**Algorithms, Spring 1997**

**FFT Assignment**

Due in one week.

We wish to compute FFT's of polynomials of degree $n-1$ using integer arithmetic instead of complex numbers. Assume $n$ is a power of 2. Suppose $t$ is a positive integer. Let

$$\omega = 2^t \qquad \text{and} \qquad m = 2^{tn/2} + 1.$$

We will argue that we can work with the integers modulo $m$, with $\omega$ playing the same role as the complex $n$ th root of unity did.

a) Show that there is an integer $\theta$ with $1 \leq \theta \leq m-1$ with $\theta\omega \equiv 1 \pmod{m}$. So, $\theta$ is the inverse of $\omega$. Show also that both $\omega$ and $\theta$ are primitive $n$ th roots of unity (see definition 9.9 of Manber) modulo $m$.

b) Show that if an $n \times n$ matrix $V(\omega)$ is defined as in 9.10 of Manber (so, $V(\omega)$ times the vector of coefficients of a polynomial of degree $n-1$ gives the values of the polynomial at $\omega^0.\omega^1, \ldots \omega^{n-1}$), then we have

$$\frac{1}{n} V(\omega)V(\theta) \equiv I \pmod{m},$$

where $I$ is the identity and the above equation is modulo $m$.

So, both forward and inverse FFT's can be done modulo $m$.