DFT over Finite Rings or Fields

\[ W_N = e^{-j \frac{2\pi}{N}} \]
DFT in GF(41)

- Consider the integer ring $\mathbb{Z}_{41}$
- Since $p = 41$ is prime, this is actually GF(41)
- For the DFT of length $d$ to exist, we need to have $\gcd(d, 41) = 1$
- Since 41 is prime, for all $d \in [2, 40]$ $\gcd(d, 41) = 1$
- The second condition: $d$ must divide $p - 1 = 40 = 2^3 \cdot 5$
- Therefore, we can have $d = 2, 4, 8, 10, 20, 40$
- Let us take $d = 8$, which gives $d^{-1} = 36 \pmod{41}$
- We find the principal 8th roots of unity as 3, 14, 27, 38
- The element $\omega = 3$ is indeed a principal 8th root of unity

$$
\begin{align*}
3^1 & = 2 & 3^2 & = 9 & 3^3 & = 27 & 3^4 & = 81 = 40 \\
3^5 & = 243 = 38 & 3^6 & = 729 = 32 & 3^7 & = 2187 = 14 & 3^8 & = 6561 = 1
\end{align*}
$$
We prefer to have \( \omega \) as a positive or negative power of 2, however, no such \( \omega \) exists for \( p = 41 \).

The DFT of length 8 in \( \text{GF}(41) \) is computed by matrix-vector multiplication where the entries of the transformation matrix \( T \) is given by

\[
T = [\omega^{i \cdot j} \mod p] = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 3 & 3^2 & 3^3 & 3^4 & 3^5 & 3^6 & 3^7 \\
1 & 3^2 & 3^4 & 3^6 & 3^8 & 3^{10} & 3^{12} & 3^{14} \\
1 & 3^3 & 3^6 & 3^9 & 3^{12} & 3^{15} & 3^{18} & 3^{21} \\
1 & 3^4 & 3^8 & 3^{12} & 3^{16} & 3^{20} & 3^{24} & 3^{28} \\
1 & 3^5 & 3^{10} & 3^{15} & 3^{20} & 3^{25} & 3^{30} & 3^{35} \\
1 & 3^6 & 3^{12} & 3^{18} & 3^{24} & 3^{30} & 3^{36} & 3^{42} \\
1 & 3^7 & 3^{14} & 3^{21} & 3^{28} & 3^{35} & 3^{42} & 3^{49}
\end{bmatrix} \mod 41
We obtain the DFT matrix as

\[ \mathcal{T} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 3 & 9 & 27 & 40 & 38 & 32 & 14 \\
1 & 9 & 40 & 32 & 1 & 9 & 40 & 32 \\
1 & 27 & 32 & 3 & 40 & 14 & 9 & 38 \\
1 & 40 & 1 & 40 & 1 & 40 & 1 & 40 \\
1 & 38 & 9 & 14 & 40 & 3 & 32 & 27 \\
1 & 32 & 40 & 9 & 1 & 32 & 40 & 9 \\
1 & 14 & 32 & 38 & 40 & 27 & 9 & 3 
\end{bmatrix} \]
Consider two vectors:

\[ a = (1, 2, 3, 4, 0, 0, 0, 0) \]
\[ b = (4, 3, 2, 1, 0, 0, 0, 0) \]

In polynomial notation:

\[ a = 1 + 2x + 3x^2 + 4x^3 \]
\[ b = 4 + 3x + 2x^2 + x^3 \]

Their direct convolution is the polynomial \( c(x) = a(x)b(x) \)

\[ c(x) = 4 + 11x + 20x^2 + 30x^3 + 20x^4 + 11x^5 + 4x^6 \]

In vector notation: \( c = (4, 11, 20, 30, 20, 11, 4, 0) \)
Convolution

- Given two vectors of dimension $d$, the direct convolution requires $O(d^2)$ arithmetic operations.
- However, the DFT method (using the Fast Fourier Transform Algorithm) requires $O(d \log d)$ arithmetic operations.
- The DFT method performs the sequence of operations:
  Step 1a: $A = \text{DFT}(a)$
  Step 1b: $B = \text{DFT}(b)$
  Step 2: $C = A \circledast B$
  Step 3: $c = \text{IDFT}(C)$
- The operation in Step 2 is the point-wise multiplication of the spectral vectors $A$ and $B$, in other words $C_i = A_i \cdot B_i$ for $i = 0, 1, \ldots, d - 1$.
The steps of the DFT method can be performed using complex arithmetic implemented in floating-point arithmetic, we will use the DFT over the finite field GF(41)

All computations (multiplication and additions) are performed in GF(41), i.e., mod 41 arithmetic
DFT Steps

The computation of $A = \text{DFT}(a)$

$$A = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 3 & 9 & 27 & 40 & 38 & 32 & 14 \\
1 & 9 & 40 & 32 & 1 & 9 & 40 & 32 \\
1 & 27 & 32 & 3 & 40 & 14 & 9 & 38 \\
1 & 40 & 1 & 40 & 1 & 40 & 1 & 40 \\
1 & 38 & 9 & 14 & 40 & 3 & 32 & 27 \\
1 & 32 & 40 & 9 & 1 & 32 & 40 & 9 \\
1 & 14 & 32 & 38 & 40 & 27 & 9 & 3
\end{bmatrix} \begin{bmatrix}
1 \\
2 \\
3 \\
4 \\
0 \\
0 \\
0 \\
0
\end{bmatrix} = \begin{bmatrix}
10 \\
19 \\
21 \\
40 \\
39 \\
37 \\
16 \\
31
\end{bmatrix}$$
The computation of $B = \text{DFT}(b)$

$$B = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 3 & 9 & 27 & 40 & 38 & 32 & 14 \\
1 & 9 & 40 & 32 & 1 & 9 & 40 & 32 \\
1 & 27 & 32 & 3 & 40 & 14 & 9 & 38 \\
1 & 40 & 1 & 40 & 1 & 40 & 1 & 40 \\
1 & 38 & 9 & 14 & 40 & 3 & 32 & 27 \\
1 & 32 & 40 & 9 & 1 & 32 & 40 & 9 \\
1 & 14 & 32 & 38 & 40 & 27 & 9 & 3
\end{bmatrix} \begin{bmatrix}
4 \\
3 \\
2 \\
1 \\
0 \\
0 \\
0 \\
0
\end{bmatrix} = \begin{bmatrix}
10 \\
17 \\
20 \\
29 \\
2 \\
27 \\
25 \\
25
\end{bmatrix}$$
The computation of $C = A \odot B$

$$C = \begin{bmatrix}
10 \\ 19 \\ 21 \\ 40 \\ 39 \\ 37 \\ 16 \\ 31
\end{bmatrix} \odot \begin{bmatrix}
10 \\ 17 \\ 20 \\ 29 \\ 2 \\ 27 \\ 25 \\ 25
\end{bmatrix} = \begin{bmatrix}
10 \cdot 10 \\ 19 \cdot 17 \\ 21 \cdot 20 \\ 40 \cdot 29 \\ 39 \cdot 2 \\ 37 \cdot 27 \\ 16 \cdot 25 \\ 31 \cdot 25
\end{bmatrix} = \begin{bmatrix}
18 \\ 36 \\ 10 \\ 12 \\ 37 \\ 15 \\ 31 \\ 37
\end{bmatrix}$$
The Inverse DFT Computation

To perform the Inverse DFT computation, we need the inverse of $\mathcal{T}$ matrix in GF(41)

The entries of the inverse of $\mathcal{T}$ are given as

$$\mathcal{T}^{-1} = d^{-1} \cdot \begin{bmatrix} \omega^{-i,j} \end{bmatrix}$$

for $i, j = 0, 1, \ldots, d - 1$, which is obtained as

The Inverse DFT Computation

- The IDFT computation of the vector $C$ is accomplished using $\mathcal{T}^{-1}C \pmod{41}$ which is given as

$$
\begin{bmatrix}
36 & 36 & 36 & 36 & 36 & 36 & 36 & 36 \\
36 & 12 & 4 & 15 & 5 & 29 & 37 & 26 \\
36 & 4 & 5 & 37 & 36 & 4 & 5 & 37 \\
36 & 15 & 37 & 12 & 5 & 26 & 4 & 29 \\
36 & 5 & 36 & 5 & 36 & 5 & 36 & 5 \\
36 & 29 & 4 & 26 & 5 & 12 & 37 & 15 \\
36 & 37 & 5 & 4 & 36 & 37 & 5 & 4 \\
36 & 26 & 37 & 29 & 5 & 15 & 4 & 12
\end{bmatrix}
\begin{bmatrix}
18 \\
36 \\
10 \\
12 \\
37 \\
15 \\
31 \\
37
\end{bmatrix}
= 
\begin{bmatrix}
4 \\
11 \\
20 \\
30 \\
20 \\
11 \\
4 \\
0
\end{bmatrix}
$$

- Therefore, we find the same $c$ vector or $c(x)$ polynomial as

$$
c = (4, 11, 20, 30, 20, 11, 4, 0)
$$

$$
c(x) = 4 + 11x + 20x^2 + 30x^3 + 20x^4 + 11x^5 + 4x^6
$$
If we take radix as $r = 10$, and consider the polynomials $a(x)$ and $b(x)$ as integers representing $a(r)$ and $b(r)$, as $a(10) = 4321$ and $b(10) = 1234$

The final product $c(x)$ evaluated at $x = r = 10$ is found as

$$c(x) = 4 + 11x + 20x^2 + 30x^3 + 20x^4 + 11x^5 + 4x^6$$
$$c(10) = 4 + 11 \cdot 10 + 20 \cdot 10^2 + 30 \cdot 10^3 + 20 \cdot 10^4 + 11 \cdot 10^5 + 4 \cdot 10^6$$
$$= 5332114$$

This is the same as $a(10) \cdot b(10) = 4321 \cdot 1234 = 5332114$
However, the computations can overflow without our knowledge, i.e., without detection.

Consider

\[
d = (1, 2, 3, 7, 0, 0, 0, 0)
\]
\[
e = (7, 3, 2, 1, 0, 0, 0, 0)
\]

We obtain the spectral vectors as

\[
D = (13, 18, 35, 8, 36, 38, 2, 22)
\]
\[
E = (13, 20, 23, 32, 5, 30, 28, 28)
\]

The point-wise product vector is found as

\[
F = D \odot E = (5, 32, 26, 10, 16, 33, 15, 1)
\]
From $F$, we obtain $f$ using the IDFT as

$$f = (7, 17, 29, 22, 29, 17, 7, 0)$$

This is incorrect since $f(x) = d(x)e(x)$ is actually equal to

$$(1 + 2x + 3x^2 + 7x^3)(7 + 3x + 2x^2 + x^3)$$

$$= 7 + 17x + 29x^2 + 63x^3 + 29x^4 + 17x^5 + 7x^6$$

What we obtained was $f(x) \pmod{41}$, which is

$$7 + 17x + 29x^2 + 22x^3 + 29x^4 + 17x^5 + 7r^6$$

During the computations the coefficients of the polynomial went outside their range and were reduced mod 41
The FFT versus DFT

- We performed the DFT computation, i.e., the computation of $T a$ using a matrix-vector product which requires $O(d^2)$ arithmetic operations.
- However, the Fast Fourier Transform (FFT) Algorithm requires only $O(d \log d)$ arithmetic operations.
- It performs exactly the same computation $T a$ using fewer arithmetic operations.
- The reduction in the complexity is due to the properties of the root of unity $\omega$ and the matrix-vector product.
The 4-point FFT

- Assume that \( d = 4 \) and \( \omega \) is the 4th root of unity in some finite field or ring
- The DFT matrix would be given as

\[
T = \begin{bmatrix}
\omega^0 & \omega^1 & \omega^2 & \omega^3 \\
\omega^0 & \omega^1 & \omega^2 & \omega^3 \\
\omega^2 & \omega^1 & \omega^2 & \omega^3 \\
\omega^3 & \omega^2 & \omega^1 & \omega^2 \\
\end{bmatrix} = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 \\
1 & \omega^2 & \omega^4 & \omega^6 \\
1 & \omega^3 & \omega^6 & \omega^9 \\
\end{bmatrix}
\]

- Since \( \omega^4 = 1 \), we can write \( T \) as

\[
T = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 \\
1 & \omega^2 & 1 & \omega^2 \\
1 & \omega^3 & \omega^2 & \omega \\
\end{bmatrix}
\]
The 4-point FFT

In order to apply the DFT to a vector of dimension of 4, such as \( a = (a_0, a_1, a_2, a_3) \), we perform the matrix-vector product \( \mathcal{T} a \), as

\[
A = \mathcal{T} a = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 \\
1 & \omega^2 & 1 & \omega^2 \\
1 & \omega^3 & \omega^2 & \omega
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
a_2 \\
a_3
\end{bmatrix}
\]

As we explained, if we perform this matrix-vector product which is dimension \( d \), its arithmetic complexity would be given as \( O(d^2) \).

The FFT algorithm utilizes the properties of the root of unity \( \omega \) and performs only \( O(d \log d) \) arithmetic operations.
The 4-point FFT

To illustrate the above 4-point DFT using the FFT algorithm, we write each element of the spectral vector $A = (A_0, A_1, A_2, A_3)$

\[
\begin{align*}
A_0 &= a_0 + a_1 + a_2 + a_3 \\
A_1 &= a_0 + a_1\omega + a_2\omega^2 + a_3\omega^3 \\
A_2 &= a_0 + a_1\omega^2 + a_2 + a_3\omega^2 \\
A_3 &= a_0 + a_1\omega^3 + a_2\omega^2 + a_3\omega
\end{align*}
\]

We know separate the odd and even index components and rewrite

\[
\begin{align*}
A_0 &= a_0 + a_2 + a_1 + a_3 \\
A_1 &= a_0 + a_2\omega^2 + a_1\omega + a_3\omega^3 \\
A_2 &= a_0 + a_2 + a_1\omega^2 + a_3\omega^2 \\
A_3 &= a_0 + a_2\omega^2 + a_1\omega^3 + a_3\omega
\end{align*}
\]
The 4-point FFT

- These equations can be written as two linear systems of equations of dimension $2 \times 2$ as

$\begin{bmatrix} A_0 \\ A_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & \omega^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_2 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ \omega & \omega^3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_3 \end{bmatrix}$

$\begin{bmatrix} A_2 \\ A_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & \omega^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_2 \end{bmatrix} + \begin{bmatrix} \omega^2 & \omega^2 \\ \omega^3 & \omega \end{bmatrix} \begin{bmatrix} a_1 \\ a_3 \end{bmatrix}$

- We notice that the first component of the above sums is the same for both spectral vectors $(A_0, A_1)$ and $(A_2, A_3)$

$\begin{bmatrix} 1 & 1 \\ 1 & \omega^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_2 \end{bmatrix}$

- This computation can be performed once and used for both spectral vectors $(A_0, A_1)$ and $(A_2, A_3)$
The 4-point FFT

Furthermore the second component for the first and second equations, given as

\[
\begin{bmatrix}
1 & 1 \\
\omega & \omega^3 \\
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_3 \\
\end{bmatrix}
\quad \text{and} \quad
\begin{bmatrix}
\omega^2 & \omega^2 \\
\omega^3 & \omega \\
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_3 \\
\end{bmatrix}
\]

It turns out these computations are also similar, as the second one can be written as

\[
\begin{bmatrix}
\omega^2 & \omega^2 \\
\omega^3 & \omega \\
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_3 \\
\end{bmatrix} = \omega^2
\begin{bmatrix}
1 & 1 \\
\omega & \omega^3 \\
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_3 \\
\end{bmatrix}
\]

This is due to the fact that $\omega^2 \cdot \omega^3 = \omega$ since $\omega^4 = 1$
The 4-point FFT

Therefore, we can write the computations of the spectral vectors \((A_0, A_1)\) and \((A_2, A_3)\) as

\[
\begin{bmatrix}
A_0 \\
A_1
\end{bmatrix} = \begin{bmatrix}
1 & 1 \\
1 & \omega^2
\end{bmatrix} \begin{bmatrix}
a_0 \\
a_2
\end{bmatrix} + \begin{bmatrix}
1 & 1 \\
\omega & \omega^3
\end{bmatrix} \begin{bmatrix}
a_1 \\
a_3
\end{bmatrix}
\]

\[
\begin{bmatrix}
A_2 \\
A_3
\end{bmatrix} = \begin{bmatrix}
1 & 1 \\
1 & \omega^2
\end{bmatrix} \begin{bmatrix}
a_0 \\
a_2
\end{bmatrix} + \omega^2 \begin{bmatrix}
1 & 1 \\
\omega & \omega^3
\end{bmatrix} \begin{bmatrix}
a_1 \\
a_3
\end{bmatrix}
\]

The conclusion is that a 4-point DFT involving \(a = (a_0, a_1, a_2, a_3)\) can be performed by first performing two 2-point DFT computations with the even and odd indexed components \((a_0, a_2)\) and \((a_1, a_3)\) and then adding these components to obtain the final spectral vector.
The 4-point FFT

- Step 1: Compute the DFT of the even components
  \[
  \begin{bmatrix}
  a'_0 \\
  a'_2
  \end{bmatrix}
  =
  \begin{bmatrix}
  1 & 1 \\
  1 & \omega^2
  \end{bmatrix}
  \begin{bmatrix}
  a_0 \\
  a_2
  \end{bmatrix}
  \]

- Step 2: Compute the DFT of the odd components
  \[
  \begin{bmatrix}
  a'_1 \\
  a'_3
  \end{bmatrix}
  =
  \begin{bmatrix}
  1 & 1 \\
  \omega & \omega^3
  \end{bmatrix}
  \begin{bmatrix}
  a_1 \\
  a_3
  \end{bmatrix}
  \]

- Step 3: Multiply the DFT of the odd components by \(\omega^2\)
  \[
  \begin{bmatrix}
  a''_1 \\
  a''_3
  \end{bmatrix}
  = \omega^2
  \begin{bmatrix}
  a'_1 \\
  a'_3
  \end{bmatrix}
  \]
Step 4: Add these components to obtain the elements of the final spectral vector

\[
\begin{bmatrix}
A_0 \\
A_1
\end{bmatrix} = \begin{bmatrix}
a'_0 \\
a'_2
\end{bmatrix} + \begin{bmatrix}
a'_1 \\
a'_3
\end{bmatrix}
\]

\[
\begin{bmatrix}
A_2 \\
A_3
\end{bmatrix} = \begin{bmatrix}
a''_0 \\
a''_2
\end{bmatrix} + \begin{bmatrix}
a''_1 \\
a''_3
\end{bmatrix}
\]
The \(d\)-point FFT

- This is true in general, i.e., \(d\)-point DFT can be performed by first performing two \(d/2\)-point DFT computations with the even and odd indexed components and then adding these components to obtain the final spectral vector.
- The FFT is a recursive algorithm: The FFT function keeps calling itself with half size vectors, until the vector size becomes 1.
- Therefore, the arithmetic complexity of the DFT as a function of the length \(d\) can be given by the recursion

\[
T(d) = 2T(d/2) + c \cdot d
\]

for some constant \(c\).
- The solution of this recursion is \(T(d) = O(d \log d)\).
Time-Frequency Dictionary

- We will now illustrate the properties of the time and spectral vectors.
- To illustrate these properties, we will use the 8-point DFT over the field $GF(41)$ and the time vector $a = (1, 2, 3, 4, 0, 0, 0, 0)$.
- We obtained the spectral vector as $A = (10, 19, 21, 40, 39, 37, 16, 31)$.
- The computation of $A = DFT(a) = Ta$

$$
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 3 & 9 & 27 & 40 & 38 & 32 & 14 \\
1 & 9 & 40 & 32 & 1 & 9 & 40 & 32 \\
1 & 27 & 32 & 3 & 40 & 14 & 9 & 38 \\
1 & 40 & 1 & 40 & 1 & 40 & 1 & 40 \\
1 & 38 & 9 & 14 & 40 & 3 & 32 & 27 \\
1 & 32 & 40 & 9 & 1 & 32 & 40 & 9 \\
1 & 14 & 32 & 38 & 40 & 27 & 9 & 3
\end{bmatrix}
\begin{bmatrix}
1 \\
2 \\
3 \\
4 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
= 
\begin{bmatrix}
10 \\
19 \\
21 \\
40 \\
39 \\
37 \\
16 \\
31
\end{bmatrix}
= A
$$
Property 1: Sum of Sequence

- The sum of the coefficients of a time vector equals to the zeroth coefficient of its spectral vector.
- The sum of the elements of the time vector in GF(41) is

$$1 + 2 + 3 + 4 = 10 \pmod{41}$$

which is equal to the 0th element of the spectral vector.

- Conversely the sum of the spectrum coefficients equals to $d^{-1}$ times the zeroth coefficient of the time vector.
- The inversion is performed in the ring or field, in our case GF(41).
- Here, $d^{-1} = 8^{-1} \pmod{41}$ is found as 36

$$36 \cdot (10 + 19 + 21 + 40 + 39 + 37 + 16 + 31) = 1 \pmod{41}$$

which is equal to the 0th element of the time vector.
Property 2: One-Term Right Circular Shift

- Given the time polynomial \( a(x) \), its one-term right circular shift is defined as the polynomial \( b(x) \) such that

\[
\begin{align*}
a(x) &= a_0 + a_1 x + \cdots + a_{d-2} x^{d-1} + a_{d-1} x^{d-1} \\
b(x) &= a_1 + a_2 x + \cdots + a_{d-1} x^{d-2} + a_0 x^{d-1}
\end{align*}
\]

- In the vector notation we have

\[
\begin{align*}
a &= (a_0, a_1, \ldots, a_{d-2}, a_{d-1}) \\
b &= (a_1, a_2, \ldots, a_{d-1}, a_0)
\end{align*}
\]

- Also assume that \( A \) and \( B \) are the spectral transformations of the time vectors \( a \) and \( b \), with \( A = (A_0, A_1, \ldots, A_{d-1}) \) and \( B = (B_0, B_1, \ldots, B_{d-1}) \) such that \( A = \mathcal{T} a \) and \( B = \mathcal{T} b \).
If we have $A$ available, we can obtain $B$, by going back to the time domain with $a = \mathcal{T}^{-1}A$ and performing the one-term right circular shift in the time domain, and then returning back to the spectral domain to get $B$.

As we have discovered, there is another more direct method: the spectral vector $B$ of the one-term right circular shift of $a$ is given as

$$B = A \odot \Gamma$$

such that $\Gamma = (1, \omega^{-1}, \omega^{-2}, \ldots, \omega^{-(d-1)})$

Let us verify this property in our 8-point DFT system over GF(41)
Property 2: One-Term Right Circular Shift

- Given \( A = (10, 19, 21, 40, 39, 37, 16, 31) \), we compute \( B = A \odot \Gamma \)
- First we need to compute the vector \( \Gamma \):
  \[
  \Gamma = (1, \omega^{-1}, \omega^{-2}, \ldots, \omega^{-d+1})
  = (1, 14, 32, 38, 40, 27, 9, 3)
  
\] since \( \omega = 3 \), \( \omega^{-1} = 14 \), \( \omega^{-2} = 32 \), etc. in GF(41)
- Now, we obtain \( B = A \cdot \Gamma \) as

\[
B = \begin{bmatrix}
10 \\
19 \\
21 \\
40 \\
39 \\
37 \\
16 \\
31 \\
\end{bmatrix}
\begin{bmatrix}
1 \\
14 \\
32 \\
38 \\
40 \\
27 \\
9 \\
3 \\
\end{bmatrix}
= \begin{bmatrix}
10 \\
20 \\
16 \\
3 \\
2 \\
15 \\
21 \\
11 \\
\end{bmatrix}
\text{(mod 41)}
\]
Property 2: One-Term Right Circular Shift

To verify that $B$ is indeed the spectral vector of the one-term right circular shift of $a$, we apply inverse DFT to this vector in GF(41)

$$b = \mathcal{T}^{-1}B = \begin{bmatrix} 36 & 36 & 36 & 36 & 36 & 36 & 36 & 36 \\ 36 & 12 & 4 & 15 & 5 & 29 & 37 & 26 \\ 36 & 4 & 5 & 37 & 36 & 4 & 5 & 37 \\ 36 & 15 & 37 & 12 & 5 & 26 & 4 & 29 \\ 36 & 5 & 36 & 5 & 36 & 5 & 36 & 5 \\ 36 & 29 & 4 & 26 & 5 & 12 & 37 & 15 \\ 36 & 37 & 5 & 4 & 36 & 37 & 5 & 4 \\ 36 & 26 & 37 & 29 & 5 & 15 & 4 & 12 \end{bmatrix} \begin{bmatrix} 10 \\ 20 \\ 16 \\ 3 \\ 15 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Thus, we obtain the same vector
Property 3: One-Term Right Logical Shift

Given the time polynomial $a(x)$, its one-term right logical shift is defined as the polynomial $b(x) = (a(x) - a_0)/x$ such that

$$a(x) = a_0 + a_1x + \cdots + a_{d-2}x^{d-1} + a_{d-1}x^{d-1}$$

$$b(x) = a_1 + a_2x + \cdots + a_{d-1}x^{d-2}$$

In the vector notation we have

$$a = (a_0, a_1, \ldots, a_{d-2}, a_{d-1})$$

$$b = (a_1, a_2, \ldots, a_{d-1}, 0)$$

Also assume that $A$ and $B$ are the spectral transformations of the time vectors $a$ and $b$, with $A = (A_0, A_1, \ldots, A_{d-1})$ and $B = (B_0, B_1, \ldots, B_{d-1})$ such that $A = \mathcal{T}a$ and $B = \mathcal{T}b$
Property 3: One-Term Right Logical Shift

- If we have $A$ available, we can obtain $B$, by going back to the time domain with $a = T^{-1}A$ and performing the one-term right logical shift in the time domain, and then returning back to the spectral domain to get $B$.

- As we have discovered, there is another more direct method: the spectral vector $B$ of the one-term right circular shift of $a$ is given as

$$B = (A - a_0) \odot \Gamma$$

such that $a_0$ is the 0th element of the time vector (which is obtained using Property 1) and $\Gamma = (1, \omega^{-1}, \omega^{-2}, \ldots, \omega^{-d+1})$.

- Let us verify this property in our 8-point DFT system over GF(41).
Property 2: One-Term Right Circular Shift

Given \( A = (10, 19, 21, 40, 39, 37, 16, 31) \), we first compute \( a_0 \) using the Property 1:

\[
\begin{align*}
a_0 &= d^{-1} \cdot (A_0 + A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7) \\
&= 36 \cdot (10 + 19 + 21 + 40 + 39 + 37 + 16 + 31) \\
&= 1
\end{align*}
\]

Now we compute \( (A - a_0) \) vector, which is accomplished by subtracting \( a_0 \) from every element of \( A \)

\[
A - a_0 = (9, 18, 20, 39, 38, 36, 15, 30)
\]
Property 2: One-Term Right Circular Shift

Finally, we obtain $B$ using $(A - a_0) \odot \Gamma$ as

\[
B = \begin{bmatrix}
9 \\
18 \\
20 \\
39 \\
38 \\
36 \\
15 \\
30
\end{bmatrix} \odot \begin{bmatrix}
1 \\
14 \\
32 \\
38 \\
40 \\
27 \\
9 \\
3
\end{bmatrix} = \begin{bmatrix}
9 \\
6 \\
25 \\
6 \\
3 \\
29 \\
12 \\
8
\end{bmatrix} \pmod{41}
\]

To verify that $B$ is indeed the DFT of one-term right logical shift $a$, we apply inverse DFT to the vector $B = (9, 6, 25, 6, 3, 29, 12, 8)$
Property 2: One-Term Right Circular Shift

- We compute $T^{-1}B$ is computed using

$$b = T^{-1}B = \begin{bmatrix}
36 & 36 & 36 & 36 & 36 & 36 & 36 & 36 \\
36 & 12 & 4 & 15 & 5 & 29 & 37 & 26 \\
36 & 4 & 5 & 37 & 36 & 4 & 5 & 37 \\
36 & 15 & 37 & 12 & 5 & 26 & 4 & 29 \\
36 & 5 & 36 & 5 & 36 & 5 & 36 & 5 \\
36 & 29 & 4 & 26 & 5 & 12 & 37 & 15 \\
36 & 37 & 5 & 4 & 36 & 37 & 5 & 4 \\
36 & 26 & 37 & 29 & 5 & 15 & 4 & 12
\end{bmatrix} \begin{bmatrix} 9 \\ 6 \\ 25 \\ 6 \\ 3 \\ 29 \\ 12 \\ 8 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- Thus, we obtain the same vector