Student Name: **A Z Hasnain Kabir**
Student ID: **200042102**

CSE 4403

Algorithms

Assignment on Fast Fourier Transform

Introduction

The Fast Fourier Transform (FFT) is a widely used algorithm that plays a fundamental role in various fields, including signal processing, image analysis, data compression, and audio/video processing. The FFT algorithm exploits the inherent symmetry and periodicity properties of the DFT (Discrete Fourier Transform), allowing efficient computation of the frequency spectrum of a discrete signal. By decomposing a time-domain signal into its constituent frequencies, the FFT enables us to extract valuable information and gain insights into the underlying characteristics of the signal.

Benefits of using FFT

Fast Fourier transform algorithm is known for its computational efficiency. The traditional DFT algorithm has a complexity of O(N^2), where N is the number of samples in the input signal. In contrast, the FFT algorithm reduces the complexity to O(N log N), making it feasible to analyze large datasets in real-time.

Application in improving Polynomial Time Complexity

The Fast Fourier Transform (FFT) algorithm has a significant application in improving polynomial multiplications. Polynomial multiplication involves multiplying two polynomials together to obtain a resulting polynomial. The traditional polynomial multiplication algorithm has a time complexity of O(n^2), where n is the degree of the polynomials. However, the FFT algorithm reduces this complexity to O(n log n), resulting in substantial speedup for large polynomials.

Example

Suppose we have two polynomials $P(x) = 3x^2 + 2x + 1$ and $Q(x) = x^2 - x + 1$

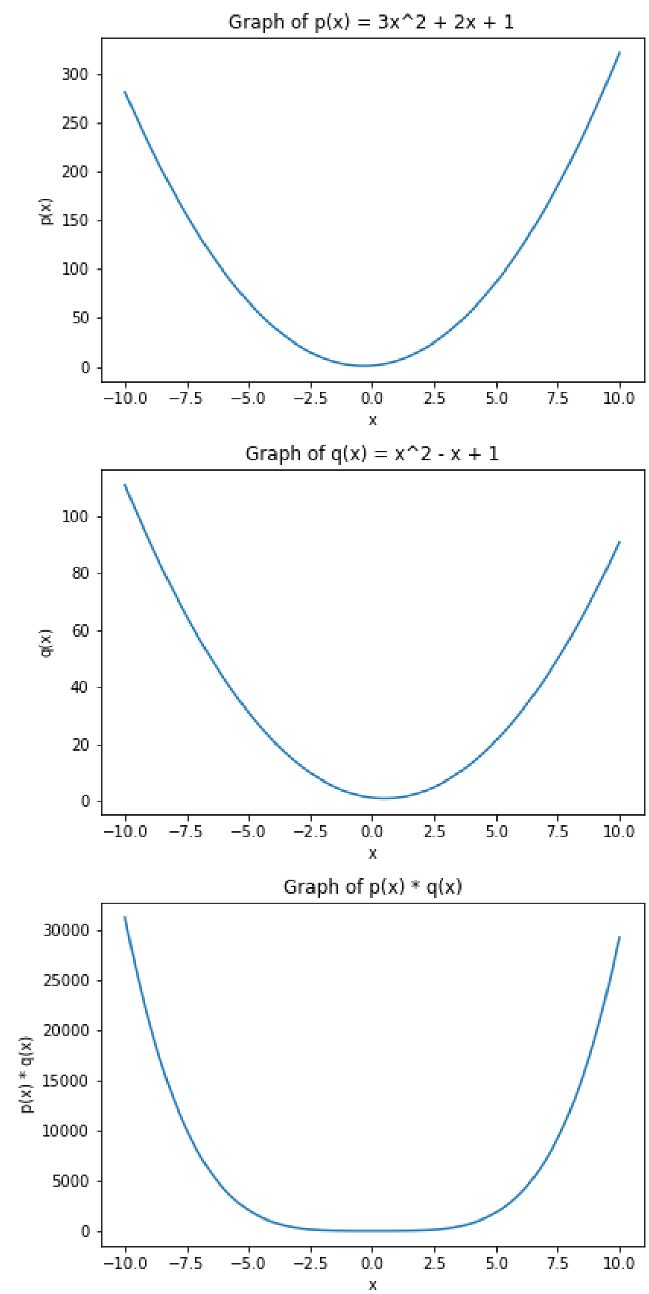
Traditional approach

The traditional computation of multiplying two polynomials, P(x) and Q(x), involves using the distributive property to multiply each term of P(x) with each term of Q(x). This process is commonly known as polynomial long multiplication. Here's a step-by-step explanation of the traditional computation:

1. Initialize an empty result polynomial R(x) with all coefficients set to zero.

2. For each term in P(x):
   - Multiply the term with each term in Q(x).
   - Add the resulting terms to the corresponding degree positions in R(x).
3. The resulting polynomial R(x) is the product of P(x) and Q(x).

The traditional polynomial multiplication algorithm has a time complexity of $O(n^2)$, where n is the degree of the polynomials. Each term in P(x) needs to be multiplied by every term in Q(x), resulting in a large number of multiplications and additions.



Fast Fourier Transform

In the FFT approach,

1. Zero-padding and FFT: To apply the FFT algorithm, both P(x) and Q(x) are zero-padded to the same degree, which is a power of 2. This step ensures that the FFT algorithm works efficiently.

2. Frequency Domain Multiplication: By transforming the polynomials from the time domain to the frequency domain using the FFT, the polynomial multiplication is converted into a pointwise multiplication of the frequency spectra. Multiplying the corresponding coefficients in the frequency domain is equivalent to multiplying the terms in the time domain.

3. Inverse FFT (IFFT): After the pointwise multiplication in the frequency domain, the resulting polynomial is obtained by applying the Inverse FFT algorithm (IFFT) to the product's frequency spectrum. This step brings the polynomial back to the time domain.

The FFT algorithm significantly reduces the number of multiplications required compared to traditional long multiplication. Instead of multiplying every term with every term, the FFT-based approach performs a pointwise multiplication in the frequency domain, reducing the complexity to O(n log n).