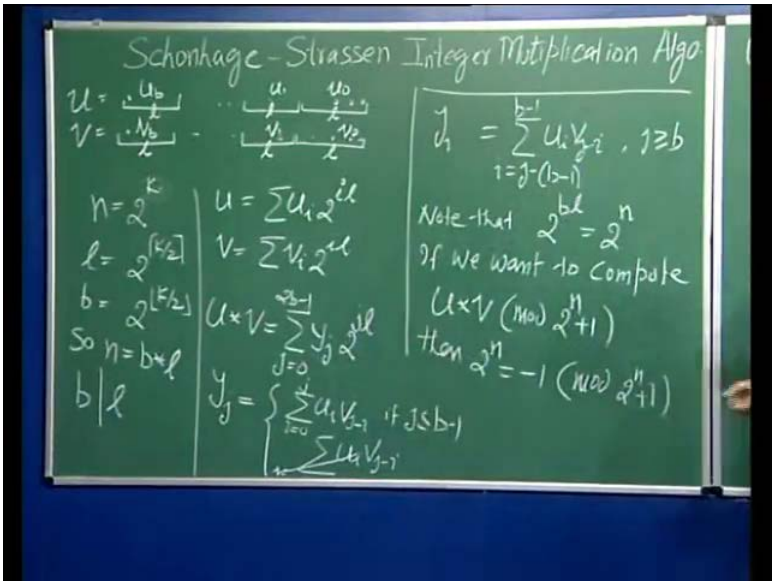
Computers Algorithms-2
Prof. Dr. Shashank K. Mehta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 26
Schonhage Strassen Algo

Hello today, we will discuss the famous Schonhage Strassen integer multiplication Algo.

(Refer Slide Time: 00:25)



Let suppose, we have two integers; u and v each has n bits. So, we have n bits, we have n bits in this one and n is a power of 2. So, we will first define say n is 2 power K, then I am going to define l as 2 power K by 2 sealing and b as 2 power K by 2 flow. So, n is b times l, one more thing you may want to notice is that b divides l; of course, b n l both divide n, but also divides l, because of this time. We will now partition this into groups of l bits and the number represented by these l bits will be let us say u 0, u 1, and so on. So, it would be u 0, because there are d groups here, simply because n is b times l. similarly, we will have groups here and the corresponding numbers will be v 0, v 1, v b.

Now, this gives me u as a sum of u i 2 to the power i l, the reason is that the place value of this is 2 to the power l. So, each of these will have weight age of 2 power i l for respective i and v is v i 2 to the power i l. So, u times v, this number will be sum y j, notice that we will be accumulating all the coefficients for a given power of 2 and those

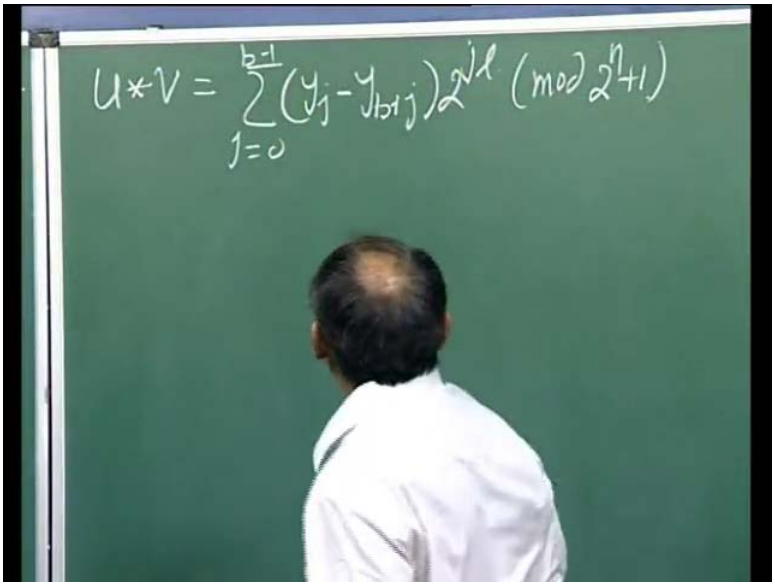
coefficients will be y j and each of those coefficients will be product of one of these and one of these, each of this numbers is at most 2 to the power l minus 1.

So, now, this will look like 2 to the power j l where, j will go from 0 to 2 b minus because the largest one of course, will be 0 and the non zero term the largest non zero term will be corresponding to j equal to 2 b minus 2. Now, let us look at what this y j is. So, y j is depending on whether j is less than equal to b minus 1 or greater than equal to b will be sum u i v j minus i for i going from 0 to j.

If j is less than or equal to b minus 1 and in case it is greater than b minus 1, we will have the sum where, again this will be u i v j minus i, the sum will go from j sorry i, it will go from j minus so maybe I will write here. So, y j will be some u i v j minus i for j greater than or equal to v i at most it will go to b minus 1 and at the other end this will be j minus b minus 1. So, this is where, this is what y j's are…

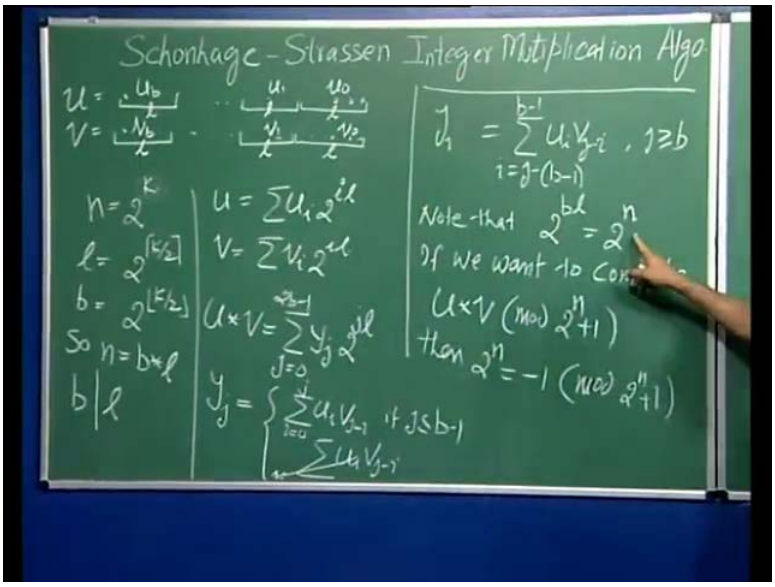
Now, notice that, note that 2 to the power when, j is b or more, we will have exponent to 2th power j l, which is b l plus something, but 2 to the power b l will be 2 to the power n. So, if we have a j equal to b or more than 2th power b l term will be common factor. So, those terms and this is nothing but 2 to the power n. Now, if we want to compute u times b v modulo 2 to the power n plus 1, then 2 to the power n is minus 1 modulo 2th power n plus 1 and that case this will as good as minus 1.

(Refer Slide Time: 07:10)



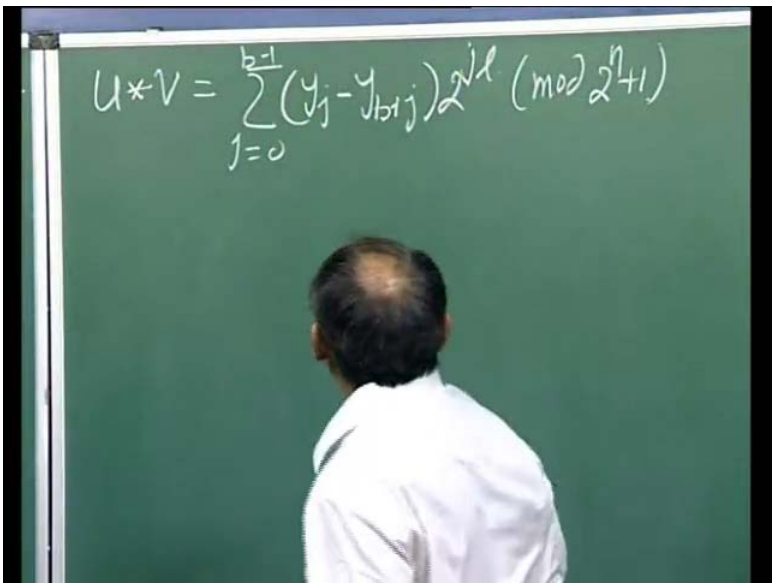
So, what, we can now say is that u times v will be sum y j minus y b plus j 2 to the power j l, when j goes some 0 to b minus 1 modulo 2 power n plus 1.

(Refer Slide Time: 07:42)



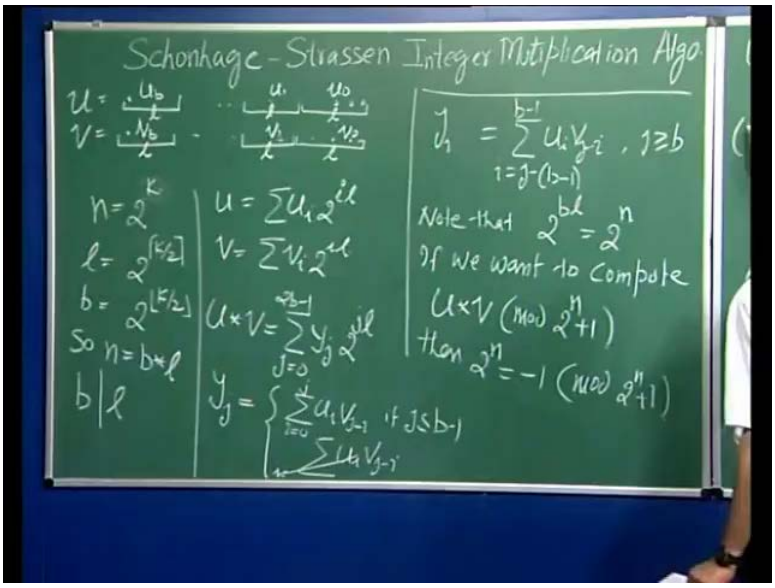
The reason as you notice is that for the larger j's, in this sum, we will get, if j is let us say b plus j prime, then this term becomes 2 to the power n times 2th power j prime l and 2 to the power n will be minus 1 here. So, the second half of this sum will have a minus sign.

(Refer Slide Time: 08:09)



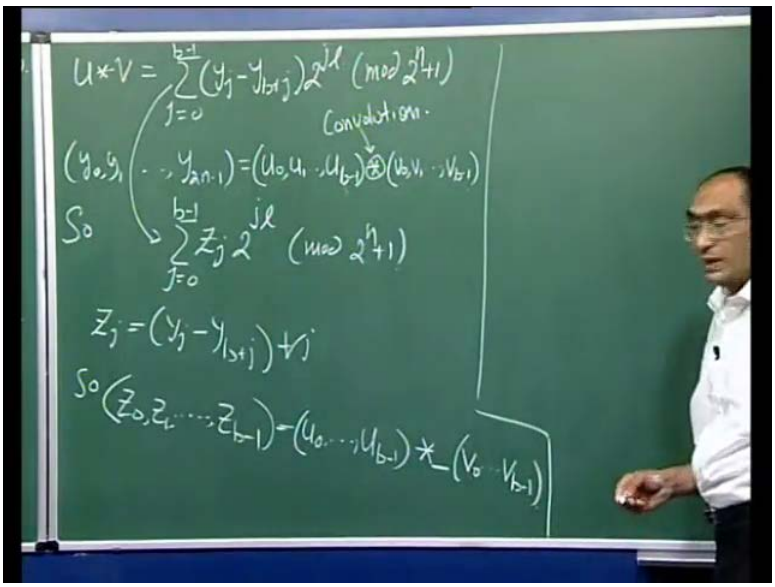
And we will have only js running from 0 to b minus 1.

(Refer Slide Time: 08:15)



Now, let us go back and look at y's are...

(Refer Slide Time: 08:18)

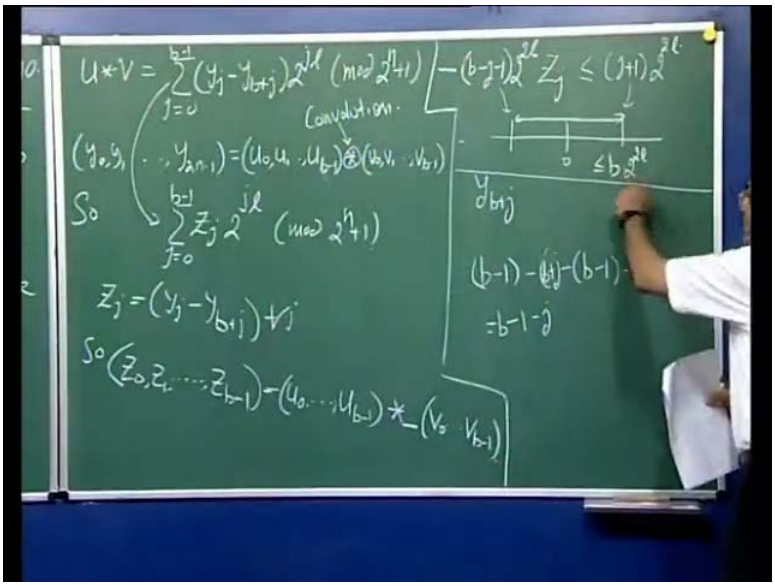


So, the vector y 0 y 1 y 2 n minus 1 is precisely the convolution of the 2 vectors is nothing but the convolution of vector u 0 u 1 u n minus b minus 1. So, v 0 v 1 v b minus 1, this is the convolution, this is the definition that, we gave for y, where each y is the component of the convolution of these 2 vectors. And so what, we see here is that, if we write this as sum j going from 0 to b minus 1 z j 2 to the power j l for mod 2 to the power

n plus 1 then z j is y j minus y b plus j for all j. So, the vector z 0 z 1 z d minus 1 is precisely, the negatively rapped convolution of this 2 vectors.

So, that is u 0 through u b minus 1 with a minus sign v 0 through v b minus 1. So, the moral of this story is up to this point is that instead of computing u times v, if we are interested in computing u times v modulo 2th power n plus 1 then all, we need to do is compute the negatively rapped convolution of these 2 vectors. And these 2 vectors are precisely the components of those vectors are these numbers. So, now for the rest of the lecture, we are going to find a way efficient way to compute these z's or the components of this convolutions negatively rapped convolution. Now, before, we do that let us try to pin down the range of the values that z can take.

(Refer Slide Time: 11:08)



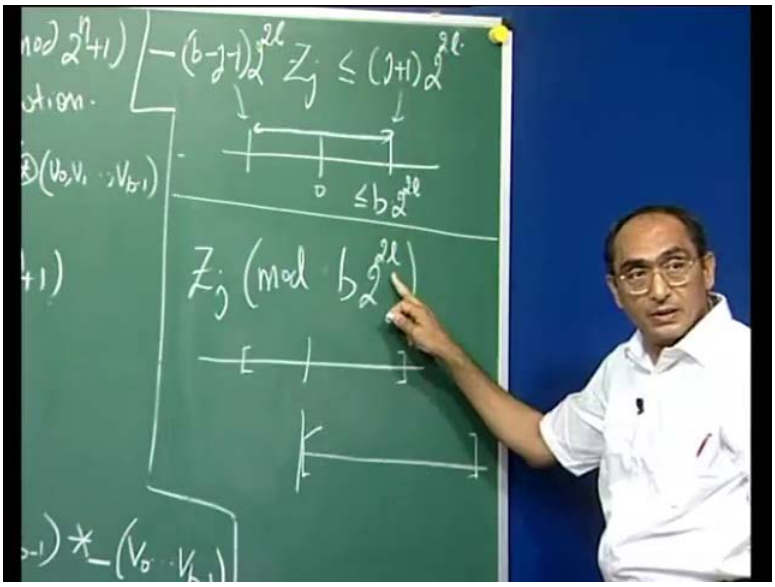
So, the maximum value of z j notice that, that will come for the maximum value of y j, which from the expression of y j, here we have j plus 1 terms for y j and each term has maximum value of 2 power 2 l, because each of them is 2 power l. So, the largest value that this can capture is j plus 1 power 2 power 2 n, on the other hand the smallest value of this is sorry, of j z is the is due to the largest value of this term.

So, y b plus j is given by this expression and here this j is b plus j and the range the number of terms in this will be b minus 1 minus j minus b minus 1 minus 1 and this j has to be replaced by b plus j. So, I will write down b plus j, this is now minus 1 b cancels

out sorry, b minus 1 and we have minus j. So, the maximum value for this is b minus j minus 1 into 2 to the power 2 l with a minus sign here.

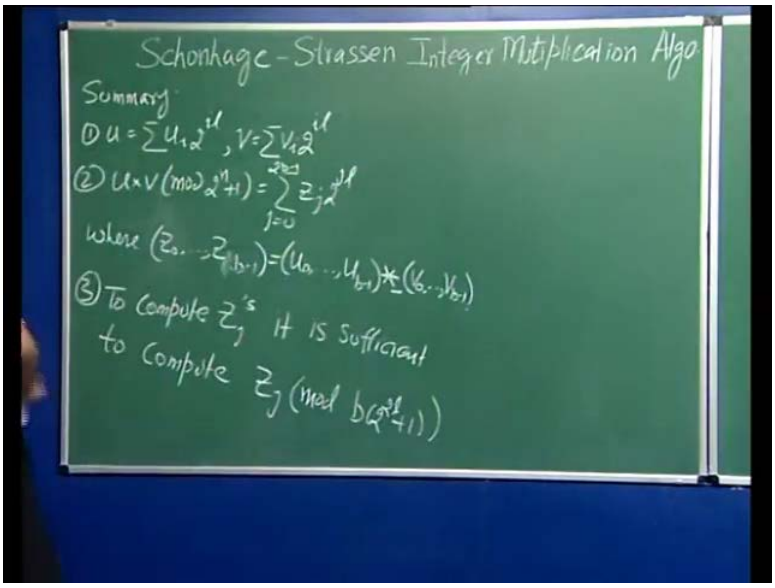
So, the maximum range, that the entire range of this z j, if this is the real axis and this is 0 then if this point is this and this is this point that the total range in, which the value varies of course, this range may vary with j. But, the total length of this range is, if you cancel out the plus 1 is less than equal to b times 2 to the power 2 n. So, that the entire each z can vary only in this range, now that tells us, that in case, we compute.

(Refer Slide Time: 14:06)



Instead of computing z j, we will compute this modulo b 2 to the power 2 l, if we do this, then we will not lose any information the only thing that will happen is that z j, which is ranging in this space will now be will occur somewhere in here. Because, this is modulo b times 2 power 2 l. So, the moment it exceeds this number, if our z j modulo this exceeds, this value what we have to do is subtract b 2th power 2 l from that and we will get the exact value as per this range.

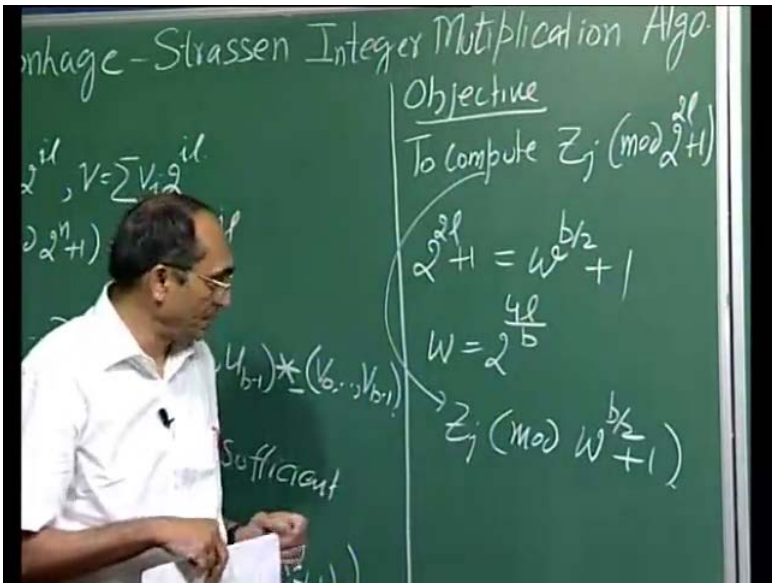
(Refer Slide Time: 15:00)



So, let me quickly summarize, what we have done is taken the n bit numbers u and v and broken them down into the blocks of l bits l and b, we have defined earlier and then this number looks like this. We have also shown that u times v modulo 2 power n plus 1 then looks like this sum where z, this is not. So, this is just b component vector and this is nothing but negatively rapped convolution of this vector of u i is and this vector of v i is both of them are b component vectors. And last thing, we notice this that to compute the z j's, it is sufficient to compute z j modulo, what we had said is b modulo b times 2th power 2 l. But, we are taking slightly bigger range, which does not hurt b times 2th power 2 l plus 1.

This will be enough to be able to get exact values of z j's, now let us go back and recollect, what chinese remaindering tells me is that, if I want to compute a number modulo this product of these 2 numbers. And if they are co prime to each other then I chose actually, it is sufficient to compute z j modulo b and z j modulo 2th power 2 l plus 1 and then we can built z j out of it. Now, notice that b is a power of 2, it is an even number, this is an odd number and easy to verify that they are co prime. So, we have now a good setting to apply chinese remaindering.

(Refer Slide Time: 17:03)

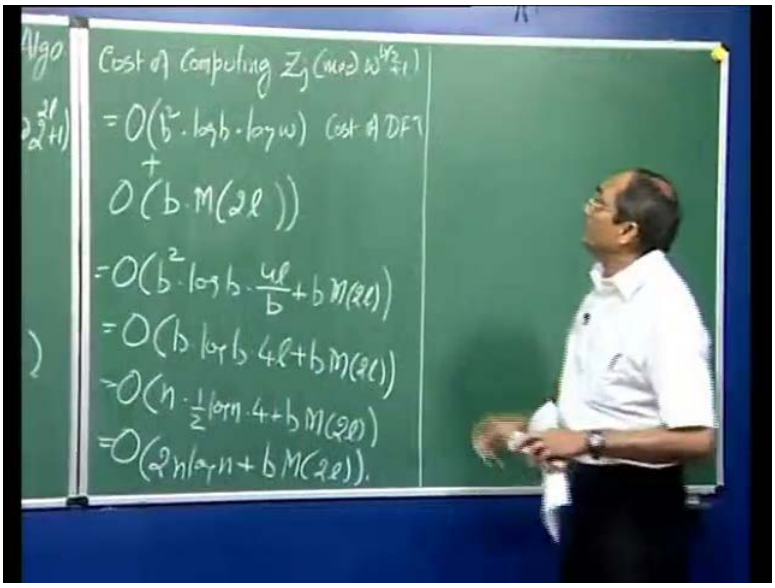


So, first objective, now, is to compute z j mod 2 to the power 2 l plus 1, now in the last lecture, we had seen how to deal with rings like this. If I chose a suitable w then I can remodel this in to the form that, we have seen yesterday and then we should be able to compute both d f t and convolution negatively rapped convolution, positively rapped convolution, etcetera.

So, let us just say that, we need 2 to the power 2 l plus 1 look like w to the power b by 2 plus 1 note that, today the vector lens are b. So, this will be w th power b by 2 plus 1. So, what we need is w should be 2 to the power 4 l by b, w to the power b by 2. So, b by 2 should be equal to 2 l. So, w will be 4 l by b note that b divides l, hence this is an integer, which will define a number. So, with this w this task now reduces 2 computation of z j modulo w to the power b by 2 plus 1.

Now, since we know how to compute and the complexity of the computation. So, in this ring, the computation of the negatively rapped convolution will involve the computation of d f t and computation of componentive component multiplication. So, let us directly asses the cost of the task, I do not need to now re do this whole thing, because we have already seen, how to compute the d f t in this.

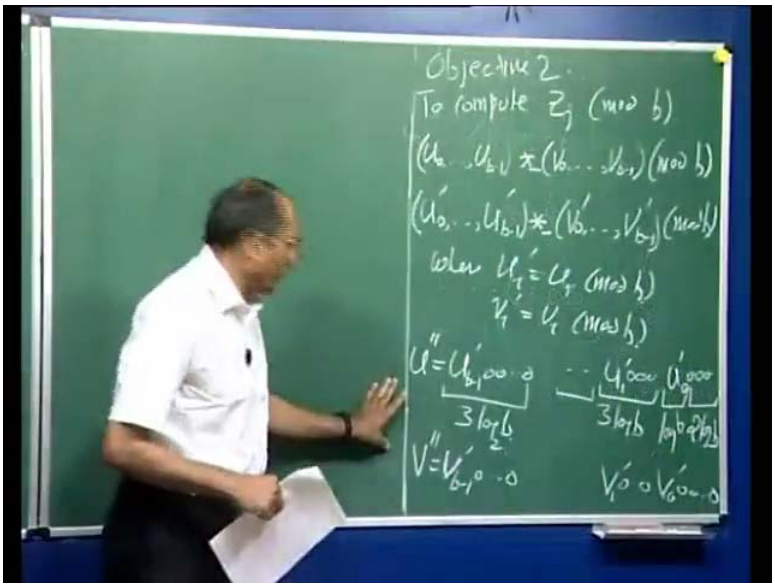
(Refer Slide Time: 19:29)



The cost of computing z j mod w power b by 2 plus 1 is order b square remember in place of n, what we have used yesterday in place of n, we have b, because the components number of components is b. So, this will be log b times log of w. this is the cost of d f t computation and the cost of point to point multiplication, we have b multiplications to perform, because there are b components. Each computation is a multiplication of 2 numbers in this ring, times the cost of multiplication of 2 numbers of size log of 2 to the power 2 l, which is just 2 l, because the number is 2th power 2 l. So, this is the number of bits terms.

So, this is now coming out to be order b square times log of b, which is, we will just leave it, as it is w is over here 2 to the power 4 l by b. So, that will be 4 l over b plus b times m of 2 l. So, this is b times log of b times 4 l plus b M of 2 l b times l is n. So, this is order n log b, b is slight is either square root of n or slightly less than that. So, this is nothing but 1 half log of n times 4 plus b M of 2 l. So, we are getting 2 n log n plus b m 2 l, this is the cost of computation of z j modulo w power b by 2 M plus 1. Now the second computation that, we have to perform is objective number 1.

(Refer Slide Time: 22:30)

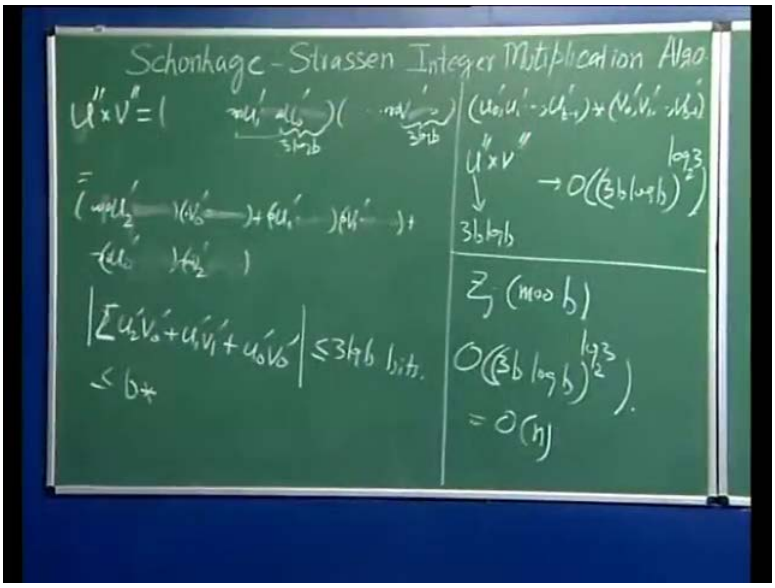


Objective 2 is to compute z j mod b. So, this is the negatively rapped convolution of u 0 through u b minus 1 v 0 through v b minus 1 modulo b. So, let us compute the modulo b of each of these numbers. So, let us say these are u 0 prime through u b minus 1 prime v 0 prime through v b minus 1 prime where, u i prime is u i mod b, which is the reminder of division by b and v i prime is v i mod b.

Now, each of these numbers has size log b, because it is reminder after b. So, let us now generate a number u double prime in the following fashion, u put u 0 prime and several 0s here will give the details in a minute u 1 prime and 0s and so on. U b minus 1 prime and 0s, now each of these numbers has size log b. So, we are going to this as size log b and we are going to put 2 times log b 0s in it, 2 log b.

Similarly this whole thing now will be 3 log b the next 1 and so on. Each of these is now 3 log base 2 or b 2 log base to b is 0s and log b is log b number of bits for this number. Now, we have created this unusual number and let us take a similar version of v. So, v double prime looks similar v double prime is, similarly v prime and b minus 1 and 0s and so on. V 1 prime and 0s and v 0 prime and 0s, so now what do, we do with this 2 numbers.

(Refer Slide Time: 25:40)



So, let us come back to look at this product of these 2 numbers u double prime times v double prime, what is this number, we have these 2, the bit representation of these 2 is like this, we have u 0 prime and 0s u 1 prime and 0s, similarly v prime 0s and so on. Now, let us write down this number x, suppose at it bit expansion, what will happen is that these 3 log b bits with theses 3 log b bits will give me the lowest the set of bits, because this in the value, these are the lowest 1.

Then let us take a look at the second set of bits, these 3 log b bits with this 3 log b bits and the next 1 with this, that will give me the next block, the third 1 will give me this with the of least 1, this 1 with the second one and this 1 with the third 1. So, what exactly, we are getting, in each of these blocks, we are getting precisely the convolution of the numbers. Because, these numbers when for example, if you look at u 3 prime 0 0 0, this times v 0 sorry, let us take 2, because that will be 2, otherwise this with 0 0 0 plus u 1 prime 0 0 0, v 1 prime 0 0 0 plus u 0 prime and v 2 prime 0 0. The total sum actually, I should have put the 0s to the left.

So, I will correct myself. So, all the 0s are to the left, similarly here, they will be on the left. So, we have here, the buffer 0s are to the left, similarly this are on this side. So, when you try to multiply and add, you are going to get nothing but sum u 2 prime, this sum will be u 2 prime v 0 prime plus u 1 prime v 1 prime plus u 0 prime v 0 prime. The largest value of these will be when, you have the maximum that, you can have in this

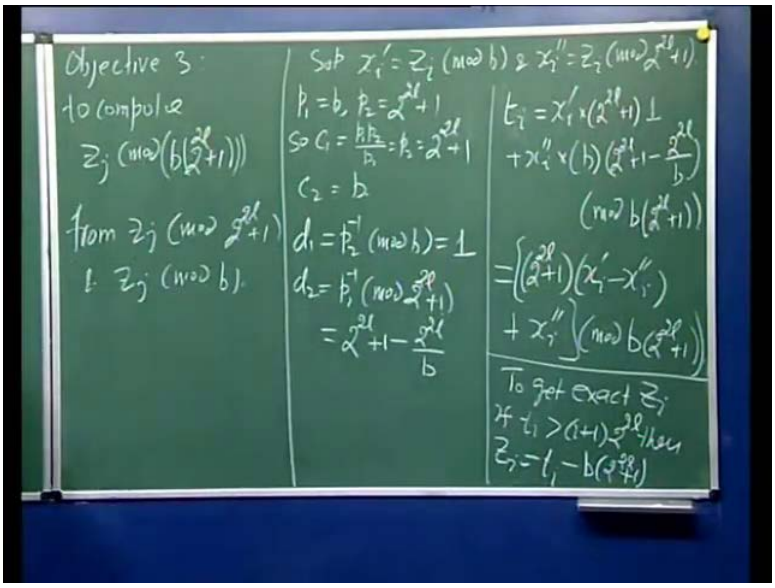
will be b such sums, you will have b such sums. So, that will be less than equal to b times, one of these products, this product each of them is of log b bits. So, the product will have 2 log b bits and when, you multiply by b, you will get the total size is 3 log b bits.

So, the total size of this number is less than equal to 3 log b bits and because we have put extra 2 log b bits and so on, on this side, similarly on this side. So, the total buffer that, we have kept will ensure that the entire result will be observed in this space without overflowing into the next block. So, we can now compute, if we can compute this product, we can compute the components of the convolutions of the 2 vectors, which are u 0 prime u 1 prime u b minus 1 prime and v 0 prime v 1 prime and v b minus 1 prime. So, we can get the convolution of these 2 by this trick of putting buffers of size 2 log b 0s, on the left of each of these numbers.

Now, this means, we have to some how find out the product of these 2 numbers well, in this case since these 2 numbers are very small, notice that the total size of these numbers is three b log b. Each of these numbers has size 3 b log b, each of this is log is 3 log b and there are b bit b blocks in the entire thing. So, this is not a big number, b is the order of square root of n. So, we can offered to compute the product directly with our tribunal algorithm and that will cost as order 3 b log b to the power log base 2 of 3 that, we had yesterday I described a simple algorithm, which can compute the product with this exponent.

Now, once you get the convolution, you can get the negatively rapped convolution by just subtracting 0s element sorry, the b's element from 0'th element b plus first element form the first element and so on. So, that total cost of computing z j mod b for each j will be order 3 b log b power log base 2 of 3, notice that this is the less than 2. So, this whole thing is less than n. So, I can very well say this is order n, now let us go to the step of computing the combined number from these 2.

(Refer Slide Time: 33:00)



Now, the objective number 3 is to compute z j mod b times 2 power l, 2 power 2 l plus 1 from z j mod 2 power 2l plus 1 and zj mod b right. So, now let us say suppose, x i prime is our z or rather x z i or mod, I am going to use the, and here x i double prime is z i mod 2 to the power 2 l plus 1. Now our p 1 is b is 2 power 2 l plus 1. So, C 1 is nothing but p 1 p 2 divided by b 1, which is just p 2 and the 2 power 2 l plus 1, C 2 similarly simply b.

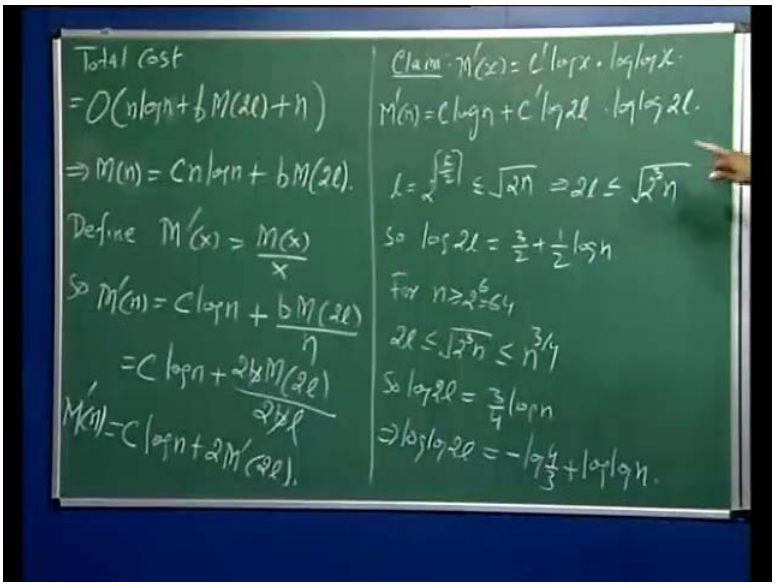
Now, the multiplicative inverse of C 1, now that I am not going to reduce, you can verify that this is p 2 inverse mod b is just 1, 1 can verify that by simply multiplying, you multiply this to this, you are going to get 2th power 2 l plus 1 and then take modulo b remember b is power of 2, so this term goes away, we left with 1. And d 2 is p 1 inverse mod 2 to the power 2 l plus 1 that is, so our p 1. So, this turns out to the 2 to the power 2 l plus 1 minus 2 power 2 l divided by b, modulo 2 power 2 l plus 1. So, now, we have d 1 d 2 C 1 C 2 p 1 and p 2.

So, z j z i is x i prime into C 1, which is 2 to the power 2 l plus 1 into d 1, which is 1 x i prime times C 1 into d 1 plus x i times double prime b times, this term 2th power 2 l plus 1 minus 2 power 2 l divided by b. This modulo mod b times 2 power 2 l plus 1, 1 can simplify, notice that b times, this term will just go away. And this b will multiply to this and turns out to be simply, I am just going to put this as 2 power 2 l plus 1 x i prime minus x i double prime plus x i double prime mod d times 2 power 2 l plus 1. You can

also verify by simply taking modulo of b of this term, you should get x i prime, if you take modulo 2 power 2 l plus 1 of this, you should get x i double prime.

So, now, we have finally, we have to just compute this expression to get our z i modulo this number and recall to get exact z i, remember this is modulo this, to get a z i let us call instead of calling this z i. I am going to call it t i, then to get exact z i, if t i is greater than i plus 1 into 2th power 2 l then z i is t i minus d into 2th power 2 l plus 1 else, it will be just. So, now we are ready to estimate the total cost.

(Refer Slide Time: 38:44)



So, the total cost is order m log n log n plus b m 2 l, this is the cost of the first step that is z i modulo 2th power 2 l plus 1, for computing z i modulo b, we have seen that cost set order n. It was order n and all the remaining computation that, we have done also cost you order n. So, the entire cost is just this, which this can be observed in this, so I will simply say that the cost t n, the cost sorry, let us just m n here, m of n is C n log n plus b m 2 l, for some constant C. Now, we have to solve this to do that let me first define m prime of x as m of x by x.

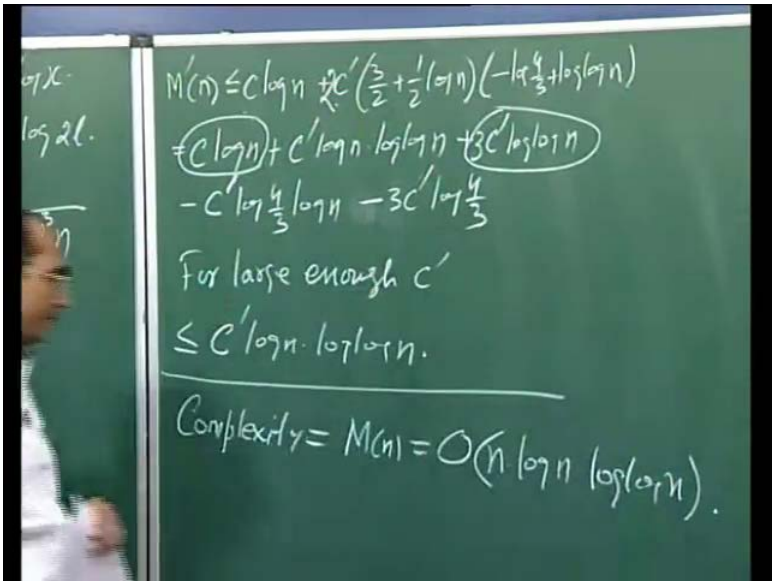
So, m prime of n is this, whole thing divided by n. So, we get C log of n plus this divided by n b m 2 l by n, that n is nothing but b times l. So, I am going to multiply by 2 of this in down there, we are going to get C log n plus 2 b m 2 l over 2 b l, b cancels out m of 2 l divided by 2 l is m prime of 2 l. This is C log m plus 2 m prime 2 l. This is m prime of l, now this is a recurrence relation that, we want to talk, now we are going to solve by

simply substituting. So, our claim is that m prime of l is x sorry, log C prime log of x plus sorry, times log log of l times log log of l, this is the claim. So, to establish that, let us substitute, this for the smaller number 2 l and show that this whole thing is as expected for m prime time.

So, m prime of n should b C log of n plus C prime log of 2 l times log log 2 l, let us find out what 2 l what is 2 l. So, notice that l is nothing but 2 to power k by 2 seeing, which is less than equal to square root of 2 l, because 2 power k is l . So, this is implies that 2 of l is less than equal to square root of 2 power 3 n. So, log of 2 l is 3 by 2 plus 1 half log of n, the logarithm of this number. Now, since we need to compute log of log of 2 l, I am going to find a slightly different expression for 2 l for, which we can take 2 log twice. So, notice that for n greater than equal to 2th power 6, which is 64.

So, larger n of n 2 l, which is less than equal to square root of 8 n is less than equal to n to the power 3 by 4, that is to say m power 1 source should, where power 2 power 3 by 2, which is what this is. So, log based on this log of 2 l is 3 by 4 log of n and log log 2 l is minus log of 4 by 3, notice that, it will log 3 by 4, which is a negative number and plus log log n, let us taken these 2 values in our expression.

(Refer Slide Time: 44:18)



So, we are going to get m prime of n, n rather n is C log or rather less than equal to C log of n plus C prime the expression over here is 3 over 2 plus half log of n times minus log 4 by 3 plus log log n. Expanding this, that C log n plus I should have had their by say

factor of 2 somewhere, it was 2 m prime 2 l, this was a 2 3 and I will have a 2 here. This times this times this, we are getting C prime log of n log log n minus this is C prime log log, it could be 3 3 C prime log log n, then I am going to get C prime minus C prime log of 4 by 3 log of n. And finally, a min plus 3 C prime, what have I missed this is a plus term this into this in this, we have seen last 1 is, the last 1 happens to be minus this into this is 3 C prime log of 4 by 3.

So, note that this term, this term and this term, these terms, you have C prime here, but this is C prime of log n times this number, which is positive, large n of C prime can cancel this number as well as this number. If you chose C prime large enough this is because it is a very small number, this is not going to grow very, very fast. This will of faster and will take care of this term, because C is a constant C prime is for us to chose.

So, we can for large enough C prime this thing is less than equal to just C prime log n log log n, this n is a negative term. So, that does not matter. So, finally, we have shown that m prime is m is also satisfying this requirement, hence what we have found is with the complexity. So, complexity is m n is order n times, this n times log of n times log log of n. So, this is significantly improved performance over the previous 2 algorithm, which was n square and the other one was n n to the power log 3 base 2, here the term n occurs with power 1.