Algorithms, May-June 2020 at CIS

Homework 3

1. Suppose N is a power of 2. Recall a primitive N-th root of unity is a possibly complex number g for which $g^0, g^1, g^2, \ldots, g^{N-1}$ are all distinct and $g^0 = g^N = 1$, and the N-th roots of unity are $e^{-2\pi ij/N}$ for $j = 0, 1, 2, \ldots, N-1$. How many primitive N-th roots of unity are there?

2. Show that if $\mathrm{DFT}_N[j,k] = \omega^{kj}$, where ω is a primitive N-th root of unity, then show that $\mathrm{IDFT}_N[j,k] = \frac{1}{N}\omega^{-kj}$ is the inverse matrix.

3. Suppose you have a string $s = (s_0, s_1, ..., s_{n-1})$ of length n with entries in $\{-1, 1\}$ and a string $t = (t_0, t_1, ..., t_{10n-1})$ of length $10n$ with entries in $\{-1, 1\}$. We say position k of t is a match if $t_k = s_{n-1}, t_{k-1} = s_{n-2}, \ldots, t_{k-n+1} = s_0$. Note if $k - n + 1 < 0$, there is no match at position k. Show how to use the FFT to find all matches in $O(n \log n)$ time. Hint: if you have two polynomials $p = \sum_{i=0}^{n-1} a_i x^i$ and $q = \sum_{i=0}^{10n-1} b_i x^i$, then the k-th coefficient of $p \cdot q$ is $\sum_{i \le k} a_i b_{k-i}$.