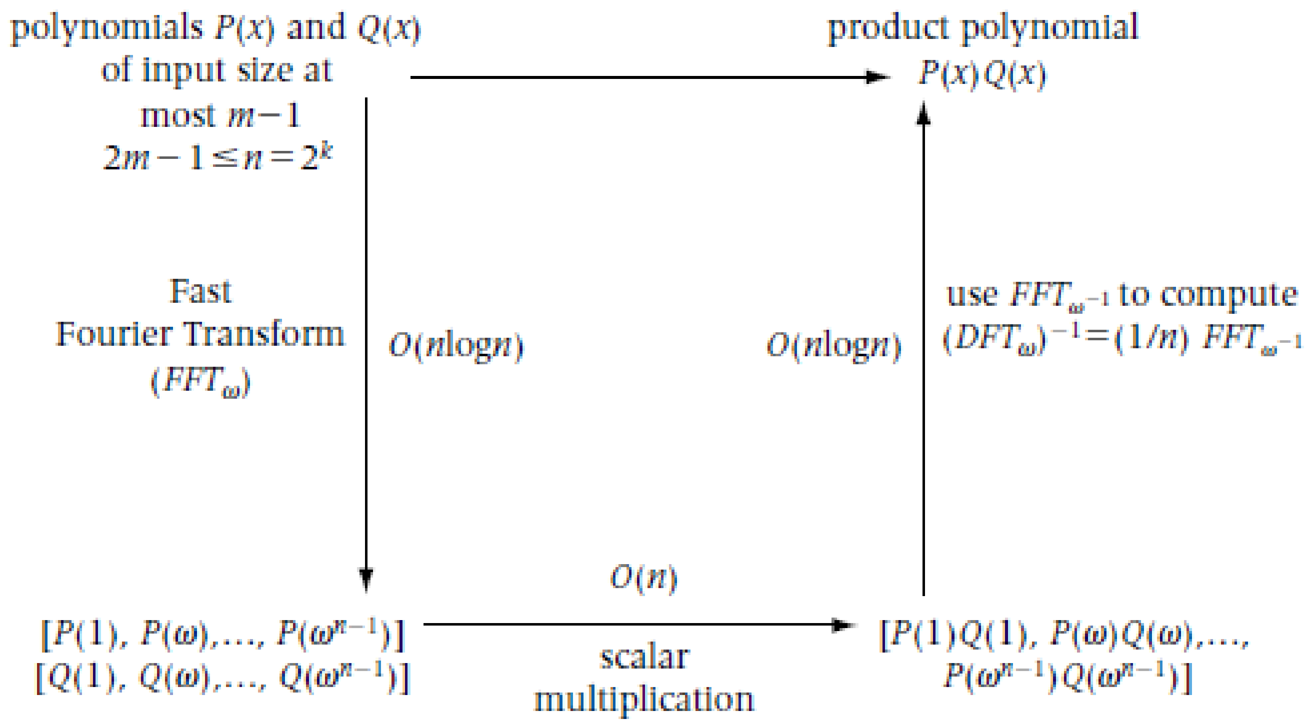
n (a positive integer) // $n = 2^k$

ω (a primitive n^{th} root of unity)

Output: $b[0{:}n-1]$ (an array of values $b[j] = P(\omega^j)$, $j = 0, \ldots, n-1$)

```
call ReverseBinPerm(R[0:n – 1])    // bit reversed order
for j ← 0 to n – 1
        b[j] ← a[R[j]]
endfor
for j ← 1 to k do
    for s ← 0 to n – 2^j by 2^j do
        for m ← 0 to 2^(j–1) – 1
            Temp[s + m] ← b[s + m] + (ω^(n/2^j))^m * b[s + m + 2^(j–1)]
            Temp[s + m + 2^(j–1)] ← b[s + m] – (ω^(n/2^j))^m * b[s + m + 2^(j–1)]
        endfor
    endfor
    for i ← 0 to n – 1 do
        b[i] ← Temp[i]
    endfor
endfor
end FFT
```

Commutative diagram summarizing $O(n \log n)$ polynomial multiplication using FFT_ω



Example showing correctness

$P(x) = 5 + 2x \leftrightarrow [5, 2, 0, 0]$

$Q(x) = 1 + 10x + 4x^2 \leftrightarrow [1, 10, 4, 0]$

$$\begin{aligned} DFT_i\, P(x) &= (5 + 2 \times 1) + (5 + 2\,i)x + (5 + 2(-1))\, x^2 + (5 + 2(-i))x^3 \\ &= 7 + (5 + 2i)\, x + 3x^2 + (5 - 2\,i)x^3 \\ &\leftrightarrow [7, 5 + 2i, 3, 5 - 2\,i] \end{aligned}$$

$$\begin{aligned} DFT_i\, Q(x) &= (1 + 10 \times 1 + 4 \times 1^2) + (1 + 10 \times i + 4 \times i^2)\, x + \\ &\quad (1 + 10 \times -1 + 4 \times (-1)^2)x^2 + (1 + 10 \times -i + 4 \times (-i)^2)\, x^3 \\ &= 15 + (-3 + 10\,i)\, x + (-5)\, x^2 + (-3 - 10i)\, x^3 \\ &\leftrightarrow [15, -3 + 10i, -5, -3 - 10i] \end{aligned}$$

$$\begin{aligned} DFT_i\, P(x)\, Q(x) &= 7 \times 15 + (5 + 2\,i)(-3 + 10\,i)\, x + 3 \times (-5)\, x^2 \\ &\quad + (5 - 2\,i)(-3 - 10\,i)\, x^3 \\ &= 105 + (-35 + 44i)\, x + (-15)\, x^2 + (-35 - 44i)\, x^3 \\ &\leftrightarrow [105, -35 + 44i, -15, -35 - 44i] \end{aligned}$$

Applying DFT inverse

Applying result $DFT_{\omega}{}^{-1}Q(x) = \frac{1}{n}DFT_{\omega^{-1}}Q(x)$

$DFT_i^{-1}\, P(x)\, Q(x) = \frac{1}{4}\, DFT_{-i}\, P(x)\, Q(x)$

$DFT_{-i}\, P(x)\, Q(x)$

$= DFT_{-i}\,(105 + (-35 + 44i)\, x + (-15)\, x^2 + (-35 - 44i)\, x^3)$

$= 105 + (-35 + 44i)\, 1 + (-15)\, 1^2 + (-35 - 44i)\, 1^3$

$\quad + (105 + (-35 + 44i)\, (-i) + (-15)\, (-i)^2 + (-35 - 44i)\, (-i)^3)\, x$

$\quad + (105 + (-35 + 44i)\, (-1) + (-15)\, (-1)^2 + (-35 - 44i)\, (-1)^3)\, x^2$

$\quad + (105 + (-35 + 44i)\, i + (-15)\, i^2 + (-35 - 44i)\, i^3)\, x^3$

$= 20 + 208x + 160x^2 + 32x^3$

Therefore,

$\frac{1}{4}\, DFT_{-i}\, P(x)\, Q(x) = 5 + 52x + 40x^2 + 8x^3$

Check: $P(x)\, Q(x) = (5 + 2x)(1 + 10x + 4x^2) = 5 + 52x + 40x^2 + 8x^3$

Previous example showing action of FFT

Problem: Use *FFT* to compute the product of two polynomials

$$P(x) = 2x + 5 \text{ and } Q(x) = 4x^2 + 10x + 1.$$

Step 1: Convert to coefficient arrays:

$$P(x) = 5 + 2x \leftrightarrow [5, 2, 0, 0]$$

$$Q(x) = 1 + 10x + 4x^2 \leftrightarrow [1, 10, 4, 0]$$

Step 2. Compute permutation of coefficients to order they occur in tree involving even-odd splits.

Reversing bits of binary numbers 00 01 10 11, we obtain 00 10 01 11, which is 0 2 1 3 in decimal, so we start with the constant polynomials of $[a_0]$ $[a_2]$ $[a_1]$ $[a_3]$.

$$P(x) \leftrightarrow [5]\ \ [0]\ \ [2]\ \ [0]$$

$$Q(x) \leftrightarrow [1]\ \ [4]\ \ [10]\ \ [0]$$

Step 3. Compute $DFT_i\, P(x)$ and $DFT_i\, Q(x)$ using *FFT*

$n = 1, \omega = 1$: [5] [0] [2] [0]

$n = 2, \omega = -1$: $[5 + 0, 5 - 0] = [5,5]$ and $[2 + 0, 2 - 0] = [2,2]$

$n = 4, \omega = i$: $[5 + 2, 5 + 2i, 5 - 2, 5 - 2i] = [7, 5 + 2i, 3, 5 - 2i]$

$n = 1, \omega = 1$: [1] [4] [10] [0]

$n = 2, \omega = -1$: $[1 + 4, 1 - 4] = [5,-3]$ and $[10 + 0, 10 - 0] = [10,10]$

$n = 4, \omega = i$: $[5 + 10, -3 + 10i, 5 - 10, -3 - 10i] = [15, -3 + 10i, -5, -3 - 10i]$

Step 4: Perform component-wise multiplication

$[7, 5 + 2i, 3, 5 – 2i] \times [15, -3 + 10i, -5, -3 – 10i]$

$= [7 \times 15, (5 + 2i) (-3 + 10i), 3 \times (-5), (5 – 2 i) (-3 – 10i)]$

$= [105, -35 + 44i, -15, -35 – 44i]$

Note this component product represents $DFT_i\ P(x)\ Q(x)$. To obtain $P(x)\ Q(x)$ we need to perform the transformation $DFT_i^{-1} = \frac{1}{4} DFT_{-i}$.

Step 5. Apply *FFT* to [105, -35 + 44i, -15, -35 – 44i] with ω = -i

$n = 1$, $\omega = 1$:   [105] [-15] [-35 + 44i] [-35 – 44i]

$n = 2$, $\omega = -1$: [105 + (-15), 105 – (-15)]   [-35 + 44i + (-35 – 44i), -35 + 44i – (-35 – 44i)]
= [90, 120]   = [-70,88i]

$n = 4$, $\omega = -i$:   [90 + (-70), 120 + (-i)88i, 90 – (-70), 120 – (-i)88i] = [20, 208, 160, 32]

Step 5: Divide by 4

¼ [20, 208, 160, 32]

= [5, 52, 40, 8]

$\leftrightarrow 8x^3 + 40x^2 + 52x + 5$

Check: $P(x)Q(x) = (2x + 5)(4x^2 + 10x + 1)$

$= 8x^3 + 40x^2 + 52x + 5$

Integer Multiplication

Multiplying two 3-digit numbers using the algorithm taught in school involves 9 digit multiplications. In general for n-digit numbers it involves n^2 digit multiplications.

$$
\begin{array}{r}
502 \\
\times\ 336 \\
\hline
3012 \\
15060 \\
+\ 150600 \\
\hline
168672
\end{array}
$$

We presented a divide-an-conquer algorithm for multiplying two n-digit number in time $O(n^{\log_2 3})$. We will now do it using FFT in time $O(n \log n)$!

Integer Multiplication

Given two n-digit (decimal) integers $a = a_{n\text{-}1} \ldots a_1 a_0$ and $b = b_{n\text{-}1} \ldots b_1 b_0$, compute their product $c = a \times b$. Note that this problem is similar to polynomial multiplication because for **x = 10**

$$a = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1}$$

$$b = b_0 + b_1 x + \ldots + b_{n-1} x^{n-1}$$

with the caveat that we may need to adjust slightly because the multiplication of digits in applying FFT could lead to a coefficient of a power of 10 consisting of a few digits.

Thus, we can perform multiplications of two n-digit integers in time $O(n \log n)$ using *FFT.*

Example

$$34 \times 288 = 9792$$

Apply *FFT* to multiply 34 and 288

Solution

$U = 34$ and $V = 288.$

Step 1: Convert to coefficient arrays:

$$U = 34 \leftrightarrow [4, 3, 0, 0]$$

$$V = 288 \leftrightarrow [8, 8, 2, 0]$$

Step 2. Compute permutation of coefficients in order of occurrence of leaf nodes in tree involving even-odd splits.

Reversing bits of binary numbers 00 01 10 11, we obtain 00 10 01 11, which is 0 2 1 3 in decimal, so we start with the constant polynomials of $[a_0]$ $[a_2]$ $[a_1]$ $[a_3]$.

$$U \leftrightarrow [4]\ \ [0]\ \ [3]\ \ [0]$$

$$V \leftrightarrow [8]\ \ [2]\ \ [8]\ \ [0]$$

Step 3. Compute $DFT_i\, P(x)$ and $DFT_i\, Q(x)$ using FFT

$n = 1, \omega = 1$: [4] [0] [3] [0]

$n = 2, \omega = -1$: $[4 + 0, 4 - 0] = [4,4]$ and $[3 + 0, 3 - 0] = [3,3]$

$n = 4, \omega = i$: $[4 + 3, 4 + 3i, 4 - 3, 3 - 3i] = [7, 4 + 3i, 1, 4 - 3i]$

$n = 1, \omega = 1$: [8] [2] [8] [0]

$n = 2, \omega = -1$: $[8 + 2, 8 - 2] = [10,6]$ and $[8 + 0, 8 - 0] = [8,8]$

$n = 4, \omega = i$: $[10 + 8, 6 + 8i, 10 - 8, 6 - 8i] = [18, 6 + 8i, 2, 6 - 8i]$

Step 3: Perform component-wise multiplication

$[7, 4 + 3i, 1, 4 – 3i] \times [18, 6 + 8i, 2, 6 – 8i]$

$= [7 \times 18, \ (4 + 3i)(6 + 8i), 1 \times 2, \ (4 – 3i)(6 – 8i)]$

$= [126, 50i, 2, -50i]$

Note this component product represents $DFT_i\, P(x)\, Q(x)$. To obtain $P(x)\, Q(x)$ we need to perform the transformation $DFT_i^{-1} = \frac{1}{4} DFT_{-i}$.

Step 4. Apply *FFT* to [126, 50i, 2, -50i] with $\omega = -i$

$n = 1, \omega = 1$: [126] [2] [50i] [-50i]

$n = 2, \omega = -1$: [126 + 2, 126 – 2] [50i + (-50i), 50i – (-50i)]

= [128, 124] = [0,100i]

$n = 4, \omega = -i$: [128 + 0, 124 + (-i)100i, 128 – 0, 124 – (-i)100i] = [128, 224, 128, 24]

Step 5: Divide by 4

$\frac{1}{4}\,[128, 224, 128, 24]$

$= [32, 56, 32, 6]$

$\leftrightarrow 6 \times 10^3 + 32 \times 10^2 + 56 \times 10 + 32$

$= 9792$

Why did the polynomial tree fall over?

It didn't have any real roots.