Spring 2019 CS170 FFT Guerrilla Section

Antares Chen, Tiffany Chien, Vishnu Iyer, Ida Liu, Rahul Malayappan, Jiazheng Zhao

February 10th, 2019

1. Complex Numbers Problem

(a) Guiding questions

i. Consider adding, multiplying, and squaring complex numbers. Which representation of complex numbers is it easier to do each operation in?

Solution: In general, it is easier to add in rectangular form and multiply in polar form.

- Addition: add in rectangular form because you can add the real and complex components separately.
- Multiplication: multiply in polar coordinates. In particular if $z_1 = r_1 e^{i\theta_1}$ and $z_2 = r_2 e^{i\theta_2}$ then

$$z_1 z_2 = r_1 e^{i\theta_1} \cdot r_2 e^{i\theta_2} = (r_1 r_2) e^{i(\theta_1 + \theta_2)}$$

- Squaring: square in polar coordinates. In particular, squaring is multiplying the complex number by itself. In polar coordinates, this amounts to squaring the radius and doubling the angle

$$z_1^2 = \left(r_1 e^{i\theta_1}\right)^2 = r_1^2 e^{i(2\theta_1)}$$

Note that $\sqrt{z_1}$ then amounts to square rooting r_1 and halving the angle θ_1.

(b) Represent the following numbers in polar form $re^{i\theta}$ for the appropriate r and θ.

i. $-\sqrt{3} + i$

Solution: $2e^{(5\pi/6)i}$

ii. The fourth roots of unity

Solution: $e^0, e^{(\pi/2)i}, e^{\pi i}, e^{(3\pi/2)i}$

(c) Find $\sqrt{\omega}$ where ω is the fourth root of unity. You can express this in rectangular, polar, or complex exponential notation.

Solution: The square roots of the fourth roots of unity are the 8-th roots of unity.

$$\sqrt{\omega} \in \{e^{(k2\pi)/8} : 0 \leq k \leq 7\}$$

Notice that this means that when we square the 8-th roots of unity, we get back the 4-th roots of unity.

2. Manual FFT Problem

(a) Guiding questions

i. How do we use the FFT to multiply two polynomials?
Solution: Given polynomials $p(x)$ and $q(x)$, we compute the product $(pq)(x)$ using the following procuedure

- Choose ω to be the 2^n-th root of unity where $2^n \geq \deg(pq)$.
- Evaluate p and q at points $1, \omega, \omega^2, \ldots, \omega^{2^k-1}$ by computing the FFT on inputs p and q with ω.
- Multiply each evaluation of p and q together to get the set of points $p(x)q(x)$ where $x \in \{1, \omega, \omega^2, \ldots, \omega^{2^k-1}\}$.
- Interpolate pq from the set of points $p(x)q(x)$ by computing the inverse FFT

(b) Use the FFT to compute the product of $p(x) = 1 + x$ and $q(x) = 1 + x^2$. Make sure to pick the appropriate power of 2, and compute the FFT and inverse FFT of polynomials p and q.

Solution: Let's compute the FFT of polynomials p and q. Since $\deg(pq) = 3$, we choose ω to be the 4-th roots of unity. We then use $M(\omega)$ to evaluate p and q

$$p : \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1+i \\ 0 \\ 1-i \end{pmatrix} \qquad q : \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 2 \\ 0 \end{pmatrix}$$

Afterwards, we multiply the results of the of the two matrix multiplications together component-wise to get $(4, 0, 0, 0)$. Finally, we convert this back into the coefficients of pq by performing an inverse FFT. We multiply this vector by $\frac{1}{n}M(\omega^{-1})$.

$$\frac{1}{4} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \begin{pmatrix} 4 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Thus the product pq is given by $x^3 + x^2 + x + 1$.

(c) Use the FFT to compute the product of $p(x) = 2 + x$ and $q(x) = 1 + x + x^2$. Make sure to pick the appropriate power of 2, and compute the FFT and inverse FFT of polynomials p and q.

Solution: Let's compute the FFT of p and q. Again we choose ω as the 4-th root of unity

$$p : \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 2+i \\ 1 \\ 2-i \end{pmatrix} \qquad q : \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ i \\ 1 \\ -i \end{pmatrix}$$

Multiplying the two vectors component-wise we get $(9, -1 + 2i, 1, -1 - 2i)$. Computing the inverse FFT

$$\frac{1}{4} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \begin{pmatrix} 9 \\ -1+2i \\ 1 \\ -1-2i \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 3 \\ 1 \end{pmatrix}$$

Indeed $(pq)(x) = 2 + 3x + 3x^2 + x^3$.

3. Extending FFT Problem

(a) Guiding questions

i. In the FFT algorithm, why do we choose to evaluate the input polynomial $A(x)$ at $x = \pm x_i$?

Solution: If we choose to use $\pm x_i$ then we notice that if split our input polynomial into chunks $A(x) = A_e(x^2) + xA_o(x^2)$, then evaluating $A(x_i)$ and $A(-x_i)$ is equivalent to

$$A(x_i) = A_e(x_i^2) + x_i A_o(x_i^2)$$
$$A(-x_i) = A_e(x_i^2) - x_i A_o(x_i^2)$$

In particular, we reduce evaluating $A(x)$ at two separate points to two recursive calls since $(\pm x_i)^2 = x_i^2$.

ii. If we use the FFT algorithm to multiply polynomials p and q, why do we choose ω to be the 2^k-th root of unity where $2^k \geq \deg(pq)$?

Solution: We choose ω to be a 2^k-th root of unity because squaring the 2^k-th roots of unity calculates the 2^{k-1}-th roots of unity. Choosing a power of 2 ensures that we have enough inputs to recursively evaluate the polynomials on for each root of unity. We choose $2^k \geq \deg(pq)$ because this ensures we evaluate pq at enough points to interpolate it uniquely.

(b) Cubing the 9-th roots of unity gives the 3-rd roots of unity. Next to each of the third root below, write down the corresponding 9-th roots which cube to it. The first has been filled for you. We will use ω_9 to represent the primitive 9-th root of unity, and ω_3 to represent the primitive 3-rd root.

Solution: We have the following

$$\begin{array}{cccc} \omega_3^0 : & \omega_9^0 & \omega_9^3 & \omega_9^6 \\ \omega_3^1 : & \omega_9^1 & \omega_9^4 & \omega_9^7 \\ \omega_3^2 : & \omega_9^2 & \omega_9^5 & \omega_9^8 \end{array}$$

(c) You want to run FFT on a degree-8 polynomial, but you don't like having to pad it with 0s to make the (degree+1) a power of 2. Instead, you realize that 9 is a power of 3, and you decide to work directly with 9th roots of unity and use the fact proven in part (b). Say that your polynomial looks like $p(x) = a_0 + a_1x + a_2x^2 + \ldots + a_8x^8$. How do you split $p(x)$ to use the fact proven in part (b) to your advantage?

Solution: The idea is to split $p(x)$ into three polynomials using x^3 as its variable. We have for $p(x)$ above

$$p_0(x^3) = a_0 + a_3x^3 + a_6x^6$$
$$p_1(x^3) = a_1 + a_4x^3 + a_7x^6$$
$$p_2(x^3) = a_2 + a_5x^3 + a_8x^6$$

We then compute the original polynomial by $p(x) = p_0(x^3) + xp_1(x^3) + x^2p_2(x^3)$

(d) Suppose you implemeneted a new FFT algorithm using your answer in part (b). Does this yield a faster algorithm than what was shown in class?

Solution: No, the recurrence relation for this algorithm would be

$$T(n) = 3T\left(\frac{n}{3}\right) + O(n)$$

Each recursive layer splits the input polynomial into three even chunks and recursively calls the next layer using each chunk as input. The layer does linear amount of work in order to multiply the polynomials together. By Master Theorem, we calculate $T(n) = O(n \log n)$.

4. Spaced Out Polynomial

(a) Find the FT of the polynomial $p(x) = 3x^{12} - 5x^8 - 4x^4 + 1$.

Solution: To employ FFT techniques, we will have to use the 16th roots of unity. The first split of the polynomial is $p_e(x) = 3x^6 - 5x^8 - 4x^4 + 1$ and $p_o(x) = 0$. Therefore we see that our evaluation process in the FFT will give us $p(x) = p_e(x^2)$ as $p_o(x)$ is always 0. Now, we split $p_e(x)$ into odd and even polynomials again. This gives $\tilde{p}_e(x) = 3x^3 - 5x^2 - 4x + 1$ and $\tilde{p}_o(x) = 0$. The evaluation process again gives us $p_e(x) = \tilde{p}_e(x^2)$ and, further, that $p(x) = \tilde{p}_e(x^4)$ At this point we could just evaluate the polynomial at the 4th roots of unity.

$$\tilde{p}_e(1) = -5, \tilde{p}_e(i) = 6 - 7i, \tilde{p}_e(-1) = -3, \tilde{p}_e(-i) = 6 + 7i$$

Now, to evaluate $p(x)$ at the 16th roots of unity $\omega_{16}^0, ..., \omega_{16}^{15}$, we apply the relation we derived earlier: $p(x) = \tilde{p}_e(x^4)$. However we can exploit a very nice symmetry here:

$$\left(\omega_{16}^0\right)^4 = \left(\omega_{16}^4\right)^4 = \left(\omega_{16}^8\right)^4 = \left(\omega_{16}^{12}\right)^4 = \omega_4^0$$
$$\left(\omega_{16}^1\right)^4 = \left(\omega_{16}^5\right)^4 = \left(\omega_{16}^9\right)^4 = \left(\omega_{16}^{13}\right)^4 = \omega_4^1$$
$$\left(\omega_{16}^2\right)^4 = \left(\omega_{16}^6\right)^4 = \left(\omega_{16}^{10}\right)^4 = \left(\omega_{16}^{14}\right)^4 = \omega_4^2$$
$$\left(\omega_{16}^3\right)^4 = \left(\omega_{16}^7\right)^4 = \left(\omega_{16}^{11}\right)^4 = \left(\omega_{16}^{15}\right)^4 = \omega_4^3$$

This is powerful - every fourth element of the FT of $p(x)$ is the same! Thus our FT procedure looks something like this:

$$p(\omega_{16}^0) = p(\omega_{16}^4) = p(\omega_{16}^8) = p(\omega_{16}^{12}) = \tilde{p}_e(\omega_4^0) = -5$$
$$p(\omega_{16}^1) = p(\omega_{16}^5) = p(\omega_{16}^9) = p(\omega_{16}^{13}) = \tilde{p}_e(\omega_4^1) = 6 - 7i$$
$$p(\omega_{16}^2) = p(\omega_{16}^6) = p(\omega_{16}^{10}) = p(\omega_{16}^{14}) = \tilde{p}_e(\omega_4^2) = -3$$
$$p(\omega_{16}^3) = p(\omega_{16}^7) = p(\omega_{16}^{11}) = p(\omega_{16}^{15}) = \tilde{p}_e(\omega_4^3) = 6 + 7i$$

Putting everything together, our FT is

$$[-5, 6 - 7i, -3, 6 + 7i, -5, 6 - 7i, -3, 6 + 7i, -5, 6 - 7i, -3, 6 + 7i, -5, 6 - 7i, -3, 6 + 7i]$$

(b) Find the inverse FT of the following vector:

$$[3, 4 - 3i, 5, 4 + 3i, 3, 4 - 3i, 5, 4 + 3i, 3, 4 - 3i, 5, 4 + 3i, 3, 4 - 3i, 5, 4 + 3i]$$

Solution: As always, we will consider our inverse FT operation as interpolating a polynomial from its value representation. We notice that, similar to the coefficient representation in the previous problem, the vector is a series of four repeats of the same four entries - that is, it is periodic. This tells us that the final polynomial is of the form $p(x) = ax^{12} + bx^8 + cx^4 + d = \tilde{p}(x^4)$. We will find $\tilde{p}$ using the inverse FFT function on the four repeated entries. From the identity

$$\text{iFFT}(p, \omega, n) = \frac{1}{n}\text{FFT}(p, \omega^{-1}, n)$$
$$\text{iFFT}([3, 4 - 3i, 5, 4 + 3i], \omega_4, 4) = \frac{1}{4}\text{FFT}([3, 4 - 3i, 5, 4 + 3i], \omega_4^{-1}, 4)$$

Thus amounts to evaluating the polynomial $q(x) = \frac{1}{4}((4 + 3i)x^3 + 5x^2 + (4 - 3i)x + 3)$ at the points $[1, -i, -1, i]$ (notice the order, as we flip our usual entries of the roots of unity when taking the inverse FFT). This computation gives us that

$$\text{iFFT}([3, 4 - 3i, 5, 4 + 3i], \omega_4, 4) = [4, -2, 0, 1]$$

Hence, we deduce that $\tilde{p}(x) = x^3 - 2x + 4$ and since we stated at the beginning that $p(x) = \tilde{p}(x^4)$, we have that $p(x) = x^{12} - 2x^4 + 4$ and the inverse FT that we are seeking is

$$[4, 0, 0, 0, -2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]$$

Indeed, using a process similar to the one in part (a), we can verify that the FT of the above vector yields our original vector.