COMP3121 Homework Q3

Arth Sanskar Patel
z5228942

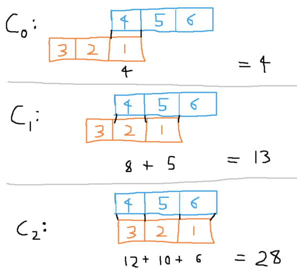
June 29, 2020

1 Answer

Since after doing IDFT of a sequence C (convolution of A and B), we get

$$C = \langle \sum_{i=0}^{j} A_i B_{j-i} \rangle$$

Since there is multiplication of $A_i B_{j-i}$ to get a coefficient, we hav e to reverse the sequence B in order to get the proper answer as shown in the image below. This image is taken from the instructor's answer on piazza post 406.



Here in the question, we have

$$A = \langle A_0, A_1, A_2, \dots, A_{100n-1} \rangle$$

And sequence N of net with holes as, sequence of n 1's and 0's where 0 represents holes. Let N' be the reverse sequence of N. Since the length of N is n lets pad A with n-1 0's in front and back.

$$A = \langle \underbrace{0, 0, \dots, 0}_{n-1}, A_0, A_1, \dots, A_{100n-1}, \underbrace{0, 0, \dots, 0}_{n-1} \rangle$$

The 0's we padded are only imaginary which will help up visualize the answer as shown in lecture slide 30 of Fast Fourier Transform Lecture. Now we need to calculate $C = A * N'$ Let $P_A(x)$ be the polynomial for A and $P_B(x)$ be the polynomial for N'. The polynomials can be calculated in $O(n)$ time.

$$P_A(x) = A_0 + A_1 x + A_2 x^2 + \dots + A_{100n-1} x^{100n-1}$$

$$P_B(x) = B_0 + B_1 x + \dots + B_{n-1} x^{n-1}$$

where B_i represents either 1 or 0. DFT of $P_A(x)$ and $P_B(x)$ can be calculated in $O(nlogn)$ time. As the result of convolution of sequence A of length 100n and N' of length n will be of length 101n - 1, we need to evaluate DFT for 101n -1 roots of unity

$$DFT(A) = \langle P_A(1), P_A(\omega_{101n-1}), P_A(\omega_{101n-1}^2), \dots, P_A(\omega_{101n-1}^{101n-2}) \rangle$$

$$DFT(B) = \langle P_B(1), P_B(\omega_{101n-1}), P_B(\omega_{101n-1}^2), \ldots, P_B\omega_{101n-1}^{101n-2})\rangle$$

Product of DFT(A) and DFT(B) can be done in $O(n)$ time which gives us DFT(C).

$$DFT(C) = \langle P_A(1)P_B(1), P_A(\omega_{101n-1})P_B(\omega_{101n-1}), P_A(\omega_{101n-1}^2)P_B(\omega_{101n-1}^2), \ldots, P_A\omega_{101n-1}^{101n-2})P_B\omega_{101n-1}^{101n-2})\rangle$$

Using IDFT on DFT(C) we get the polynomial C in $O(nlogn)$ time. This can be converted to the sequence C in $O(n)$ time and we can search for the highest coefficient in $O(n)$ time to get us the index at which we have to place the net to maximise our gains for the fishing. Since the highest time taken throughout out our process is $O(nlogn)$, the total time for this algorithm is $O(nlogn)$.