The Alien's Multiplication Challenge: DFT to the Rescue!

CSE 220 Online on DFT
Section A

Time: 40 minutes

Scenario

One peaceful evening in the small village of Digiton, a strange alien descended from the sky. The alien presented the villagers with a challenge:

”Here are two gigantic numbers. Multiply them, or I'll capture your village!”

The numbers were so large that no one in Digiton could solve the problem—not even the village mathematician with their trusty calculator. Desperate villagers turned to you, the Wizard of Discrete Fourier Transform, to save them.

Will you rise to the occasion, or will the villagers fall to the alien's wrath? Solve it with the elegance of Fourier. The universe awaits your answer.

Your Mission

You will be given 2 numbers as input. With your secret technique, DFT-based multiplication, you must:

- **Break the alien numbers into smaller ”digit pieces”**: Convert the large numbers into arrays of their digits.
- **Use the (DFT)**: Apply the Discrete Fourier Transform (DFT) using **FFT** to transform the digit arrays into the frequency domain and apply appropriate dft property . See the **hint** part for proper understanding.

- **Use the Inverse DFT**: Apply the Inverse DFT using **IFFT** to transform the result back into the time domain.
- **Carefully combine any extra parts (carry-overs)**: Process the resulting array to handle carry-over values and construct the final product.

Hint

Multiplication problem can be thought of extension of polynomial multiplication. Suppose we want to multiply 123 & 456. Represent the numbers 123 and 456 as polynomials, where the digits of the numbers become the coefficients of powers of x:

$$123 = 1 \cdot x^2 + 2 \cdot x^1 + 3 \cdot x^0 \quad \Rightarrow \quad [1, 2, 3]$$

$$456 = 4 \cdot x^2 + 5 \cdot x^1 + 6 \cdot x^0 \quad \Rightarrow \quad [4, 5, 6]$$

Polynomial multiplication can be achieved by **linear convolution.** You can calculate linear convolution with the help of circular convolution just by padding appropriate number of 0s (How many?find it!) to both of the arrays and by taking circular convolution of that. You must use FFT and IDFT to calculate it. After taking IDFT, you need to round the result to the nearest integer number. After polynomial multiplication, you will get,

$$4 \cdot x^4 + 13 \cdot x^3 + 28 \cdot x^2 + 27 \cdot x + 18$$

. To calculate the final answer from here, you need to just carry over from the rightmost part.

$$18 \quad \Rightarrow \text{keep } 8, \text{ carry over } 1$$

$$27 + 1 = 28 \quad \Rightarrow \text{keep } 8, \text{ carry over } 2$$

$$28 + 2 = 30 \quad \Rightarrow \text{keep } 0, \text{ carry over } 3$$

$$13 + 3 = 16 \quad \Rightarrow \text{keep } 6, \text{ carry over } 1$$

$$4 + 1 = 5 \quad \Rightarrow \text{keep } 5$$

$$\text{Final result: } 56088$$

Test Case

65767879797907 x 765454532435435345 = 50342321679976816694244164822915

Restriction

You cannot use naive DFT to solve it. Use FFT & IFFT.

Submission

Just submit a single python file after renaming it with your student ID: 2105XXX.py