**6.S897 Algebra and Computation** Feb 18, 2015

# Lecture 5

*Lecturer: Madhu Sudan* *Scribe: Ilya Razenshteyn*

## 1 Multiplication of polynomials

In this lecture we will study, how to multiply two polynomials in time $O\big(n \log^{O(1)} n\big)$, where $n$ is the degree of a product. First let us define two problems that are useful for fast multiplication, and, at the same time, very important by themselves.

In the *multipoint evaluation* we are given a polynomial $p \in R[x]$ and several points $\alpha_1, \ldots, \alpha_n \in R$. The goal is to compute $\beta_1, \ldots, \beta_n \in R$, where $\beta_i = p(\alpha_i)$. The naïve algorithm works in time $O(\deg p \cdot n)$, but the hope is that for special sets of points the evaluation can be sped up substantially.

The complementary problem is that of *polynomial interpolation.* Here we are given $n$ distinct points $\alpha_1, \ldots, \alpha_n \in R$ and $n$ arbitrary values $\beta_1, \ldots, \beta_n \in R$. The goal is to construct a polynomial $p \in R[x]$ *of degree less than* $n$ such that $p(\alpha_i) = \beta_i$ for every $i$. A caveat is that such a polynomial does not necessarily exist. A conceptually simple (if slow) algorithm is to solve the corresponding system of linear equations. This algorithm implies, in particular, that over a field interpolation is always possible.

Our overall strategy to multiply two polynomials $p$ and $q$ will be the following:

- Evaluate $p$ and $q$ on some set of points $\alpha_i$
- Multiply the corresponding values "pointwise"
- Interpolate the corresponding values to recover $pq$.

We need $\alpha_i$'s to satisfy the following two conditions. First, the points should be nice enough to be able to evaluate and interpolate quickly. Second, there should be enough of them to be able to interpolate.

### 1.1 Discrete Fourier Transform

The first simple case is when a ring of interest $R$ contains a primitive root of unity $\omega$ of sufficiently high degree. Namely, suppose that $\omega^n = 1$, where $n = 2^k$, for every $0 < i < n$ one has $\omega^i \neq 1$, and that $\deg p + \deg q < n$, where $p$ and $q$ are the polynomials we are willing to multiply.

In this case, it turns out, one can quickly evaluate and interpolate for the set of points $\alpha_i = \omega^i$, $0 \leq i < n$ using *Fast Fourier Transform (FFT).* The key observation is that the map $x \mapsto x^2$ is 2-to-1 between $\{\omega^0, \omega^1, \ldots, \omega^{n-1}\}$ and $\{\omega^0, \omega^2, \omega^4, \ldots, \omega^{n-2}\}$. This observation allows the following clean recursive procedure for the evaluation of $p$ with $\deg p < n$ on $\omega^i$:

- Decompose $p(x) = x \cdot p_1(x^2) + p_0(x^2)$
- Recursively evaluate $p_0$ and $p_1$ on $\{\omega^0, \omega^2, \ldots, \omega^{n-2}\}$
- Compute $p(\omega^i)$ using evaluations of $p_0$ and $p_1$.

After a moment of reflection, it is clear that we can perform decomposition and the final computation in the linear time $O(n)$. Since $p_0$ and $p_1$ are of degree less than $n/2$, the overall running time is $O(n \log n)$. This consists of $O(n \log n)$ additions and multiplications of the special form (by a power of $\omega$).

To perform the interpolation on $\omega^i$, one can just observe that we can literally invert the FFT going bottom-up.

## 1.2 Evaluation on subspaces

If we live in characteristic 2, we can instead evaluate a polynomial on all points of some *additive* subspace $V = \mathrm{span}\langle \alpha_1, \dots, \alpha_k \rangle$ that consists of $2^k$ points.

Here one can observe that the map $x \mapsto q(x) = x^2 - \alpha_1 x$ is a 2-to-1 map from $V$ to $\mathrm{span}\langle q(\alpha_2), \dots, q(\alpha_k) \rangle$, since $q$ is a linear map ($q(0) = 0$, $q(x+y) = q(x) + q(y)$).

The only difference between FFT and evaluation on $V$ is we need to decompose $p$ as follows: $p(x) = x \cdot p_1(q(x)) + p_0(q(x))$. The existence of such a decomposition is not obvious. In addition to it, we need to compute $p_0$ and $p_1$ quickly. Both of these issues can be addressed: one can show that such $p_0$ and $p_1$ exist and can be computed in time $O(n \log n)$. Thus, the overall time is $O(n \log^2 n)$.

Moreover, recently Lin, Chung and Han showed how to evaluate a polynomial on $V$ in time $O(n \log n)$, if one is given coefficients in a carefully chosen basis $P_V^0, \dots, P_V^{2^k - 1}$ that depends on $V$. This turns out to be useful for encoding and decoding certain Reed-Solomon codes.

## 1.3 Schönhage-Strassen

What if we do not have a good root of unity, but still want to multiply polynomials quickly? This was addressed by Schönhage and Strassen, who showed how to multiply two polynomials over a ring $R$ in time $O(n \log n \log \log n)$, provided that 2 is a not a zero divisor. That is, we barely assume anything about our ring!

The algorithm is pretty intricate, so we only sketch a high-level idea of it.

The first naïve approach would be to consider an extension of $R$ that contains a good root of unity and perform FFT there. Unfortunately, if done naïvely, we may end up with an extension, where it is very expensive to perform multiplications.

We will be considering an extension of the form $A = R[y]/(y^l + 1)$. A new variable $y$ ends up being a $2l$-th root of unity. So, the larger $l$ is, the more we would be able to "reduce" our polynomial of degree $n$. On the other hand, elements of $A$ are polynomials of degree $l - 1$, so the smaller $l$ is, the cheaper operations in $A$ end up being.

Turns out the optimal choice of $l$ is around $\sqrt{n}$. We can represent any polynomial $p(x)$ as $p(x) = Q(x, x^l)$, where $\deg_x Q(x, y) < l$ and $\deg_y Q(x, y) \le n/l$. Now we map $p$ into $Q(x, y) \in A[x]$. This map is invertible.

Then, to multiply polynomials in $Q(x, y)$ we will be using powers of $y$ for evaluation and interpolation. It turns out that performing FFT and Inverse FFT is cheap (since we only need multiplications of the special form), and the bottleneck is pointwise multiplication. But this can be done recursively, since points in $A$ are polynomials over $R$ of degree less then $l$!

For the real description and all the details filled see excellent notes of Zachary Abel from the previous offering of the class.