Number-theoretic transform (integer DFT)

Introduction

The NTT is a generalization of the classic DFT to finite fields. With a lot of work, it basically lets one perform fast convolutions on integer sequences without any round-off errors, guaranteed. Convolutions are useful for multiplying large numbers or long polynomials, and the NTT is asymptotically faster than other methods like Karatsuba multiplication.

Review of the complex DFT

The classic discrete Fourier transform (DFT) operates on vectors of complex numbers:

1. Suppose the input vector has length n. The output vector will also have length n.
2. Let ω (omega) be a primitive nth root of unity. In other words, $\omega^n = 1$, but $\omega^k \neq 1$ for all integers $1 \leq k < n$. The standard choice for the DFT is $\omega = e^{-2\pi i/n}$.
3. Each output element equals a particular weighted sum of all input elements, using some powers of ω as weights. Denoting the input vector as X and the output vector as Y, we have (with 0-based indexing): $Y(k) = X(0)\omega^{0k} + X(1)\omega^{1k} + X(2)\omega^{2k} + \ldots + X(n-1)\omega^{(n-1)k}$.
4. The inverse transform, which restores the original vector, is given by: $X(k)n = Y(0)\omega^{-0k} + Y(1)\omega^{-1k} + Y(2)\omega^{-2k} + \cdots + Y(n-1)\omega^{-(n-1)k}$.

Additional notes:

- Linear algebra guarantees that if ω is indeed a primitive nth root of unity of the working field, then the inverse transform is correct.
- One application of the DFT is to compute a circular convolution of two same-length vectors X and Y. To do so, compute the forward DFT on X and Y, then point-wise multiply the two vectors together, compute the inverse transform on this vector, and finally divide by a scaling factor of n. Again, this procedure will work in any suitable field of numbers.
- The DFT's most popular application of analyzing/representing a signal as a sum of sine waves doesn't work in the number-theoretic transform.

Procedure for the NTT

1. Suppose the input vector is a sequence of n non-negative integers.
2. Choose a minimum working modulus M such that $1 \leq n < M$ and every input value is in the range $[0, M)$.
3. Select some integer $k \geq 1$ and define $N = kn + 1$ as the working modulus. We require $N \geq M$, and N to be a prime number. Dirichlet's theorem guarantees that for any n and M, there exists some choice of k to make N be prime.
4. Because N is prime, the multiplicative group of $\mathbb{Z}_N$ has size $\varphi(N) = N - 1 = kn$. Furthermore, the group must have at least one generator g, which is also a primitive

$(N-1)$th root of unity.

5. Define $\omega \equiv g^k \bmod N$. We have $\omega^n = g^{kn} = g^{N-1} = g^{\varphi(N)} \equiv 1 \bmod N$ due to Euler's theorem. Furthermore because g is a generator, we know that $\omega^i = g^{ik} \not\equiv 1$ for $1 \leq i < n$, because $ik < nk = N-1$. Hence ω is a primitive nth root of unity, as required by the DFT of length n.

6. The rest of the procedure for the forward and inverse transforms is identical to the complex DFT. Moreover, the NTT can be modified to implement a fast Fourier transform algorithm such as Cooley-Tukey.

Additional notes:

- When convolving two vectors of length n where each input value is at most m, the upper bound on each output value is $m^2 n$. Choosing a minimum working modulus of $M = m^2 n + 1$ is sufficient to always avoid overflow in the worst case.

- For the prime field $\mathbb{Z}_N$, we can either find a generator of the field then derive a primitive nth root of unity (as mentioned in the procedure), or directly find a primitive nth root of unity.

  - Such a field has $\varphi(\varphi(N)) = \varphi(N-1) = \varphi(kn)$ generators but $\varphi(n)$ primitive nth roots of unity. The first number is greater than or equal to the second number. So if we are sampling random candidates in the range $[0, N)$, we are more likely (or equally likely) to succeed in finding a generator than finding a primitive nth root.

  - To find a generator, first fully factorize $N-1$ and collect its set of unique prime factors. A candidate $a \in (1, N)$ is a generator of $\mathbb{Z}_N$ if and only if {for each p in that set of unique prime factors, $a^{(N-1)/p} \not\equiv 1 \bmod N$}.

  - To find a primitive root directly, first fully factorize n and collect its set of prime factors. A candidate $a \in (1, N)$ is a primitive nth root of unity in $\mathbb{Z}_N$ if and only if {$a^n \equiv 1 \bmod N$, and for each p in that set of unique prime factors, $a^{n/p} \not\equiv 1 \bmod N$}.

- Although the procedure only describes prime fields for simplicity, it might also be possible to operate on composite rings (e.g. $\mathbb{Z}_{100}$) if a primitive nth root of unity exists. This could be useful if a composite modulus is much smaller than a prime modulus of the form $N = kn + 1$.

- Furthermore, it should be possible to operate on prime-power fields such as $\mathrm{GF}(2^8)$. However, this is only useful if the input vector is composed of elements from that field, instead of being plain integers.

- Computing an NTT requires many modular multiplications. It is possible to apply Montgomery reductions (or the less efficient Barrett reductions) to speed up the modular arithmetic in an NTT.

Examples

- Input vector $X = (6, 0, 10, 7, 2)$ with length $n = 5$. Compute a forward transform.

  1. Define the minimum working modulus of $M = 11$. We have $1 \leq n < M$, and every input value is in the range $[0, 11)$.

  2. Select $k = 2$ so that with $N = kn + 1 = 2 \times 5 + 1 = 11$, we have $N \geq M$, and N is prime.

3. Try to find a generator of $\mathbb{Z}_{11}$. Fully factorize $N-1=10=2\times 5$. Choose a candidate $a=6$. We have $\{a^{10/2}=a^5\equiv 10\not\equiv 1 \bmod 11\}$ and $\{a^{10/5}=a^2\equiv 3\not\equiv 1 \bmod 11\}$, so $g=6$ is indeed a generator.
4. Calculate $\omega=g^k=6^2\equiv 3 \bmod 11$. This is our primitive 5th root of unity.
5. Compute the output vector Y:
$Y(0)=X(0)\omega^{0\times 0}+X(1)\omega^{0\times 1}+X(2)\omega^{0\times 2}+X(3)\omega^{0\times 3}+X(4)\omega^{0\times 4}\equiv 3 \bmod 11.$
$Y(1)=X(0)\omega^{1\times 0}+X(1)\omega^{1\times 1}+X(2)\omega^{1\times 2}+X(3)\omega^{1\times 3}+X(4)\omega^{1\times 4}\equiv 7 \bmod 11.$
$Y(2)=X(0)\omega^{2\times 0}+X(1)\omega^{2\times 1}+X(2)\omega^{2\times 2}+X(3)\omega^{2\times 3}+X(4)\omega^{2\times 4}\equiv 0 \bmod 11.$
$Y(3)=X(0)\omega^{3\times 0}+X(1)\omega^{3\times 1}+X(2)\omega^{3\times 2}+X(3)\omega^{3\times 3}+X(4)\omega^{3\times 4}\equiv 5 \bmod 11.$
$Y(4)=X(0)\omega^{4\times 0}+X(1)\omega^{4\times 1}+X(2)\omega^{4\times 2}+X(3)\omega^{4\times 3}+X(4)\omega^{4\times 4}\equiv 4 \bmod 11.$

- Input vector $Y=(3,7,0,5,4)$ with length $n=5$. Compute the corresponding inverse transform for the example above.

1. We must use the same values of $k=2$, $N=11$, and $\omega=3$ as in the forward transform. Note that $\omega^{-1}\equiv 4 \bmod 11$.
2. Compute the unscaled output vector Xn:
$X(0)n=Y(0)\omega^{-0\times 0}+Y(1)\omega^{-0\times 1}+\cdots+Y(4)\omega^{-0\times 4}\equiv 8 \bmod 11.$
$X(1)n=Y(0)\omega^{-1\times 0}+Y(1)\omega^{-1\times 1}+\cdots+Y(4)\omega^{-1\times 4}\equiv 0 \bmod 11.$
$X(2)n=Y(0)\omega^{-2\times 0}+Y(1)\omega^{-2\times 1}+\cdots+Y(4)\omega^{-2\times 4}\equiv 6 \bmod 11.$
$X(3)n=Y(0)\omega^{-3\times 0}+Y(1)\omega^{-3\times 1}+\cdots+Y(4)\omega^{-3\times 4}\equiv 2 \bmod 11.$
$X(4)n=Y(0)\omega^{-4\times 0}+Y(1)\omega^{-4\times 1}+\cdots+Y(4)\omega^{-4\times 4}\equiv 10 \bmod 11.$
3. Finally, multiply each element by $n^{-1}=5^{-1}\equiv 9 \bmod 11$ to get back the original values: $(8,0,6,2,10)\times 9\equiv(6,0,10,7,2) \bmod 11$

- Input vectors $X=(4,1,4,2,1,3,5,6)$ and $Y=(6,1,8,0,3,3,9,8)$ with length $n=8$. Compute their circular convolution.

1. Define the minimum working modulus of $M=9\times 9\times 8+1=649$, because each input value is a single-digit integer (less than 10) and we want to avoid the convolution output from overflowing.
2. Select $k=84$ so that with $N=84\times 8+1=673$, we have $N\geq M$, and N is prime.
3. Try to find a primitive 8th root of unity directly. Fully factorize $n=8=2\times 2\times 2$. Choose a candidate $a=326$. We have $\{a^8\equiv 1 \bmod 673\}$ and $\{a^{8/2}=a^4\equiv 672\not\equiv 1 \bmod 673\}$, so $\omega=326$ is indeed a primitive 8th root of unity.
4. Compute the forward transforms:
$X'=\mathrm{NTT}(X)=(26,338,228,115,2,457,437,448)$
$Y'=\mathrm{NTT}(Y)=(38,594,224,157,14,201,433,406)$
5. Compute the point-wise multiplication of the two vectors modulo N: $Z'=X'Y'=(315,218,597,557,28,329,108,178)$
6. Compute the inverse NTT with scaling: $Z=\mathrm{INTT}(Z')=(123,120,106,92,139,144,140,124)$
7. This result represents a circular convolution because, for example, $Z(0)=X(0)Y(0)+X(1)Y(7)+X(2)Y(6)+\cdots+X(7)Y(1)$. That is to say,

$$\text{each } Z(i) \equiv \sum_{j=0}^{n-1} X(j)Y((i-j) \bmod n) \bmod N.$$

Source code

Python

- numbertheoretictransform.py
- numbertheoretictransform-test.py

Java (size)

- SmallNumberTheoreticTransform.java
- SmallNumberTheoreticTransformTest.java

Java (size)

- BigNumberTheoreticTransform.java
- BigNumberTheoreticTransformTest.java

Library features:

- Finding suitable prime modulus
- Finding primitive nth root of unity
- Computing forward and inverse transforms (naive $\Theta(n^2)$ algorithm)
- Computing fast forward transform (Cooley-Tukey $\Theta(n \log n)$ algorithm)
- Computing circular convolution (using naive algorithm)
- Unit tests for all functions (some hard-coded vectors, some randomized cases)

Proof of DFT/NTT correctness

1. The forward NTT is given by the following matrix:
$$A = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}.$$

2. The unscaled inverse NTT is given by the following matrix:
$$B = \begin{bmatrix} \omega^{-0} & \omega^{-0} & \omega^{-0} & \cdots & \omega^{-0} \\ \omega^{-0} & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ \omega^{-0} & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^{-0} & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)^2} \end{bmatrix}.$$

3. We want to show that their product is a scaled identity matrix, i.e. $C = AB = nI$, where
$$I = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

4. Examine an arbitrary cell on the main diagonal:
$$C_{i,i} = [A \text{ row } i] \cdot [B \text{ col } i] = \sum_{k=0}^{n-1} \omega^{ik}\omega^{-ik} = \sum_{k=0}^{n-1} \omega^0 = \sum_{k=0}^{n-1} 1 = n$$

5. Now examine an arbitrary cell off the main diagonal, with $i \neq j$, letting $\delta = i - j$, where $-n < \delta < n$ and $\delta \neq 0$:
$$C_{i,j} = [A \text{ row } i] \cdot [B \text{ col } j] = \sum_{k=0}^{n-1} \omega^{ik}\omega^{-jk} = \sum_{k=0}^{n-1} \omega^{(i-j)k} = \sum_{k=0}^{n-1} \omega^{\delta k}.$$

   This sum is a geometric series. We know that in general,
$$1 + x + x^2 + x^3 + \cdots + x^{n-1} = \frac{x^n - 1}{x - 1}.$$

   Hence $C_{i,j} = \sum_{k=0}^{n-1} \omega^{\delta k} = \frac{\omega^{\delta n} - 1}{\omega^{\delta} - 1} = 0$, which requires a bit more explanation.

   In the numerator, $\omega^n = 1$ because ω is a (primitive) nth root of unity, so $(\omega^n)^\delta = 1$ and the overall numerator is 0.

   In the denominator, ω is a *primitive* nth root of unity and $-n < \delta < n$ and $\delta \neq 0$, thus $\omega^\delta \neq 1$, which avoids division by zero.

6. Therefore we have proven that the matrix product AB is a scaled identity matrix, which shows that the number-theoretic transform is invertible up to a scale factor.

More info

- apfloat: Number theoretic transforms
- Wikipedia: Discrete Fourier transform (general) - Number-theoretic transform
- University of California, Los Angeles: EE133A (Applied Numerical Computing) - Orthogonal matrices

Categories: Programming, Math, Python Last updated: 2017-06-07

Feedback: Question/comment? Contact me

ProjectNayuki: Like, comment, follow updates on Facebook