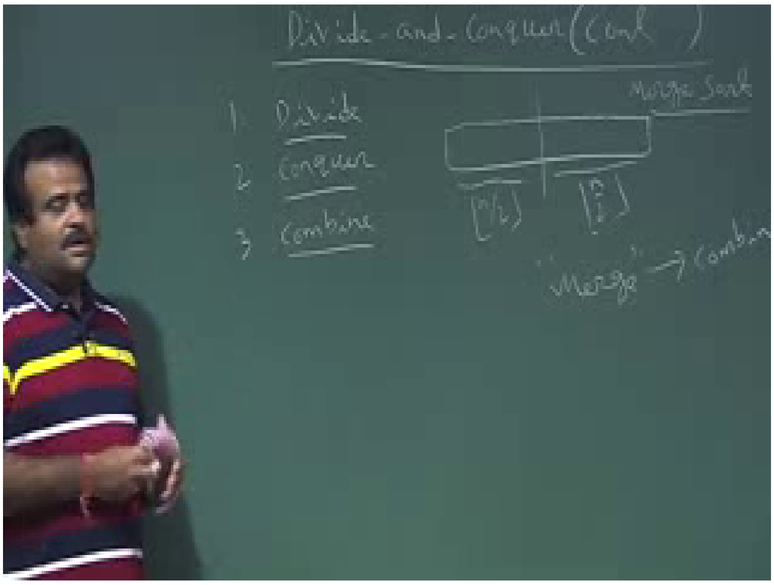
An Introduction to Algorithms

Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 08
Divide And Conquer (Contd.)

So we are talking about divide and conquer technique. So, it is a designed technique. So, it is basically has 3 steps one is divide.

(Refer Slide Time: 00:34)



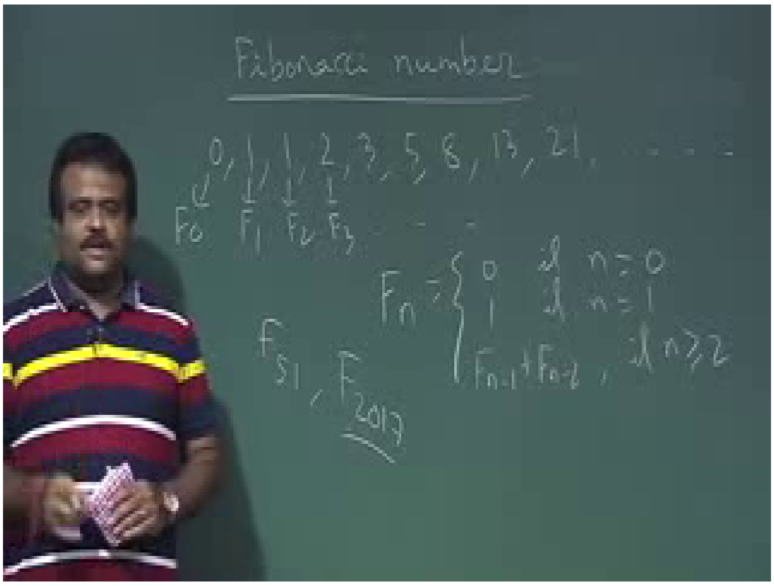
So, we have given a problem of size n, we divide the problem into sub problems. So, that is the divide step and then. So, we have the sub problems which are lesser size, which is not now in n. So, which are lesser size now we solve the sub problems by recursively solving them so that is the conquer step conquer step and then once we have the solution of this 2 sub problems then we combine the.

We combine the solution of the sub problems to get a solution of the whole problems. So, this is basically 3 fundamental steps of any divide and conquer technique like in merge sort is a example of divide and conquer technique what we are doing, we have a array of size n which need to be solve, now we divide this array into 2 sub array with equal size and similar inner ceiling and upper ceiling n by 2, then a this is the divide step is merge sort this is the divide step in merge sort and then what we are doing we have recursively sorting.

So, this is a sub problems we reduce the problem our problem is sorting problem, we reduce this problem size to from n to n by 2 n by 2 now at the that is the divide step. Now the conquer step we sort this sub array we sort this sub array recursively. So, that is the conquer step, but when you sort this sub array this is the sub problems this is also sorting problem, but the size is not n size is n by 2. So, we call same merge sort on this same merge sort on this, now that is the conquer step. Now once we have the solution for this sub array this sub array; that means, was these 2 sub array are sorted then we call what is called merge; merge sub routine.

So, that is the combine step this is the combine. So, we have a solution up to sub problems then we solve we get the solution of the whole problem by the combine step. So, this is the way by this is 3 fundamental steps for any designing conquer technique and we have seen few example like binary search powering a number. So, today will talk about 2 more example which can be can be handle by divide and conquer technique.

(Refer Slide Time: 02:56)

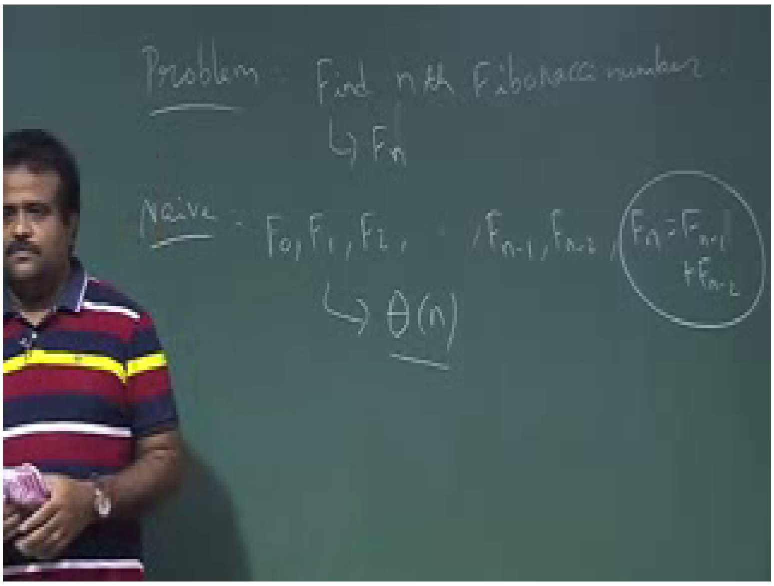


Today’s problem is first problem is Fibonacci number. So, problem is to finding Fibonacci number. So, what is the Fibonacci number? So, it is start with 0 then next Fibonacci number is one, and then next one is add this 2 previous two, 1, then 2 then 3, then 2 plus 3 5, 5 plus 3 plus 5 plus 3 8, 8 plus 5 13, and then 13 plus 8 21 like this. So, this is the sequence of Fibonacci number, this is called step 0 Fibonacci number F 0 we denote F 1 like this F 2 F 3 so on. So, what is the formula is F n is basically zero.

If n is 0 this is the are in first Fibonacci number or the 0 Fibonacci number one if n is one and then after one we have the recursive formula like F n n minus 1 plus F n minus 2 if n is greater than equal to 2. So, this is the formula for n th Fibonacci number. So, F n is basically F n minus 1 plus F n minus 2. So, last few Fibonacci number will give us the n th Fibonacci number. So, our problem is to find the nth Fibonacci number where n is another input. So, we have to find 50th Fibonacci; 51 th Fibonacci number we want to find say 2017 th Fibonacci number like this.

So, this is the problem. So, the problem is to find the nth Fibonacci number. So, how to do that? So, that we can just see how we can solve this problem what is the algorithm will use for this. So, the problem is to find n th Fibonacci number.

(Refer Slide Time: 05:06)

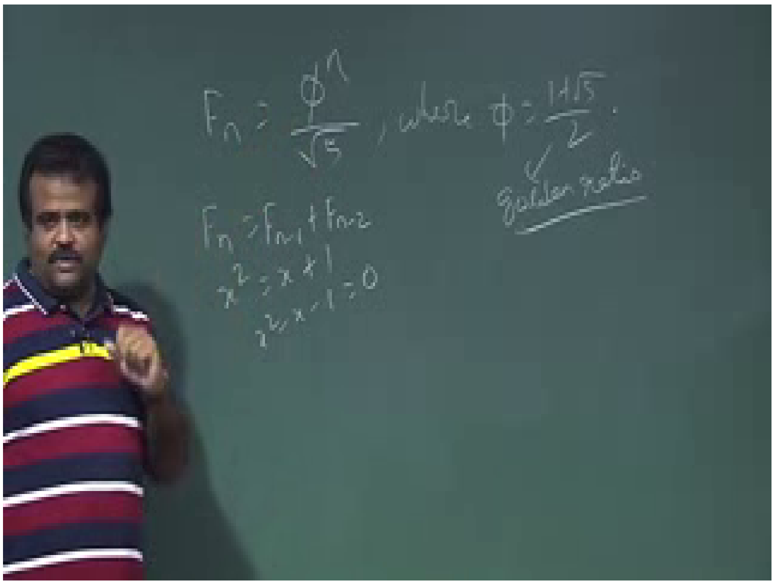


Find n th Fibonacci numbers n th Fibonacci number F n so; that means, we have to find F n where n is also an input. So, how to do that? So, what is the naive approach?

So, it is a bottom of method we start with the first Fibonacci number that is 0 I mean we just keep on calculating F 1 F 0 F 1 F 2 and every time we store the last Fibonacci number and then F n minus 1, F n minus 2, and then we got F n by adding this 2 F, F n minus 1 plus F n minus 2. So, this is our nth Fibonacci number. So, this is the bottom of method we keep on calculating the Fibonacci number until we reach to the n th Fibonacci number. So, this is basically a bottom of way and. So, this each time we store the 2 last Fibonacci number to get the next Fibonacci number.

So, what is the time complexity for this? So, this is basically linear time algorithm because we are just calculating up to nth Fibonacci number like is bottom of way. So, this is the linear time algorithms. So, now, we want to do we want to see whether we can have something better than this linear time whether we can do it in logarithm time like this, so that for that we can use some formula which we know about the Fibonacci number. So, that formula is basically telling us the nth Fibonacci number F n can be written as phi to the power n by root 5.

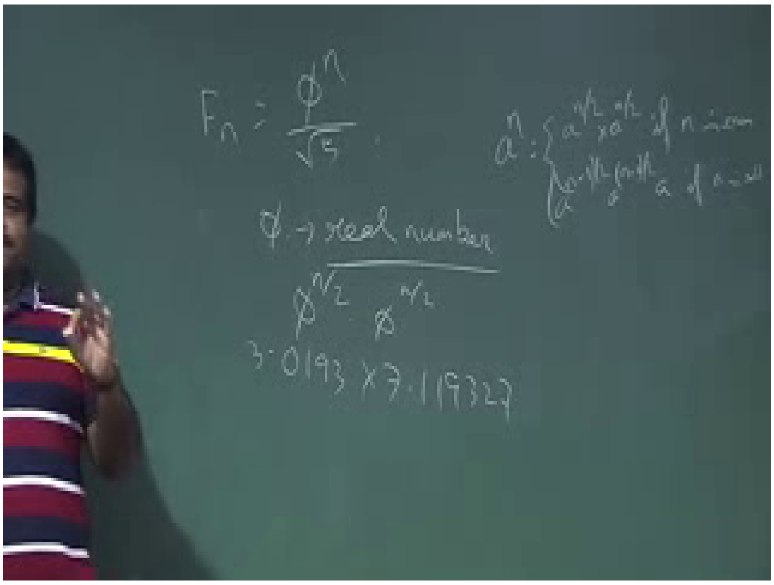
(Refer Slide Time: 07:08)



Where phi is basically 1 plus root 5 by 2, and this is basically what is called golden ratio this number is called golden ratio, where from this is coming this we can prove by index and this is coming basically by solving this recurrence. So, we know the F n is basically F n minus 1 plus F n minus 2, now if we take this as a x square this is basically x plus 1. So, if we have this x square minus x minus 1 is 0 now it has 2 one root is this, plus 1 plus minus root 5 by 2.

So, that is why this phi is coming. So, it can be shown that F n is phi to the power n by root 5. Now we want to see how this formula can help us to have a algorithm, now this is also similar to owing a number, but here the number is not a integer we have seen if we have integer like A to the power n we know this is a candidate of divide.

(Refer Slide Time: 08:27)

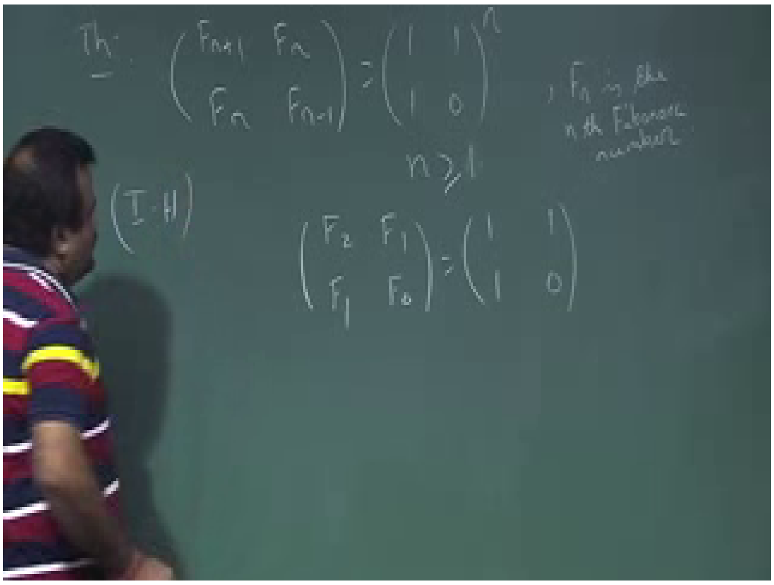


And conquer algorithm we can write this as A to the power n by 2 into A to the power n by 2, if n is even else.

We have seen this is n minus 1 by 2 into A to the power n minus 1 by 2 into a, if n is odd this formula we know this is basically give us powering a number, but here a is basically a integer, but here also this is also similar to powering a number, but this phi is a real number. So, problem with handling real number when we find out powering a real number then the problem will be in the round off or in the precision. Suppose if we calculate this phi to the power n by 2 phi to the power n by 2, suppose this is a 3.01932 like this and this is say 7.119325.

So, depending on how much round off. So, this rounding and precision will take some time to fix. So, this algorithm is not as simple as powering a integer. So, that is why we will. So, that is why this will not give us a logarithm time algorithm because this fixing will take some time more time. So, this is the reason will try to avoid this powering a real number. So, this formula is not helping us. So, now, we try to see whether we can have another kind of formula which can help us to find out the n th Fibonacci number so that formula is basically will have a theorem on that.

(Refer Slide Time: 10:28)

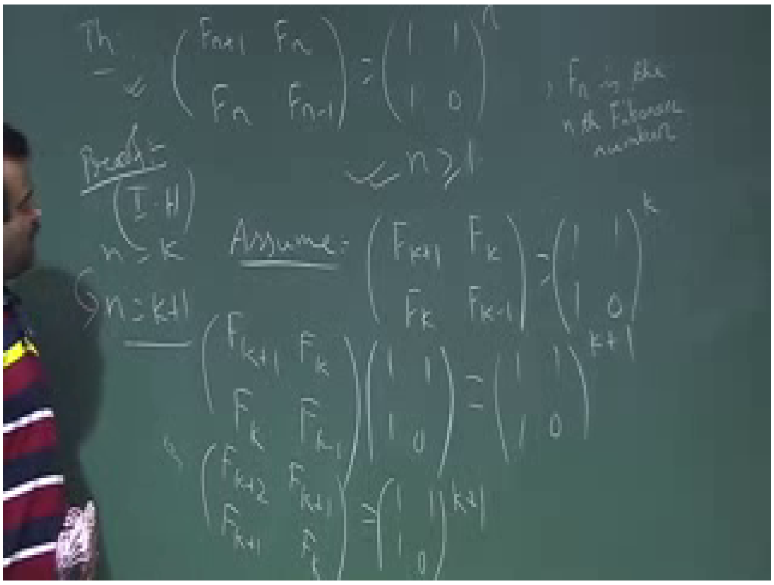


So, this is the theorem on the Fibonacci number this theorem is telling F of n plus 1 F of n, F of n, F of n minus one can be written as 11, 1 0 to the power n. So, this is the theorem where F n is the n th Fibonacci number Fibonacci number. So, it is start with F zero. So, we know F 0 is one. So, what is the base case? Base case means. So, this, this 2 this true for all n greater than equal to 0 this we have to prove now what is the base case base case is if we put n is equal to one. So, if we put n is equal to one.

So, this is true for n is greater than equal to one because if n is 0 then F minus 1 there is no F minus 1. Now Fibonacci number is staring from F 0. So, we start the base case is we put a n is equal to 1 and we see whether it is true the result is true or not. So, this will see by the method of induction. So, for n is equal to 1 what is this matrix? This is we put just n is equal to one this is F 2 F 1 F 1, n is equal to 1 this is F 0.

So, this matrix is basically we know F 2 is 1 F 1 is 1 F 0 F 0. So, this is true for n is equal to 1. So, result is true for n is equal to 1. So, the base case is satisfied now we need to take the induction hypothesis where. So, induction hypothesis (Refer Time: 12:20).

(Refer Slide Time: 12:28)



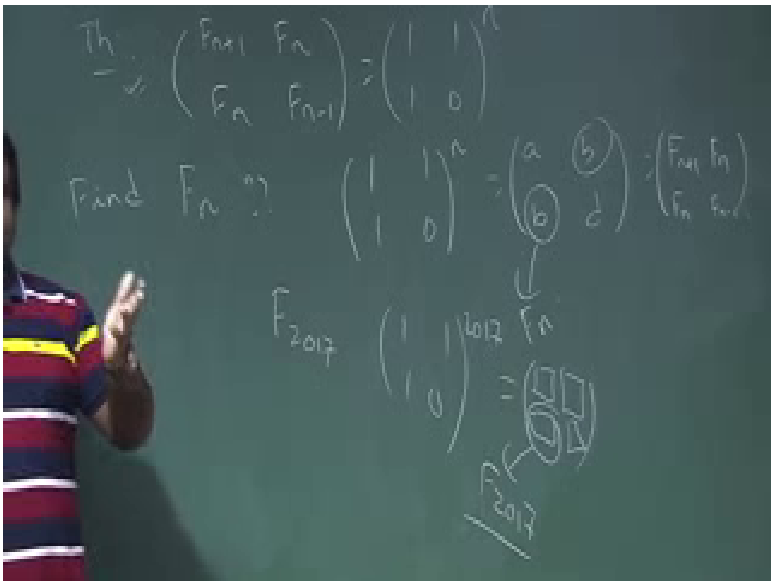
We assume that result is true for n is equal to k so; that means, we assume F of k plus 1 we put n is equal to k F of k F of k F of k minus one is equal to 1 1 1 0 to the power k. So, we assume the result is true for n is equal to k.

And now we need to prove that we need to show that the result is true for n is equal to k plus 1, then it is done for we already put the base case based by the method of induction this is true for all n now how to prove this; now to show this. So, this is our assumption induction hypothesis now we multiply both side by this right hand side matrix 1 1 1 1 0. So, F of k plus 1, F of k, F of k, F of k minus 1, 1 1 1 0 we multiply and then it will be 1 1 1 0 to the power k plus 1 ok.

So, now what is this matrix? If we multiply this with this matrix, so this will be basically. So, this with this, this will give us F of k plus 2 and this will give us, so this into this. So, this is basically F of k plus 1 and here F of k plus 1 and F of k. So, this is basically this into this F of k. So, this is to be 1 1 1 0 to the power k plus 1 so; that means, the result is true for n is equal to k plus 1. So, we assume the result is true for n is equal to k. So, from there we can show that the result is true for n is equal to k plus 1. So, on we have already prove the base case.

So that means, by the method of induction we can say the result is true for all n greater than equal to 1. So, this theorem is proved. So, this is the, prove of this theorem by the help of mathematical induction. So, this theorem is true. So, if we have this theorem how this theorem will help us to have a algorithm. So, that will see. So, for that we just. So, this is a. So, now, our goal is to find F n we need to. So, this is our goal we need to find F n what is F n ok.

(Refer Slide Time: 15:04)

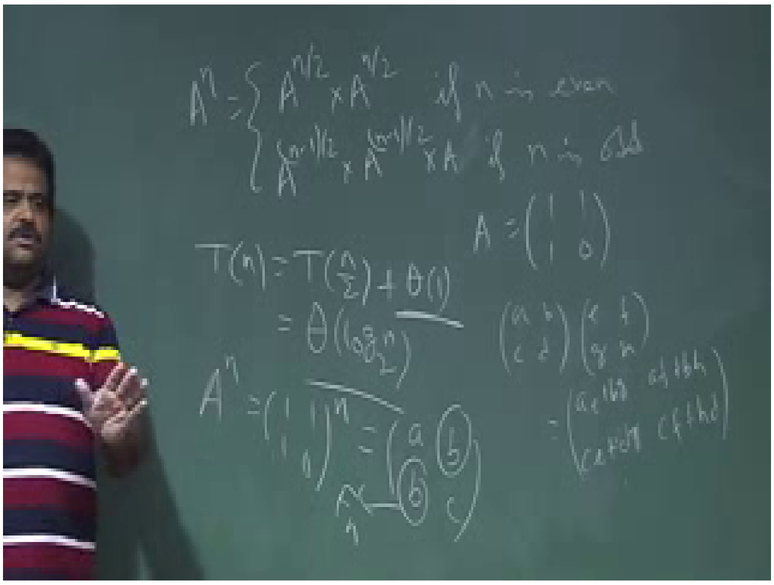


Now, to find F n we have this theorem now if we can have this to the power n, then if we can have this matrix result. So, this is a b c d then our F n is this is basically this will be same matrix, because this is basically by this theorem F n plus 1, F n, F n F n minus 1. So, from this base and we can take this as F n, so done, if you want to find F of say 2017. So, what we do we just need to get this 2017?

So, this will give us some matrix 4 by 4 matrix and this will be our F 2017. So, now, the problem is reduce to the problem of powering a matrix, but here the matrix is a 2 by 2 matrix not n by n matrix, we will see in the next problem is the matrix multiplication, but it is just a 2 by 2 matrix. So, it is very similar to powering a number. So, we can apply the divide and conquer technique to have this matrix to the power n powering a matrix how.

So, that is basically the same divide and conquer formula, which we have for powering a number here we have a 2 by 2 matrix.

(Refer Slide Time: 16:51)



So, we need to find out a to the power n, A to the power n can be written as A to the power n by 2 into A to the power n by 2, if n is even and if n is odd it is basically n minus 1 by 2 into A to the power n minus 1 by 2 into a if n is odd. So, now, this a is a just a 2 by 2 matrix 1 1 1 0 it is a 2 by 2 matrix. So, now, this is a formula for a divide and conquer approach now.

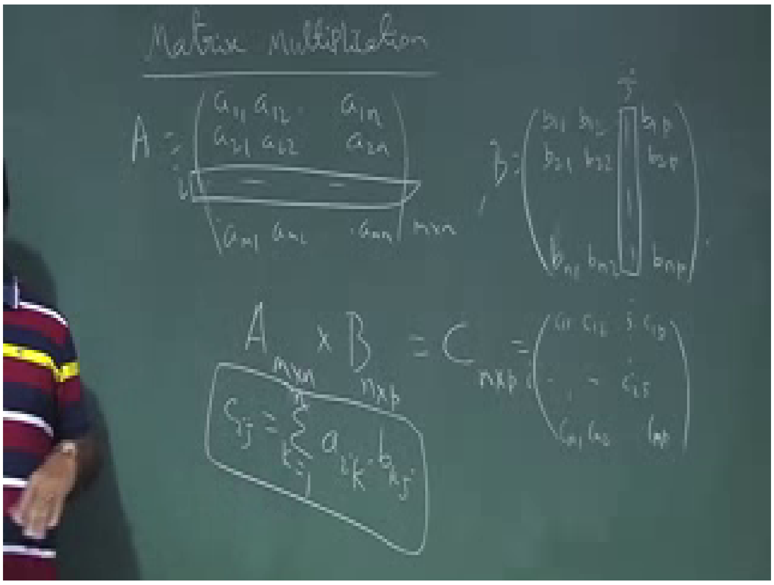
So, we have a problem of size n. So, we have to find A to the power n. So, we reduce the problem in size n by 2 now we need to find A to the power n by 2 now once we have the solution A to the power n by 2, once we have this matrix A to the power n by 2 that is conquer by conquer step then we multiply by itself to get A to the power n if it is even. Otherwise if it is odd we calculate this we multiply this with again I mean this we did (Refer Time: 18:20) a. So, either 2 multiplication or 3 multiplication then what is the recurrence, recurrence is we reduce the problem into sub problems.

But we have only one sub problems and this is the cost for doing this multiplication either 2 multiplication or 3 multiplication, but all the multiplication on the matrix of size 2 by 2 because once we get A to the power n by 2 this will be again a matrix of 2 by 2. So, if you multiply 2 matrix of size 2 by 2. So, how many had a. So, I think a 8. So, we are multiplying 2 matrix 2 by 2 matrix. So, say a b c d e f g h. So, what is the result is basically this into this, a e plus b g this into this a F plus b h this into this c e plus c e plus d g and this into this, c f plus ok.

So, this is the result. So, that base how many what. So, we are doing just this is the l number addition. So, we are doing 4 a 4 addition and this is l number multiplication and 8 multiplication. So, all together will be some constant time. So, that is why it is theta one and if it is if odd we have to take another matrix. So, that is also in constant time because it is just either 4 5 or 4 addition and 8 multiplication, but real number multiplication. So, this will all take constant and so this is a theta of one time. So, this will be same as recurrence is same as powering a number.

So, this will give us by master method log n base 2 sorry l log n base 2. So, this is the n th Fibonacci number finding. So, once we got this A to the power n in this time once we got A to the power n a to the power n is 1 1 1 0 to the power n, then this will be some 4 2 by 2 matrix. So, this is say some number like a b, b c now this quantity will give us F n. So, to find F n we just need to power this matrix this matrix is 2 by 2 matrix so, this will take logarithm time. So, this is the finding the n th Fibonacci number. So, next problem will deal with the matrix multiplication problem.

(Refer Slide Time: 21:01)



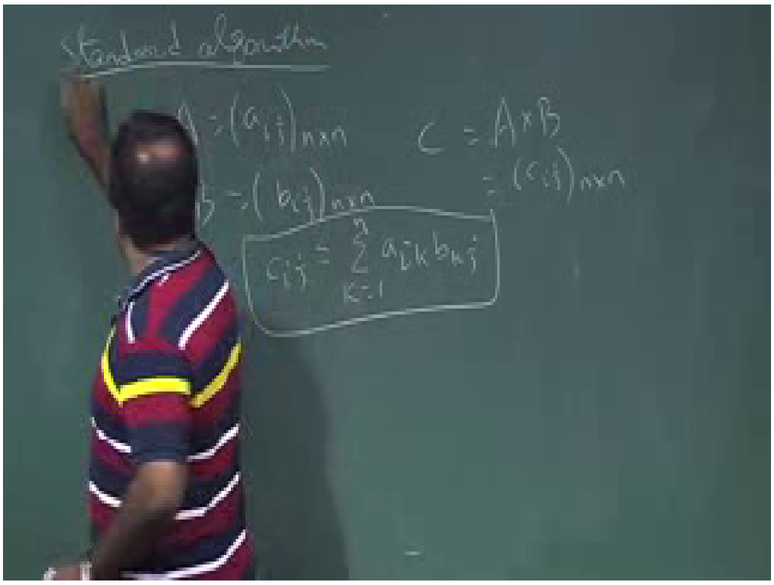
So, this is the next problem will talk about which can be solve using the divide and conquer technique this is called matrix multiplication problem matrix multiplication. So, basically suppose we have 2 matrix A B. So, a is a matrix this is general matrix say a 1 1,a 1 2, a 1 n suppose this is a m cross n matrix a 2 1, a 2 2, a 2 n now a m 1, a m 2, a n m.

So, any matrix can be written as in this form this is if you want to implement this is 2 dimensional array of size m by n m into n sorry suppose we have a matrix a and we have another matrix b which is say of size say, to have 2 multiplication enhance if we have matrix say which is size m by n and another matrix is b which is size n by p, then we can multiply this. So, this will give us a matrix C which is of size m cross p. So, what is the formula for that? So, this is basically if we have a matrix n by p.

So, this is basically b 1 1, b 1 2, b 1 p, b 2 1, b 2 2, b 2 p. So, b n 1 n row p columns b n 2 b n p suppose we have this matrix. So, then the, what is C matrix? C will be again the matrix of size m cross p. So, this is c 1 1 c 1 2 c 1 p dot dot dot c m 1, c m 2, c m p and this is basically this is if this is I th row this is j th column and this is basically c i j and what is the formula for c i j? c i j is basically we take the I th row of this and we take the j th column this is the j th column j th column are this. So, c i j is basically inner product of this two.

So, c i j small c i j is basically summation of a i k into b k j; this k is varying from the column one to m. So, this is the sorry one to m cross n n cross n. So, this is one to n. So, this is the formula for c i j. So, this we know this is the definition of matrix multiplication. So, now, we want to have a, for we want to know how we can have the algorithm to solve to have this matrix multiplication. So, here for the simplicity will take all the m n are same.

(Refer Slide Time: 24:39)

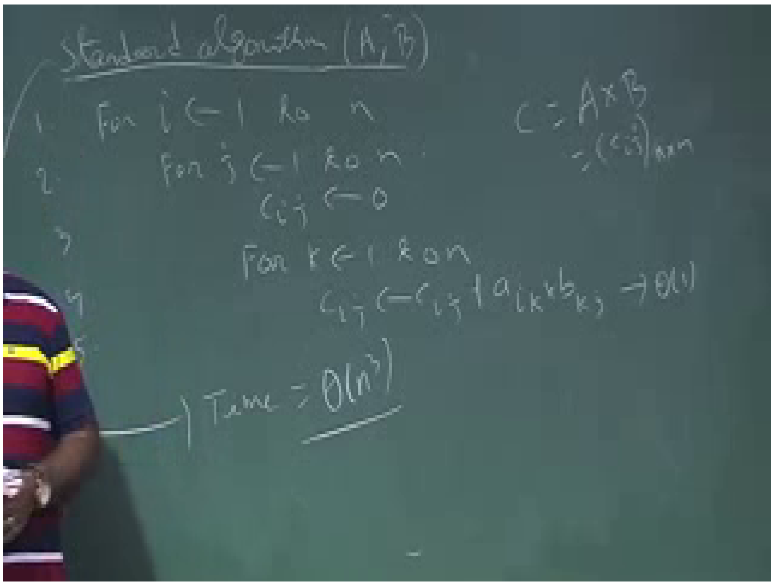


So that means, we take the order of the matrix both the matrix are n cross n and b matrix also n cross n. So, the c matrix will be also n cross n. So, basically we have 2 matrix of order n cross n and n cross n then we have to multiply.

So, this is A matrix a i j n cross n, and we have B matrix b i j n cross n and then C will be the A cross B this is the matrix multiplication. So, this is c i j this will be also n cross n. So, we are just taking for the simplicity the matrix order a n cross n. So, now, c i j is basically summation of a i k, b k j k is equal to 1 to n. So, this is the formula. So, now, what is the code the standard algorithm to find this? So, this is the standard matrix multiplication algorithm to find this standard algorithm for matrix multiplication.

So, now how we can get this c i j basically this is basically we can write in a loop. So, we have. So, basically we need to find c i j

(Refer Slide Time: 25:55)



So, we have 2 loops i is equal to 1 to j, and we have a j th loop j is equal to 1 to n and then we assign c i j to be 0 initially then we have another loop with k to find the c i j. So, this is basically for k is equal to 1 to n. So, this is basically c i j is equal to c i j plus a i k into b k j that is it. So, this is the pseudo code for finding c i js. So, we have a matrix 2 matrix A b.

So, this is the matrix standards algorithm matrix multiplication. So, this will give us a c I j. So, this will give us c matrix which is basically A into B. So, this is basically c i j, n plus n because all matrix are all n cross n matrix. So, given to matrix this is the standard code for matrix multiplication. So, what is the time complexity basically we have 3 loops each of size n and this is what we are doing constant time. So, this is basically the time for this is order of n cube.

So, this is the standard run time for matrix multiplication. So, this is the naive approach now we want to see whether we can use some divide and conquer technique to handle this problem. So, that we do in the next class.

Thank you.