Corollary. *If f is a prime number, then there are $(p^f - p)/f$ distinct monic irreducible polynomials of degree f in $\mathbf{F}_p[X]$.*

Notice that $(p^f - p)/f$ is an integer because of Fermat's Little Theorem for the prime f, which guarantees that $p^f \equiv p \bmod f$. To prove the corollary, let n be the number of monic irreducible polynomials of degree f. According to the proposition, the degree-p^f polynomial $X^{p^f} - X$ is the product of n polynomials of degree f and the p degree-1 irreducible polynomials $X - a$ for $a \in \mathbf{F}_p$. Thus, equating degrees gives: $p^f = nf + p$, from which the desired equality follows.

More generally, suppose that f is not necessarily prime. Then, letting n_d denote the number of monic irreducible polynomials of degree d over $\mathbf{F}_p$, we have $n_f = (p^f - \sum d n_d)/f$, where the summation is over all $d < f$ which divide f.

We now extend the time estimates in Chapter I for arithmetic modulo p to general finite fields.

Proposition II.1.9. *Let $\mathbf{F}_q$, where $q = p^f$, be a finite field, and let $F(X)$ be an irreducible polynomial of degree f over $\mathbf{F}_p$. Then two elements of $\mathbf{F}_q$ can be multiplied or divided in $O(log^3 q)$ bit operations. If k is a positive integer, then an element of $\mathbf{F}_q$ can be raised to the k-th power in $O(log\, k\, log^3 q)$ bit operations.*

Proof. An element of $\mathbf{F}_q$ is a polynomial with coefficients in $\mathbf{F}_p = \mathbf{Z}/p\mathbf{Z}$ regarded modulo $F(X)$. To multiply two such elements, we multiply the polynomials — this requires $O(f^2)$ multiplications of integers modulo p (and some additions of integers modulo p, which take much less time) — and then divide the polynomial $F(X)$ into the product, taking the remainder polynomial as our answer. The polynomial division involves $O(f)$ divisions of integers modulo p and $O(f^2)$ multiplications of integers modulo p. Since a multiplication modulo p takes $O(log^2 p)$ bit operations, and a division (using the Euclidean algorithm, for example) takes $O(log^3 p)$ bit operations (see the corollary to Proposition I.2.2), the total number of bit operations is: $O(f^2 log^2 p + f\, log^3 p) = O((f\, log\, p)^3) = O(log^3 q)$. To prove the same result for division, it suffices to show that the reciprocal of an element can be found in time $O(log^3 q)$. Using the Euclidean algorithm for polynomials over the field $\mathbf{F}_p$ (see Exercise 12 of § I.2), we must write 1 as a linear combination of our given element in $\mathbf{F}_q$ (i.e., a given polynomial of degree $< f$) and the fixed degree-f polynomial $F(X)$. This involves $O(f)$ divisions of polynomials of degree $< f$, and each polynomial division requires $O(f^2 log^2 p + f\, log^3 p) = O(f^2 log^3 p)$ bit operations. Thus, the total time required is $O(f^3 log^3 p) = O(log^3 q)$. Finally, a k-th power can be computed by the repeated squaring method in the same way as modular exponentiation (see the end of § I.3). This takes $O(log\, k)$ multiplications (or squarings) of elements of $\mathbf{F}_q$, and hence $O(log\, k\, log^3 q)$ bit operations. This completes the proof.

We conclude this section with an example of computation with polynomials over finite fields. We illustrate by an example over the very smallest (and perhaps the most important) finite field, the 2-element field