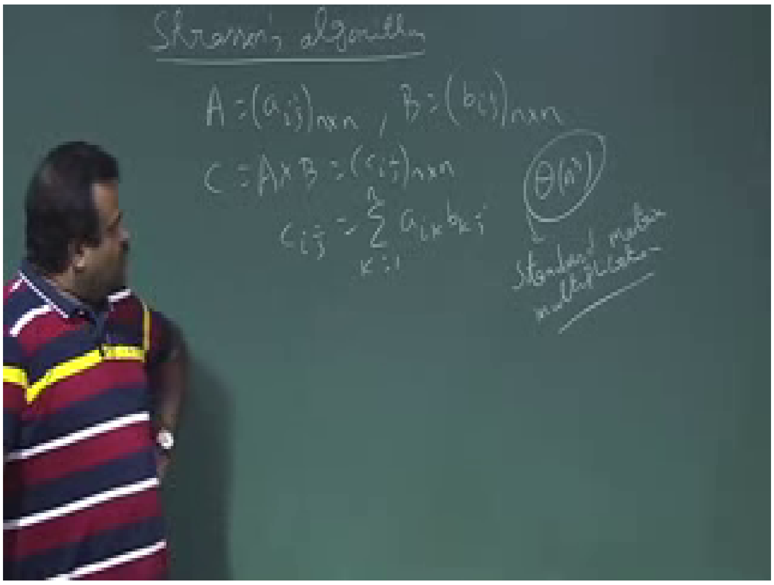
An Introduction to Algorithms
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 09
Strassen's Algorithm

So we talk about we are talking about matrix multiplications. So, that we have seen using the standard method or matrix multiplication.

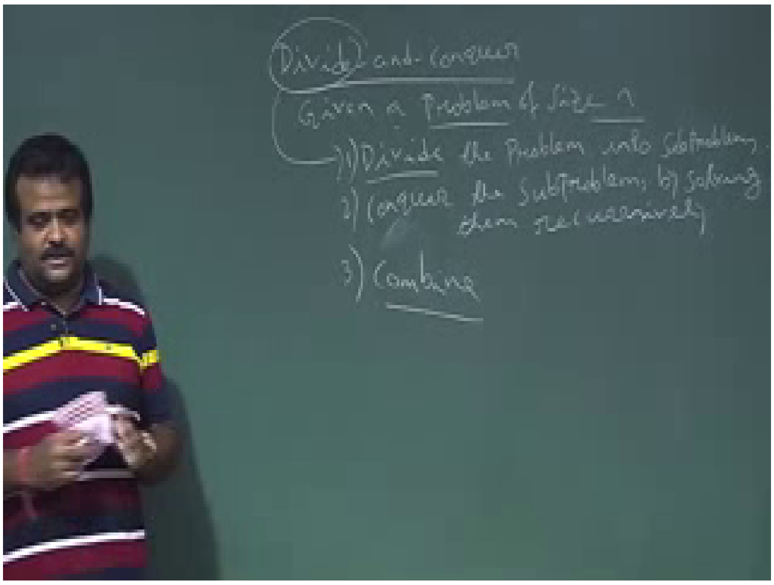
(Refer Slide Time: 00:31)



If we have 2 matrix a b which is basically size n cross n and B matrix is size n cross n and we have seen A C is basically A cross B which is of size n cross n, and c i j is basically summation of a i k b k j and we have seen the standard this is basically a using a for loop three.

For loop we can get this value, but that will of order of n cube this is the by the standard matrix multiplication formula method algorithm standard matrix multiplication. So, now, in this talk we want to see whether we can use some divide and conquer technique to reduce it further I mean whether we can apply some divide and conquer technique to handle this problem. So, for that let us just look at divide and conquer technique means if we have a problem of size n it basically has three step.

(Refer Slide Time: 01:46)

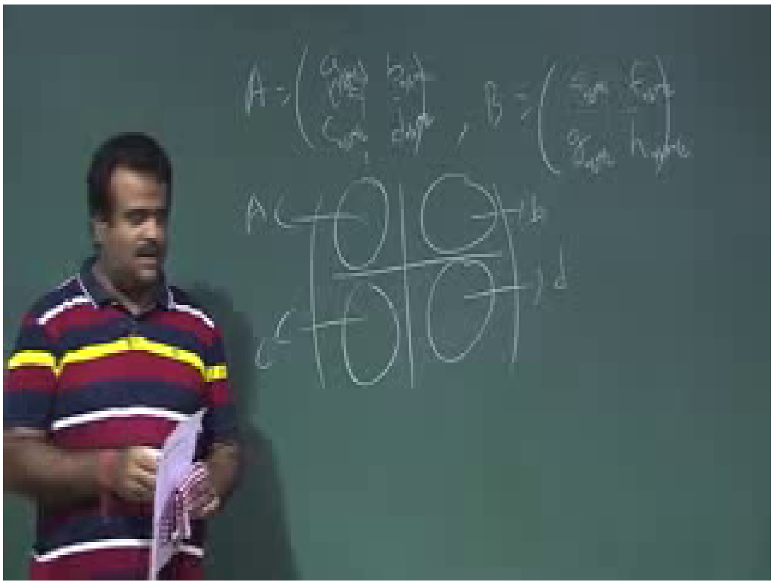


So, this is the just to recap. So, we have a we have given a problem of size n now in the first step divide step we divide the problem this is the divide step it has three steps, we divide the problem in to lesser size into sub problems and sub problems size should be less than n it could be n by 2, it could be n by 3 divide the problem into sub problems and then this is the first step of divide and conquer method then the conquer step we conquer the sub problems by solving them recursively.

We conquer the sub problems by solving them recursively. So, this is the recursive step; again we have a sub problems again we further divide into sub problems like this until the we reach to the size is 1, once the size is 1 you must stop because then it cannot be further reduce to sub problems and then we have a final step which is combine which is the basically for merge sort which is basically merge sub routine.

So, we had a solution for this 2 sub problems or 2 or 3 we have the solutions for the sub problems, now this merge step will this merge or combine step will combine the solution of the sub problems to get the solution of the whole problem. So, this is basically the divide and conquer technique. So, now, we will see how we can apply this technique for our matrix multiplication method. So, for that it is just write the matrix in to sub matrices. So, we have 2 matrix A B.

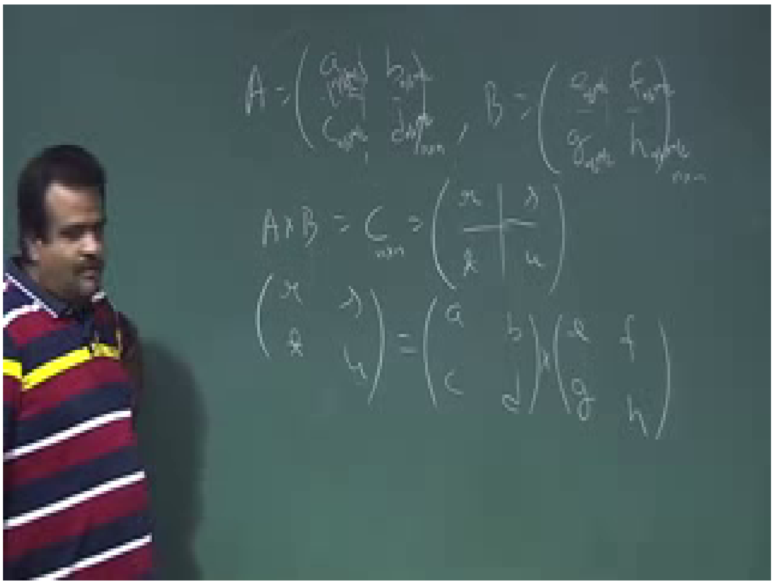
(Refer Slide Time: 04:03)



So, we write the matrix is A as a b c d and B as say e f g h. So, these are basically sub matrices. So, this is these are all size n by 2 cross n by 2. So, a is n by 2 cross n by 2. So, b is also n by 2 cross n by 2. So, if we have big matrix what we do we just partition into 4 part. So, this is basically A this is this matrix this sum matrix is B like this, this sum matrix is c and this sum matrix is d like this. So, this are basically sum matrices.

So, we have 4 part w have dividing provided n is an even number or n n is multiple up to otherwise we have a lower ceiling or upper ceiling, but anyway this we will do the asymptotic analysis. So, we are allowed to do this copyness. So, these are all sum matrices of size n by 2 cross n by 2. So, these are all n by 2 cross n by 2, these are all sum matrices similarly here we have n by 2 cross n by 2, we have n by 2 cross n by 2 this is also n by 2 cross n by 2, n by 2 cross n by 2. So, this is basically just given a matrix we divide into we just thing is a partitioning into 4 part.

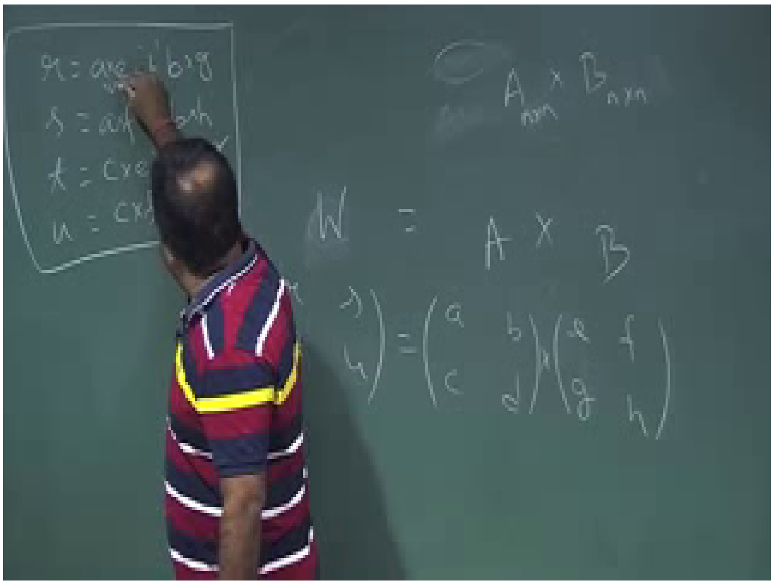
(Refer Slide Time: 05:53)



So, that is the way we represent this a b c d and for b matrix e f g h. So, now, we denote this. So, we multiplying a into b. So, we are multiply A into B and A into B we where denoting by c. So, this is also we are writing in 4 part. So, like r s T u. So, this is also this is n by 2 n by n cross n by n. So, c will be also n by n cross n matrix. So, this is also we partition into 4 part. So, these are all n by 2 n by 2. So, basically what we have we have this matrix multiplication r s T u which is basically a b c d then this is e f g h. So, this is the.

So, this is the matrix multiplication and these are in the sum matrix form. So, now, what is r? R is basically. So, we have to write this formula. So, what is r, r is basically r is getting by this into this. So, r is.

(Refer Slide Time: 07:01)



Basically a e plus these are all matrix multiplication this is sum matrices plus b g. So, these are these are all this is n by 2 matrix this n by 2 n by 2 this is also n by 2 cross n by 2 this is also n by 2 cross n by 2 this is also n by 2 cross n by 2 and this addition is. So, matrix addition, we have 2 matrix of size n by 2 cross n by 2 we add it.

So, just a addition of the elements corresponding elements. So, similarly we have s equal to s is basically this cross this. So, f plus s is a f plus b h. So, these are all matrix multiplication these are all matrix addition. So, T is similar to T is basically, T means T is basically this into this. So, c e plus d g c into e plus. So, this c is nothing to do our this a A into B is equal to c this is capital C. So, better we use this in different notation. So, do not confuse with this c and our matrix c they are different basically. So, basically we have 2 matrix a b, a b this is a matrix this is b matrix and the multiplication is this matrix.

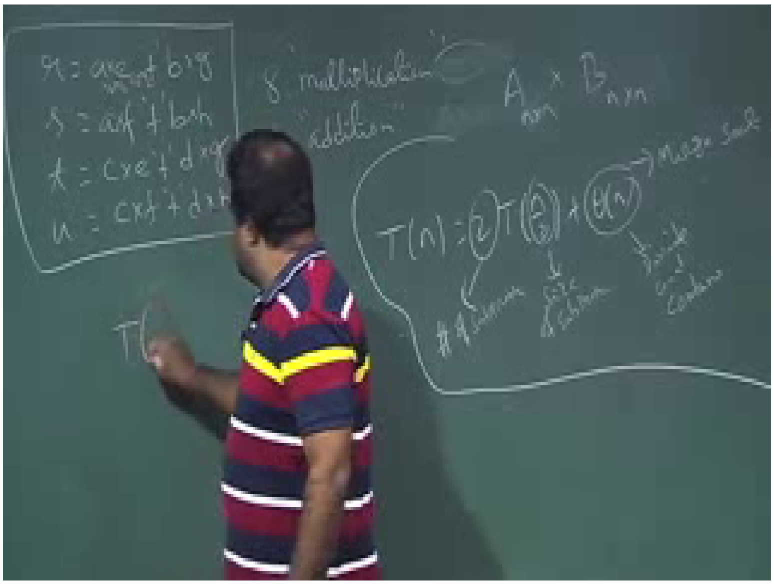
So, this is capital C. So, this is this 2 c are not same. So, if there is a any confusion we can made it to some another matrix W. So, A W is equal to A cross B if there is any confusion with this c, but anyway this is small c that is capital c c into e plus d into. So, this into, this into this d into h, so if T is basically these into this, c into c into e plus b into g. So, and the last one is u, u is basically this into this c into f; c into c into f plus d into h, d into h. So, this plus are all matrix addition. So, this is the formula we have. So, matrix multiplication is basically.

So, this is the divide and conquer step this is the divide step. So, now what is this now we have the matrix. So, we earlier we have to multiplication of matrix this cross. So, we have a A matrix we have a B matrix, and we have to multiply this is of size n cross n and this is also of size n cross n. Now here these are the all matrix multiplication, but these are all lesser size what is the size these are all n by 2 cross n by 2, these are all n by 2 cross n by 2 all the matrix multiplication. So, these are the sub matrices. So, these are all lesser size. So, this is the divide step.

We divide the problem into sub problems earlier our problem was to multiply 2 matrix of size n cross n into n cross n, now we reduce the problem into sub problems like now we have to multiply n by 2 cross n by 2 and another matrices is also n by cross by 2 cross n by 2. So, that is the divide step and now by in conquer step we must get the solution of this and we must get the solution of this once we have this solution of this 2 then we have to combine by adding this to get s so, that is the divide and conquer approach is this clear.

So, basically we have a we have 2 matrix we reduce the matrix into sum matrices and then we reduce the so that means, we reduce the problem into sub problems and then conquer step we again call the same formula, now this is of size n cross lesser size again we part the divide into sub matrices like this and again once we have a solution for this once we have solution for one once we have the result and once we have this result, we add this and that is a part of the combine step. So, how many matrix multiplication we are doing basically we are doing 8 multiplication ok.

(Refer Slide Time: 11:44)

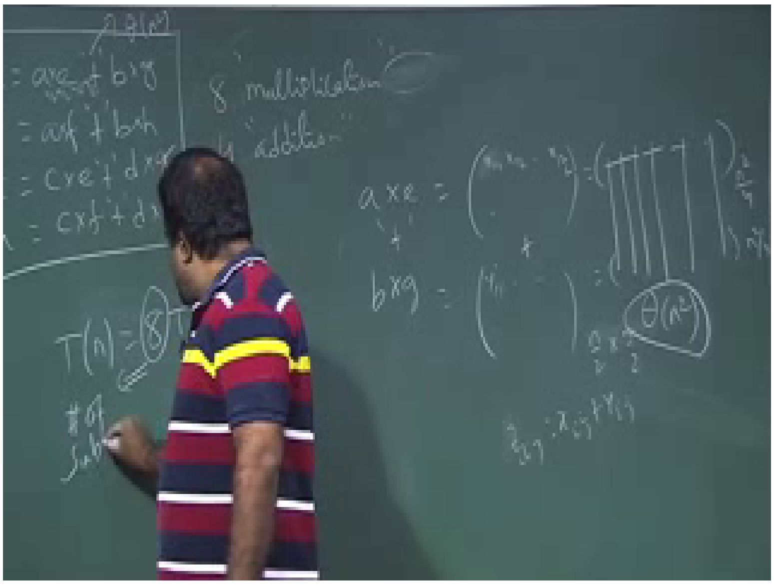


And how many addition we are doing? We are doing the 4 addition 4 matrix addition these are all matrix addition these are all matrix multiplication of lesser size. So, now, what is the recurrence we have? So, basically every divide and conquer technique have the recurrence like T n is equal to. So, merge sort we have if you remember for merge sort we have the recurrence T n is equal to 2 T n by 2 plus theta of n. So, these was the.

So, we have we reduce we have a problem of size n we reduce into the sub problems and these are the number of sub problems this is the merge sort recurrence this is the number of sub problems, and this is the size of sub problems size is reduce to half I mean we have a array we divide into 2 part half half, this is the size of sub problems and this is the cost for both divide and merging and combine, divide and combine.

So, now if we use this divide and conquer technique of matrix multiplication. So, then what is the recurrence we have? So, we have T n is basically how many sub problems. So, our problem is basically matrix multiplication.

(Refer Slide Time: 13:25)



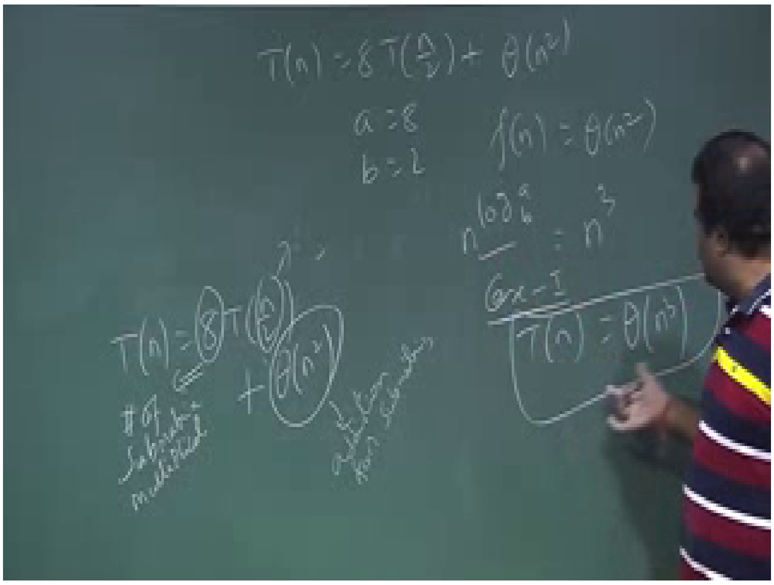
So, we reduce this into sub matrices. So, how many sub problems. So, now, here 8 sub problems because 8 matrix sub multiplication is required on the lesser size. So, that will done in the conquer step. So, that is 8 of T of size this n by 2 is the size of the sub problems. Now our problem of problem was n cross n matrix multiplication now it is n by 2 cross n by 2 and both matrices sub matrices.

So, size is reduce to half. So, that is the size of the sub problem. So, this is basically n by 2 plus the cost for cost for what cost for combine. So, combine means this additions this, this matrix addition. So, how much time it will take to add 2 matrixes. So, that is the point. So, that is the time to take the addition. So, how much time it will take? So, 2 matrix addition. So, what is the size of this matrix? So, once we get the result this by the conquer step. So, this will be a a cross e will be a matrix of size n by 2 plus n by 2 ok.

And then b cross g this is also of size n by 2 cross sorry n by 2 cross n by 2; now this addition means we have just adding the this is added with this like this. So, if this is the matrix like x 1 1, x 1 2, x 1 n by 2 like this and this is y 1 1 like this. So, this basically addition is y i. So, addition will be z i j, z i j is basically x i j plus y i j. So, we are adding just adding the corresponding n T t matrix addition. So, just we take this value adding with this value adding with this like this even you can think this as a vector. So, just this is a vector of size.

So, just first row then second row third row like this. So, this is a vector of size n square by 4, and then this is also vector of n square by 4. So, we are just adding 2 vector like this. So, if we have 2 vectors then it will be. So, how many addition? We have basically n square by 4 addition. So, this is this addition will take order of n square. So, each addition will take order of n square. So, how many addition we have each of this addition will take order of n square. So, we have basically 4 such addition. So, if we have 4 addition. So, this will take order of n square ok.

(Refer Slide Time: 16:55)



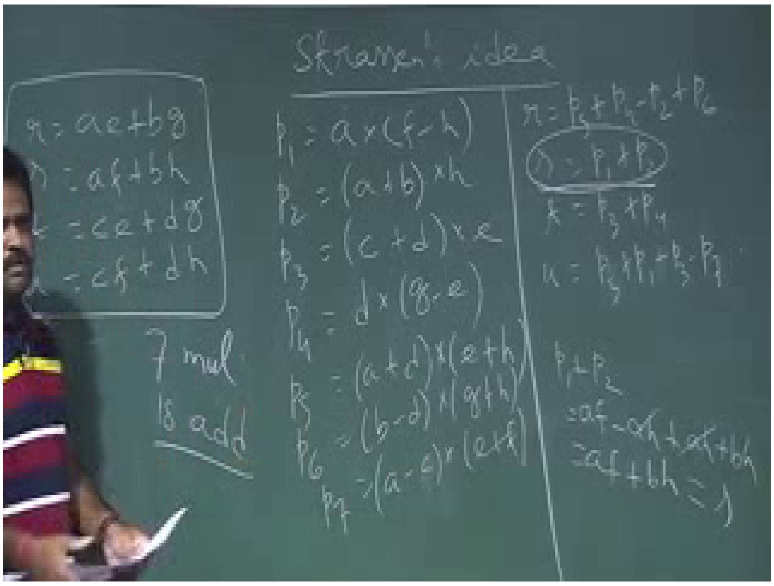
So, this is the cost. So, this is basically number of sub matrices, this is the recurrence for matrix multiplication using this divide and conquer technique this is the number of sub matrices basically we are having 8 sub matrix multiplication, number of sub matrix multiplication, and this is the size of the sum matrices, sub matrix size is reduce to n by 2 cross n by 2 size of sub matrix and this is the cost for combining the addition for addition for sub matrices ok.

So, this is the recurrence we have for this divide and conquer techniques. So, now, we want to know the solution for this recurrence. So, this is basically. So, we have this recurrence what is the recurrence T n is equal to 8 T n by 2, plus theta of n square. So, what is a? We want to fit it into master method we want to see whether this. So, a is 4, b is 2 and f n is n square. So, which case we are in. So, n to the power log a base b this is basically n cube. So, we are in case 1. So, case 1. So, what is the solution? So, solution is this term is dominating.

So, this is asymptotically a ploynomically faster than this one. So, this is basically order of n cube. So, this divide and conquer technique is giving us order of n cube time algorithm which is not better as the standard method or matrix multiplication because standard method is also giving us order of n cube. So, then the point is how we can modify it further. So, that is the idea of Strassen's. So, Strassen's give the idea of how we can modify it further like because this direct divide and conquer technique will give us order of n cube algorithm, which is same as the standard matrix multiplication algorithm. So, the idea of strassen is like this. So, instead of this recurrence suppose.

So, this is the recurrence coming from this divide and conquer techniques suppose instead of 8 multiplication.

(Refer Slide Time: 19:39)



If we can reduce a multiplication by 1 instead of 8 multiplication, if we can reduce it to 7 with the cost of more addition, because addition anyway will take order of n square, so here we are doing 8 addition what is the formula? Formula is our r is basically a plus b g our s is a plus b h our T is basically c e plus d c T is basically c plus d g and our u is basically c f plus d h. So, this is the r s p u now here we are doing 8 multiplication.

So, the Strassen's idea is if we could reduce 8 multiplication instead of 8 multiplication if we could reduce to 7 with the cost of more addition here we are doing 4 addition, but if we could do more addition that will be capture under order of n square. So, that will not affect much. So, that is the idea. So, that is why we want to make it that Strassen's idea is to make it 7 matrix multiplication. So, that is the. So, that we are going to write how we can do that. So, that is the idea by Strassen's, Strassen's idea to reduce the multiplication by 1 with the cost of more additions.

So, for that Strassen's is calculating some intermediate terms like p 1 a into f minus h, p 2 into h, p 3 is basically c d into e. So, I will explain. So, these are the term we are calculating these are intermediate terms. So, there will be 7 terms so; that means, there will be 7 matrix multiplications, but we will have more addition. So, we will we will talk about that. So, d into g minus e and p 5 is basically a plus d into e plus h.

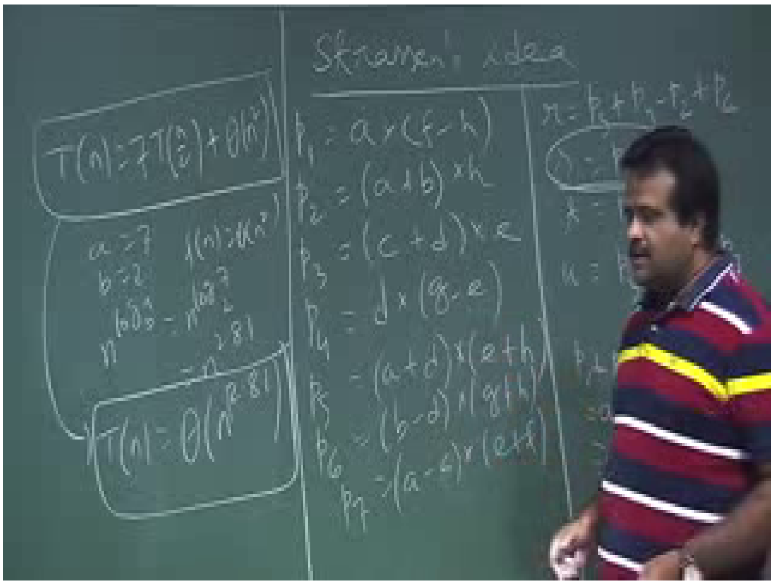
And p 6 is basically b minus d into g plus h, and the last term is p 7; p 7 is basically a minus e sorry a minus c into e plus f. So, this are the terms Strassen's is calculating these are the intermediate term. So, there are basically 7 terms so; that means 7 multiplication of lesser size. So, this is multiplication with. So, once we get this value how to get r s t u. So, that formula we have to write then we can see r is basically p 5 plus p 4 minus p 2 plus p 6 and s is basically p 1 plus p 2, T is basically.

We can verify this p three plus p 4, and u is basically p 5 plus p 1 minus p three minus p 7. So, with the help of this p is we can get this. So, how to verify this we can take one example like this one remaining you can check by yourself, p 1 plus p 3, p 1 plus p 2. So, p 1 is basically this into this. So, p 1 is a f minus a h. Now what is p 2? p 2 is h plus b h. So, h plus b h, this 2 is canceling a f plus b h what is a f plus b h a f plus b h is basically s.

So, this is verified s equal to p 1 plus p 2. So, (Refer Time: 23:54) in term also you can easily verify just putting the value of ps. So, this is the formula, these idea is given by Strassen's. So, basically instead of 8 multiplication we have 7 multiplication see there are 7 terms and each term is having one multiplication, but with the cost of more addition this subtraction is also addition, with the cost of more addition. So, how many addition? Subtraction is also addition 1 2 3 4 5 6 7 8 9 10 then 11, 12, 13, 14, 15, 16, 17, 18 with 18 addition.

So, we have we have 7 multiplication and we have 18 addition, that is because addition can be more because addition will take one addition will take order of n square. So, this will be capture in theta of n square the, what is the recurrence of this method. So, recurrence will be we have 7 multiplication.

(Refer Slide Time: 24:56)

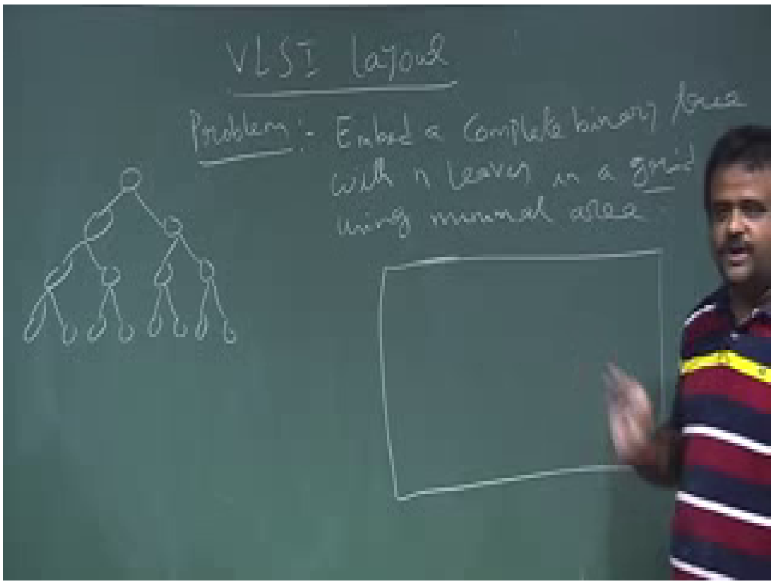


So, that is the good news. So, T n to be 7 T n by 2 plus order of n square, this is the recurrence we have for Strassen's multiplication method; here we have more addition.

But that will taking care by order of n square and we have all the instead of 8 matrix multiplication we have 7 matrix multiplication. So, these are the term is this. So, then what is the solution what is the solution for this. So, here a is 7, b is 2 f n order of n square. So, what is n to the power log a base b it is basically n to the power log 7 base 2, it is basically n to the power 2.81. So, now, this is also in case 1 and the solution for this is basically T n is equal to n to the power 2.81.

So, this is the time complexity for Strassen's matrix multiplication. So, this is much more better than the n to the power n cube which was the standard method or just the divide and conquer technique. So, this is also divide and conquer technique, but we could reduce the matrix multiplication by instead of 8 we could reduce to 7. So, this is a good gain in terms of this. So, this is the Strassen's idea. So, next problem we will talk about VLSI layout which is the another problem which can be used which in which we can use the divide and conquer technique.

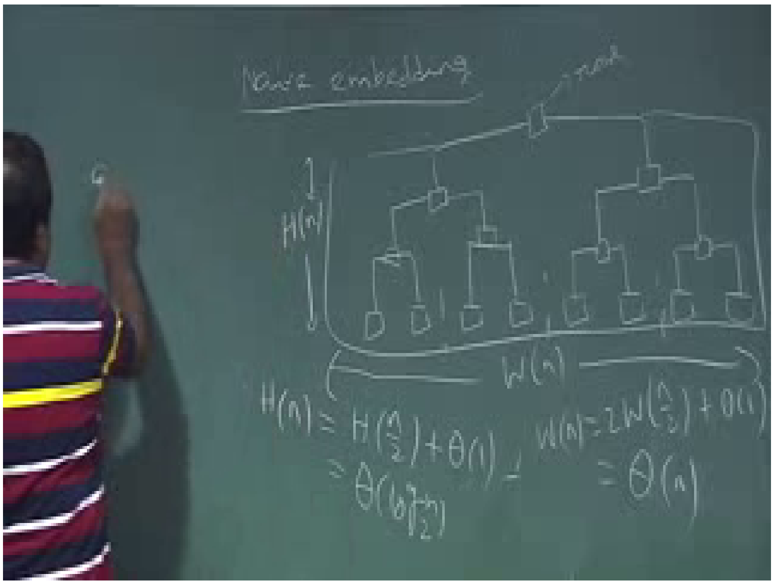
(Refer Slide Time: 26:54)



So, this is the different problem, this is this is not really algorithm problem this is called (Refer Time: 26:55) 5 minutes VLSI layout. So, the problem is. So, we have a chip we have a chip rectangular chip and we want to embed a complete binary tree that this is the problem, embed a embed a complete binary tree with n notes n leaves sorry n leaves in a great using minimum area that is the problem using minimum area. So, we have a rectangular chip. So, in which we have to embed a tree complete binary tree like this.

So, this is the tree. So, may be once more. So, this tree we have to embed in a chip a rectangular chip. So, we have the rectangular chip in a great so that we should take the minimum area. So, area means this is the length this is the width. So, the total size will be less.

(Refer Slide Time: 28:46)

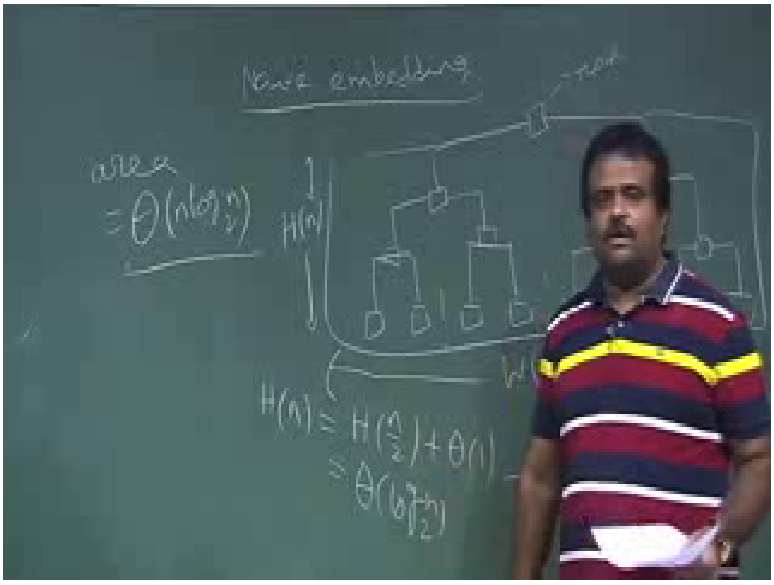


So, this we can do it like this. So, we can just start at this is the naive approach. So, what is the naive approach? So, this is the problem, the naive embedding. So, we start embed in this. So, this is. So, we start from the. So, then this is basically.

So, this is giving us like this like this then again we take this again we take this like this. So, this is the naive embedding. So, we just start with this leaf note then we slowly this is the root now this is our, this is our chip now what is the length of this how to calculate the area. So, if we denote this by the height of the tree, and if we denote by this width of the w n, now for height what is the recurrence height is basically.

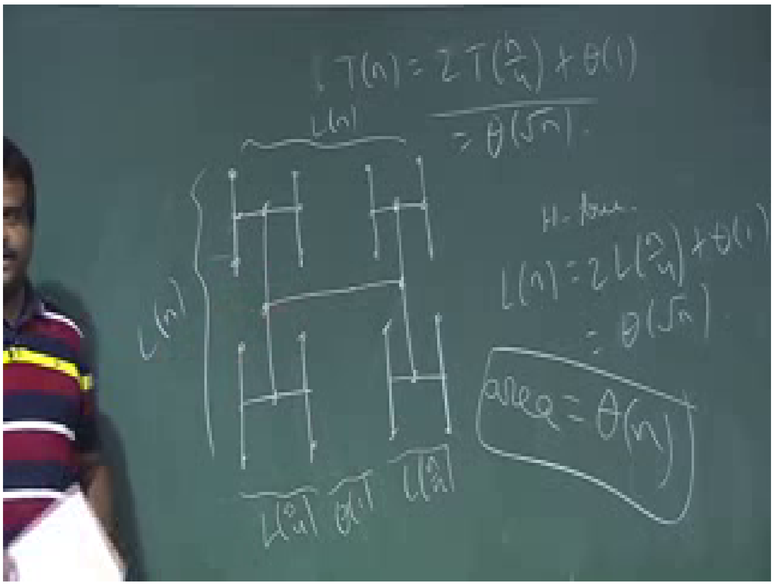
So, this is a, we have to embed n notes. So, to embed n notes we just. So, these are the running in these are in parallel. So, height for recurrence for height is T n h n equal to. So, h n by 2 plus theta 1 because this 2 are in parallel, but for width it is not parallel. So, this will count. So, basically for width we have w n is equal to 2 of w n by 2 plus theta 1. So, if we again use the master method this will give us a solution like log n, but this will give us a solution like n. So, the area is basically.

(Refer Slide Time: 31:05)



So, the area is height into width in into the width. So, area is basically order of n log n. So, this is the log 1 now do you want to make it in a order of n times n we want to do it in order of n. So, that should be the area we want. So, for that we will do something which is called h embedding h tree embedding. So, how to get order of n? To get order of n height and width should be order of root n.

(Refer Slide Time: 31:43)



So, how to get root n which recurrence give us the root n? So, if we have a T n is like this T n is equal to say 2 T n by 4 plus theta 1.

So, this type of recurrence will give us root n if we can so, that the height of both are coming like this recurrence then you are done. So, for that you will have h type embedding. So, what is that will do like this. So, these are all h this called h tree. So, we are drawing the, this thing and another one here another 1 h tree here. So, these are all the connections. So, these are all h tree, then we have this one then we have this one.

So, these are all these are all H tree H structure. So, if we have this then this is basically l of n and this is basically this is also same this is also l of n this is basically (Refer Time: 33:13) now what is the area. So, this is we have l of n by 4 and this is theta of 1 and this is again l of n by 4. So, we have a recurrence for l is like this 2 of l of n by 4 plus theta 1. So, this will give us a solution theta of n.

So, this is the one side area. So, l n and the area is basically root over of h this. So, this is the using the help of h type embedding. So, this is on application of divide and conquer technique totally in different area. So, this is called VLSI layout, we want to fit a in this chip we want to fit a complete binary tree connection. So, that is the idea.

Thank you.