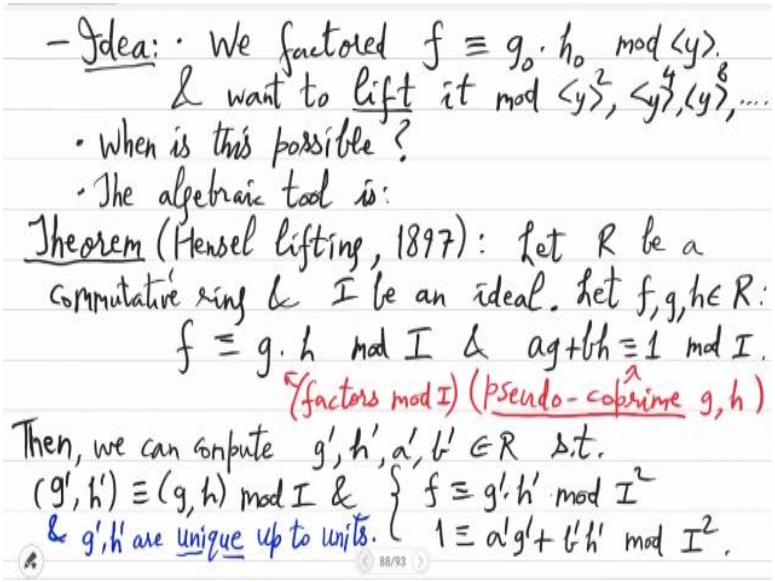
Computational Number Theory and Algebra
Prof. Nitin Saxena
Department of Computer Science and Engineering
Indian Institute of Technology-Kanpur

Lecture – 16
Bivariate Polynomial Factoring (continued)

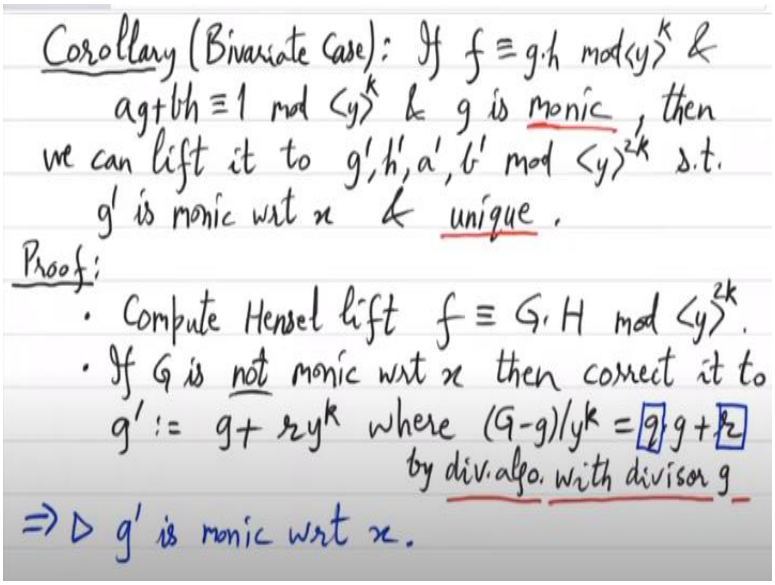
(Refer Slide Time: 00:17)



So last time we did versions of Hensel lifting, right. So main theorem of Hensel lifting is this one. For a commutative ring R in an ideal pseudo-coprime factorization can be lifted from mod I to mod I square by a closed form expression. And then as an application, there are two applications. You can either take this ring R to be the ring of integers and ideal I to be the prime ideal generated by a prime p.

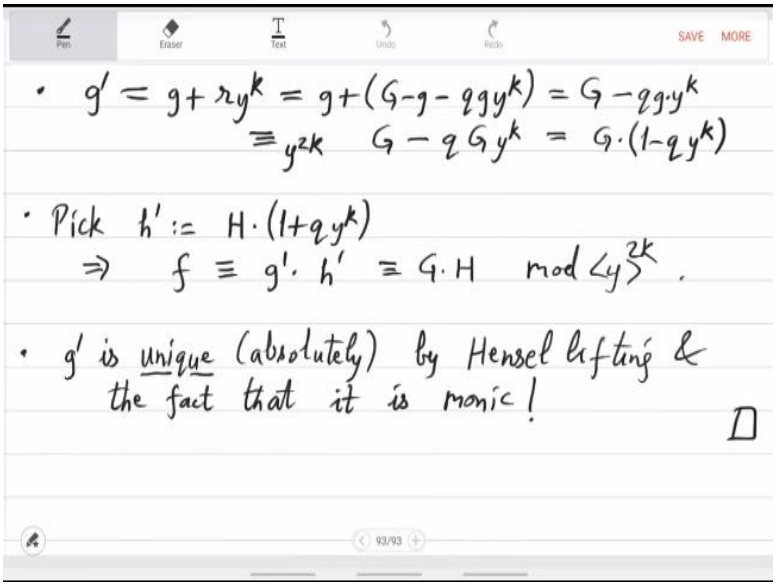
And then you can actually or you can take z x. So integral univariate polynomials and then mod p you can find the factorization and lift it mod p square and lift that mod p to the 4 and so on. Or you can take R to be a bivariate, the bivariate polynomial ring f (x, y) and I you can take to be y, y square, y to the 4 and so on. So currently we are interested in that one, in the latter.

(Refer Slide Time: 01:23)



So in the bivariate case, the application gives you mod y to the k factorization, 2mod y to the 2k factorization, which will continue to be pseudo-coprime and g will be monic and this g prime will actually be absolutely unique. Not up to units, but it will be there will be only one option for g prime. So that also we completed. Any questions?

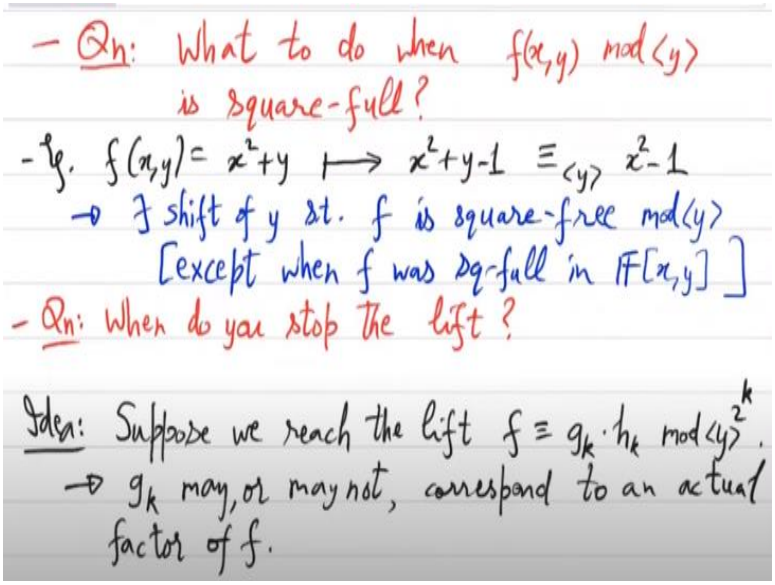
(Refer Slide Time: 01:59)



Okay. So all this remember we are doing so that we can find an algorithm to factor bivariates, right. This still is not a factorization algorithm because well in a way it is reducing to univariate factorization mod y, but then the lifting is only giving you factors mod y to the 2 raised to k. It is still not a factor in the bivariate polynomial ring R. You want to remove this mod I.

So how can, how will you remove this is the next question. The other restriction is that you want a coprime factorization. Now mod y that may not exist, right? You saw this example x square plus y. Mod y it is x square, so there is actually no coprime factorization. So what will you do in those cases?

(Refer Slide Time: 03:03)



So what to do when f (x, y) mod y is square-full, in fact a square is the worst case. So any ideas? What will you do when f (x, y) is x square mod y? Then you do not even have a starting point. So what you do in that case is you shift your starting point from mod y to something else like mod y minus alpha, mod y - 1 for example. So in that example of x square plus y if you change your origin from y - 0 to y - 1 you will get x square plus y – 1, right?

Which mod y is x square minus one which is no more a square at least in characteristic 0 it is x - 1 times x + 1 and – 1, + 1 are distinct. No right now, we are only interested in bivariate. Let us continue with bivariate. So by changing the origin, so instead of evaluating f (x, y) at y = 0, you evaluate it at some other point. And hope is that it will be square free, right. So when will that happen?

Well, if you started with an f that is already a square. Then no matter what y you use it will remain a square, right. So that is the actually only obstruction. As long as f is as a bivariate polynomial it is not square-full, it is square free, you can find a y such that it continues to be square-free. So basically we can always do that, we can always do

this. There exists a shift of y such that mod y except when f was square-full to begin with.

If as a bivariate it was square-full then you cannot do anything. Then you have to resolve it in a different way which is you take derivatives and GCD; f is given as a square-full polynomial then with GCD it will factor. So use that algorithm, use the derivatives. So you can assume that f is square-free and if f is square-free then you can find a shift. So substitute, basically look at mod y minus alpha and modulo that it will be square-free, okay.

So if it is square-free then you can also hopefully find a univariate coprime factorization and then use Hensel lifting, okay. That is the, that will be a starting point. So we will go into more details. But let us first look at the overview. This gives, always gives you a starting point. Next question is when do you stop the lift? Because you have seen this example, which where polynomial was irreducible and it kept on factoring mod y to the 2 raised to k ad infinitum.

So when do you stop and output that either the polynomial is irreducible or the polynomial is reducible, right. You cannot do this for very long. So at some point you have to stop and make a decision. This actually is a trickier point. And so here we will use the strong properties of Hensel lifting and we will also use this invariant of resultant okay to make this thing work.

So that will tell you when to stop, do some computation and output. Either output a factor or output that it is irreducible. Even though mod y powers it has been always reducible. So steps are so suppose you reach a point 2 raised to k, f g k times h k mod y to the 2 raised to k. Now, if you assume that or if f had a factorization, if f had a non-trivial factor, well the problem here is that or the question we have to ask here is whether g k corresponds to some actual factor of f like without the mod.

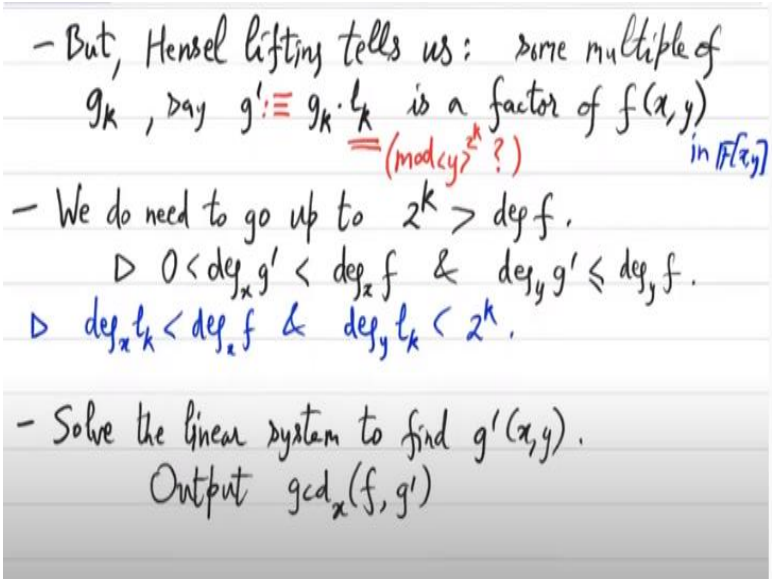
That may not be the case. But you know that if f had a actual factor, there was a starting point, which would have led to a good g k. And the converse of that is that if at this point you have a g k, maybe you can construct an actual factor by modifying g

k. Now what are the possibilities of modifying g k, what could you do? Assuming that f has a has an absolute factor, how can you get to that from g k?

So here you have to use the fact that Hensel lifting is in some sense unique. So if you start with a with g 0 monic, then the only lift possible is this g k. So if you had started with the correct factorization, then you would definitely get to g k and maybe some multiplier of this will give you the correct factor. So basically, we will be looking for that multiplier.

So you is there an l k such that g k times l k is an absolute factor of f. That is the only way we can transform g k by multiplying with something which is outside Hensel lifting. So you have to find that. So g k may or may not correspond to an actual factor of f.

(Refer Slide Time: 11:14)



But what Hensel lemma tells you or Hensel lifting theorem tells you is that some multiple of g k, say g k times l k is a factor of f. Is this point clear? Do you see we need to talk about multiple of g k instead of g k itself. So if you look at g 0, since g 0 is a factor of f mod y g 0 actually may be properly dividing an actual factor of f mod y. So f has an actual factor which further factors when you go mod y, right.

So you g 0 may just be a part of an actual factor, it may not be the full factor because you are only doing computation in this limited precision mod y, not absolutely, right. So g 0 maybe just a factor mod y of the actual factor of f. So you will in general need

to multiply with something. And so this l k we have to find separately, okay. This Hensel lifting cannot give you because the process of Hensel lifting is unique once you fix g 0.

So that will just uniquely lift to g k. But g 0 has an associated l 0 such that g 0 l 0 is the actual factor of f but now seen mod y. So that g 0 l 0 will keep on lifting and will become g k, l k, okay. That is the insight behind this. But l k is now unknown. This you have to, you have to come up with a method to find this. But you can compute this only mod limited precision which is mod y to the 2 to the k.

So that is another problem. Will this precision be enough? So in this much precision there is an l k such that g k times l k is g prime, but so this I maybe I should not use equal but congruent. How much of k should we go to so that these so that we can find an l k and g k times l k is an actual factor of f which we are calling g prime, right? So these things are still unanswered and unclear.

Well, what is the necessary condition or bound on 2 raised to k? That you definitely have to go to. g prime could have degree as high as degree of f or just slightly smaller than that. So 2 raised to k should be enough to at least compute up to that, right. So 2 raised to k should be more than the degree of f. That much is clear. So that bound is necessary. So we will, let us remember that bound.

So we do need to, 2 raised to k should be greater than the degree of f. Also we know that the degree of g prime which is an actual factor this is between since it is a non-trivial factor that we are looking for with respect to x, individual degree with respect to x should be at least 0, it should be more than 0 and should be less than the degree of f, right. This is the definition of a non-trivial factor g prime of f with respect to x.

So g prime that is currently unknown will satisfy this condition. Yes. **“Professor - student conversation starts”** g prime is a factor of f modulo y. No g prime is an actual factor. **“Professor - student conversation ends”.** g prime is a factor of f without any mod. So it is unknown because we were doing computation mod y to the 2 to the k, not absolute computation.

No, l k we are not putting any restriction. We have computed g k and we are looking for g prime. So since we are looking for g prime which is a non-trivial factor of f with respect to x this range individual degree of g prime has to satisfy. Also 2 raised to k should be large enough. If 2 raised to k is very small then there is no hope of getting information about g prime from g k.

And also you have to degree of g prime with respect to y also you can bound, right? Because g prime is an absolute factor of f. So individual degree of y cannot really exceed the degree of f. So this is also less than equal to the degree of f with respect to y. It could be equal. So for example, f could have been x, y and g prime is just y. So that is a non-trivial factor with respect to x. Well, not quite.

Maybe x square y and look at x, y. So x, y is a non-trivial factor of x square y; satisfies all these inequalities and here it is an equality, okay. So these are the bounds that you should remember. We will, these will be important in the algorithm. Okay. So now we will use the trick that we have used many times before, which is that if you look at this congruence, g prime congruent to g k, l k mod y raised to 2 raised to k, g prime is unknown, l k is unknown, but g k is known.

So how do you find g prime and l k? You have degree bounds on g prime and so you also have degree bounds on l k. In fact that also we can write here. So degree of l k with respect to x is less than degree of f and that with respect to y is less than well 2 raised to k. Because you are going mod y raised to 2 raised to k. Yeah, sure. But that is it would not be needed. The linear system has to take care of that.

So basically you have upper bounds on the degrees of g prime and l k and it is a so you get a linear system in the unknowns. Okay, it is a finite linear system in the unknown. Unknowns being the coefficients of g prime and l k. So you design that linear system and solve it. **“Professor - student conversation starts”** Sir, can you explain why there is an equality on the y? **“Professor - student conversation ends”.**

It can be equal. As I said there is no reason why they will not be equal. Sorry, no x we want a non-trivial factor with respect to x; y is somehow set with the 0; y since you started mod y think of y as being set to 0. So the only free variable one should think of

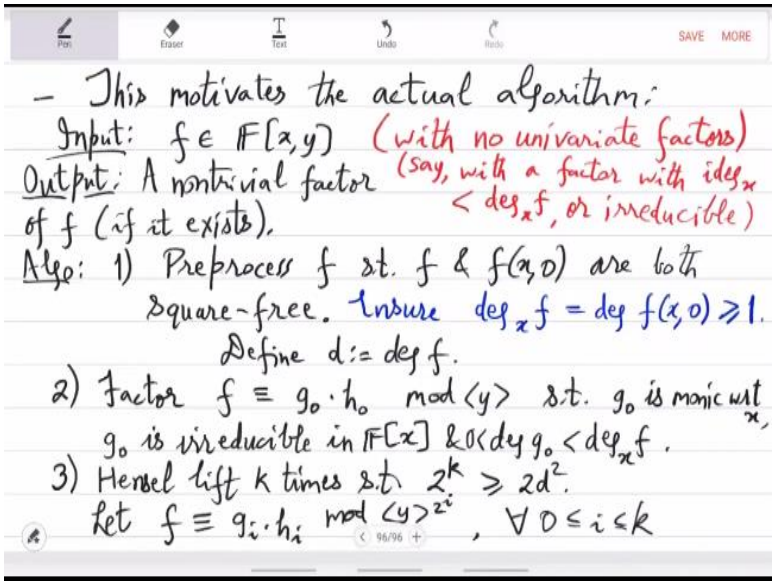
here as x. So in x, g prime has to be non-trivial. We do not care about y; y is kind of a constant in this calculation, in this algorithm.

Anyways, if it does not exist then the linear system cannot give you that. The linear system will also tell you that there is no such g prime, okay. So these are the constraints and you will solve the linear system. You will get some candidate for g prime and some candidate for l k. But then it is still totally unclear whether this g prime will be an actual factor of f, right?

Because it is coming from a linear system which was designed mod y to the 2 to the k. And there is little reason that it will just magically start dividing f absolutely without mod, right. So that magic we have to show mathematically that it actually happens. This is how it works. You solve the linear system, you get a g prime, it will divide f. Okay, but that needs a detailed proof. Well there is some modification.

What you do is solve the linear system to find g prime x, y. Then output the GCD of f with g prime with respect to x. Now if the GCD is trivial, which is 1 that means, f is actually irreducible. If the GCD is something non-trivial then you have a factor, okay? That is essentially the algorithm. The main steps of the algorithms are these. So yeah. Yeah, sure. So we will see the actual algorithm.

(Refer Slide Time: 23:21)



So this idea sketch motivates the following algorithm for bivariate factorization. So in the input you are given f bivariate. Does not matter what field. We will assume that

there are no univariate factors. If there are univariate factors then you can find anyways, right. If there is a univariate in x factor, then you can just set y and factorize by using the univariate factoring algorithm that you have as a sub routine.

And symmetrically if there is a univariate in y factor you factorize with respect to that by fixing x. So we will not worry about the univariate factors; f we were interested in the bivariate factor. So the bivariate factors have both x and y, right. So in terms of one variable, the individual degree is strictly less because it is a non-trivial factor. In fact that one we want to be x. So maybe that is another assumption.

Say with a factor like g prime, with a factor with individual degree with respect to x less than that of the degree of f. This we can assume. I mean either it has a bivariate factor where the individual degree of x is less or y is less. You can do, you can try both. Okay, so that is not really a problem. It is an assumption without loss of generality. So that is your input. In the output, well okay I have to correct that.

It could also be irreducible. It is either irreducible or there is a non-trivial factor where the individual degree of x is smaller. We have to distinguish these two cases. So in the output we will get a non-trivial factor of f if it exists. If it does not exist, then we will output that f is irreducible, okay? And this will be a deterministic polynomial time algorithm. So in the first step preprocess f to make it square-free, okay.

This we briefly discussed before. We will again go into the details later. But the preprocessing is simply this that if f is square-full already, then you can take GCD of f with appropriate f prime, either derivative with respect to x or derivative with respect to y and the GCD will factor f. And the bivariate GCD computation will, you can do because you can do long division. It would not be expensive.

It may happen that f is square-free, but at 0 it is square-full. So in that case we have to shift y by and make it y minus alpha. And that alpha we have to find. Now why will it exist? How will we find it? Those things we will see in the detail. Another thing we can do in the achieving the preprocessing is ensure that the degree of f with respect to y does not change. f at y = 0, what does it mean?

So if the degree of f with respect to x is more than the degree of f at y = 0, it means what? In the leading monomial with respect to x there is a y, y divides it. So when you set y to zero then that vanishes. This is again a problem which can be solved by appropriately shifting y okay. So both so all these problems here in the preprocessing have a common solution which is appropriately choose alpha and work with y minus alpha instead of working with y - 0.

So we will see the common solution later. These things are easy to ensure. And with all that preprocessing now we call the degree to be d; d is the total degree and well that we could assume earlier. We could assume that this is at least 1. If the degree of x with respect to x is not at least 1, then it means that it is free of x, right. That is not our case. Okay, so this was essentially for free.

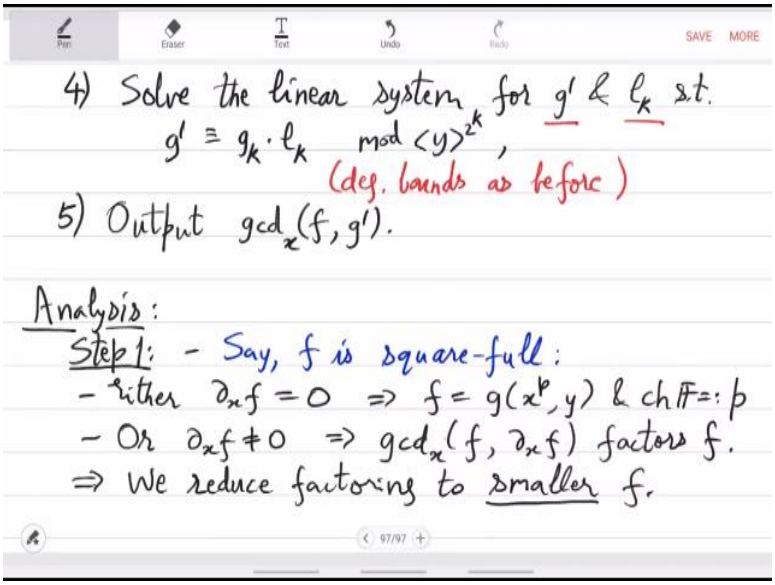
Now we have a moderately nice bivariate polynomial, which at y = 0 is square-free. And setting y to 0 does not change the degree with respect to x. Now if there was a bivariate non-trivial factor of f, how will it manifest in this Hensel lifting that we are doing, right? That is what the algorithm has to capture. So but the algorithm has no idea. So the algorithm will just do the following.

It will factor f in some way mod y such that g zero is monic. Mod y there is actually no y; y has been set to 0. It is just univariate factorization with respect to x. So you pick a monic factor, call it g 0. It is irreducible; g 0 is irreducible over f or in F x and its degree is strictly less than degree of f with respect to x. And obviously, it is not a constant. So its degree is positive as well.

Okay, so you factorize f at y = 0 and get this atomic factor g 0. Now on G 0 you do Hensel lifting sufficiently many times. So that bound actually I will now increase slightly. So Hensel lift k many times such that 2 raised to k is at least 2d square. Okay, so previously I had said d, but now I want a bit more. I want to 2d square. If you remember 2d square is also the bound on the resultant of basically two polynomials of degree d bivariate.

So that is the connection of this lower bound. So let us say these Hensel liftings are g i h i mod y to the 2 to the i for all i 0 to k. Okay, so these g i h i you have collected. I mean your algorithm actually computes this. What is the next step?

(Refer Slide Time: 33:56)



So next step is the linear system solver. You try to find l k. So g prime is g k times l k, g prime l k are unknowns, mod y to the 2 raised to k. Degree bounds as before. So g prime and l k I have already given you the degree bound, so I would not repeat. Let me just say degree bounds as before. For g prime and l k. Okay, essentially g prime I want degree with respect to x to be strictly smaller.

Degree of g prime with respect to y and degree of or degree of g prime with respect to y I want less than equal to that of f. Degree of y of l k, we do not care, less than 2 raised to k. And degree of x of l k, again strictly smaller than degree of f, right? Those four things you remember and solve the linear system. And last step is as promised. Just the GCD. So compute the GCD and output it.

So this is the full algorithm and the promise is that this works, okay. So given f the promised output will actually come. Either this will factorize f or it will just give GCD equal to 1, which will indicate that f is irreducible. Any questions about the steps of the algorithm? Yeah, so how hard do you think the proof is? Can you guess the proof? Why does it work?

So the only bad case is when f has a factor, but this GCD comes out to be 1, right. So what does it mean? So at this point actually you should move to resultant. So you look at the resultant of f and g prime with respect to x. So that would be? GCD 1 means that the resultant is 0, right. But then what is the next step in the argument? So remember that all the above computation that we have done that information is mod y raised to 2 raised to k.

So this resultant equal to zero equality you reduce mod y raised to 2 raised to k. And then use all the congruences that you have computed. And that will give you a contradiction okay. So that is the line of argument. It will lead to a contradiction. So the only possibility is that f is Irreducible if GCD is 1. So yeah, let us look at the steps now. Hensel lifting we have already seen in great detail.

We have to look at step 1 more carefully now. What is this preprocessing step? So let us look at step 1. And the next thing then to look at would be step 4, step 4 and 5 basically. In step 5 what is the meaning of GCD 1, okay. So there are two things to analyze. Time complexity should be fine. This is routine to check that it is deterministic poly time, right.

You are just doing some trivial p processing. Alpha will be found and then you will just use y minus alpha. Factorization of univariate is against somebody else's responsibility. Based on the field f there will be a univariate factorization sub routine. So that we assume to be fast. Hensel lifting you have seen, one step is very simple. Since linear system is not too big, so you can solve it.

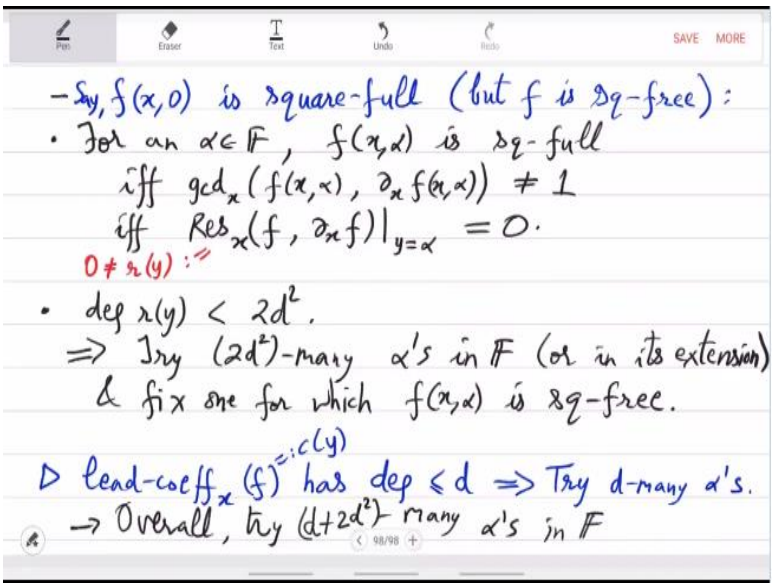
And finally, the GCD is just Euclid GCD using division. So that also is fast, right. So time analysis I will not do. Let us just look at the correctness. So in step 1 say, so either f is square-full, that is one case. Other case is f is square-free. But at y = 0 it is square-full. So if f is square-full, then yeah, then there are these simple cases. Either the derivative of f vanishes.

If it vanishes, then it means what? So if it vanishes, then it means that f is equal to some g of x to the p, y and characteristic is p, p is a prime. So then you can actually reduce f. The degree of f can be reduced. So instead of working with x you work with

x to the p replaced by x, right? So it is a kind of a reduction to a simpler polynomial. Other case is the derivative is not zero. If it is not zero then what do you do?

Compute the GCD of f with it, right? And what is this? Since f was square-full, this GCD is a factor. It factors f, okay. So square-full case is done. Any questions? So basically, we reduce to a simpler f, okay.

(Refer Slide Time: 41:38)



Second case is f (x, 0) is square-full, but f is not. So f is square-free. Okay, so this case is a bit trickier. What do you do here? This is the motivating example x square plus y, right? x square plus y is square-free but at y = 0 it becomes square-full. So how do you get out of that problem? What origin should you use? Yeah, so you will try several possibilities.

You can think of your field as somehow ordered and just look at the first few elements of the field. So in terms of numbers, you think of 0, 1, 2, 3, 4 and try these shifts and the guarantee is that one of them will work. So why is that? How do you bound the number of attempts for shift. Of degree of what? No not degree of zero. You have to recall the how do you check square fullness actually or square freeness?

Is the same thing, GCD of f with f prime. So you should look at the resultant which is also called discriminant degree of that. So for an alpha in f, f (x, alpha) is square-full if and only if the GCD with respect to x is non-trivial, which is if the resultant is,

resultant at y equal to alpha vanishes. So this is each, let us call it r y. What is the degree of r y? That is less than 2d square.

So either r is absolutely 0 or it has less than 2d square roots. Is r absolutely 0? If r was absolutely 0 then there would have been a f would have been then square-full right. So by assumption this is nonzero. So because of square freeness of f, r is a nonzero polynomial of degree less than 2d square. So it cannot have so many roots. So we just try the first 2d square field elements. One of the alpha works.

Yeah, that is true. Why not? Yeah, this is just the GCD property. Right. So GCD resultant if and only y. Okay, right. Yeah, I do not see that. Anyways, we can, even if it is true, that you can handle. So basically pick first few alphas. So in F or in its extension, yeah if you have to go to an extension then that might become a randomized component in your algorithm.

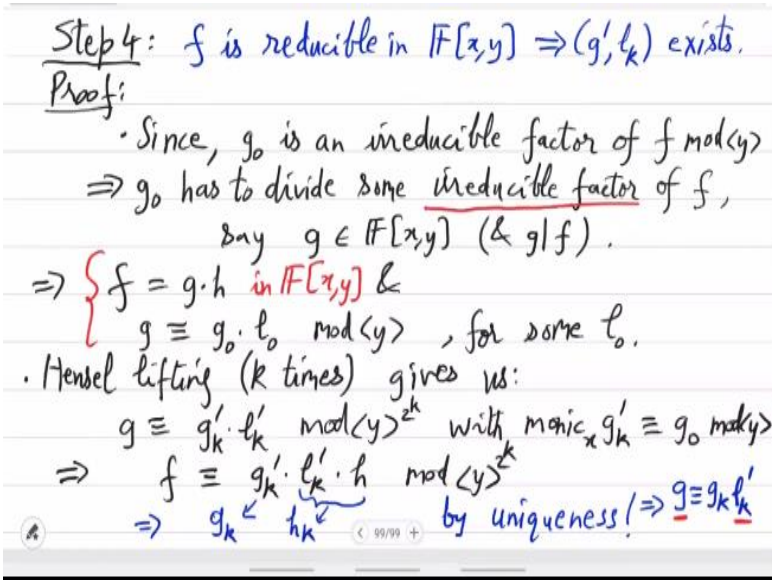
But if your field is big enough, then you will have 2d square many elements. So either way somehow find these 2d square many elements and try all the alphas. So trying means that fix an alpha and then check whether the resultant is vanishing. And pick the alpha for which it is not. So try these many alphas and fix one for which f (x, alpha) is square-free, okay. So that is how you find alpha.

This is just by enumeration. And continuously check. Okay, this is and one last thing is the degree falling when you fix at y equal to alpha. For that you just observe that the leading coefficient with respect to x (f) has degree how much? Cannot exceed d, d is the total degree. So and this leading coefficient of f with respect to x is a polynomial in y, c y we can call it, right. So c y is a nonzero univariate with degree at most d.

So if you try out again d plus 1 many alphas. For one of them the leading coefficient will not vanish. So you will be interested in that in those alphas, right. So combine all these observations. In this case try d many alphas. Yeah, so the basically the bad alphas they are contained in two polynomials. One is r y and the other is c y. So you multiply the two. So r times c has degree 2d square plus d.

So that collects all the bad alphas. Everything else is good. Okay, so you just have to look at 2d square plus d. So overall try d plus 2d square. It is just additive. Okay, so this completely describes step 1. So now let us move to the most interesting part of this, which is step 4. Step 4 gives a good g prime. No matter what g prime you pick from the linear system solver it will work.

(Refer Slide Time: 50:53)



So what we will show here is reducible, if f is reducible in the bivariate polynomial ring not mod then yeah, for now forget about step 5. Actually, even in step 4, you have to first prove something. Suppose f is reducible, how are we certain that the linear system solver will be able to output something. There may not be any solution. The linear system may be infeasible, right.

So you have to actually prove that when f is reducible the linear system is feasible, right. That what g prime it gives is another matter. That we will show later, but why should some solution of the linear system exist? Why is the system feasible? So how do you show this? This is not difficult to show. So f is reducible. So you pick a as I was saying before, let us say big G is a factor of f and big G mod y will factorize.

So pick an irreducible factor, call it **0,** g 0. And yeah, but you may say that Hensel lifting was completely oblivious to those things. So, Hensel lifting started with some arbitrary g 0. Okay, so you break f into g big G times big H. Big G will factor mod y, big H will factor mod y. And g 0 is in one of these, right?

g 0 is either an irreducible factor of big G or it is an irreducible factor of H, right irrespective of what your Hensel lifting did or what was the starting point of the Hensel lifting. g 0 is an irreducible factor of either big G or big H. And then based on that, we will actually show that the linear system has a solution. So let us prove this. Since g 0 is an irreducible factor of f mod y so g 0 has to divide some irreducible factor of f, say we call it g.

It is a bivariate factor. Is this clear? You assumed f to be reducible. So this g 0 that Hensel lifting picked as a starting point it has to divide some actual factor g, obviously mod y, right? Any questions? So this means that now you can write these consequences. So f is equal to g times h over f. And this particular g, this factors into g 0 and say l 0 mod y. That is the starting point right.

F is factoring as g times h in the polynomial ring and modulo why this g is further factoring into g zero and something else l 0. But obviously, you did not know neither did you know g nor did you know l zero when you did Hensel lifting, right? That was oblivious to all this. But this is happening anyways. Implicitly it is happening, that when you are when you do Hensel lifting, g 0 l 0 will also lift to g 1 l 1 and eventually to g k l k.

Because Hensel lifting as a theorem says that all these things are unique, right? So even if you do not see it happening, implicitly it is happening. So that is all. So Hensel lifting k times gives us that g is g k prime. Were we using prime before? Yes, I am trying to do it formally. So this is some g k prime times l k prime mod y to the 2 to the k with monic g k prime. Monic with respect to x.

And g k prime is a lift of g 0, right? So it is congruent to g 0 mod y. So that means that f is congruent to g k prime, l k prime times h, okay. So just look at this last equation. So this is an alternate factorization for what Hensel lifting gave you which is not possible, right. So this is g k prime, which is actually a lift of g 0 and the rest which is l k prime times h.

So by the uniqueness of Hensel lifting, you deduce that g k prime is actually the g k you already computed and l k prime times h is the h k that you computed. So, this is

then g k. And this thing is h k by uniqueness, okay. So that means what? So that means g is g k l k prime. So g is g k times l k prime mod y to the 2 raised to k which gives you what? That the linear system is feasible.

So there is a multiple of g k which satisfies all the degree constraints that you put in the linear system, right namely g, l k prime. That is a feasible solution. Yeah, so it seems to be a bit roundabout, but in the end, what we are saying in step 4 is that if f is reducible, there is a decent non-trivial factor called g, which will actually be a solution of the linear system. Okay, so the linear system is solvable, it is feasible.

Now, whether your linear system solver will give you g that is not being claimed, okay. All that we are claiming here is that the linear system will have some solution. In particular, it has g as a solution. But then since you do not know as a user, you may get some other solution of the linear system because it has many solutions. What you will do with that happens in step 5. Okay that we will see next. Yeah. Any questions? Okay.