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

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

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 $\text{FFT}(1, 0, 0, 0) = (1, 1, 1, 1)$ 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 $(1, 0, 0, 0) = (1/4, 1/4, 1/4, 1/4)$.

(b). $\text{FFT}(1, 0, 1, -1) = (1, i, 3, -i)$. 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 $\text{FFT}(1, 1, 0, 0) = (2, 1 + i, 0, 1 - i)$.
The FFT of $x^2 + 1$ is $\text{FFT}(1, 0, 1, 0) = (2, 0, 2, 0)$.
The inverse FFT of their product $(4, 0, 0, 0)$ corresponds to the polynomial $1 + x + x^2 + x^3$.

Types of binary search
Solution:

(a). Let's begin the binary search by dividing the array into two subarrays defined as follows $B_1 = \{10, 23, 36, 47, 59, 64, 71, 82\}$ and $B_2 = \{95, 100, 116, 127, 138, 141, 152, 163\}$. Since the number of entries is even we take the last element of B_1 as our middle element. Since, $36 < 82$ we take B_1 and discard B_2.

Next step, we divide B_1 into two arrays. $C_1 = \{10, 23, 36, 47\}$ and $C_2 = \{59, 64, 71, 82\}$. Now since $36 < 47$, we keep C_1 and discard C_2.

We follow the same process for C_1 by dividing it into $D_1 = \{10, 23\}$ and $D_2 = \{36, 47\}$. Since $36 > 23$, we keep D_2 and discard D_1.

Finally, we divide D_2 into $E_1 = \{36\}$ and $E_2 = \{47\}$. Since 36 is equal to the $E_1[1]$ we have found 36 and the process is over.

(b) If we have an array of length n we expect the numbers $\{1, 2, 3, \ldots, n\}$ to be in the array. Since, one number is missing this means there is at least one number such that its position does not match its value. If mid is the position of the middle element, we first check to see if $A[mid] = mid$. Since the array is sorted, if $A[mid] = mid$ it means there are not numbers missing from $1, 2, \ldots, mid$, so we have to check the right half of the array. If there was a missing number, it would have been replaced by a bigger number. This means, if $A[mid] > mid$, we have to search the left half. As you reduce the input, track what the starting value should be (initially it's 1), call that S. If $A[1] \neq S$ then the missing value is S. If they match, then you divide the current input in half based on the value of $A[mid]$, update S accordingly, and recurse. If you get to a single element and $A[i] = S$ then the missing value is $A[i] + 1$. To check the running time is logarithmic, note that the recurrence relation is $T(n) = T(\frac{n}{2}) + O(1)$ which solves to $O(\log(n))$ by the Master Theorem.