Solutions to Homework Practice Problems

[DPV] Problem 2.7 – Roots of unity

Solution:

For the sum, use the geometric series equality to get

$$1 + \omega + \omega^2 + \cdots + \omega^{n-1} = \frac{\omega^n - 1}{\omega - 1} = 0.$$

For the product, since $1 + 2 + \cdots + (n-1) = \frac{(n-1)n}{2}$ we get

$$1\omega\omega^2 \dots \omega^{n-1} = \omega^{\frac{(n-1)n}{2}}$$

which equals 1 if n is odd and $\omega^{\frac{n}{2}} = -1$ for n even (remember that $\omega = e^{\frac{2\pi i}{n}}$).

[DPV] Problem 2.8

Solution:

(a). Given four coefficients, the appropriate value of ω where $n = 4$ is $e^{(2\pi i)/4} = i$.

We have FFT$(1, 0, 0, 0) = (1, 1, 1, 1)$ Here's the calculation:

$A_e = (1, 0) = 1 + 0x$, $A_o = (0, 0) = 0 + 0x$

$$\begin{aligned}
A(\omega_4^0) = A(1) &= A_e(1^2) + 1(A_o(1^2)) &&= 1 + 0(1^2) + 1(0 + 0(1^2)) &&= 1 + 1(0) = 1\\
A(\omega_4^1) = A(i) &= A_e(i^2) + i(A_o(i^2)) &&= 1 + 0(i^2) + i(0 + 0(i^2)) &&= 1 + i(0) = 1\\
A(\omega_4^2) = A(-1) &= A_e((-1)^2) - 1(A_o((-1)^2)) &&= 1 + 0((-1)^2) - 1(0 + 0((-1)^2)) &&= 1 - 1(0) = 1\\
A(\omega_4^3) = A(-i) &= A_e((-i)^2) - i(A_o((-i)^2)) &&= 1 + 0((-i)^2) - i(0 + 0((-i)^2)) &&= 1 - i(0) = 1
\end{aligned}$$

The inverse FFT of $(1, 0, 0, 0) = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$.

(b). FFT$(1, 0, 1, -1) = (1, i, 3, -i)$. Here's the matrix form of the calculation:

$$\begin{bmatrix} A(\omega_4^0) \\ A(\omega_4^1) \\ A(\omega_4^2) \\ A(\omega_4^3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ i \\ 3 \\ -i \end{bmatrix}$$

[DPV] Problem 2.9(a)

Solution:

We use 4 as the power of 2 and set $\omega = i$.
The FFT of $x + 1$ is FFT$(1, 1, 0, 0) = (2, 1 + i, 0, 1 - i)$.
The FFT of $x^2 + 1$ is FFT$(1, 0, 1, 0) = (2, 0, 2, 0)$.
The inverse FFT of their product $(4, 0, 0, 0)$ corresponds to the polynomial $1 + x + x^2 + x^3$.

FFT Design

Let $A(x) = 1 - 6x - 2x^2 + 7x^3$. You wish to run FFT to evaluate this polynomial.

(a) What is $A_{odd}(y)$ and $A_{even}(y)$?

$A_{odd}(y) = -6 + 7y$, $A_{even}(y) = 1 - 2y$

(b) What is the appropriate root of unity to use?

We have a polynomial of degree 3; the smallest power of 2 where $2^k \geq d + 1$ is 4, so we use the *4th* roots of unity.

(continued on next page)

FFT application

Design an algorithm that takes as input a set $S = \{s_1, s_2, \ldots, s_n\}$ of n distinct natural numbers such that $0 \leq s_i \leq 100n$, and a natural number N, and outputs True if the equation $s_i + s_j + s_k = N$ has at least one solution, and return False otherwise. There is a simple solution that runs in $O(n^3)$ time (can you find an $O(n^2)$ solution?), but you can improve on these times using FFT! Your faster algorithm should use FFT as a black box. That is, describe how to form the inputs to FFT and what can be done with the output from FFT to solve the problem.

Example: For $N = 6$ and $S = \{1, 2, 3, 5, 10\}$ your design should output True since $1+2+3 = 6$. For $N = 20$ and the same set S the answer should be True again since $5+5+10 = 20$ (yes, you can have $s_i = s_j = s_k$) but for $N = 19$ the answer is False since no three numbers add up to 19.

Solution:

The key idea is to note that the exponents of the monomials of the polynomials multiplied by FFT add up to make the monomials of the resulting product.

For set S, define our polynomial

$$P(x) = x^{s_1} + x^{s_2} + \cdots + x^{s_n}.$$

This polynomial has degree at most $100n$, by the condition of the problem. We next get P^3 via FFT where our value for ω is the smallest 2^k roots of unity such that $300n + 1 \leq 2^k$ and k is a natural number.

- Run FFT on $P(x)$ using ω as described above, giving vector of point values P'
- Element-wise multiplication to obtain interim point values of P'^2
- Second round of element-wise multiplication, $P' \times P'^2$ to get point values for P'^3
- Run IFFT on P'^3 to recover coefficients for P^3

The resulting polynomial looks like

$$\begin{aligned} P(x)P(x)P(x) &= (x^{s_1} + x^{s_2} + \cdots + x^{s_n})(x^{s_1} + x^{s_2} + \cdots + x^{s_n})(x^{s_1} + x^{s_2} + \cdots + x^{s_n}) \\ &= \sum x^{s_i + s_j + s_k}. \end{aligned}$$

where the resulting exponents are the cartesian sum of the set S. Hence, if we are looking for a solution to $s_i + s_j + s_k = N$ we check if the monomial x^N has a positive coefficient in P^3 : if it does, return TRUE, otherwise FALSE. This check takes $O(1)$.

The total running is dominated by two calls of FFT on polynomials of degree at most $300n$, yielding $O(n \log(n))$.