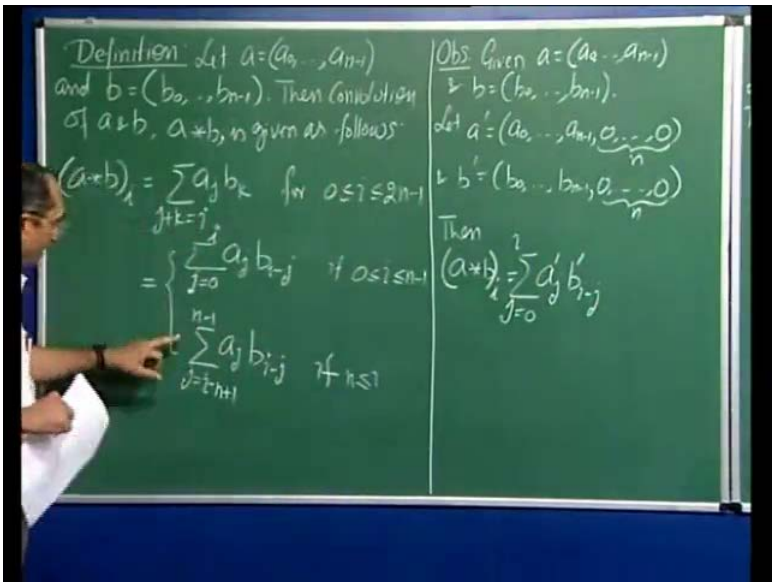
Computer Algorithms – 2
Prof. Dr. Shashank K. Metha
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 24
Approximation Algo III

Hello, in the last lecture we had introduced the discrete period transform and the other than the fast Fourier transform to compute d F t.

(Refer Slide Time: 00:28)



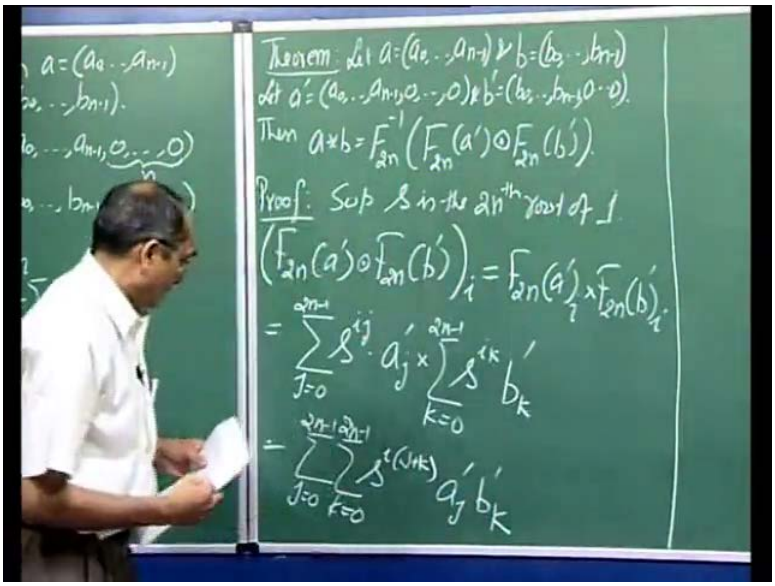
After that, we have introduced the notion of a convolution of 2 vectors, given 2 vectors of and components each the convolution denoted by a star b as define in the following fashion. If i a component of a star b was given as the sum of products of the components of a and b such that this is add up by 2 i. So, this was the original definition of the convolution. Here, shown that this expression can be broken down into this two parts, in the two cases when i index, i is between 0 and n minus 1 then this is simple a j b i minus j where j goes to 0 to i and when i is greater than n. When the difference between this two is let the j ranges from i minus n plus 1, 2 n minus 1, in this same sum is now take into this 2 behind an observation which show.

That this 2 sums can be than combined into this single sum of this kind they going from 0 to i, let the difference is, let they have to be redefine vectors 2 n length. Each a prime was original vector a followed by n zeros, b prime was originally similar with followed

by n zeros, so both of them are now 2 and length vectors. Then there up some of the some of the products of the terms a prime j and b prime i minus j they going from 0 to i gives you exactly this term, now one thing was to remember is let the convolution is actually a vector for length 2 n minus 1.

Precisely the reason is that there are only 2 n minus 1 possible ways we can have this values when both of them are 0, so i can be 0 and both in this 0 larges they can have n minus 1 plus n minus 1. So, 2 n minus 2, but we actually extended to 2 minus 1 by really finding the value 0 only, so we will thing that this as vector of length to 2 n. Here, we have extended these two base vectors to 2 n and then we find the following expressions for the convolution.

(Refer Slide Time: 03:46)



The last thing we did was state stated this theory which is a way to compute the convolution using discrete Fourier a transform and inverse discrete Fourier transform. So, let says through this is again saying that they have a and b which are further extended to this to a length by studying n zeros here and n here zeros then we take discrete Fourier transform of both of them. So, this are again vectors of length to an end, each this symbol indicates let your multiplying by 0 component to 0 component that could be the 0 component of this product.

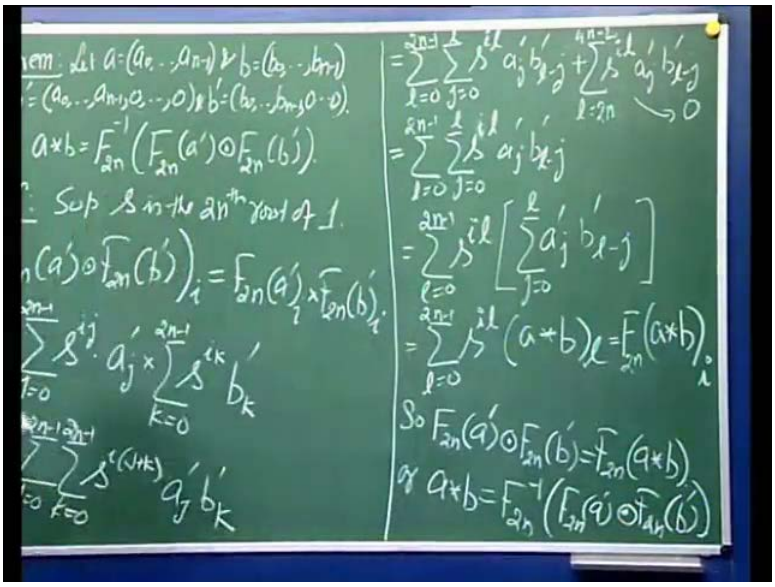
First component to the first component will be the first component of this product and so on, this component wise multiplication and this is again as vector of length to an end.

Then you compute the inverse discrete Fourier transform and the claim is that that is precisely the convolution of the two original vectors. So, let us go through proof the definition of this we have already shown is this product my mistake, so we have to talk about the discrete Fourier transform of 2.

So, let us take the i-th component of the product which is F 2 n a prime F 2 n b prime the i-th component of this as we mention this is the nothing but point 2 point multiplication. So, this is nothing but F 2 n a prime i into F 2 n b prime i, now we called that discrete Fourier transform of a prime was assuming that the w t denotes or rather we will use that the another symbol F, let suppose F is the 2 n F primitive root of unit.

So, we will have s power i j times a prime j and j is going from 0 to n minus 1, this times another sum s i K they going from 0 to 2 n minus 1, b prime K and then write down suppose s is the 2 n-th root of unity. So, that is what s denotes we can combine this together, they going from 2 n minus 1, here also changing from 0 to 2 n minus 1 s i, j plus K a prime a, j b prime K, so let me define a new variable which denotes a plus K as l.

(Refer Slide Time: 07:47)



So, that would be some l j plus K will go from 0 to 4 n minus 2, but I am again going to extended 4 n minus 1 setting at the last element of 0, so that under the split into 2 sums. First, from 0 to 2 n minus 1 j from 0 to l s power i l this is i, this is l, a prime j, b prime l minus j plus the remaining range of l which is 2 n minus 2 n took fine. Now, let us write

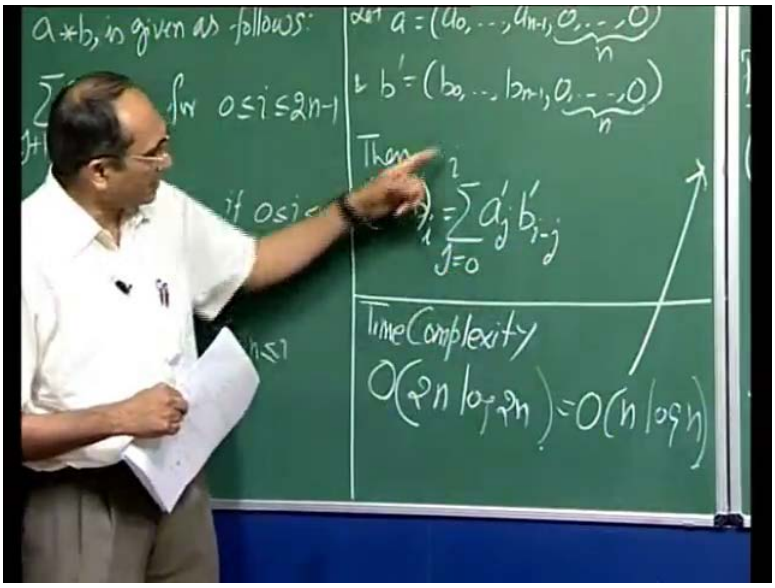
down 4 n minus 2 that is the largest possible value of our variable l s, i l, a prime j b prime l minus j, so let us look at the second term, the sum of the 2 in this is l, and that is at least 2 n.

Hence, at least one of this is has to be n or more and what we notice it that the value of our vector behind n minus is 0 both of this 0, so this products will always vanish. So, this will be 0 and we can only look upon this slide, so we had, now l going from 0 to 2 n minus 1, sum l 0 to l s, i l a prime j, b prime l minus j let me take this sum an you have l 0, 2 n minus 1 s forward i l sum 0 to l, a prime j b prime l minus j. Now, let us put us on this equation, instead of i if I plug in l here then this expression is say as this expression, so what we are seeing here is the l-th component of the convolution of a and b.

So, this is 2 l from 0 to 2 l minus 1, s i l and that is lesser, but the Fourier transform of this vector, notice that this is at 2 n, length vector and s is 2 n root of primitive the 2 n root of primate s i l, hence this precisely the i-th component of Fourier transform of a star b. So, we have F 2 n to a prime 3 n b prime equal to F 2 n, a star b that is an i-th component of this equal 2 i-th component of this, hence the 2 vectors are equal. So, we can say if we take inverse transform on both sides we are going to get a star b the right hand side will be a star b and this side will be F 2 n inverse of F 2 n a prime F 2 n b prime.

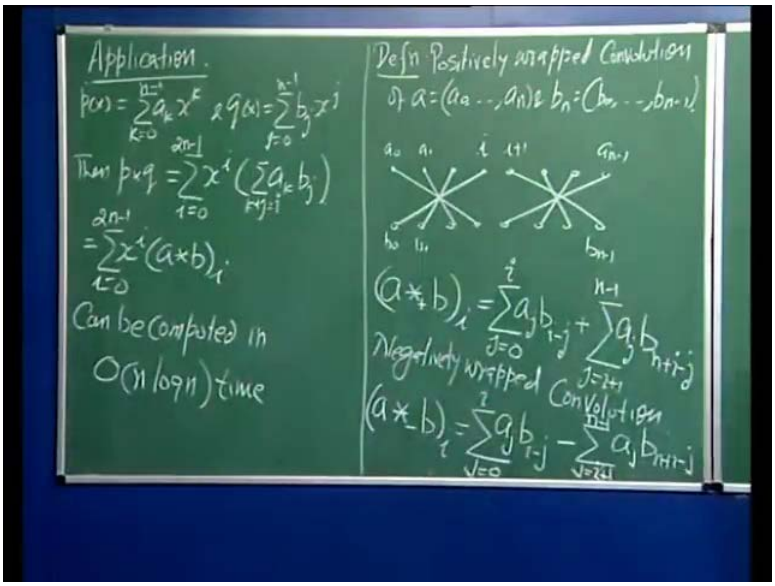
So, that is what the claim was and this let me compute the convolution, by computing two transforms and one inverse transforms we call that the complexity of computing Fourier transform was n log n. When this case it is 2 n log of 2 n which is same as order n log n, so the total cost of, so further time complexity of this the time complexity is computation of three Fourier transformer. So, remember inverse transform is also a transform by just taking this suitable root, here we will get the universal transform. So, three operations of the complexity, n log n and over here we have this 2 end multiplication, but 2 and multiplication take time 2 n because we are assuming that all basic arrangement operations take unique time throughout.

(Refer Slide Time: 15:26)



So, total cost is nothing but 2 n log of 2 n times 3 given and this is same as order n log of n, this is a very efficient way to compute the convolution because if you try to do in a group force manner, then you are going to perform one multiplication for each of this component. So, there are somewhere around 0 to 2 n defined multiplications, so almost n square multiplications is not almost precise n square multiplications. So, the brute force also become it takes order n square times.

(Refer Slide Time: 16:23)



Now, let us take an application of convolution, let suppose we have 2 polynomials p of x which is sum a i x power i, i going through 0 to n minus 1 and q x this sum b j x power 0 j going through 0 to n minus 1. Then the product p times q the coefficient of x k just to align myself with the efficient let me use another index, so let say the index of the, sorry coefficient x power i in the product would be off course i will go from 0 to 2 n minus 2. But, again let it go to 1, here we are going to have the sum of all the products using this add up to i, so I will write down sum a k b j let K plus j is i this is exactly be the result of multiplying this two polynomials.

But, this is exactly play the convolution, so what we have here is x i a star b i, so what we notice here that to compute the product of two polynomials is essential computing the convolution that. Hence, the product of two polynomials which are given expressive that is to polynomials as given coefficient of each of the monomials than the product can be computed in order n log n time.

So, this is in the efficient logarithm to compute the product of the polynomials and as we just see a group force method, we take order n square log n time. Now, I am going to describe another form of convolution which will be, hence in some other applications and that is we describe to such convolutions. The first one is we call positively wrapped convolution of, again we have this an component vectors, now in this case what we do is let me just visually scribe this. So, we have this a 0, a 1, a n minus 1, b 0, b 1, to b n minus 1 suppose this is i, then the i-th component in normal convolution the multiply this two and we multiply this two and this two and this 2 and we had the 1.

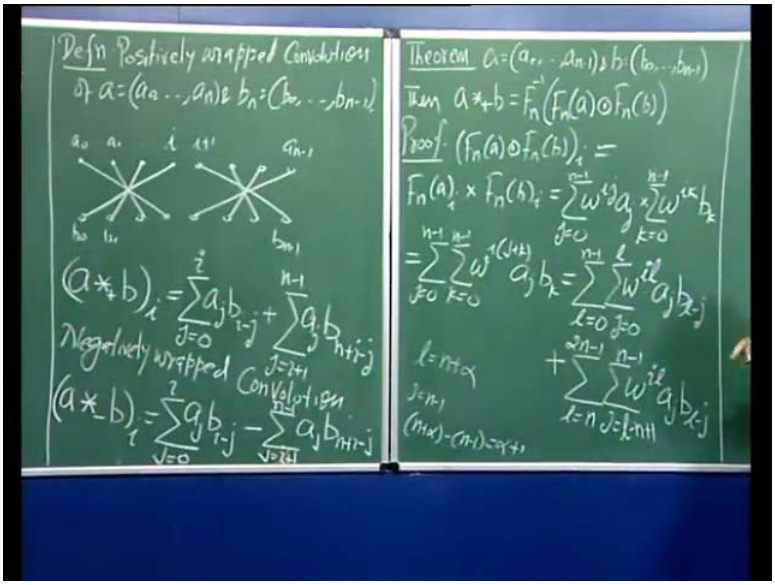
But, in the positively wrapped convolution we continue this is index i plus 1, so this should be multiply with a minus 1 that is index minus 1 which can wrapped is this. So, this will be multiplied with this and this will go by this and this will go by this, so we will have actually several terms together i-th component of it. Now, I have a complicated notation this indication positively wrapped convolution of a and b the i-th component is I am going to write down equation in this 2 terms separately. We will have a j b i minus j, well first off all we keep in mind this unlike the regular convolution is only an n component vector there are only one components.

So, i only ranges to 0 to b n minus 1, j will go from 0 to i, 0, 1, 2, 3 up to i and then the second term we had j going from i plus 1 to n minus 1, i plus 1 to n minus 1 a j b. Now,

the index for b is negative, now all we done is we added n to it, so it is n plus i minus j, so this comes for this terms similarly, now we can define negatively wrapped convolution, this point is make sure n plus i minus j. So, negatively wrapped convolution is exactly the same thing, but the convolution is minus side, so that we will expresses a star minus b, i is j going from 0 to i a j b i minus j minus j going from l, sorry i plus 1 i plus 1, 2 n minus 1.

So, the ranges says this is the same term m going to write this is the same term as this only difference we have the minus signs a j b n plus i minus j, now our objective is to compute this using this Fourier transforms. So, that is what we will do next and what am going to do is effectively use the logarithm for this to compute this, so is to compute this and then using that will get this one, now just let me take the claim.

(Refer Slide Time: 25:20)



Here, theorem which says that a is a 0 through a n minus 1 and b is b 0 through b n minus 1 then a star plus b is F inverse n F n of a everything look like similar. But, this kind of a dealing with the original vector themselves not the extended vectors we just taking the regular Fourier transforms of a that of b. Then we do point to point multiplication and then take to inverse Fourier transform and this we are calming nothing but the positively wrapped convolution of the two vectors.

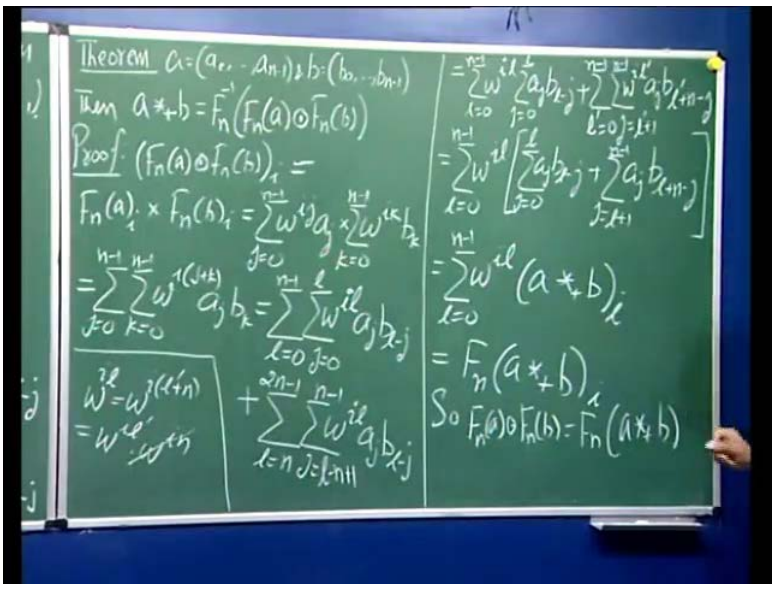
So, let us see how is that once again we going to follow the same steps am going to take the i-th component of this F n a F n b i is F n a i into F n b i that. Now, I am going to use

W denote the n-th primitive proof of the one, so that would be j from 0 to n minus 1 W i j a j and similar terms K 0 to n minus 1 W i k b k again it play the same trick i will combined this two, j from 0 to n minus 1 K 0 to n minus 1 a j b k. Again we going to find variable j plus, k so let say l denotes the variable, now that variable will have range comes 0 to n minus 1 last term we have.

So, let us write down l and I think am going to write them into two parts 0, 2 n minus 1 first this is j 0 to l W i l is a j b l minus j a j b l minus j and then we have second term lets write the term that plus. So, let us say, now l goes across from n to 2 n minus 1 this kind, suppose l lets say l n plus alpha i is 0 on words this is the range then j at the highest j will be n minus 1. At the upper most limit because we have the actually vectors of only the n components, so j can be at most of n minus 1 at this point at the lowest it will be it will be lowest when k becomes the second index this becomes n minus 1.

So, that would be n plus alpha minus n minus 1 which is alpha plus 1, now the range of j which is alpha plus 1 to n minus 1 what is alpha is n min is l. So, I will take n minus l plus 1 to n minus 1 and W i l a j b, now l minus j, so this what we get by substituting variable l for j plus K. Let me verify I did make 1 mistake a were alpha is l minus n not in l minus n, now to put l minus n here l minus n plus 1, so let us on the right hand side I can continue I can again put in this here.

(Refer Slide Time: 32:06)



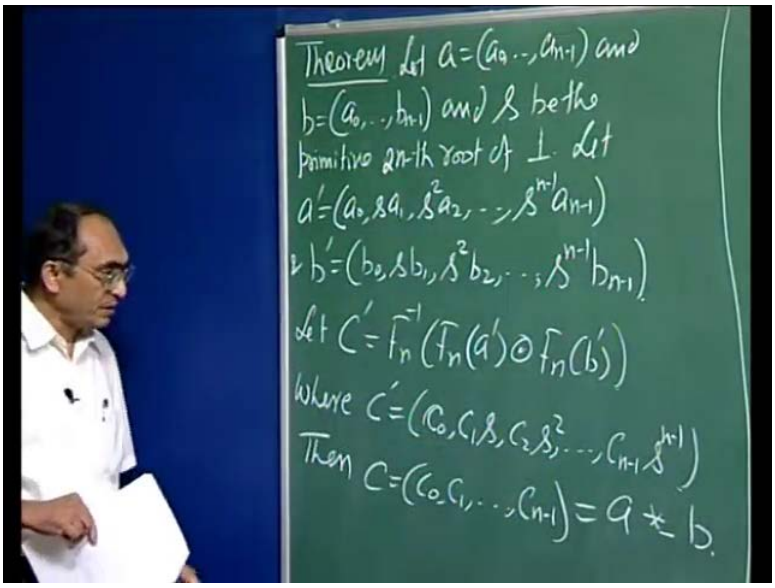
To get sum l from 0 to n minus 1 will sum j going from 0 to l a j b l minus j plus, now look at this expression let me substitute for l another variable l prime which stands for l minus n, n is a constant. So, I am going to write everything in consign assuming that l prime is l minus n, so this is l prime which takes from 0 because l is n to 2 n minus 1, so this will go from go to n minus 1 j l prime plus 1 to n minus 1, W time a power i takes l plus i prime. Now, let just we look at this w power i l is W power i l prime plus n which is W power i, l prime in to W power i n, notice that W power n is 1, so W power is i, n is also 1.

So, this goes away this is gone and this is just i l prime I am going to write w i l prime a j and b l prime plus n minus j, now say l prime is a demo variable, I can replace all again by variable. So, this is l 0 to n minus 1 if I replace l prime by l again notice says that l is different to the old this is just l prime re written to itself, so we are going to get the same term this ranges same as this. So, I can combined the 2 and I can write down this, W i l and, now am going to put this terms as j 0 to l a j b l minus j plus sum j going to l plus 1 to n minus 1 a j b l plus n minus j.

Now, we have an expression for positively wrapped convolution, notice that this expression is same as expression we have calculated there this is same as this. So, we are going to get sum W i l, l going from 0 to n minus 1 a positively wrapped convolution of the n b l and that makes it F n of a star plus b i-th component. So, we get the combined 2 and get F n a, F n b is F n of a star plus b, once again all we do is take F n inverse of both sides and you get here design.

Now, finally here we going to state how to compute the negatively wrapped convolution as I had mentioned earlier we are going to use this particular method. But, we will have to we will have a trick, here to able to introduce the minus sign for the second term of the convolution, so what we are going to do is that following.

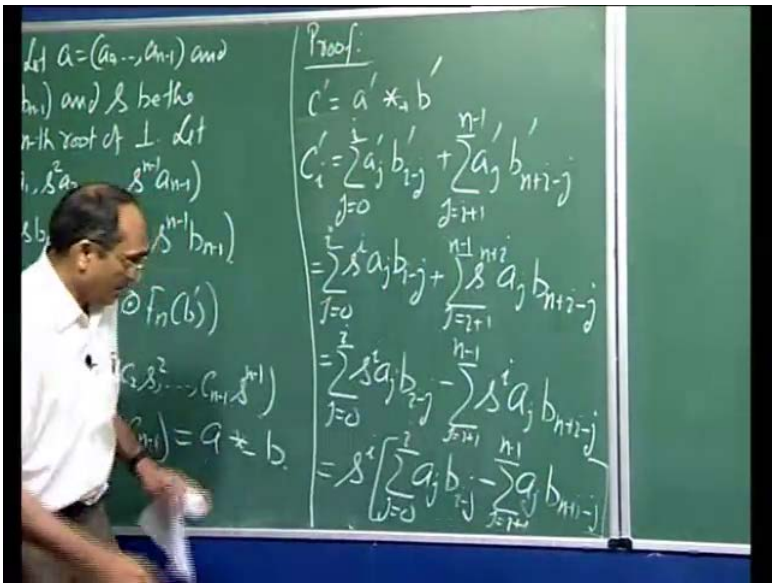
(Refer Slide Time: 37:23)



Let us just take the claim in the form of this theorem, so let a be as this 0 to a n minus 1 and b is also n component to this vector and s be the be the primitive 2 n-th root of 1. Now, define two vectors a prime as a 0 s a 1, s square a 2, s n minus 1, a n minus 1 and b prime, similarly b 0, s b 1, s square b 2, s power n minus 1, b n minus 1. So, instead of dealing this vectors we are inserted this powers of s, increasing powers of s with each component. Now, we claim in that let C prime be the F inverse of F of a prime this is the same computation, only the vectors if changed and let says we are going to get where C prime is let us say C 0 to C 1 s, C 2 s square, C n minus 1 s n minus 1.

So, this vector in this form then our claim is C which is nothing but C 0, C 1 to C n minus 1 is nothing but the negatively wrapped convolution of and b. So, all we have to do modified the original vectors into this 2 vectors, do the computation of that we would do that positively wrapped convolution. Then if I divide the respectively component by power of s, I am going to get the negatively wrapped convolution of the two original vectors a and b. Now, let us take the what exactly C prime is C prime is nothing but positively wrapped convolution of a prime and b prime that is what we have proven earlier.

(Refer Slide Time: 41:19)

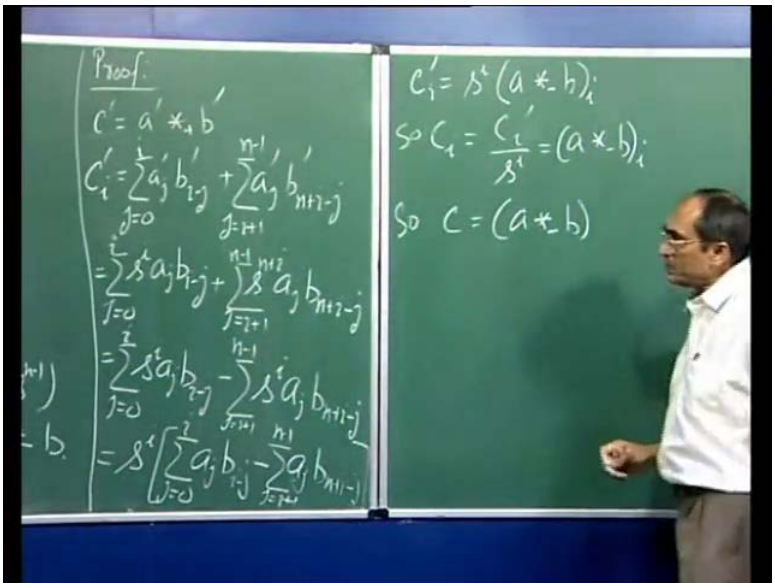


So, let just look at proof what we are getting C prime is a prime positively wrapped convolution of prime and b prime, so let us write down the expression for that which take sum expressed into two parts where our i goes. So, I need to write down the i-th component C prime, i is sum j going from 0 to i, a prime j b prime i minus j plus j going from.

So, let just copy that the range for the second part is i plus 1 to n a prime j b prime n plus i minus j, so we have this the prime variable components are nothing but, ahh the corresponding s values is 0 we can plug in here we are going to get, so we have s power j and i minus j. So, this is s power i a j b i minus j i, sorry j going from 0 to i plus sum, now this time j to behave s this contributing of power j and this is contributing s power n plus i minus j, so we have n plus i.

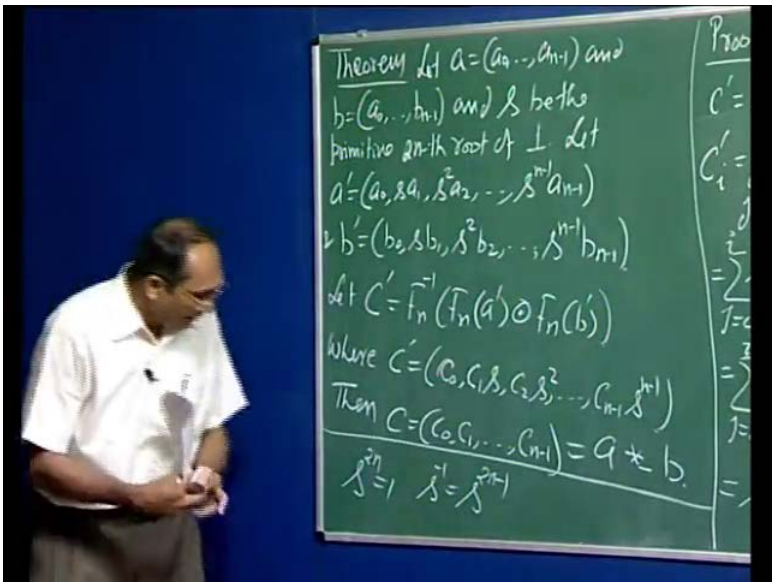
Here, a j b and n plus i minus j and you notice that we have an s power n which is minus 1, so i can, now write down this it has sum s i a, j b it minus j minus sum s i j goes from 0 to i and j goes from i plus n minus 1. Here, n minus 1, here also a j b n plus i minus j, so we have this, now we take s n common and we are getting s i sum j going from 0 to i a j b i minus j minus sum again j goes from i plus 1. So, j from i plus 1 to n minus 1 a j b n plus i minus j, now look back at the definition of the negatively wrapped convolution is this precisely the i-th component of the negatively wrapped convolution of the at the.

(Refer Slide Time: 45:56)



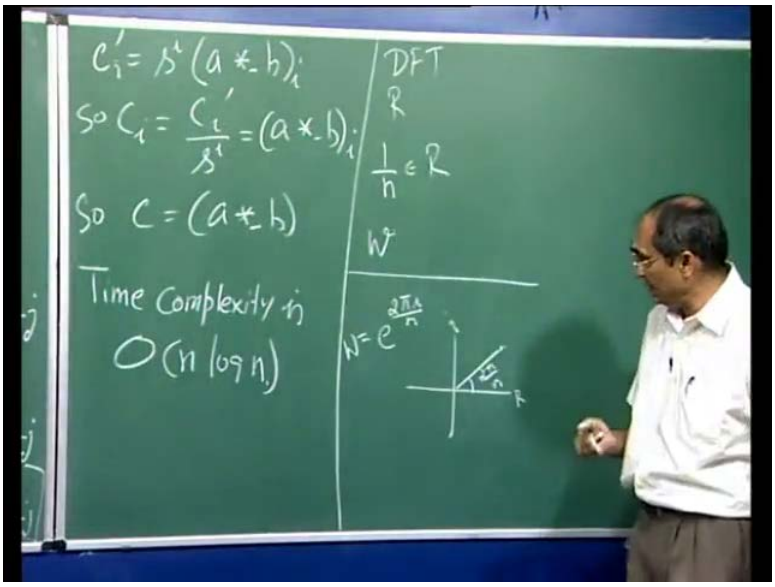
So, we are getting C i prime as s i and a star minus b i, now we have defined our C components of C prime component of the divided by power, so C i this C i prime by s power i. Hence, this is nothing but a star minus b i, so what we have formed is that the vector C is precisely a star minus b, now how much time would take to compute this to compute the powers of all s takes order n time in each step i could take the next power. So, I got this, I can then in order, then to multiply them to individual components, so this is order n otherwise we have just 3 Fourier transform to compute over here we have to divide by s.

(Refer Slide Time: 47:35)



Now just we remark there by division by s, notice that s power 2 n is 1, hence division is s like, but multiplying but s power by minus 1 which is s minus is nothing but s power 2 n minus 1. So, if I have a s power 2 n minus 1 then if I have just multiply successive power of that the term, so each of this get read of this, so that takes order n times again, so the entire process takes i order than plus order log n.

(Refer Slide Time: 48:19)



So, the time complexity is again order n log n, so we compute all the 3 convolutions in this same time complexity I had given the application of the regular convolutions in computing the product of two polynomials. We will see an application of the this convolutions in the next lecture, but before we will talk about next lecture I would like to make some remark about the range in which Fourier transform can be computed. We call that, we said that Fourier transform, discreet Fourier transform can be computed in a range r which satisfies to conditions besides being a range competitive way being done.

What we wanted was one over n belong to range and to there should be a primitive n F root of unity primitive n F root of unity, but if you notice our computations of we also needed primitive 2 n F root of unity sometimes. So, exactly we have a such a range, so one of the range obviously nothing but the field of complex numbers in complex numbers we have e to the power 2 pi i 4 n.

In terms of the picture this is real axis, this is imaginary axis and this angle is 2 pi by n, so this is the n F root of unity and it is primitive n F root of unity because when you

multiply it with itself will make the twice the angle and so on. So, the n F root the n F power of that will take to this before that it will be something else will never come back to unity before that, so this will be the w you can use because complex numbers form of field one over end is all of is here. So, both these conditions are satisfied and we can compute the Fourier transform, discreet Fourier transform and the main, but sometimes we are interested in.

Notice that we working with complex numbers means there is always problem of precision we use certain precision computations are not going to be exact. So, we are sometime interested working with intelligent, hence what we will do next time is describe another range where we can work with the intelligent laws of any precisions provided. In the large ring we will have range of ranges enough range of infinite ranges and then we will actually discuss the Fourier transform in that.