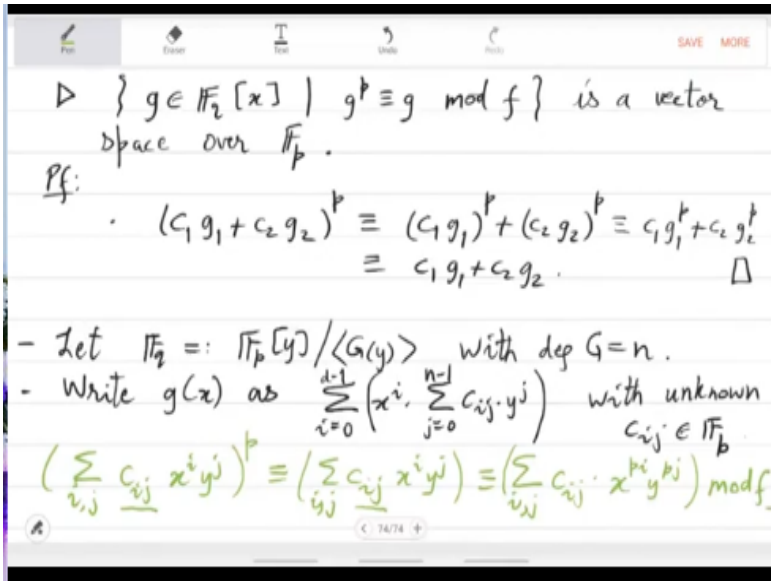
**Computational Number Theory and Algebra**
**Prof. Nitin Saxena**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Kanpur**

**Lecture No -11**

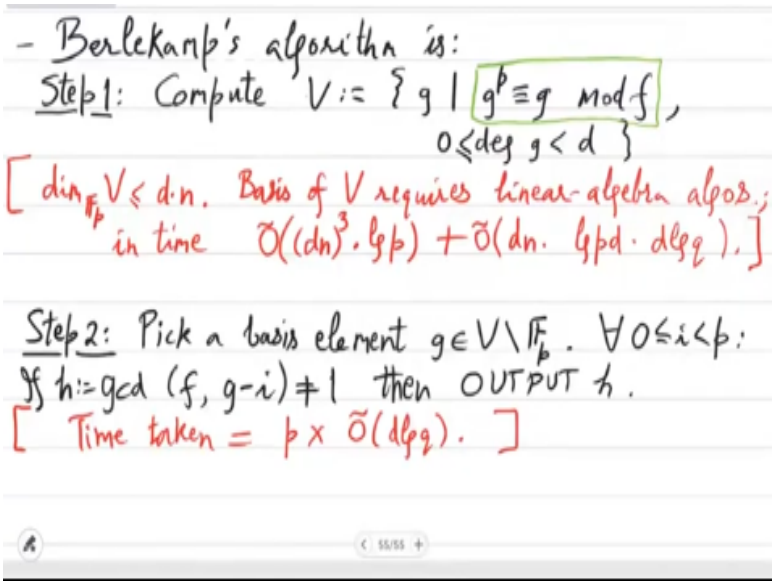
**(Refer Slide Time: 00:16)**




So last time we were doing Berlekamp and so we deduce that we want to solve, we want to find the g that such that g is to p is equal to g mode F. F is the polynomial that is given to you to be factored over a finite field whose characteristic is p. So we observe that the solutions of j is to p equal to g. They form a F p vector space and the vector space now you want to find. Every element in the vector space we want which will be through some basis.

Otherwise the vector space is exponentially large, but basis elements are few. So what we will do is this green part so C i j are the unknown coefficients, x is for the polynomial f so it goes from 0 to d – 1 and y is for the finite field, so it goes from 0 to n – 1, so there are dn unknowns. So LHS is x i y j and RSH is x p i y p j and you want to now compare the monomials both sides, you will get linear constraints C i j and then you find C i j by using a linear system solver. So that part is straight forward.

**(Refer Slide Time: 01:47)**




Based on this let us now write down the formal algorithm. So Berlekamp's algorithm is mainly 2 steps. So step 1 is compute the vector space through basis elements. So g is to p is g and degree of g should be limited, well for vector space you also want degree 0. So 0 to d - 1 you find all the g's. So before going to the next step let us first analyze the time complexity of this. So what is the dimension of V over F p.

That is at most d times n, because here dn unknowns, so you will basically this system will give you a dn cross dn matrix of constants which are just numbers 0 to p - 1 and essentially that matrix you have to then in a way inward to find solutions of the linear system. So basis of V requires linear algebra algorithms, so what is the time complexity of that? So if you nively do it then time complexity will just be the same as matrix multiplication, which is dn whole cube, F p operations.

So that is dn cube times log p, so this much time dn cube F p operation, so you might so that is that each operation is linear in log p and then voted tilted for the log factors in the fast algorithms. But even before you apply linear systems solver you had to construct this matrix. Right so the construction of matrix also costs a bed because, you have to compute this x to the p i, y to the pj and then you have to divide mod f compute the remainder so all that also cost something but it will not be as dominating as this so this is the dominant term.

But just for satisfaction we can also write down that expression, so that is so what is that, see you at around pd and pn. So that you can do by let us say repeated squaring, so log pd log pn and so that many repeated squaring then you have to multiply them which will be well, maybe this is over optimizing. So you first of all, the number of monomials already is dn in your polynomial, so you cannot save on that.

And that much space you need plus what you need is the computing the monomials x to the p i, y to the p j. So all that can be done in around, I would say d log pd and log q. So dn is for the number of i, j and once you fix i j then you have to compute x to the p i, y to the p j. So that is by repeated squaring you can do this in log pd and then you have to also divide by f. So, maybe I collect them together like this.

D log q is finally the mod f computation in the finite field. It is log pd times d log q, because you have to compute mod f over in the finite field. So the d log q will hang around. Is that clear? But whatever it is, it is always smaller than the dominant term so this will not matter. This is really maximum quadratic while there you have already paid cubic time or super quadratic time. So, that is the cost of computing V. All the way down to bit operations number of bit operations.

Yes, you do it by repeated squaring? Repeated, the number of squaring will only be logged pd. So there is an important point here the point is that linear algebra is actually it is truly polynomial time in the input size, because as you can see, there is no factor of p and there is no factor of q, So even for very large characteristic p this is truly polynomial time. Linear algebra is not the bottleneck.

So this you can do quite fast. Irrespective of what the characteristic is, the problem will happen in the second step. So in the second step, now, you have a g, so you pick a g in V. In fact, you can pick it to be a basis element pick up basis element g in V which is non constant and what do you do with this? Is non constant polynomial g, how will this factor f? So you basically compute many g c d's with f.

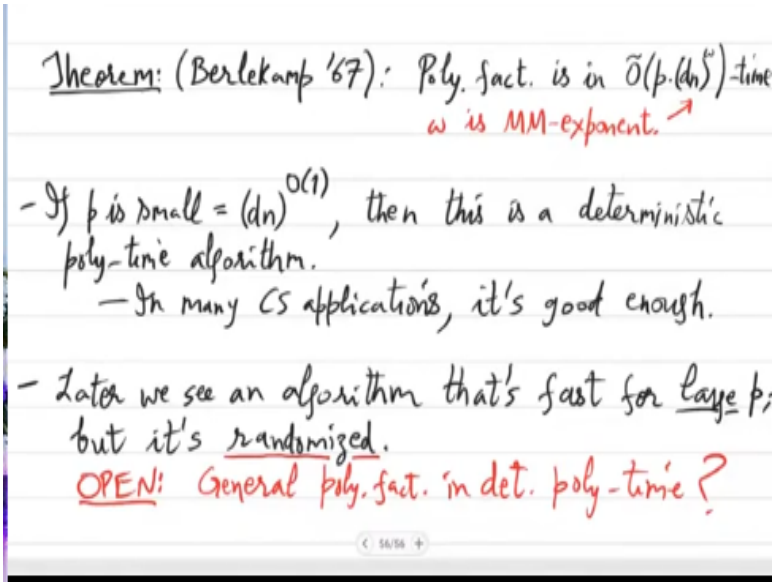
You compute this g c d, f with g – i and quality h. So this h is a proper factor is not 1 basically. Then it is a proper factor of f, because it has degree, less than d. If it is non constant then it has degree at least 1, so this will be a proper factor of f. So you can output it, so you will be done. This is a 4 loop so you are doing this for all i's numbers 0 to p - 1. How do you know that one of them will succeed?

So, the key thing is that equation g raise to p is g mod f, so remember that g satisfies that equation. So since it has satisfies this box f divides g raise to p - g so f divides 1 of the, I mean some part of f has to divide some g - i. Because g is to p - g factors as product of g – i. So something in f divides some g - i when you take g c d then that part of f comes out as each, f is irreducible, but that you have checks separately that already in the pre-processing you checked that f is reducible.

So the only option is that you will get an output each here each here in may or may not be. Now our problem of factorization is only to find a non trivial factor. That is how we define the problem everything else, you can just repeat this subroutine. So the time taken here is so there is a for loop p times and each time how much is the g c d cost? G c d is soft linear, so linear means d log q. So this is the place where p appears.

So the sum of step 1 and step 2 is the overall time complexity and it will make sense only when p is small if p is exponentially large. Then this will be a useless algorithm. But when the characteristic is small it is a good algorithm. In fact right now, this is the only algorithm, you know to factor.

**(Refer Slide Time: 13:37)**




So what we have proved is this theorem, any questions? What the algorithm? Just two simple steps now. So this is Berlekamp's theorem from 67, it is an old theorem. Polynomial factoring is in O tilde, p times dn omega time. So omega is the matrix multiplication exponent, so p times dn omega is this correct? So in step 1 you can replace the standard linear algebra with methods using fast matrix by multiplication.

So dn cube will become dn omega. So that sum is bounded by p times dn omega, that is correct because omega is at least 2. So again second term will not matter and the third term so step 2's complexity is also below p times dn omega. So, that is the final result, so assuming that f p is small it depends on what is the field extension degree of f q, which is n and what is the polynomial degree which is d.

So in that it is dn raise to some 2.4 is super quadratic, but sub cubic. So if p is small which means dn to some polynomial, then this is polynomial time algorithm. Then this is a Deterministic poly time algorithm to factor polynomials over finite fields for low characteristic, where low means this poly dn. So in many CS applications, this is already good enough, because in many CS applications, the characteristic is not that large.

One works over a large finite field with low characteristics. So in the next lecture we will see an algorithm which will work also for large characteristics. So in particular if you have a large

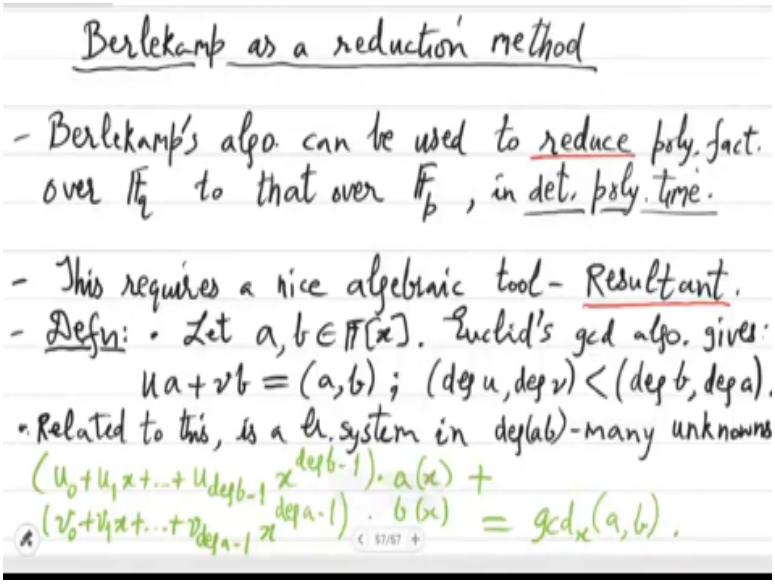
prime field, then this algorithm will not help you. So Q is equal to P and P is large, so in that case, you have no algorithm really no practical algorithm. I mean Berlekamp is not practical. So a different idea is needed which will see in the next class.

But it will not be deterministic, so for large p we will see a different algorithm. It will be polynomial in log p instead of p, but then it will be randomized. It will be randomized polynomial time algorithm for general polynomial factoring. So, do you know what is the randomized algorithm? Anybody here would does not understand the difference between deterministic and randomized?

So randomized is enough or it is okay for practical implementations, but theoretically you want to deterministic algorithm as well. That currently we do not know. So, what is open is general polynomial factoring in deterministic polynomial time. So this is still an open question, although in practice how this is solved we can we can see already in the next, either next class. But doing it really in deterministic polynomial time is open. Any questions about Berlekamp?

So then we will build a bit on Berlecamp, because it gives some interesting tools which will be helpful also in other places.

**(Refer Slide Time: 20:26)**



So, let me call it Berlekamp as a reduction. So what it means will be clear later? But we will see Berlekamp as giving two major tools. So first tool will be a polynomial invariant, which is called resultant and second major tool will be the Berlekamp algorithm gives us a way to reduce polynomial factoring from finite fields to prime fields with very little cost. So in deterministic polynomial time, it actually reduces given input polynomial to one which is in the prime field, which means that the coefficients are only 0 to p - 1 for a prime p.

And if you can find a factor of this polynomial then it will give you a factor for the original input polynomial, this is the meaning of reduction. So Berlekamp algorithm actually gives you these two major tools. So let us discuss these two before we move to the next algorithm. So Berlekamp's algorithm can be used to reduce polynomial factoring over a general finite field F q to that over F p in deterministic polynomial time.

So, why is this nontrivial? Reducing F q to F p is nontrivial because your input polynomial may not have anything to do with F p. It may have factors which have coefficients from F q but not F p. So you are input polynomial may start with coefficients in F q and its factors also may have coefficients only in F q. So it is not clear what is even the connection with factorization over a prime field.

So what Berlekamp does is that it will change your input polynomial to a completely different polynomial possibly, which will have F p coefficients and it will have F p factors as well if the input polynomial had F q factors. So it connects potentially very different problems, because factorization you should remember always depends on the field. So F q and F p should be seen as totally different fields, these are two different factorization questions.

p is always the characteristic p is always a prime, does not matter, polynomials over F q by still be reducible when seen in F p and may not even be defined there. So the factorization questions are really different questions, but Berlekamp will connect them. So it will connect by a polynomial invariant, which is called resultant. So what is a resultant? Is there anybody here who has seen resultant or resolvent term before, what does it mean? Good so you will learn something new.

The term may not suggest anything what is going to come? So suppose you are given two polynomials over some field F, so Euclid's g c d algorithm that analysis give you this Bezout identity, so u a + v b is equal to the g c d of them. If this was the identity and there were additional properties restrictions that degree of u and v are also limited. So both of them are respectively smaller than degree of b and a.

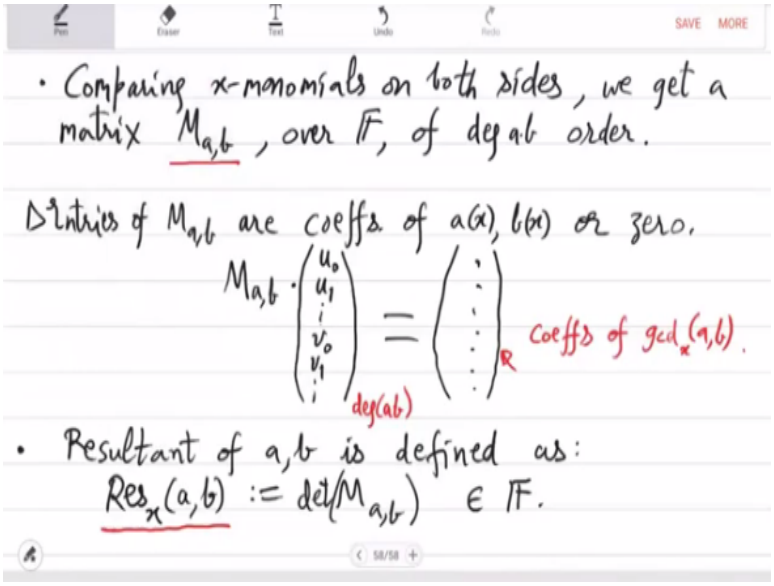
So this was the result of Euclid so g c d algorithm. So you know that u and v exist with this limited degree without going through Euclid g c d algorithm, if you want to find them given a and b so you can see this equation as a linear system in u and v. Right because if you assume the coefficients of u to be u i's and v to be v j's is then in u i and v j you have a linear system by comparing the coefficients would monomials both sides.

Assuming we have computed a, b g c d. Yes, so that linear system basically corresponds to a matrix and that matrix has a determinant. So the question is, what is that determinant? So that determinant we will call resultant, related to this is a linear system of or in how many unknowns? So degree of a + degree of b which is degree of the product. So in degree of a, b many unknowns.

So let us write that equation. So you have. u 0 + u 1 x to u degree b – 1, so u times a is this + you have same thing for v up to degree of a – 1 times b x and the sum is equal to the g c d with respect to x variable. So, this is the equation that or the constraint that u and v have to satisfy and as you can see since a is known b is known and g c d is known the u i's and the v i's they satisfy merely a linear constraint.

I mean, they satisfy a sequence of linear constraints, so the way you will extract those constraints is by comparing coefficient of x to the 0 and both sides. Then x on both sides x square on both sides and so on. So you will get around the degree of a b many linear constraints in that many variables. So it is a square matrix, so the question is, what is the determinant of this square matrix?

**(Refer Slide Time: 30:37)**




So comparing x monomials on both sides, we get a matrix. Let us call it m sub e comma v. It has entries where entries are all constants, of course. So F entries of degree e times b order. The matrix size is degree may be cross degree a b and what is the relationship between the entries of the matrix and the coefficients of a and b? Sorry degree, you compare the x monomials as I said. So what is the relationship between the matrix entries and the coefficients of a and b?

Can you see? So the coefficients of a and b will appear in the linear constraint as it is. So, they will appear in the matrix as entries. So simply the coefficients of a and b they become entries of m, obviously in the correct locations. So we will not go into the exact location. Location details but just make this observation that its entries are coefficients of a b. Or when not applicable they are just 0.

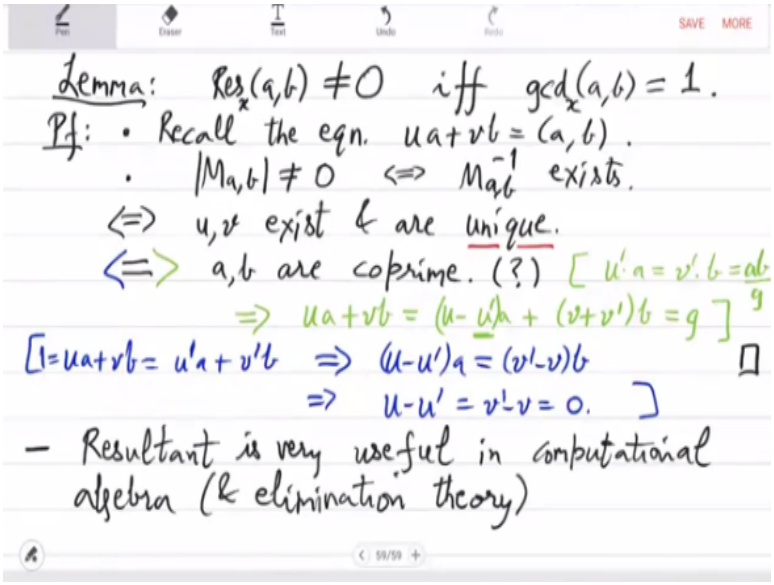
So basically the input polynomials given to you over a and b and look at the coefficient vectors and just place them suitably in a matrix. That is the matrix M a, b. And the linear constraint you get is M a comma b times these unknowns u 0 u 1 and so on and then v 0, v 1 so on. So u M times this is equal to a vector of constants. Whose entries come from the RHS, which is g c d. Again, these are just coefficients.

So it is on the LHS, you have a matrix multiplying this vector of unknowns which is degree a b. many. So M is a square matrix and on the RHS, you have a vector of constants they are coming

from the g c d polynomial, that is the linear constraint we were talking about. And finally resultant is just the determinant, that is the definition. So resultant with respect to x of a and b is defined as the determinant.

We will come to that so this determinant lives where what is it? It is a constant so it is in the field. So it is so you are basically mapping 2 given polynomials to the field element, that is what it does. Now it has a huge number of properties. So one of which will answer your question. But any questions about the definition or the setting? So what is this good for?

**(Refer Slide Time: 35:44)**




So we will show the first major property of this which is that if the resultant is non-zero. So remember that the where resultant is defined given two polynomials without doing any computation you can just define the matrix and compute the determinant, which is the resultant. So this is simply as this is just one determinant computation, no extra computation required and you can check whether it is zero or non-zero.

So suppose, it is non-zero, what can you say about the g c d? Now unique u and v means what? In Bezout identity when is or when our u and v unique? So actually, they are unique only if g c d is 1. So, that is the, so this is an alternative to Euclid's algorithm. You have a completely different way of deducing something that you deduce by Euclid g c d which is that you just take these two polynomials and compute the appropriate determinant. Again by linear algebra

algorithms, and if the value is non-zero, then you know that the polynomials are co prime.

And if the resultant is 0, then it means that there is a non-trivial g c d.. So this is the basic property and it is some major property because you can also do this with multivariate polynomials. So instead of one variable if you had 10 variables in a and b then you can extract the resultant with respect to x 1. So if you extract resulting with respect to x 1. What do you get? What remains? The remaining n - 1 variables, right?

So resultant is multivariate but this property will still hold. So resultant with respect to x 1 is non-zero if and only if the g c d with respect to x 1 is 1. So also for multivariate you can apply the same argument. So, it actually gives you relationship between common 0's of a b and the projected 0's. So when can a projected 0 between of a b in n - 1 variables, when can you lift it to n variables?

So those questions actually get related to resultant. So one such question is there in your assignment. So let us first prove this, so recall the equation, u a + v b = g c d. Now If the matrix determinant is non zero, it is equivalent to saying that unknown vector is unique because matrix inverse exists. Since the inverse exists the unknowns are unique and will they exist and they are unique.

And so now this uniqueness, what does it say about g c d that is the questions of claim is that this means that a b are co prime. Do you agree? Why is that true? So suppose a b were not co prime then you can find the u times a equal to v times b. For u prime times a equal to v prime times b. Where u prime v prime have their degrees limited. So this you can do just by so for example, what can you take u prime as?

So you exactly, so you basically want this to be equal to a b something like that a b by g c d I guess so which is LCM. Here a b by g c d, which is also LCM, so you basically want to get to LCM and they will be a u prime whose degree will be smaller than b that will give you this because the g c d is a non constant. So u prime is equal to b by g, which is degree less than b and similarly v prime is e by g, which is degree less than a.

So u prime v prime exist, but clearly they are not unique, why is that? Actually I should be careful you are talking about that equation, so what does it tell you about u a + v b = g. You want to show that u and v are not unique. So, do you know how to modify u v to some other solution. Exactly, so you use u prime for that so, another solution is u - u prime and here is v + v prime. So the degrees limited; the degree of u prime and v prime is equally limited.

So when you do this perturbation, you get a different solution than u a + v b. So, that is the simple property. So limited degree u v are still not unique, this is what happens. This completely fails when g is 1. Then the limited degree really gives you a unique solution. Well, which you can prove by looking at the determinant? So by the uniqueness of u v you can reduce that a b have to be co prime.

So that is one direction, what if a b are co prime? If a b are co prime then why are, well u v exist by the Euclid algorithm, why are the unique? Yes, so you go you jump this and go to the to this matrix and by studying the matrix you can reduce that. So this is all if and only if you basically have two views of Euclid g c d. So one is this Bezout identity view and the other is the matrix or resultant view and you can go back and forth.

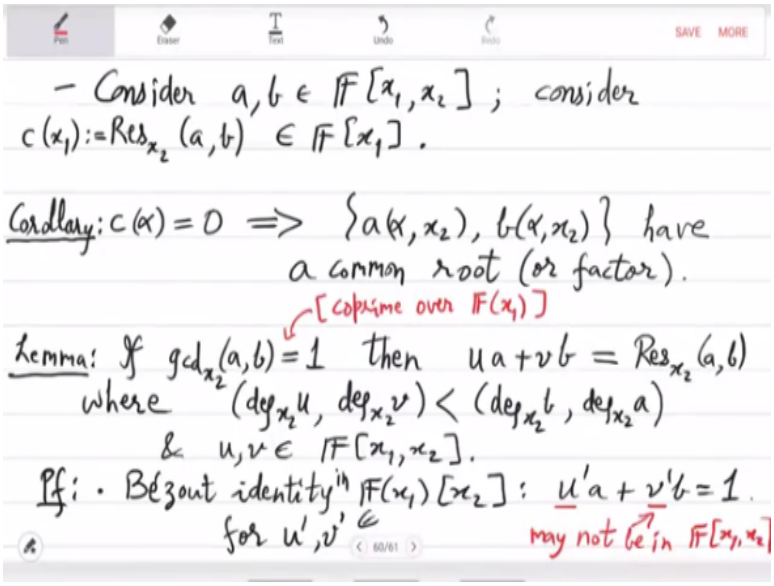
And these two views correspond to actually two different algorithms. One is the g c d algorithm usual recursive algorithm other is the determinant computation algorithm. So everything is all that is actually contained in this single lemma. Is that clear? Any questions? The other side, so suppose that a, b are co-prime, so you have u a + v b = 1. Well I can do it by calculation, it is actually that is easier.

Now let me just do it straight a straightforward way so suppose ua1 is equal to u a + v b = u prime a + v prime b. So then you take the difference of these equations. So you get u - u prime a is = v prime - v b. But you have assume that a and b are co prime. So this means that I mean this leaves you with very few options, it actually means that you both u - u prime and v prime - v is 0, because this a has to then divide v prime - v.

But then by degree only you can only divide 0. So that is the converse that is actually simpler than going by a matrix. So you use, this was for the green one is for the forward and for the converse you use this other calculation. So then that gets connected to the determinant or matrix inverse. So with this understood now let us move to the other major properties of resultant which is useful in the case of multivariate polynomials and also in which is similar to actually elimination theory.

In computational algebra and what is called elimination theory? So let me give you a taste of that.

**(Refer Slide Time: 47:54)**




So suppose you consider now bivariate a and b. Now, bivariate polynomials are extremely different from univariate polynomials. This is a much harder case, for example here now the roots may have no connection with the degree, in fact roots can be infinitely many common roots. That you can think of the integral case in the integral case common roots may be infinitely many.

So now for these possibly infinitely many roots, you can look at the question of projection. So you project the root to the first coordinate and you can ask the question of what are the possibilities of the first coordinate? In other words fixing x 1, how many lifts are possible to x 2? Or is there actually any lift possible to x 2? So those questions you can ask and they will be

answered by resultant.

So, a point where the first coordinate is your favorite, fixing. So reverse I mean fibers of projection pre images of projection. So you can consider here resultant with respect to x 2 of bivariate a b which will now be a polynomial in only in x 1. So this is why elimination theory, so you are actually eliminating a variable you are given 2 equations, so combining them, you can eliminate one variable.

So now you are in univariate in the univariate world, where things are better. So you in particular this resultant. Can only have finitely many roots, is that true? What if the resultant is 0 it will also be 0. So either this is absolutely 0 if not then x 2 has only finitely many possibilities. But what do these possibilities entail? So if you take a root of the resultant and fix your x 2 to be that. What is the relationship between resultant and roots of the resultant and common roots of a and b?

That is the question you can ask, so let us give this thing a name c x 1. So, let me make this observation. So C alpha = 0 implies that in A and B, if you fix x 2 to be alpha, sorry x 1 in a and b, if you fix x 1 to be alpha, then what? These two polynomials have a common root, do you see this? So suppose not suppose these two polynomials do not have a common root but these two are univariate polynomials.

So not having a common root could only mean that they are co prime. So you have then you this substituted a and plus v times substituted b = 1. Though the lemma was this resultant non zero means resultant 0 means that there is a common root so you read this lemma as a resultant is 0 if and only if the g c d is non one which means that there is a common root. So actually resultant if you kill the resultant then there is a common root.

So from this lemma just by reading the lemma correctly, you can reduce this property, this is a direct corollary of that lemma. That a root of this resultant will definitely lift to a common root of original bivariate, so this is the meaning of lift and elimination. So you are eliminating a variable which is x 2 finding a root in x 1 and then you know that this can be lifted. So this kind

of gives you a way to solve also bivariate polynomials using the univariate machinery.

So this is actually the starting point of very big theories. So say it as a common factor, you can replace this by factor. If F is not algebraically closed then you will only get a non-trivial factor. But intuition is really from the roots. So let me so resultant with respect to x 2 let us continue on that this has a nice Bezout identity, actually. So let me prove this as a lemma that, If the g c d of these a b with respect to x 2 is 1.

Then there is this bivariate Bezout identity, but it will be slightly modified from what you have seen before. So what will say now is? u a + v b, do you know this already? So visible identity says that u a + v b is = 1 right but that was in the univariate case, what will you get in the bivariate case? You will actually get resultant that is the beauty of it, where u v or these are polynomials limited degree.

It was a field element non-zero, so invertible. So u and v will have limited degree talking only in terms of x 2. So individual degree of x 2 in u and v is limited by whatever you saw before which is for u it is b and for v it is a and goes without saying that and v are looking for polynomials. So in the bivariate case we seem to be stating a new identity which is when the even a, b are co prime with respect to x 2 variable.

By the way, what is the meaning of this? We should have defined this what is the meaning of g c d of 2 by variance respect to x 2? Sorry, no, I would say the opposite push x 1. Yes, so what this is saying is that there is no common factor of a and b dependent on x 2, just like in univariate we say a and b are co prime, if there is no common factor if function of x. So here we will make the same statement that a and b do not have a common factor function of x 2. So, maybe this this needs a definition.

Although the, it is a natural extension of core primality from univariates. So if you want let me just say co prime over F x 1, the field. So it is the same notion of co primality over a field but the field here is a function field F x 1 and so with that there is only one variable, which is x 2. So we are talking about co primality over function field now. So it means the, intuitively correct thing,

so if that is satisfied then there is a Bezout identity except that the 1 will be replaced by the resultant.

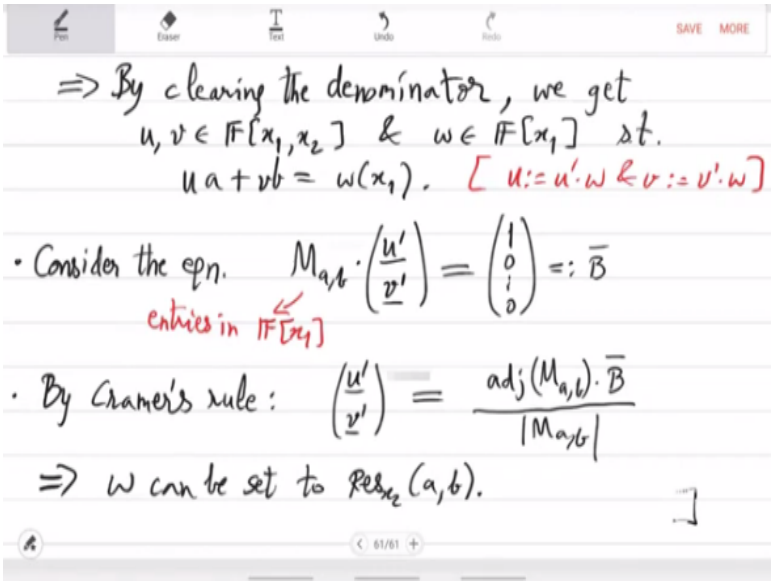
The reason is that, intuitive reason is that this resultant is a polynomial in x 1 and co primality is not able to see these polynomials. So basically any polynomial in x 1 is thought to be as equal to 1, where it is in the function field. So this is a more precise identity than what you got before. So here we are saying that actually if you restrict u v to these degrees then u a + v b will be equal to exactly the resultant polynomial which is free of x 2.

And the proof of this is actually quite straight forward. So we can finish the proof and then wait for the next class, so this is just by so this Bezout identity over F x 1. So, let me say the field is f x 1 function field and the variable is just 1, which is x 2. So over this, in this so you will get u prime a + v prime b = 1. So, by the by what you have seen before the univariate case, so think of the variable to be only x 2 and the field to be F x 1.

So in that polynomial, you will have u prime v prime giving you u prime a + v prime b = 1, that is just invocation of Euclid g c d standard Euclid g c d, but the problem is that this u prime v prime have maybe you are dividing by univariates in x 1, they are not polynomials. So you want to make u prime a polynomial in x 1, x 2 in both the variables. So to do that, you have to multiply by the denominator.

And that denominator is guaranteed to be a factor of resultant, Why is that? Do you see that? Well, this part is g c d part. This is just invoking Euclid g c d for univariates. But u prime may not be a polynomial. Now, they may not be in F x 1, x 2 because they are by definition only guaranteed to be in F x 1 field with the square bracket on x 2.

**(Refer Slide Time: 1:02:33)**




So to make them polynomials, you have to multiply by the denominator. So by clearing away we get u v polynomials and the w which is univariate in x 1. Such that u a + v b is = w. So on the standard Bezout identity, you clear away the denominator, so denominator can only be a function in x 1 a polynomial in x 1, so that is w and so u is just u prime times w. That is what is meant here.

Thus making u and v polynomials in bivariate what is the relationship between this w and resultant? That is the content of the lemma. So, how do you relate these two? This seems to be arbitrary. Why should it be something so special as the resultant? u prime and v prime are actually you look at the linear system which gives you prime v prime. So you look at the matrix that gives that too. So consider the equation, M, a b times you have u prime written down here expanded and v prime expanded as vectors.

And this is equal to 1, so this is the equation that gives you u prime and v prime the coefficient vectors of u prime v prime. And this resultant matrix appears here that we saw by the definition of resultant this will be the equation giving you u prime v prime. What are the entries of this matrix? Coefficients with respect to x 2, so this is actually a matrix over its entries are in F x 1. So this is a matrix whose entries are actually polynomials, they are univariate polynomials in x 1, they are free of x 2 in particular.

And the determinant of this is the resultant with respect to x 2. Now, can you say something about w. What can you say about the denominator that we created? So, actually, you have to recall what is called Cramer's rule in linear algebra? So when you want to solve this matrix equation, how do you get the solution? By actually here I here it is simpler, so you simply have to compute M inverse times elementary after 100.

And that will give you the first entry of M inverse and the, what is that? So, that is to do with the adjoint of the matrix divided by the determinant of this. So basically matrix inverse is some matrix divided by the determinant. And that matrix happens to get adjoint. Entries of adjoint are, what are the entries? They are the same as before so they are again univariate polynomials in x 1. So the numerator here is a polynomial in x 1, every entry is a polynomial x 1 denominator is also a polynomial x 1, but it is the resultant.

So you cannot clear anything worse than this.. So w in particular you can take to the resultant. So, that is why you get the, identity in the lemma. So this means that w can be set to resultant, that is it. But that is out of control, that we cannot say in general. In general, you can be as bad as resultant. So the generic cases resultant some specialized cases may give you a factor of resultant, but that is not important. All our results will be based on this identity. Any questions? So we will discuss more about resultant next time.