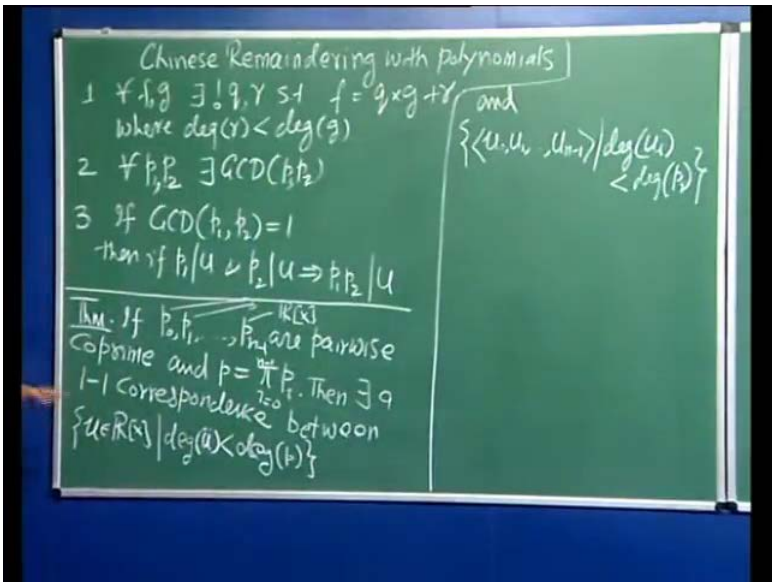
Computer Algorithms – 2
Prof. Dr. Shashank K. Mehta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 22
Chinese Remainder II

In the last lecture, we began the discussion Chinese remaindering theorem in context with the polynomials. We gave the basic background. We had established a few results.

(Refer Slide Time: 00:35)



First one was the division property that polynomials have that is for all polynomials f and g. There exists unique and r such that f can be written as q g plus r, where the degree of r is strictly less than degree of g. This is the property very similar to the integers. The second property that we had was the existence of the GCD. We said that for all p 1 and p 2 p 2, their existence GCD of p 1 and p 2. This is the polynomial, which divides both of them. Anything else that divides these two also divides this. We also notice that the GCD of two polynomials is 1 that is to say p 1 and p 2 are co prime.

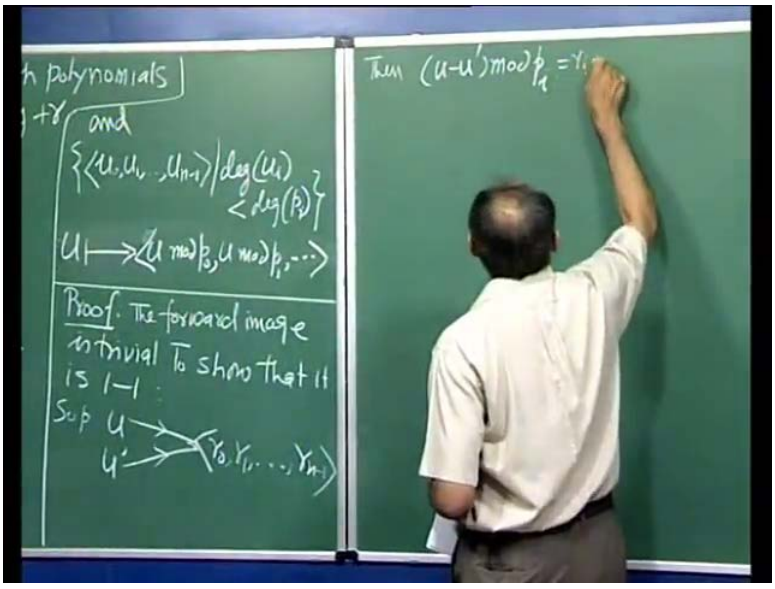
If p 1 divides another polynomial u p 2 divides the same polynomial then, p 1 and p 2 their product divides u. This we can extend to n polynomial. If you have a set of n polynomial, which are pair wise co prime; each of them divides u then, the product of all the n polynomial. We had also written some properties of modulo computation. Now, we

are in a position to take the Chinese remaindering theorem and the statement as well as the proof.

Most identical, there is no difference. Only difference is in case of integer, we were looking at the value. Here, we will be looking at the degree of polynomial. The theorem says that if we have n polynomials are pair wise coprime, p is the product of all this polynomial 0 to n minus 1 say. Then, there exist a one to one correspondence between them. On one hand, we have the polynomials u. So, I would simply say, of course, these are all members of polynomial over one variable.

So, all those polynomials, which have degree strictly less than, sorry, degree of u is strictly less than degree of p. So, on one we have a set of polynomial. In addition to this, we have this touple of polynomial. In touples of polynomial, we call them u 0, u 1, u n minus 1. The degree of u i is less than strictly less than the degree of p i. The 0 is also included in this set. Similarly, 0 is included here as well this one to one correspondence we can show in the following way.

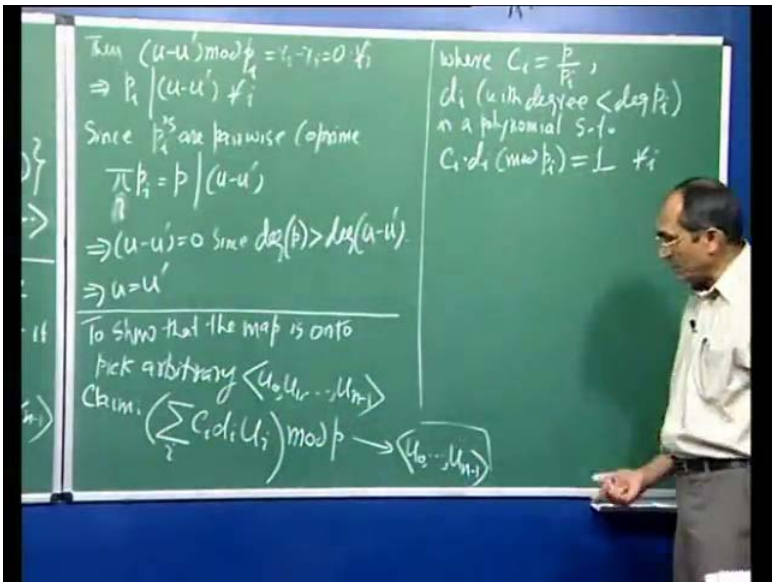
(Refer Slide Time: 06:13)



Sorry, I have to put a condition that this correspondence such that the image u actually goes to touple where the first element is u modulo p naught. The second is u modulo p 1. So, this correspondence is unique and the proof is easy to establish. For every u, we have such a touple because all we have to do is compute the remainder of u with respective p 0, p 1, p 2 and so on.

So, the forward image is trivial. It is the thing that we have to show that this map from u to this is one to one. Then, we will say this on to show that it is one to one. We will use one of the results that we have suppose u and u prime both near to the same touple. Let us take that as r 0, r 1, r n minus 1 both map to the same thing. This simply means that u modulo p 0 is r 0. Similarly, u prime modulo is 0 and so on.

(Refer Slide Time: 08:37)



Then, u minus u prime modulo, any p i will be r i minus r i is the 0. This is for all i that implies that p i divides u minus u prime for all i. So, every one of this p i is able to divide this polynomial. But, our assumption is that all p i's are pair wise coprime. We have earlier argued that in that case. Since, p i's are pair wise coprime, the product of p i, sorry, product of p i's which is same as p divides u minus u prime.

Now, u and u prime belong to this set. This set contains all the polynomials of the degree strictly less than the degree of p. You have a polynomial p here and polynomial degree of p is strictly less than of p. Then, p divides simply means u minus u prime is 0. Since, degree of p is greater that the degree of u minus u prime that means, u is equal to u prime. That establishes the one to one nature of this map. So, this was exactly the same argument we had occurred for the integers.

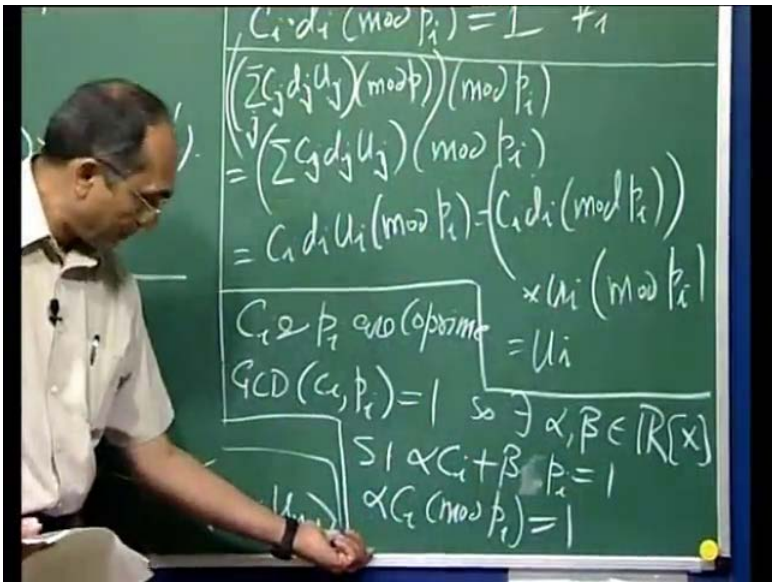
Similarly, to show that this is one two, what we have to do is to show that the map is 1, 2. We are going to pick the arbitrary element from this step and give a polynomial u such that the image of that u is 1 that we had picked. So, let us take pick arbitrary polynomial,

sorry, n touple of polynomial u 0, u 1 through u n minus 1 where each has degree less than a degree of a corresponding p.

So, u i has degree less than a degree of p i. We claim that this sum c i d i u i mod p, the image of this polynomial is this. Now, let me just explain what is this because as in the case of the integers c i is nothing but p by p i. So, let us just write down where as c i is p divide by p i d i. This has degree strictly less than the degree of p i; is a polynomial such that c i times d i modulo p i is 1 for every i.

In other words, d i is the multiplicative inverse of p i such a d i exists. Then, this sum modulo p is a polynomial whose image is again this. So now, there are two points again we have to argue that d i is exists until the image of, this is indeed this. But, image of this is this simply means that when I compute modulo p i with this polynomial, I should get u i.

(Refer Slide Time: 14:28)



Then, right down now, let us first take that the sum i p i d i u i mod p and the modulo p i of this of this from our last lectures exercise. We have noticed this is the product of p i and some other p. So, this thing is same as sum c i d i u i mod p i because p i is a factor in p i. p i is also a factor in every c j other that c i. Remember, in c I, we do not have p i. Every other c j has p i up there. That makes modulo computation easy because that would be 0 for this one.

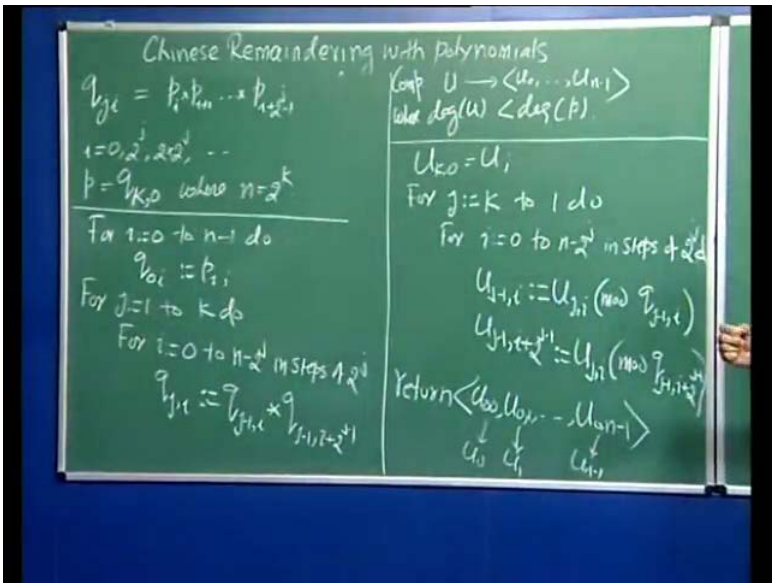
This will reduce to simply c i d i u i. I would probably apologize for this is. Let me make them j here just to not to confuse these is with this here. So, I just make them j. All the j's other than I will vanish and left with this term in the sum. Then, of course, we still have modulo p I, but this be, again yesterday we saw c i d i. I can always say this as mod p i into u i. Then again, mod p i and this is 1.

So, this simply reduces to u i. The reason is the polynomial u i has less than that of p i. So, it does not really effect p i. This is what we wanted to show that the image of this in this. What we have started with the existence of this d i comes from the fact that c i and p i are coprime. They do not share any factor or any non reveal polynomial because p i has no polynomial factor with any p j. Hence, it does not have any factor with entire product of p j without p i. Therefore, the GCD of c i and p i is 1.

Once again, we have seen yesterday that in this case as well there exists some polynomial alpha and beta. So, there exists alpha and beta in x such that alpha c i plus beta p i, p i should be 1. I should be able to express the GCD of the two as this linear combination. Then, if i take modulo p i of both sides; we get alpha c i mod p i equal to 1. This and this is all we wanted to show.

This alpha would serve as c i. Notice that, you can have many alphas. But, we replace this by modulo p i of alpha that is say you can reduce, divide this alpha by p i. Take the remainder that will also serve this part first and giving actually p i. It has minimum degree reduce. That establishes the one to one correspondence between this touples; all these polynomials, which have a degree less than the degree of p. Now, we look at the polynomial algorithms for computing the touple; touples for a given polynomial. Conversely, conclude the polynomial from given touple. Then, we will look at a special case today of a special p i gives you an interesting result. So, once again the algorithms also are identical.

(Refer Slide Time: 20:15)



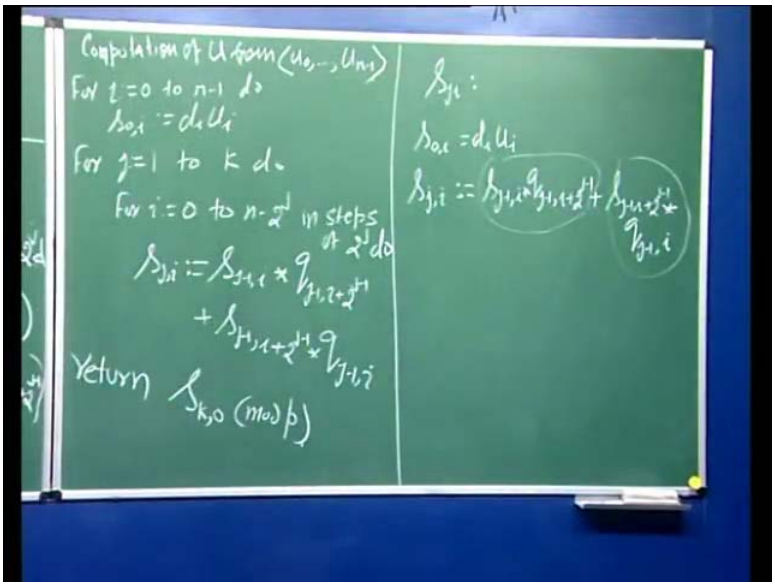
It will be, recall that we define q j i to be the product of 2 to the power these polynomials p i plus 2 power 0 minus 1. We define these terms for i equal to 0 and 2 power j, 2 times 2 power j and so on. Hence p was nothing but q p 0. I think we had used k let us say, k 0 where n was 2 power k. So, first we have to compute all these q i's, q, q j i's. For that we start, with p, sorry q 0 i are p i.

So, i would probably have to write down for 0 to n minus 1, do q 0 i is p i. Then, we will start computing q's of higher level that is the higher j's value. So, for j equal to 1 to k and for i equal to 0 to n minus 2, power j in steps of 2 power j. q j i is q j minus 1 i times q j minus 1 i plus 2 power j minus 1. That computes us the q i a j i's. Next, let us compute the end touple corresponding the polynomial u. Let the degree of u. So, I am computing u 2. Compute the touple for n where degree of u is less than the degree of p. So, we will start with u k 0 u i. Once again, we are going into the levels. We put this again into j loop and an i loop j equal to k to 1 do. For an i again, 0 to n minus 2 power j in steps of 2 power j.

Do this u j minus 1 i and u j minus i plus 2 power j minus 1. These are the two terms we have to compute. This is nothing but u j i and modulo with q j minus 1 i and u j minus 1 i plus 2 power j minus 1. This is u j i mod q j minus 1 i plus 2 power j minus 1 and determinate this loop we have used 0 i. So, we return. Then, we return the touple u 0, 0, u 0, 1, u 0 n minus 1. These are exactly the u 0 and u one and so on. Let me also right

down the computation of u from the n touple. Then, you look at the time complexity of each of these this time.

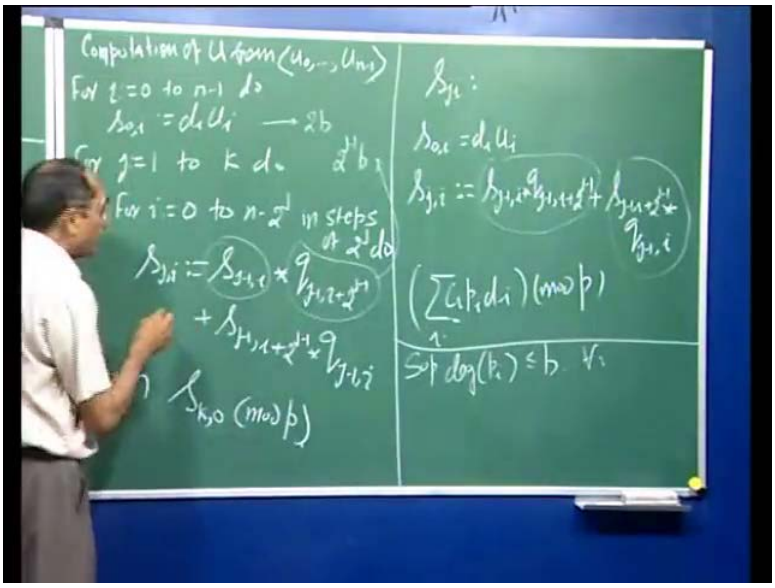
(Refer Slide Time: 26:17)



Once again, let me say computation of u from the touple u 0 to u n minus 1. Here, we are going to initialize. This is the reverse process that we have to do. You call that the way we compute is by computing d i's, but I am assuring that d i's are given to you. Then, we are going to compute the u. So, let us just recollect that we have S j i. The definition of j i was that s 0 i was nothing but d i u i. Then, the recursion S j i was S j minus 1 i q j minus 1 i plus 2 power j minus 1 and s j minus 1 i plus 2 power j minus 1. This is time this. Then, this is times q j minus 1 i. This product of plus this product of this is what we have to compute and differentiation.

So, we will start with initializing for i equal to 0 to n minus 1 S 0 i is d i u i. So, we have this. Therefore, j equal to 1 so, k and for i 0 to n minus 2 power j in steps of 2 power j. We are just going to write down that relation S j i is S j minus 1 i into q j minus 1 i plus 2 power j minus 1 plus s j minus 1 i plus 2 power j minus 1 into q. Return S k 0 mod p. The final polynomial that we get is S k 0.

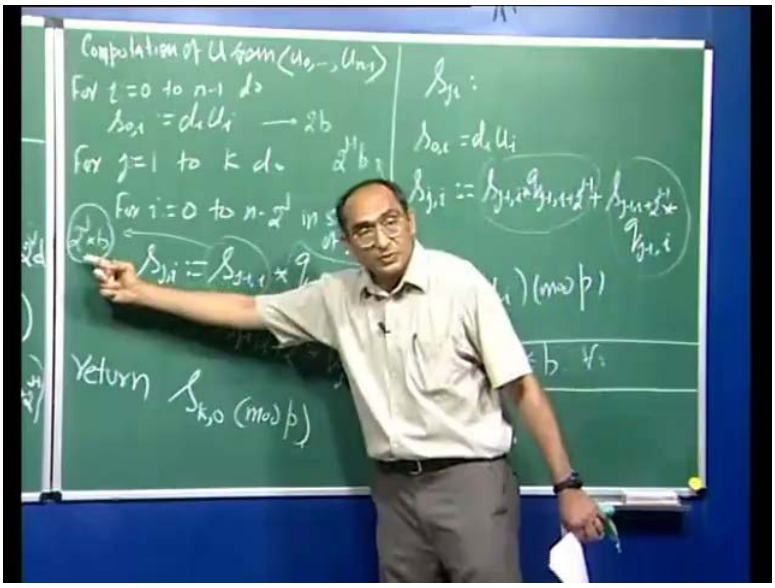
(Refer Slide Time: 30:02)



It is simply; S k 0 computed here is nothing but c i p i d I, but you also need to compute mod p of that. Hence, in the last lecture, we are computing the module p of this. Now, let us go through the algorithms for the time complexity. Degree of these polynomials will depend on the degrees of p i. So, let us suppose the degree of every p i is less than equal to b; for all whatever be the b.

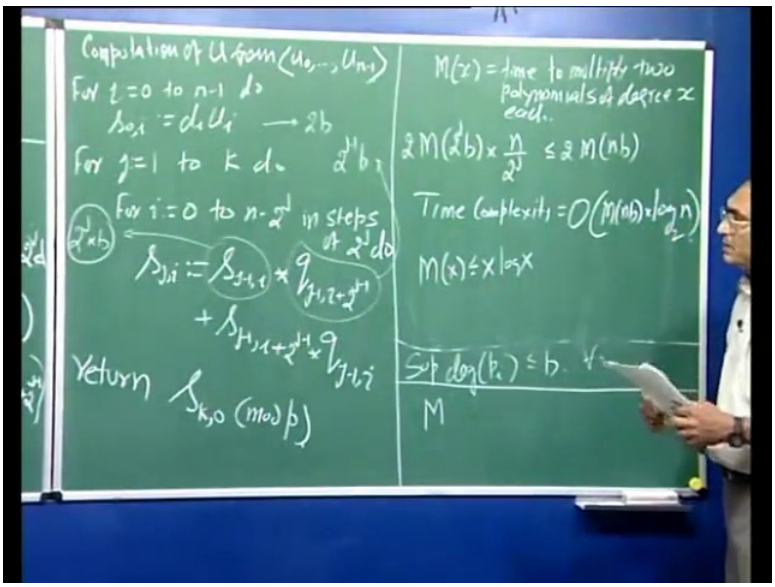
Then, this has degree 2 b because both of them have degree less than the degree of p i. Degree of these terms will be 2 to the power j minus 1 b 2 power j minus 1 b. There are 2 power j minus 1 terms multiplies here. That terms are p i's. It has degree at most b. So, many of them to this term; these are all of degrees for j minus 1. These recursively; one can establish have degree 2 power j. Degree of this is 2 power j time b.

(Refer Slide Time: 32:12)



I will multiply this by this. I have a polynomial of degree of 2 power j plus 1 times b. So, that is the argument one. This one of these computation involved this computation, involves one computation of size 2 power j time b m. I would probably clean this.

(Refer Slide Time: 32:55)

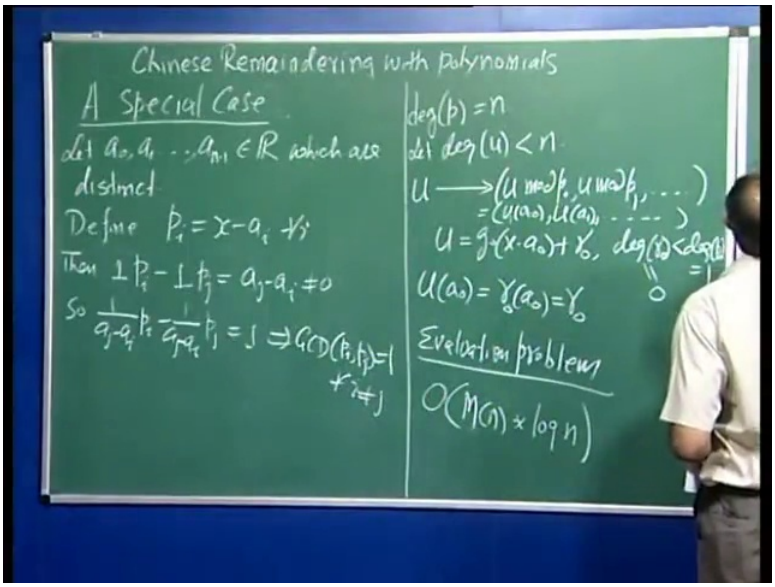


Denote by M x; the time to multiply two polynomials of degree x each. So, this step takes M 2 power 2 times 2 power j b. One of these steps takes and then, there are n by 2 power j. This one is less than equal to 2 M n b once again. This is at least linear. This is independent of j and we have k such that k is nothing but log of n. So, the total cost is

time complexity is order M n b into log n. This is what we take to compute. Assuming that the d i's are given as coprime, it is very easy to verify that the complexity of this algorithm.

The complexity of this algorithm is also the same; exactly the same is order M n b log M n. All of them cost us the same amount, in lecture later on we will show that M x is less than equal to x log x. Later on, we show this that indeed shows that the method of computing these more expensive than this. So, now, I am ready to discuss a special case. In this special case, I am going to take every specific set of p i and show the consequence of their computation. Now, we are coming to a special case of p i's.

(Refer Slide Time: 35:49)



Let a 0, a 1, a n minus 1 are some real numbers, which are distinct. Now, define p i as x minus a I, for all i than 1 times p i minus 1 times p j is a j a i. That is not 0 because these are distinct. So, 1 over a j minus a i times p i minus 1 over a j minus a i times p j is 1. This implies GCD of p i and p j is 1, for all i not equal to j. This is the condition we want to satisfy.

Indeed these p i's are pair wise coprime. Now, let us take this and find out what. First of all, notice that the degree of p is equal to exactly n because we have degree one polynomial. You multiply n of that to get the polynomial of degree n. Now, let degree of the polynomial is less than n. Then, the Chinese remainder theorem says that the image of u under this correspondence is u mod p 0, u mod p 1.

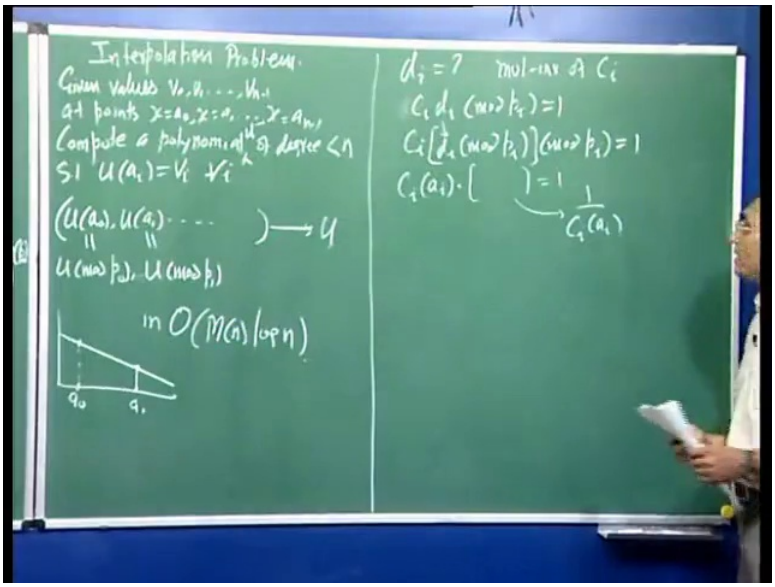
So, what are these? This is remainder of u and divided by x minus k 0. So, u divided by so, we are going to u equal to some g times x minus a 0 plus r. This is from division. Here now, we evaluate both sides at a 0. So, we get this polynomial, u evaluated at a 0. Here, I am going to get this r a 0. Now, notice that the degree of r has to be strictly less than the degree of this polynomial. Degree of r is less than the degree of p 0, which is what the degree of p 0 is 1.

Hence, this is 0. This means the r is a constant to evaluate r 0, a 0. It does not give anything else. It is same as r. So, all we notice is that the remainder, let me call it r 0. r 0 remainder is nothing but the value of u and a 0. Hence, this is same as u evaluated at a 0 u a 1. So, what we will get for this special choice of k i is the forward computation; will give you the value of u at n distinct point. The points are of your choice. You can choose your point. So, this is, I will call it as evaluate problem. So, what we see is that this problem of computing the value of n degree of polynomial at n distinct points; we can do this in order.

Notice, our p is 1 because, all our polynomials have degree 1 so, this takes M n in to log n time. As I have told you earlier, we will show that we can multiply two polynomials in n log n time. The entire time will be n log square n. This is an efficient algorithm because a brute force computation of algorithm computing the value of polynomial at each point will take order n time. Hence, computing will take order n square. So, this is more efficient.

The second problem is interpolation; the reverse one. What does it say? It says that given values let us say, v 0, v 1, v n minus 1 at points x equal to a 0, x equal to a 1, x equals to a n minus 1, compute a polynomial of degree less than n. That polynomial, let us call it u such that u and a i is v i for all i v. This is exactly the reverse problem. Now, we are given u at a 0, u at a 1, which is same as u modulo p 0 and so on. This would be u modulo p 1 and so on. Given this, we will give you a polynomial u of degree less than the degree of p, which is n.

(Refer Slide Time: 41:03)

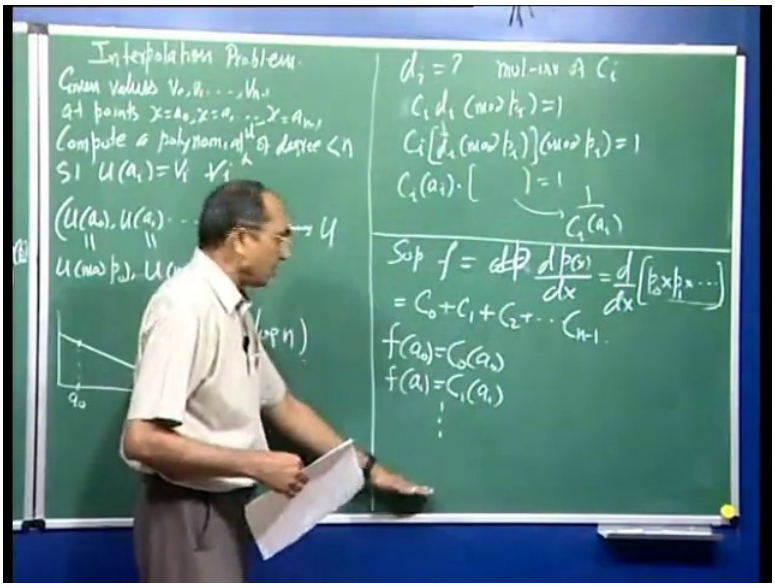


So, this has degree at most n minus 1. We know that if I have 2 points, a 1 and I know the values of a polynomial at 2 points. Then, I can give the polynomial degree 1. Newly, a straight line which will pass these 2 so, this is the interpolation this. In more general sense, we can compute in this fashion; again, in order M n log n, but there is one catch. Now, we have to figure out what are out d i because that is one unknown entity we did not discuss. For general k now, we have a specific p i's.

So, I now want to know what are d i's. How do I compute them? Once we know that then, we should be able to compute the polynomial from its value point. This is supposed to be a multiplicative inverse of c i. Furthermore, we will assume that the degree of this is less than the degree of p i. We can always do that. Our goal is that, sorry, c i d i mod p i should be 1.

So, I can always place this by d i mod p i. Take this polynomial. Do this and to p i. I will try to compute a constant. Notice, d i modulo p i is of degree p i. I i suppose to find a constant so that time c i is 1. Such modulo p i is 1, but this is same as c i at a i. When we compute modulo p I, we are computing at point p i that is the modulo is nothing but computing at that value a i. This time, this entity is 1. So this entity is nothing but 1 over c i a i. All we need to do is compute c i at a i. Now, that is very easy.

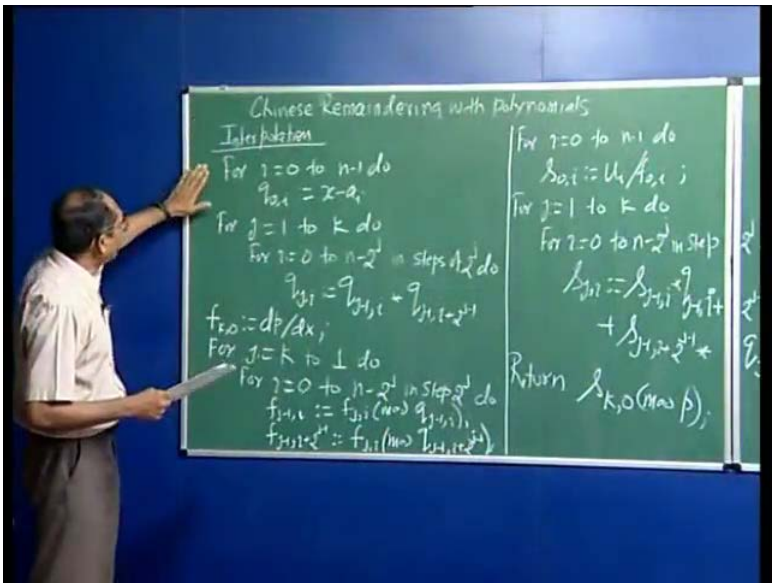
(Refer Slide Time: 46:04)



Now let us say suppose f is nothing but d u, sorry, d p. d p by d x. Notice that, p is the product of p i. This is same as d by d x of p 0 times p 1. So, if I differentiate by parts, when I differentiate irrespective one, I get p 0. This is nothing but x minus a 0. What is left; this part which is nothing but p 1. Hence, this is nothing but c 0 k. Multiplication of these steps comes as c 0 plus c 1 plus c 2 c n minus 1. Now, when you evaluate the f at a 0; f at a 0 is nothing but c 0 at a 0 c 1 has a p 0 in it. At a 0, it vanishes. Hence, all this term will vanish. Similarly, f of a 1 is nothing but c 1 at a 1.

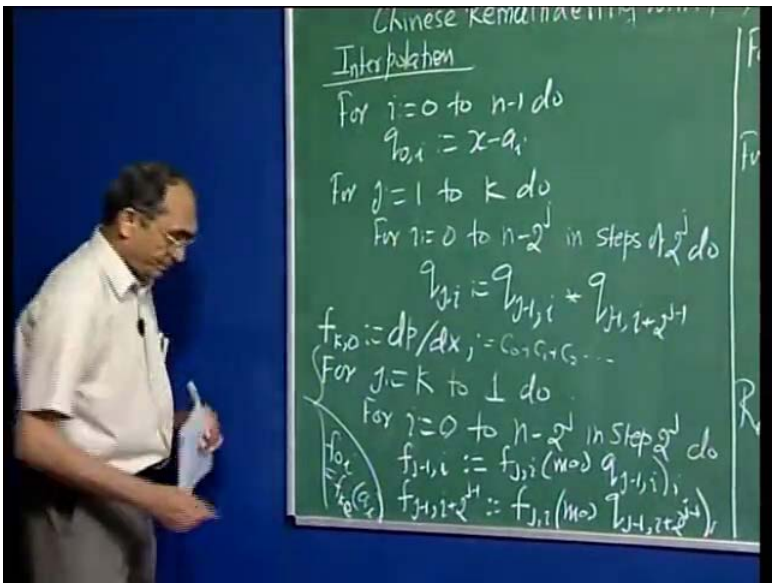
So, to compute c 0 at a 0 and c 1 at a 1 and so on, all we need to do is evaluate f at all these endpoints. Then, take their inverses reciprocals. Then, I am going to get my d i. So, all we have to do is compute f, determine its value at all these points; I have d i's. Then, I will go head and compute all u i. So, I am combining all the steps to compute the interpolation.

(Refer Slide Time: 48:22)



The first step is I initialize q 0 to p i's, the next step, I am computing all the j i. These are the same old steps here. I order to compute the d i's. As I discussed here, you first compute the derivative of p with respective x.

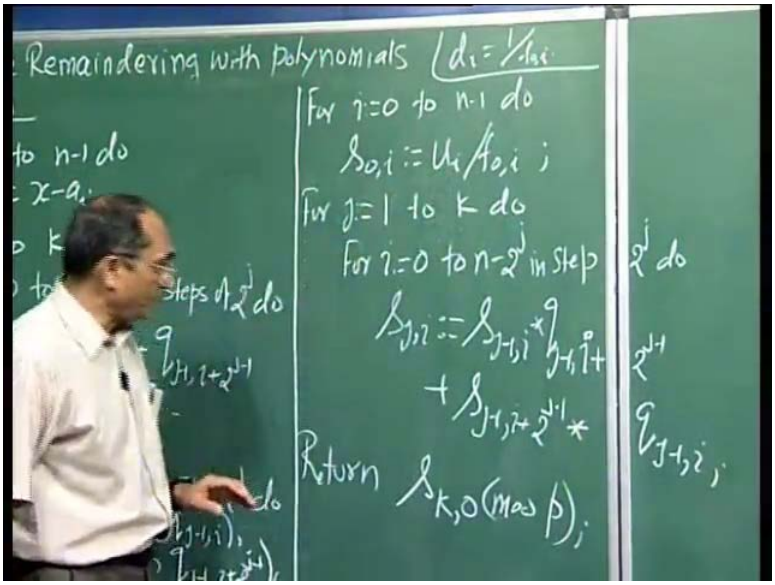
(Refer Slide Time: 48:50)



We know nothing but c 0 plus c 1 plus c 2. We have to compute this polynomial at a 0, a 1, a 2. Those will give us the d i's and their inverses. So now, I am here. I am computing the evaluation problem. This is where i started j equal to k at 1 in those steps f j minus 1 i is f j modulo q j minus 1 i and f j minus 1, i j minus 1 i j 2 power j minus 1 is f j i modulo

q j minus 1 i plus 2 j minus 1, but this is nothing but computing the f k 0 at point a 0, a 1, a 2 and a 3. Those will be nothing but f 0 i's. So, we have f. Note that f 0 i is nothing but k i at sorry, f k 0 at k i. Now, I have j 0 i, which recall nothing but 1 over d i.

(Refer Slide Time: 50:22)



d i is nothing but 1 over f 0, i. This is a constant. This is nothing but u i, d i. Once we have u i, d I, this is the same old step. It leads to at the end S k 0 and we compute modulo p. So, the final polynomial we get is of degree less than m. Its values at a i's are given to be; where is my a i, is this, u i. These are our value at various a i. This polynomial will evaluate to this u i. This is where we complete the discussion of Chinese remaindering theorem.