

**Computational Number Theory and Algebra**  
**Prof. Nitin Saxena**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology - Kanpur**

**Lecture – 09**  
**Polynomial Factoring Over Finite Fields – Irreducibility Testing**

(Refer Slide Time: 00:16)

Polynomial Factorization

- Problem: Given  $f(x) \in \mathbb{F}[x]$  of degree  $d$ .  
 Compute  $g(x) \in \mathbb{F}[x]$  of  $\deg \in [d-1]$  s.t.  $g|f$ .  
 $\rightarrow$  in  $\text{poly}(d)$ -many  $\mathbb{F}$ -operations?

Fact:  $\mathbb{F}[x]$  is a unique factorization domain.  
 I.e. each  $f$  factors as  $f = \prod f_i^{e_i}$  uniquely,  
 where  $f_i$  is irreducible & are mutually  
coprime.

Okay, so we started factorization of polynomials, univariate polynomials over a field, right because that polynomial ring is a unique factorization domain.

(Refer Slide Time: 00:28)

- Factorization pattern depends on the field.  
 (So do its algorithms)

- Eg.  $f = x^2 + 2$  is irreducible over  $\mathbb{Q}$ .  
 but is reducible over  $\mathbb{F}_3$ :  
 $f \equiv_3 (x-1)(x+1)$ .  
 $\equiv_2 x^2$ .

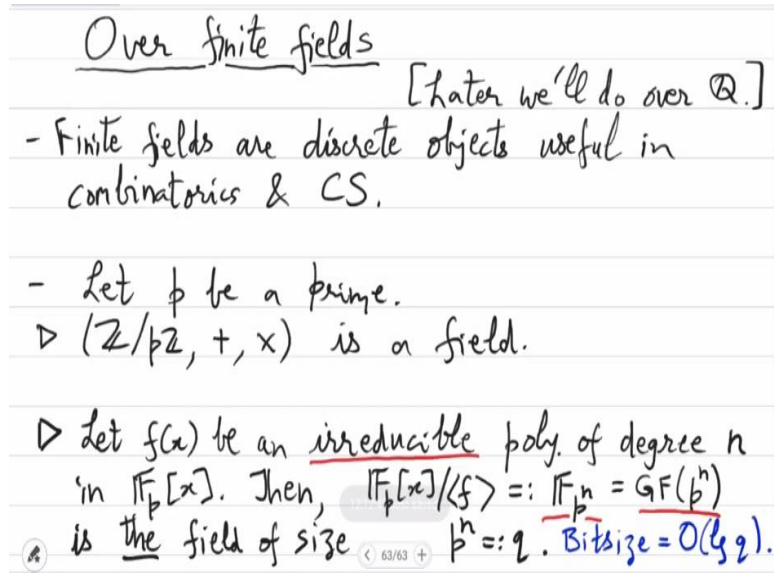
$\triangleright$  [Gauss] Over  $\mathbb{C}$ , every polynomial factors!  
 [so, completely splits]

- Defn: Algebraically closed:

We saw some simple examples manifesting that roots are very closely related to field. So, as you change the field, the roots and the factorization pattern changes. In fact, there may not be

any factors if you change the field. Over complex, it is the other extreme, every polynomial actually factors nontrivially all the way up to degree 1, such fields are called algebraically closed, but now we will actually come down to something very different a place where there is no geometry which is finite field.

(Refer Slide Time: 01:13)



Over finite fields [later we'll do over  $\mathbb{Q}$ .]

- Finite fields are discrete objects useful in combinatorics & CS.
- Let  $p$  be a prime.
  - ▷  $(\mathbb{Z}/p\mathbb{Z}, +, \times)$  is a field.
  - ▷ Let  $f(x)$  be an irreducible poly. of degree  $n$  in  $\mathbb{F}_p[x]$ . Then,  $\mathbb{F}_p[x]/\langle f \rangle =: \mathbb{F}_{p^n} = \text{GF}(p^n)$  is the field of size  $p^n =: q$ . Bitsize =  $O(\log q)$ .

So, this is the example which computer science cares the most about, almost all the practical applications are actually based on finite fields. So, be it combinatorial constructions or error correcting codes and so on. So, we will see some examples of those applications once we are done with some basic polynomial factorization. So, yeah later we will see factorization over  $\mathbb{Q}$ .

So, this order we will follow because believe it or not factorization over  $\mathbb{Q}$  or over integers will actually need factorization over finite fields. Okay so once you know these methods, then we will use these roots or these factors to actually get integral factors, so that has to be done in this order. So, finite fields you must have seen enough properties and prove some of them in this first assignment.

Finite fields are, so these are discrete objects useful in combinatorics and computer science and the construction of this as you have seen is basically based on a prime, right. So, you start with a prime characteristic  $p$ . So, you are also allowed to take  $p = 2$ , so that will be a field with only 2 elements right, 0, 1 and you can add them and you can multiply them and arithmetic is more too.

So in general, you have shown that  $\mathbb{Z}$  modulo  $p$  which we will denote, actually let us not use this, we will denote it by quotienting. So,  $\mathbb{Z} \bmod p$  ideal is a well first of all it is a ring arithmetic and then it is actually a field. So, how do you show that every element is invertible in this except 0 of course. Right, so this is based on just gcd. So, for nonzero element  $a$ , you take the gcd of  $a$  and  $p$  to be 1, so you have  $ua + vf = 1$  which means that  $u$  is inverse of  $a$ .

So, that is a simple proof, More interesting things happen when you want to field of size  $p$  square. So, this is a field of size  $p$ , what is a field of size  $p$  square that you have constructed in the first assignment. So, that is done by picking irreducible polynomials, so in this case of degree 2 be an irreducible polynomial of degree let us say  $n$  with  $\mathbb{F}_p$  coefficients, so, it is in the polynomial ring  $\mathbb{F}_p[x]$ . So, this will be key to pick an irreducible polynomial.

Then, now when you do arithmetic mod this polynomial it will be a field and it will be a bigger field. So,  $\mathbb{F}_p[x] \bmod f$ , this is a field because every element here is invertible again by Bezout identity. Because  $f$  is irreducible, so it behaves the same as you saw in the proof of the above statement  $f$  is kind of a prime. So, any polynomial  $a$  will again be the gcd with  $f$  will be 1 or  $a$  itself is 0 in this ring in this arithmetic.

So, for a non-zero  $a$ ,  $\gcd(a, f) = 1$ , so, you get  $ua + vf = 1$ , so every element has an inverse. So, this is a field and this field is called we will write it as  $\mathbb{F}_{p^n}$ . It is also written as in the literature Galois field, so GF of size  $p$  raised to  $n$  okay, but usually we write  $\mathbb{F}_{p^n}$  raised to  $n$ , so either of these definitions. **“Professor - Student conversation starts.”** But if  $r$  is a ring then will  $r$  be smaller than  $f$  for an irreducible  $f$  possibly a field or a ring.

No, you if you talk about general rings, then behavior will depend on  $r$ . I mean if you start with an  $r$  which has 0 divisors, then the 0 divisors will remain no matter what you do. So, it cannot be a field. So, you have to start with a field to have any hope of getting a field extension. **“Professor – Student conversation ends.”** So, this in fact in the assignment you must have shown that this is unique up to field isomorphism, right.

So, this is actually the field. So, this is the field of size  $p$  raised to  $n$ . There is only one field of this size, okay. So, for every prime power, there is a unique field and vice versa. So, every finite field is of size prime power okay. So, this is a nice characterization that you have shown and so we will usually denote prime powers by  $q$ , and so yeah in this property, I also

want to embed something about the bit size.

So, how many bits will a single element require when you want to represent it practically?  $n \log p$ , yeah, so or if in terms of  $q$  it is  $\log q$ . So, the representation requires  $\log q$  many bits, so that needs to be remembered okay. So, finite fields of size  $q$  one element will not need one bit, it will actually mean  $\log q$  many bits. Also what is the structure of  $F_p$  to the  $n$  over  $F_p$ ? It is a vector space, it is a field extension.

Hence it is also a vector space and how many basis elements are there,  $n$  basis elements are there, right. So that is another way to represent an element in this field, you can be given  $n$  numbers, right. So these  $n$  numbers, each number should be thought of as an element in  $F_p$ , so it is a number between, yeah so hence this number will have magnitude 0 to  $p-1$  and bitesize  $\log p$  and there are  $n$  of those.

So, it is  $n \log p$ , which is also the same as  $\log q$ , like so this actually is a standard representation. You think of an element as  $n$  numbers and when  $n$  is 1, it is only one number which is clear because it is an integer between 0 to  $p-1$ , when  $n = 2$  then you are talking about 2 numbers, right. So, that is how we will, whenever we say a field is given or we are working over a field that is the representation okay.

Without exception this is the underlying representation we will talk about and we will assume and then the time complexity will be in terms of this representation, so it is completely explicit discussion, this is not abstract, okay.

**(Refer Slide Time: 11:26)**

$$\begin{aligned}
 & -19. f = x^2 + x + 1 \in \mathbb{F}_2[x] \text{ is irreducible.} \\
 & \text{So, } GF(4) = \mathbb{F}_2[x] / \langle x^2 + x + 1 \rangle \\
 & \quad = \{0, 1, x, 1+x\}. \\
 & -19. F(x) = x^2 + x \in GF(4)[x] \\
 & \quad = (x + x + 1)^2 \\
 & \therefore \mathbb{F}_q^* := \mathbb{F}_q \setminus \{0\} \text{ is a cyclic group of size } (q-1). \\
 & \Rightarrow \\
 & \triangleright \forall a \in \mathbb{F}_q^*, a^{q-1} \equiv 1 \\
 & \triangleright \forall a \in \mathbb{F}_q, a^q \equiv a. \quad [\text{Fermat's little thm.}] \\
 & \quad [\text{Frobenius action}]
 \end{aligned}$$

So, for example, if you look at  $x^2 + x + 1$  interpreted in  $\mathbb{F}_2[x]$ , then is this polynomial irreducible or reducible? **“Professor – Student conversation starts.”** Irreducible. How do you prove it? It could have a root, but both 0 and 1 only. Exactly, so there are only two roots possible and both of them will give you 1. **“Professor – Student conversation ends.”** So, this is an irreducible polynomial. So, that gives you a simple quadratic extension over  $\mathbb{F}_2$ .

So,  $GF(4)$  is you can think of it as just  $\mathbb{F}_2[x]$  mod, this is the field  $GF(4)$  and it has very few elements, right. So, it has 0 and 1 from  $\mathbb{F}_2$ , what else? It has  $x$  and it has  $1 + x$ , right. So, these are the only elements in this field. Any polynomial in  $x$  will map to one of these four, right. So,  $x^2$  maps to what,  $x^2$  in this ring is equal to what? It is equal to  $x + 1$  and  $x + x^2 + 1$  is then equal to 0 and so on.

So, similarly you can think of  $x^3$ , all the other higher degree monomials reduced to this. So, hence when you multiply elements, when you multiply  $x$  with  $1 + x$ , multiplication actually requires one polynomial multiplication operation and then also division, right. So, all those things are again implicit whenever we will talk about complexity that you have to actually do division here.

There was a question in the assignment where you must have done that as well. So, those things now we will just assume when we talk about complexity because you have seen polynomial multiplication, you have seen integer division and similarly you can also do polynomial division, everything you can do in  $O(\tilde{\text{linear}})$  time. So, these are the kinds of fields you are given in the input.

And over this there is a polynomial given which you want to factor, right that is the setting that would be your input instance. So, maybe I give that example also. So, you are given maybe a polynomial  $f(x)$  which is say  $x^2 + 1$ . So, this is a polynomial  $\text{GF}_4$  with this formal variable  $x$ . So, you may be given this big  $F$  in the input,  $\text{GF}_4$  is in the input and then  $x^2 + 1$  is a polynomial over the finite field, right.

And then somebody asks you to factorize this which means find the root of this, which is also the same as talking about finding a square root of  $x$ , right, well so whether it exists in the finite field or not will be a related question or prove that this is irreducible, right. So, the question of irreducibility testing, root finding, factorizing for big  $F$  is our goal now. How will you do this efficiently? Any questions? So, we will now move towards such algorithms okay, how is that done?

So, first of all how is irreducibility testing done? Given this polynomial big  $F$  how do you quickly decide on a computer that it is irreducible? This is a very simple example, this you can actually see that it is irreducible because square root of  $x$  is nonexistent, is it? **“Professor – Student Conversation starts.”**  $x^2 + 1$  is there. Yeah it is actually there. So, this will factorize as what?  $x + 1$ . Yeah, so let us write down. It actually factorizes.

So, this is actually big  $x^2 + 1$  plus what is the factorization, say that again. Big  $x^2 + 1$  whole square. I somehow think this is false.  $x + 1$ . It splits. It is  $x^2 + 1$  whole square. Okay, yeah that is correct. So, this is how it factorizes. **“Professor – student conversation ends.”** Yes, it is actually a good example. It seems to be irreducible, but then you can see that  $x + 1$  is a repeated root of this, right.

So, now imagine this problem if your big  $F$  was arbitrary degree  $d$ , right. So, in that case actually this will be a real problem, just to check whether it is irreducible. Forget about finding root or finding factors. So, that will be our first school, maybe we will finish it today how do you check irreducibility okay. So, towards that let us now repeat some of the properties that you have seen in the homework okay, all those things will be now used.

So, you have seen that this field  $\text{GF}_q$  – 0, so  $\text{GF}_q^*$ , we call it  $\text{GF}_q^*$ , yeah because it is because it is a field this has to be multiplicative group, but there is more structure to this, it is

a cyclic group. So, this you know is a cyclic group of size  $q - 1$  and this means in particular, actually you just needed Abelian group of size  $q - 1$  also. So, what I want to deduce is that any element you take from this  $a$ ,  $a$  raised to  $q-1$  is 1, right?

Just because it is an Abelian group of size  $q - 1$  any element if you multiply that many times will give you in the end 1 because the multiplicative order of any element first of all it is finite and second you can show it will divide  $q-1$ . So, if you raise  $a$  to  $q - 1$  you will always get 1 that is just by the structure of Abelian groups. And if you also include 0, then you have to look at  $a$  raised to  $q$ , which is  $a$ , right. So, this is satisfied by all  $a$  in the finite field.

So, this is a fundamental identity, its special cases have names. So, this is also called FLT Fermat's little theorem. So, when  $q$  is a prime, then any number if you multiply it  $q$  many times you get the remainder same as  $a$ , mod  $p$ , for in this case mod  $q$ . But when  $q$  is  $p$  square, then this is different. This  $a$  is not really a number, it is an element in a bigger field. But this continues to hold. So, its special cases called Fermat's little theorem.

So, Fermat's little theorem has multiple proofs, this is one of them, this is the field based proof. Any questions? **“Professor – Student conversation starts.”** Sir, we are assuming  $q$  to be prime, then it is,  $q$ , yeah, no  $F_q$  is a finite field. Whenever we will say  $q$  there will be no assumption except there is prime power. Yeah, so for finite field you can continue calling this FLT. This this is also called Frobenius action actually, that is another name for this. **“Professor – Student conversation ends.”**

So, if you raise any element by the field size that is one of the Frobenius actions, in this case it is a trivial action because  $a$  raised to  $q$  will give you the same thing back which is  $a$ , but you could have instead raised  $a$  to  $p$ , so that is actually called the Frobenius automorphism. There are multiple names for this okay.

**(Refer Slide Time: 22:35)**

- These basic properties inspire an irreducibility test:

$$F \in \mathbb{F}_q[X], \quad (F(x), X^q - x) = ?$$

$$(F, X^{q^2} - x) = ? \text{ \& so on.}$$

(Input bitsize  $\approx d \cdot \log q$ )

Theorem:  $F \in \mathbb{F}_q[X]$  is reducible ( $\& \deg F =: d$ ) iff

$$\exists 0 < i < d, \quad \gcd_x(F, X^{q^i} - x) \neq 1.$$

Pf:  $\Rightarrow$ :

Let  $h \mid F$  be an irreducible factor of  $\deg d' \in [d-1]$ .  $\Rightarrow \mathbb{F}_q[X]/\langle h(x) \rangle = GF(q^{d'})$

$$\Rightarrow X^{q^{d'}} \equiv x \pmod{h} \Rightarrow h \mid (F, X^{q^{d'}} - x).$$

So now based on that identity, we will design an irreducibility test for an input polynomial, discuss what property we are interested in. So, suppose you are given a polynomial, we continue to call it big F okay, you are given this with coefficients in  $\mathbb{F}_q$ . So univariate polynomial. So, you want to test whether this polynomial is irreducible. So, what we intend to do is we will, suppose you want to check whether this has a root  $\mathbb{F}_q$  root, right.

So, I want to take gcd of this with something. So, by looking at the previous equation what should you take gcd of this with? So, if big F has a root, then it will satisfy a to the q - a, right. So, you should actually then take gcd with x to the q - x, right. So, this F is, okay let me make this as before big X. So, F is a polynomial in big X and you compute the gcd of these two polynomials.

So, now if the gcd is 1, then it means that big F has no root right because if it had a root a, then x - a will divide this and x - a you know also divides the second polynomial. So, x - a will definitely divide the gcd. So, this being 1 actually means that big F has no root. Now, more interestingly, this test can be extended to cover quadratic factors. So, if big F has no root, it may still have a quadratic irreducible factor, right.

So for that, the test you will do is X to the q square - X and then so on. Okay, this is the intended algorithm that you keep taking gcd of big F with these Frobenius type polynomials, Frobenius inspired polynomials. So, if you take gcd with x to the q - x, you are filtering out roots. If you take gcd with x to the q square - x, then you are filtering out quadratic, well both roots and quadratic irreducible.



So, this sequence of gcd is we want to compute, so, that is the intention, now why should such a thing work is this theorem. So, big  $F$  factorizes say degrees  $d$  if and only if there is some  $i$  such that the gcd of big  $F$  with  $X$  to the  $q$  to the  $i - x$  and gcd with respect to the variable big  $X$  both of them are unique variants over the underlying finite field, so you compute the gcd and this should not be 1.

So, if big  $F$  has a nontrivial factor, then the gcd will not be 1, and if the gcd is not 1, then there will be a irreducible factor, in fact just the gcd operation will give you a factor of big  $F$ , okay, it is a constructive proof. So, we will prove this theorem it has an elementary proof, but once you have the proof, how can you use this in an algorithm? How will you compute this gcd, in how much time?

**“Professor – Student conversation starts.”** We will have to  $\log q$  to the  $i$ . Sorry.  $i \log q$  for  $m$ ,  $m$  of  $i \log q$  for every,  $i \log q$ , but the degree of this is  $q$  to the  $i$ . That is if the degree of one. So, well take  $i = 1$ . So if you are taking gcd with  $X$  to the  $q - X$  that is a  $q$  degree polynomial, right. **“Professor – Student conversation ends.”** So, if you just use Euclid gcd directly, then this will give you  $q$  in the time complexity, but since your input size was  $\log q$  or  $d + \log q$  or  $d$  times  $\log q$ , so let me write that down.

The input size in this case is  $d$  times  $\log q$ . So, you want your time complexity to be polynomial in  $d \log q$ . If you spend time  $q$ , then it is already exponential right. So, how do you compute this gcd faster than  $q$  time? Well, you should reduce  $x$  to the  $q \bmod f$ , compute the remainder. So, there was a question in the assignment that this exponentiation can be done by  $\log q$  repeated squaring.

So, you compute big  $X$  square, then big  $X^4$ , then big  $X^8$ , 16 and so on. So, this will only take  $\log q$  iterations. So, very quickly you can actually compute the remainder of  $X$  to the  $q \bmod f$  and then you work with the remainder instead of  $X$  to the  $q$  because it does not change the gcd, like gcd is invariant if you divide one argument with the other, so that is the thing. So, this Frobenius polynomial happens to be so nice that this gives you an immediate algorithm okay.

So, that is a very lucky break, it may not have been this easy otherwise to compute. **“Professor – Student conversation starts.”** If we directly take  $i = d-1$  in the first step. Yeah. It is sufficient. No. Let us first look at the proof of this, why is this thing true, then you should answer this question. Sir if you get a quadratic irreducible of  $F$ , then in the extension  $q$  square we will be getting a root of that. Yeah, that is the idea. Yeah.

So, if big  $F$  suppose it is itself quadratic irreducible, then its root is available in  $F_q$  square in Galois field of size  $q$  square, so hence the GCD will come out, I mean  $f$  big  $F$  will actually divide the whole thing in that case, gcd cannot be one. **“Professor – Student conversation ends.”** So, yeah, that is the basic idea, so we will just build on that to finish this proof. So, let us do the forward direction.

So, we assume big  $F$  is irreducible. So, let  $h$  dividing  $f$  be an irreducible factor of degree  $d$  prime between 1 and  $d-1$ , it is a nontrivial irreducible factor of  $f$ . So, now, you look at  $F_q X \bmod h$ , so what is this? This is the Galois field right of size exactly. So, this is also a finite field it is bigger than  $F_q$ , possibly bigger than  $F_q$  and has size  $q$  to the  $d$  prime. So, what you can see now is now you use the FLT or Frobenius action identity.

So that will give you  $X$  to the  $q^d$  prime is the same as  $X \bmod h$ , right,  $h$  will divide  $X$  to the  $q^d$  prime – minus  $x$  in other words. So,  $h$  divides this polynomial and  $h$  also divides original  $f$ , sorry I should use capital  $F$  there, I have changed the notation. So,  $h$  divides your input polynomial by assumption and  $h$  also divides this by Frobenius action.

So,  $h$  divides the gcd, right. So, hence gcd cannot come out to be 1 because  $h$  is not 1,  $h$  is degree at least 1, so that part is done. Any questions? Okay.

**(Refer Slide Time: 34:22)**

$$\begin{aligned}
& \Leftrightarrow: \text{Say, } F \text{ is irreducible \& let } i \in [d-1] \\
& \text{be the least s.t. } (F, x^{q^i} - x) \neq 1. \\
& \Rightarrow F \mid x^{q^i} - x \\
& \Rightarrow x^{q^i} \equiv x \pmod{\langle F \rangle} \\
& \Rightarrow \forall a \in \mathbb{F}_q[X]/\langle F(X) \rangle \stackrel{=: \mathbb{F}_{q^i}}{=} a(X)^{q^i} \equiv a(X) \\
& [\because (y+z)^q \equiv y^q + z^q \pmod{p} \text{ by binomials.}] \\
& [x \mapsto x^q \text{ by } p \text{ is an } \mathbb{F}_p\text{-automorphism here.}] \\
& \cdot (\mathbb{F}_{q^i})^* \text{ is a cyclic group of size } q^i - 1. \\
& \Rightarrow q^d - 1 \mid q^i - 1 \Rightarrow d \leq i \Rightarrow \text{ } \\
& \Rightarrow \text{Converse holds!} \quad \square
\end{aligned}$$

Let us prove the converse now. The converse is more interesting. So, suppose for some  $i$  you get the gcd, well we will actually do it the contrapositive way. So, you want to show that the conclusion implies the premise, right? So, instead we will show that if big  $F$  is irreducible, then the gcd is 1 okay, let us prove the contrapositive. So, say big  $F$  is irreducible and  $i$  be the least number such that the gcd of these two polynomials is not 1.

So, we actually intend to add a contradiction okay. We are assuming that big  $F$  is irreducible and we are assuming that the gcd is not 1 for some  $i$ . So, if you get a contradiction, then we are done, right. So, let us see what the contradiction is. Well, so if big  $F$  is irreducible and the gcd exists what does that mean? It has to exactly divide, gcd has to be big  $F$  that is the only option, right, and so write this as Frobenius action.

So,  $X$  to the  $q$  to the  $i$  is actually  $X$ , mod big  $F$  which is irreducible. So, from this what can you say about elements in this quotient ring? Any element  $a$  that you take in this quotient ring  $\mathbb{F}_q[X] \text{ mod } F(X)$ , so first of all this ring is what, why is it a field? Yeah because we have assumed the big  $F$  to be irreducible, right. So, this is a field. **“Professor – Student conversation starts.”** What can you say about the elements in this which is again a finite field, elements  $a$  in this finite field?

There are roots of  $h$  that will not give the  $(( ))(37:31)$  Every  $a$  will satisfy the previous Frobenius action, why is that? That is not trivial, how do you show this? How do you show that  $a$  raised to  $q$  raised to  $i$  is also  $a$  by using the previous congruence? The  $a$  is a polynomial,

so  $a$  raised to  $q$  raised to power  $i$  will distribute. Yeah, so you use the fact that  $q$  raised to  $i$  distributes over sum. Binomial identity. Yeah use the binomial identity.

So, since  $y + z$  raised to  $q$  is the same as  $y$  to the  $q + z$  to the  $q$  and this is actually two simply mod  $p$ . So, your characteristic is  $p$  and  $q$  is a power of  $p$  okay. So, no matter how many powers you put over  $y + z$ , they will distribute, the exponentiation operator will distribute over sum. So, the reason is that when you do binomial expansion, the other coefficients are  $q$  choose  $i$  and  $q$  choose  $i$  is zero mod  $p$ , it is divisible by  $p$  okay.

This is true by binomials. So that is a nice property to remember. This is again a consequence of Frobenius action. So, this in particular shows that raising exponentiating by  $p$  preserves addition and multiplication So, it is actually first of all a homomorphism. You can see there is also an automorphism okay, maybe I should write that, that should also be remembered.

So here meaning in the case of finite fields whenever you exponentiate by  $p$ , you actually get a homomorphism which happens to be an automorphism and endomorphism which is actually an automorphism. Okay, this is a very nontrivial facts and also very useful one. Yeah so coming back to the last implication what you have learnt is that every field element in  $F_q$  satisfies this  $q$  to the  $i$  Frobenius action which is a trivial action.

**“Professor – Student conversation starts.”** So, which means what? Utmost  $q$  power  $i$  limits because any polynomial can have utmost degree by roots in  $f$  field. Yeah, but you can make even more precise statement. So you can actually say that  $q$  raised to  $d-1$  will divide  $q$  raised to  $i-1$ . You can actually get an equality in terms of exponents. **“Professor – Student conversation ends.”** So that argument goes as follows. So let me give it a name, let me call it  $F_{q^d}$ .

So this field except 0 is actually cyclic group of size, field size minus 1, right. What is the field size,  $q^d$  which is  $q$  to the  $d$ . So this is cyclic group of size  $q^d - 1$ . Every element here satisfies this  $q$  to the  $i$  trivial Frobenius action, so it means that generator as well, the generator has order  $q^d - 1$ , so which means that  $q^d - 1$  has to divide  $q^i - 1$ , right. Which means what?

Yeah in particular it means that  $d$  is less than or equal to  $i$ , but that is not possible, the  $i$  with which we started with was between 1 and  $d-1$ , so that is the contradiction okay. So this contradiction means that when you start with the irreducible input, then you get all gcd's 1. So, this means that converse holds. Any questions? So, this is the fundamental property and its proof as you can see uses almost all the basic properties of finite fields, right.

This is why you were prepared by assignment 1. So building on that you can actually show that an input polynomial is irreducible if and only if for some  $i$  this gcd is not 1 or equivalently an input polynomial is irreducible if all the gcd's happen to be 1. **“Professor – Student conversation starts.”** I think we can also write  $i$  is less than root  $d+1$ . The  $i$  is less than root  $d$ . Yeah, greater than root  $d$  capital  $D$ .

I do not want to say that, yeah  $d$  by 2 we can say. So if the input polynomial factors, then clearly there will be a factor of  $d$  by 2 degree or less. So, that optimization you can do, you need to only go up to  $d$  by 2. So let me just say it here. So  $d$  by 2 is a bad notation. I mean yeah, let me use the range, so  $1 \leq i \leq d/2$  works. **“Professor – Student conversation ends.”** Any questions? Okay.

Yeah so now the, the algorithms all the constituents are there, these checks are already so, they are in a good form that you can just implement them in a straight forward way using the basic arithmetic. So using the arithmetic of integers and using the arithmetic of polynomials, on top of that you can implement everything and it will be really fast.

(Refer Slide Time: 46:09)

Algorithm: (Input -  $\deg=d$  poly.  $F \in \mathbb{F}_q[x]$ )

Step 1: For  $1 \leq i \leq d/2$ :

If  $(F, x^{q^i} - x) \neq 1$  then OUTPUT Reducible.

*reduce mod F by repeated-squaring*

Step 2: OUTPUT Irreducible.

Time Complexity:  $d \times [d \log q \times \tilde{O}(d) + \tilde{O}(d)] \mathbb{F}_q\text{-ops}$

$\leq \tilde{O}(d^3 \log q) \mathbb{F}_q\text{-ops}$

$\leq \tilde{O}(d^3 \log^2 q) \text{ bit-ops}$

So, let us just for formality see the algorithm and also we want to analyze the exact time complexity. So in the input you are given a degree  $d$  polynomial  $f$  and a finite field  $F_q$ . The finite field has to be properly given in the natural representation, otherwise you cannot give the coefficient of  $f$ . So, the finite field is given and the polynomial is given. So this is just a for loop.

**“Professor – Student conversation starts.”** But before that there was a question you had right Pranav, what was the question? Just for  $i = d$  minus. Yeah maybe we address that first because that is good property in the proof. So this proof actually tells you more. So, in particular if there is an irreducible factor  $h$  of degree  $d$  prime, what are the  $i$ 's which will work? I mean the  $i$ 's where you will see not 1. Yeah that is not clear.

So actually, the correct characterization is that  $i = d$  prime will give you a certificate, so  $\gcd$  not 1, but then after  $d$  prime the next step or the next  $i$  will be  $2d$  prime okay. So actually you will get multiples of  $d$  prime. So that is a good characterization because if you look at the converse, it means that what are the factors  $x$  to the  $q$  to the  $i$  minus  $X$ ? What is the degree of the irreducible factors of  $x$  to the  $q$  to the  $i$  minus  $X$ .

Yeah so they are actually irreducible polynomials of degree dividing  $i$ , okay, so it is not that  $x$  to the  $q$  to the  $i$  minus  $X$  will be divisible by all the irreducible up to degree  $i$ , it's actually only factors of  $i$ . So, did you understand that,  $h$  here of degree  $d$  prime will divide for  $i = d$  prime, then  $i = 2d$  prime, did you understand that property? So why would not it divide for things in the middle between  $d$  prime and  $2d$  prime, what is the reason?

So that reason actually is in this slide, so you have to go through this. See in particular if you look at the last condition  $q$  raised to  $d-1$  divides  $q$  raised to  $i-1$ , what can you deduce from there? Right, you actually deduce something stronger than what I have written. You actually deduce from here that  $d$  divides  $i$  okay. So this slide is proving another interesting property over finite fields that this  $X$  to the  $q$  to the  $i$  minus  $X$ .

All its irreducible factors have a degree that divides  $i$  and they are all of them, nothing is missed okay. So, this  $X$  to the  $q$  to the  $i$  minus  $X$  actually contains all the irreducible polynomial whose degree divides  $i$ . This I mean as a sanity check you can look at  $i=1$ , so  $X$  to the  $q$  minus  $X$

contains all the irreducible polynomials of degree 1, which is just roots. When you go to the  $X$  to the  $q$  square –  $X$ , you will get all the roots and all the quadratic irreducible and so on.

So cubic would not get ordered. Exactly. **“Professor – Student conversation ends.”** So in the case of  $X$  to the  $q$  cube –  $X$ , the quadratic factors will be missing. So there will be all linear factors possible and all the cubic irreducible possible okay. So actually this slide is quite important, the parts of the proof here tell you a lot more structure over finite fields okay. So, this is just a for loop.

So in case the gcd is not 1, then you have a certificate of reducibility and this is written very abstractly, so in practice you actually have to do this by repeated squaring. So this step is loaded, it is not just direct application of Euclid gcd, but first you have to do this assignment question you should do repeated squaring to compute what is  $X$  to the  $q$  to the  $i \bmod F$ .  $F$  is low degree, so this computation is actually low-degree computation.

So you can do this efficiently and then you replace this  $X$  to the  $q$  to the  $i$  by that remainder. Then you invoke Euclid gcd okay. So this can be done. **“Professor – Student conversation starts.”** If it would be even though other property gcd property because gcd of  $a, b$  is gcd of, yeah that you can do, yes. **“Professor – Student conversation ends.”** Okay and what do you do in the next step? Well not much.

So if the program has not halted yet, so then it went from  $i = 1$  to  $i = d$  by 2 and always the gcd was 1. So it means that big  $F$  is irreducible. Okay so only when the whole four loop has gone through you will declare irreducibility that is the certificate of irreducibility. Is this clear? So the proof of correctness of this algorithm should be clear by the theorem that is straightforward. It is only the time complexity that we have to, so what is that?

The for loop is  $d$  many times the gcd computation before that you have to compute  $X$  to the  $q$  to the  $i \bmod F$  right? How much is that? That is again  $i \log q$ , so  $d \log q$  many squaring and for every squaring, so that is the single multiplication operation right, yeah so modulo this polynomial of degree  $d$  and also the finite field calculation. So actually we can separate the finite field calculations, we can just say  $O \tilde{d}$  many finite field operations, right.

So now you have brought down  $X$  to the  $q$  to the  $i$  to a degree  $d-1$  polynomial at most and you will run Euclid gcd. So that will be how much? That will be linear time soft- $O$  linear that will be  $O$  tilde  $d$ . So these many  $F_q$  operations okay. Any questions? So, this is the time complexity which will ultimately we can talk about actually bit operations as well. So this is at most  $dq \log q$  which is at most  $F_q$  operations will take how much time?

Just  $O$  tilde  $\log q$ . So this is the actual time okay. In the input you are given a binary string and in the output you will give a bit answer 0 or 1, irreducible or reducible and in the middle the number of bit operations which you can think of as a second is  $dq$  times  $\log$  square  $q$  overall with hidden  $\log$  factors also. So this is  $O$  tilde, so there is also some multiplication by  $\log d$  and  $\log \log q$ , but that we can assume as very small, it is a cubic time irreducibility test.

Any questions? **“Professor – Student conversation starts.”** It should be  $F_p$  right,  $F_p$  operations because whenever we are doing  $(())$ (58:32) normally. Yeah we are not going all the way to  $F_p$ , so let us just stop at  $F_q$ .  $F_q$  is the base field for us. We do not want to go all the way, we also do not need because  $F_q$  how you will implement in  $O$  tilde  $\log q$  bit operations, in that detail it is hidden that you convert  $F_q$  into  $n$   $F_p$  elements and then you do addition multiplication all that we do not have to go into.

So that is why we abstract these things out, simplifies the calculation. Otherwise I mean, when you look at the full implementation there is a lot of things that go on. It will not be just these many lines of code, it will be hundreds of lines if not thousands. Is there any field operations are preferred instead of this? Where? Instead of bit operations is preferred, who prefers it? No, ultimately the computer will do only bit operations. So we are doing it only for ourselves to simplify the analysis that is all.

It is just a structured way to analyze, ultimately what will happen on the computer is just bit operations. Somewhere it is written if this algorithm complex is in bit operations, it is difficult to implement over field like the  $(())$ (01:00:05) I do not understand, let us discuss after the class okay. **“Professor – Student conversation ends.”** Another thing in this algorithm is it is not just giving you a one bit answer, it is actually giving you more. So, when it says that reducible, it has worked hard, it has computed the gcd and there is actually some information you get.



So for example if big F was irreducible, in that case you will only get one bit of information, but if big F is reducible then with factors having different degrees, let us suppose big F has a linear factor and it has a quadratic irreducible factor. So those two factors in this gcd process will get separated, did you see that? Because we are running i upwards from 1 to 2 to 3. So for  $i = 1$  you would have filtered out the linear factors, all of them together.

And for  $i = 2$  you would have filtered out quadratic factors all of them together. So, yeah that will need a slight modification actually. So when you compute the gcd with  $i = 1$  and you get something, then you remove that from big F and continue the for loop okay. Then you will get maybe quadratic factors, you remove that from big F and continue the for loop. So, this way you would have actually gotten clusters of factors of big F okay.

So slight modification of step 1 you have to do, it is not exactly this, no otherwise we are breaking, the program halts. As soon as it gets something it halts, so you are not changing big F. So to achieve what I am claiming, you actually have to continue, whatever you get you remove it from big F and proceed. So, you will get the linear factors or whatever, the least i degree factors, so  $i_1$ , then degree  $i_2$ , then degree  $i_3$  and so on.

Yeah. so let me just write down that observation in the same time similar algorithm similar implementation.

(Refer Slide Time: 01:02:47)

Corollary: We factor  $F(x)$  as  $\prod g_i$  where each  $g_i \in \mathbb{F}_q[x]$  is a product of equi-degree irreducible polynomials. In time  $\tilde{O}(d^3 \log^2 q)$ .

Pf: Keep updating  $F$  as  $F / \gcd(F, x^{2^i} - x)$  & continue with  $i, \dots, (d-1)$ .  $\square$

So we factored big F as product of  $g_i$ 's where each  $g_i$  is obviously a polynomial over  $\mathbb{F}_q$ , it is a product of equi-degree irreducibles. So this code is also called equi-degree factorization of

big  $F$ . So, again the big  $F$  will be actually factored into clusters, each cluster will have equidegree irreducibles as their factors. Okay this you do by whenever you find a non-trivial gcd, you remove that from big  $F$  by dividing.

So that is a single division and proceed with the quotient in the for loop. Is that clear? So, this also takes same time. So, this in the same cubic time you can get, equi-degree factorization. Keep updating big  $F$  as big  $F$  divided by the gcd that you have computed. So whatever gcd you will compute, you remove it from big  $F$  and yeah so there is this multiplicity issue, so because of that you continue with the same  $i$ .

In this case, you should go up to  $d$  okay. So at an  $i$  when you get an gcd for the first time, you remove this part from big  $F$  and then repeat the same thing for  $i$ , may be you will again get a cluster, this may happen because of repeated factors. The same factor each may be repeating which means each square may divide  $F$ . So the first instance of gcd will give you  $h$ , the second instance will again give you  $h$ .

But after a while degree  $i$  factors would be eliminated completely, then you will go to the next degree which is probably  $i+1$  and you do this up to  $d-1$  okay. So if you carefully analyze it, it is the same time complexity, there is no change. **“Professor – Student conversation starts.”** If the same quantity, we can also get the full factorization, the same and check we get the cluster, again do it and get the same we get the whole factorization the same quantity.

What do you mean by whole factorization, you can find an irreducible factor? Means all the irreducible. I have finding clusters of. Clusters, yeah, you can find the clusters, but those clusters are reducible. But if it is rather same I want the clusters again, then we get the full factorization. No so. The same cluster again. What do you mean by full, I do not. Prime factors.

You have to, irreducible, no, you cannot, this is not giving you irreducible factorization, we are not there yet. So that will take multiple classes. **Professor -Student conversation ends.”** So in the first lecture all you have achieved is what is called equi-degree irreducible factorization or cluster factorization. We have gotten clusters where the clusters may further factorize, but if they do, all the factors are of equal degree.

But those equal degree factors how to get to them, each of them that we do not know. So, at this point, this algorithm is oblivious to that because  $X^q - X$  for example collects all possible roots. There is no way to distinguish between root 1 and root 2. So how will you distinguish them that we will see gradually. So next time we will achieve something and after that we will achieve the full factorization. Okay.