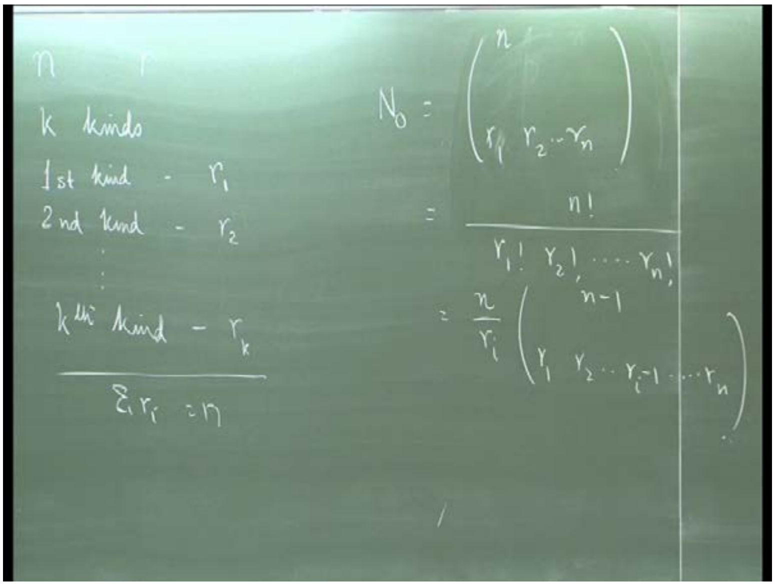
Parallel Algorithms

Prof. Phalguni Gupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 21

(Refer Slide Time: 00:27)

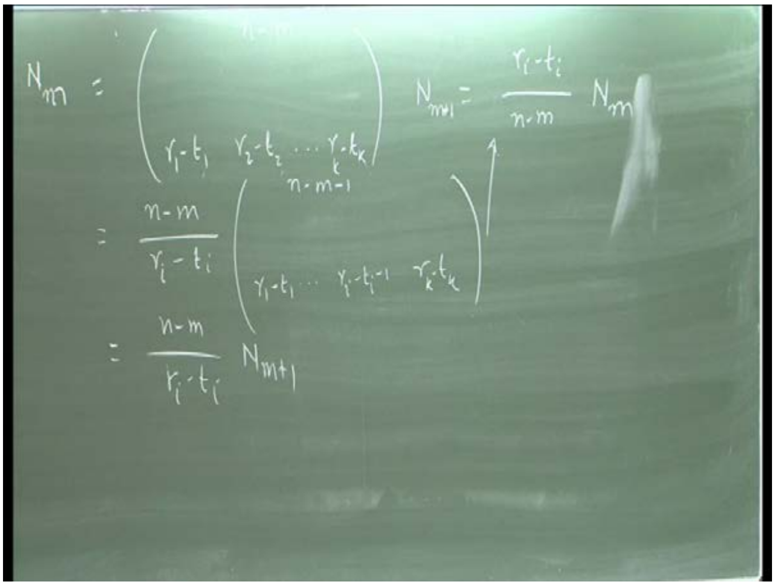


So, on electrographic, but the problem is little different, here earlier what we decide, we have the n elements, the elements are distinct and we have selected r of them or arrange using r elements, that means we have number of permutations n p r, right? Now, what happens if these elements are not distinct? These n elements are not distinct, right? Then how to generate the foundation of r elements out of n? In a ratio there are k kinds of objects k kinds of objects and first kind there are there are r 1 r 1 objects.

Second kind we have r 2 objects and k kind, we have r k objects, so that summation over r i is r, sorry is n summation of r i is n, so what is given the real objects, the objects are not distinct, the objects are of k kinds. First kind of r 1 element, second kind of r 2 element and k kind of r k elements, so that is the total kinds of this r 1 per r 2 per r k equals to n. We have to select r elements not r select. We have to make the arrangements of r elements out of this n elements, so that if N 0 is the total number of such foundations then I can write in different notation I should r 0 sorry, r 1 r 2 r n.

This is nothing but n factorial, r 1 factorial, r 2 factorial, r n factorial, right? This can be written as n by r i, n minus 1, r 1 r 2 r and r i minus 1 r n ok, so in n 0 is the total number of foundations n 0 r 1 r 2 r n, n factorial r 1 factorial r 2 factorial r n factorial. And this can be written as n by r i, n minus 1 and r 1 r 2 r i minus 1 r n, just i have taken out r i from 1 r i from here so r i minus 1.

(Refer Slide Time: 04:04)

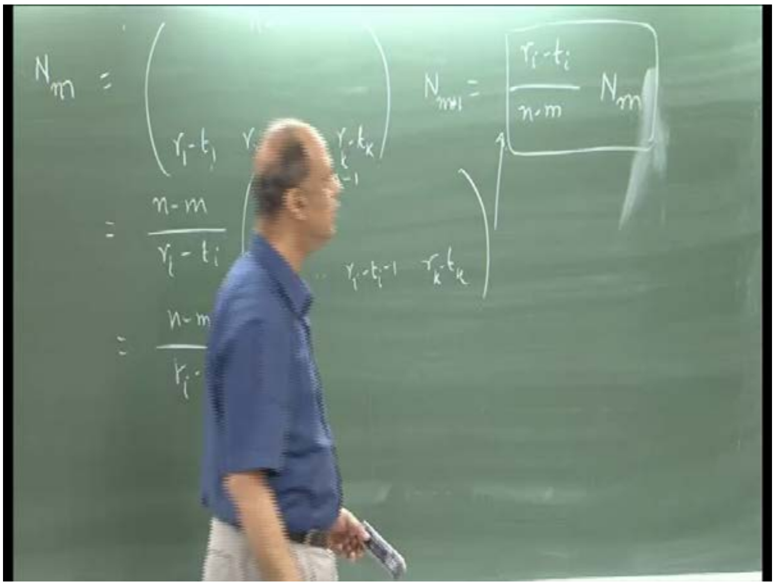


Now, suppose n m is the number of permutations where first n elements are known right, so I can write n minus m and r 1 minus t 1 r 2 minus t 2 r this must be r k minus t k. So, this is the total number of permutations you have, but first n elements are known and this n elements consist of, consisting of t 1 of first time t 2 of the second time and t k of the k th time.

So, this I can write n minus m r i minus t i, also I can write here n minus m minus 1, r 1 minus t 1, r i minus t i minus 1, r k minus t k this I can write, write this from the same one. I can also write r i minus t i, n minus m, N of (( )) so I can write from here so this is nothing but I can write n minus m, r i minus t i N of m plus 1, right?

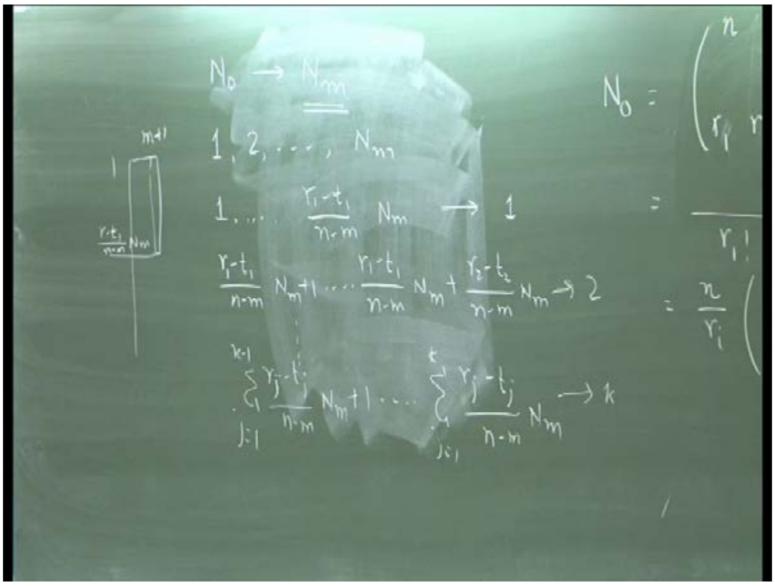
So, N of m plus 1 is equal to r i minus t i by n minus m and so what gives that if I know the first standard math, first n elements where t 1 of first time, t 2 of second time and t k of k th time. Then this many this many permutations you have whose m plus one th element is i this many permutations you have whose m plus one th element is I, is it okay?

(Refer Slide Time: 07:51)



So, this gives you the number of elements of number permutations of N m whose m plus one th element is i, so given N 0 you will get N m, right?

(Refer Slide Time: 08:04)

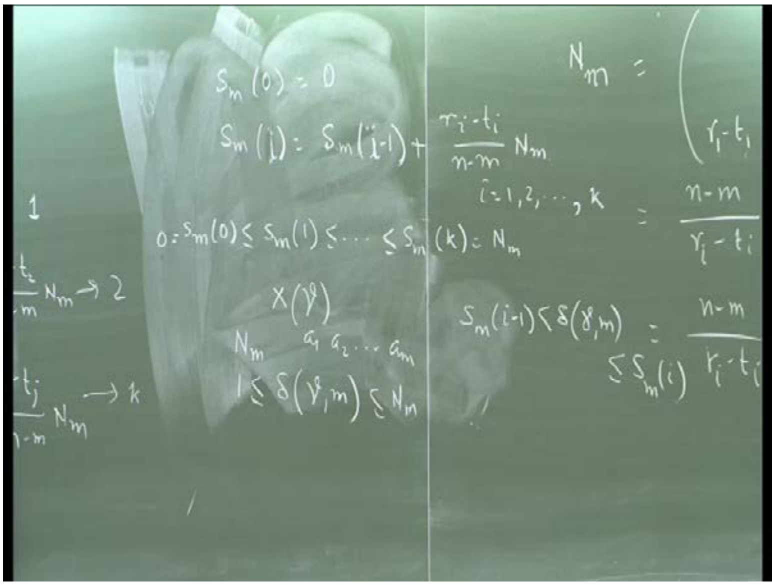


So, N m is a number of permutations whose first elements are defined k 0, k 1, k 2, k n. Now, you are thinking how to generate the m plus 1 element so that the elements are in lexographic order. Now, this N m permutation can be numbered from 1, 2, 3 up to N m right because this three permutation first element is fixed and this can be numbered 1 to Nm. Now, 1 to r 1 minus t 1 by n minus m, N m, so these are numbered the permutations are numbered 1 to n m.

Now, 1 to r 1 minus t 1 by n minus m N m, that many that many permutations whose N m th element is 1, m plus 1 th element is 1. Because they are in lexographic order 1 to r 1 minus t 1 divided by n minus m and N m that many m plus one th element would be object one first time. Now, r 1 minus t 1 by n minus m, N m plus 1 dot dot r 1 minus t 1 n minus m, n m plus r 2 minus t 2 by n minus m, N m this will give you the object of type 2.

Similarly, if I go for the summation over r i minus t i or r j minus t j n minus m N m plus 1 j is 1 to k minus 1,so this would of be object k right and I know lexographic order, since I want to arrange them so this will be obtained the thing.

(Refer Slide Time: 11:12)



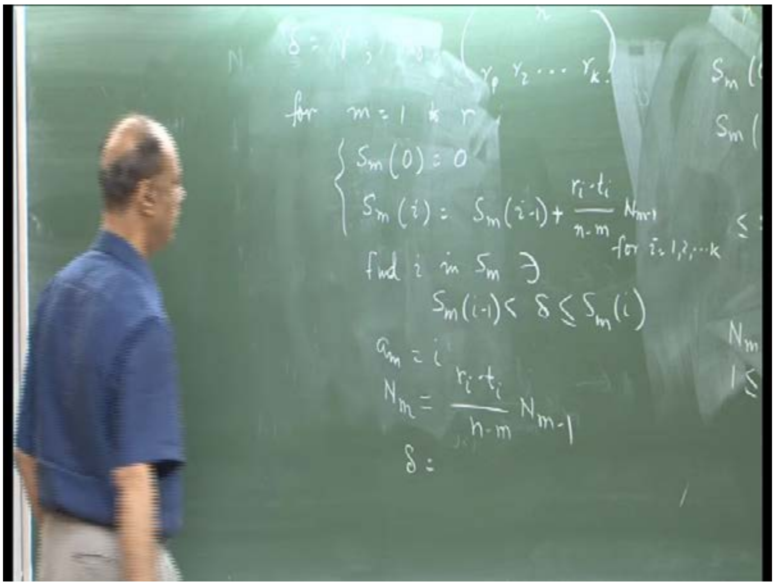
Now, both t array I can define an array S m 0 is 0 and S m, j is equal to S m j minus 1 or I can write i, i plus r i minus t i by n minus m N m I can define S m i is equal to S m of i minus 1 plus r i minus t i, n minus m N m, right? Just summation sign and basic nothing else and i can be 1, 2 to up to k up to k and so on, so from this no this I should put 0, otherwise (( )) so from this I can derive a relationship S m 0 less than equals to S of m 1, S m 0 less than equal to S m 1 less than equal to these are all positive numbers less than equal to S m k and s m k is nothing but this total number which is N m.

Now, come back to our gamma permutation I have X gamma, whose first n elements are known a 1, a 2, a n as find out the m plus one th element. Now, in this sequence there will be several sum, then we N m permutations whose first n elements are say a 1, a 2, a n and gamma is also having a first n elements so gamma must belongs to this one. So, let us assure that the delta gamma m is the index of X gamma this lexography index of X gamma in this N m.

So, delta must be lying between Nm right, delta is lying between 1 Nm and this S also lying between 1 Nm. Now, since I have to arrange in the lexographic order I have to look for an i in this sequence, such that S of m i minus 1 less than delta. That means I am looking for a smallest i such that S of i minus 1 less than delta gamma m less than equal to S m of i and that i is the object you are selecting as an m plus one th element for X gamma.

Now, clearly since this is lying between 0 to Nm, this is also lying between 0 to Nm such an i must exit, because this lying between this would be a (( )) and more word since this is in increasing order. So, you can use the binary search to find such an i, so if it is there case then I can have an algorithm to generate X gamma.

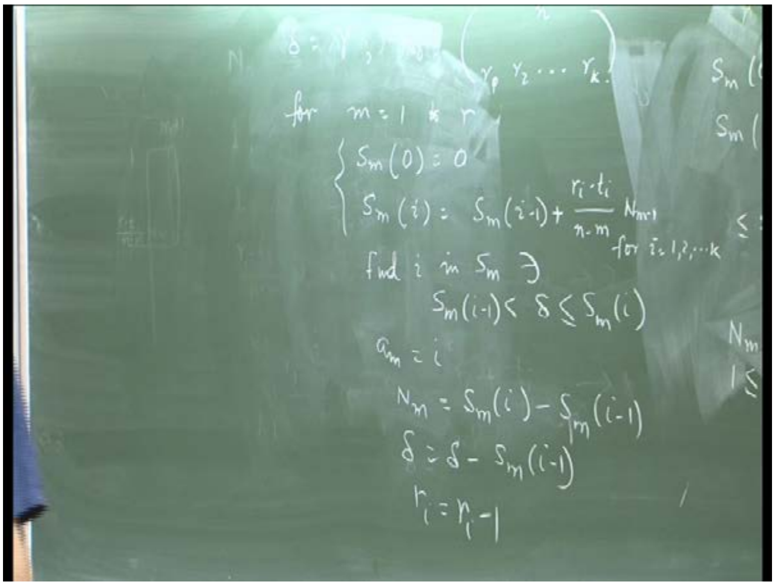
(Refer Slide Time: 15:39)



So, initially when gamma is delta, I have to obtain the gamma in lexographic order and your n 0 of competent and r 1, r 2, r k. Now, you have to find out for m equals to 1 to r, that r elements you have to right? So, first you have to define what is S m, S m 0 is 0, S of m i is equals to S of m i minus 1 plus r i minus t i by n minus m, Nm minus 1.

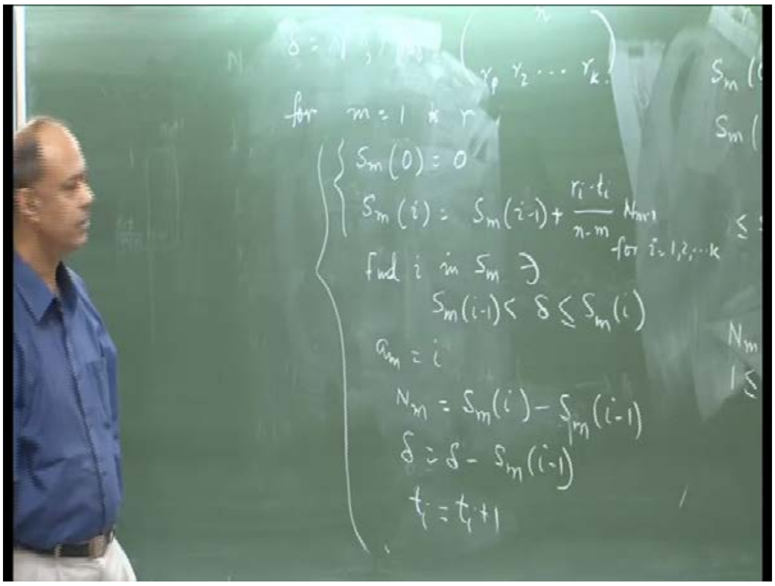
So, this is you have to find the value of S m, now once the S m is known for i equals to for i equals to i equals to1, 2, k. Now, you have to find out, find a j or find the i find the i in S m such that S of m, i minus 1 less than delta less than equals to S m of i, right? That i is important and that i is your result S of a of m is your i, now once a m is defined then now you have to modify the value of N m what will be the value of N m? Value of N m is nothing but r i minus t i divided by n minus m, N of m minus 1, so this N m will be used for next iteration, then what is the value of delta? Actually, this I do not have the compute I can straight away get.

(Refer Slide Time: 19:03)



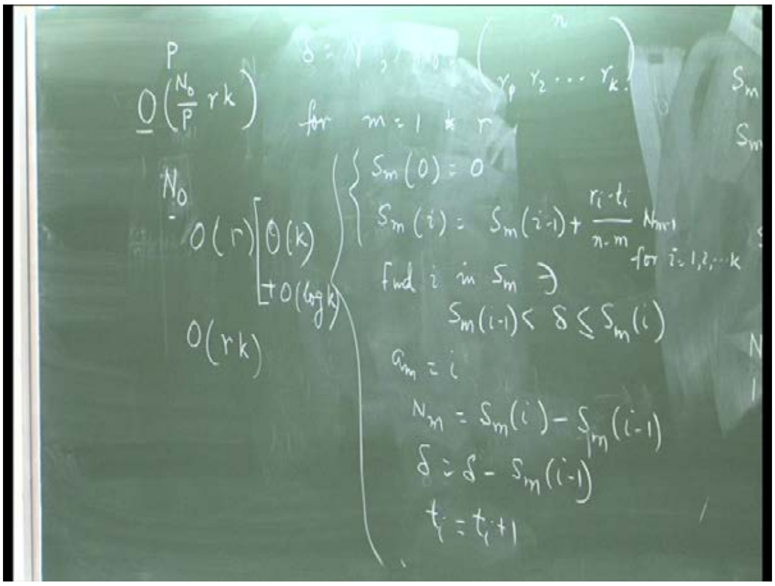
N m is equals to S of m i minus S of m i minus 1 that will give you (( )) and what is the value of delta? Delta minus S of m i minus 1, that will give you the position of delta in for the next iteration, so anything else you need? You need for, you need now the value I you selected, so r i is reduced by 1 or because r i or basically t i is increased by 1, then you increase t i, t i is increased by 1. So, the next time just we have to use t i is increased by 1 anything else? No so everything is taken care.

(Refer Slide Time: 20:12)



This is your algorithm to generate r elements of lexographic gamma X permutation, so you the you observe that is indifferent in nature, so it is take forward and another algorithm you can write.

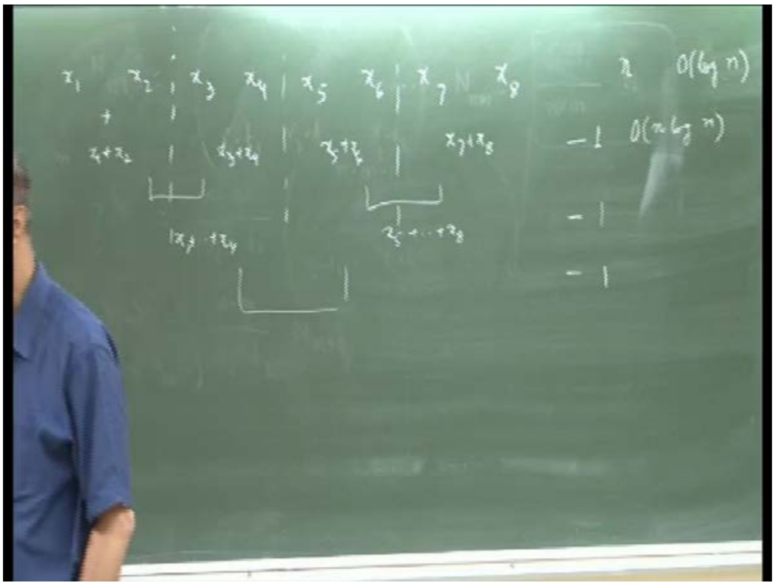
(Refer Slide Time: 20:35)



So, suppose I have n p r processor not n p r, N 0 processor that total processor you have, then you need order of time for (( )) yes. Now, the line is 1 to r. Now, observe there is an addition is included, so that addition is of size k, so order r into order k and rest of the thing this is plus order of log k, this is a binary search so order of k.

So, the complexity comes if I have N 0 processors, then order r into k because this is log k comes much less compared to order of k. Now, but since since this is this can be written in adaptive manner, so suppose I have P processor, then the time complexity comes N P r into k, correct? Now, you can look at the time complexity further by increasing the number of processor. Basically what happens let us go back to our initial finding the sum of n numbers using first we tried with using n processor.

(Refer Slide Time: 22:33)

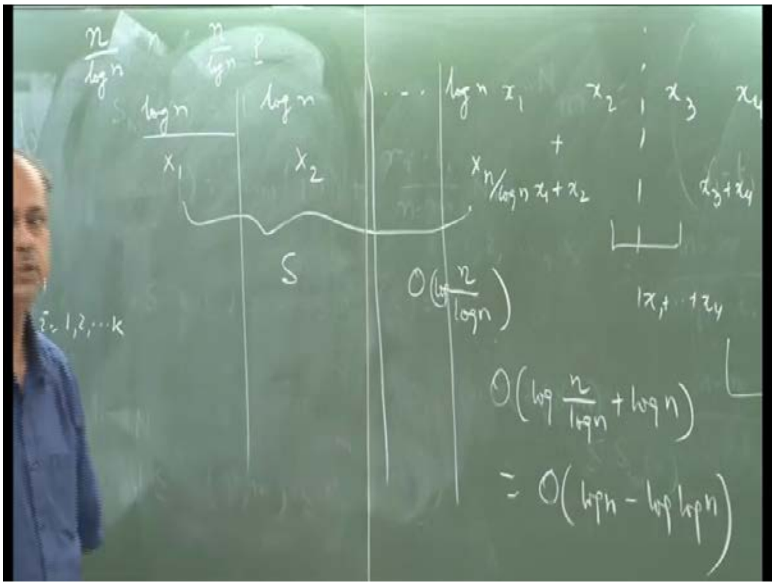


So, what happens if I have x 1, x 2, x 3, x 4, x 5, x 6, suppose there are eight number and I have the n by 2 processor right? What I need is sum of n numbers divided into four groups, each group is having two elements and I just add them, so I get x 1 plus x 2, I get x 3 plus 4, I get x 5 plus x 6, x 7 plus x 8, right? So, this is one integration then I divide into two groups and instead of four processors. Now, I am using only two processors, I add this two I get sum of first four elements and I get sum of next four elements, which is another parallel steps you need. Now, one processor I am applying and at this I get the sum of eight terms, so you need log n iterations log n additions to find the sum of eight elements.

This log n parallel additions remember do not conflicts to each other parallel, the sequential addition is number of sequential addition is n minus 1, but the number of parallel condition is 3 using four processor. And you observe you observe that every stage and discarding 50 % of the processor, so first I use four, then the two processors and then one processor.

So, if I see intense of n intense of n if you have n elements then you read order log n additions to find the sum of n numbers and the cost is becoming the order n log n, right? This into this order n log n, which is not a cost of (( )).

(Refer Slide Time: 25:17)

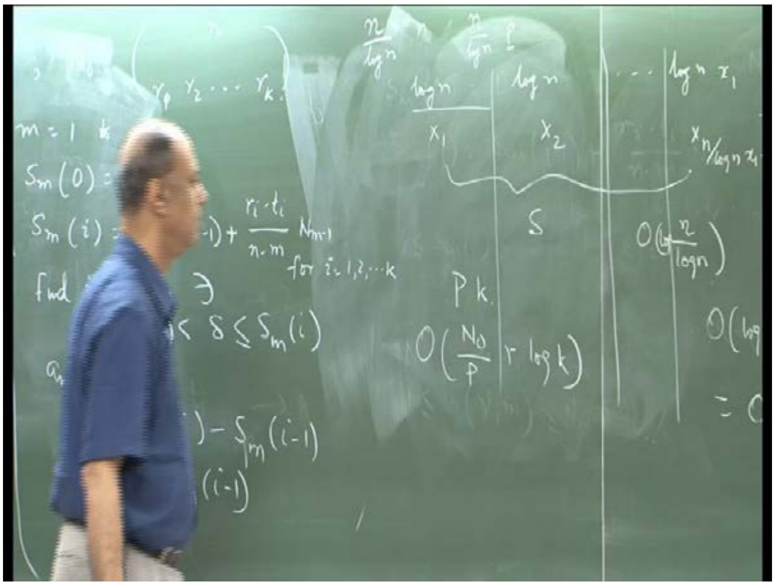


So, in real world that what we have to discussed to make a cost algorithm, I divided this n elements into n by log n groups, n by log n groups at each group is of size log n, n by log n group each group is of size log n and n log n the size log n. And I employ n by log n processors that will be processed (( )) each process is addressed to one row and it points the sum of these elements right, sum of these elements which is x 1, sum of these elements x 2, sum of these elements x n, x n by log n, n by log n, right?

Now, you have n by log n elements and you have n by log n processor and you follow the same thing same only you will be getting the sum of all these things sum of all these things in order n b y log n time, log of log of n by log n time. And also, you did the sequential relation which was taking log n time so order log of n by log n plus log n time, right? So, this I can write order log n minus log log n all are a log n time, now you observe that cost of this algorithm is order n by log n into log n which is other n time which is cost optimal.

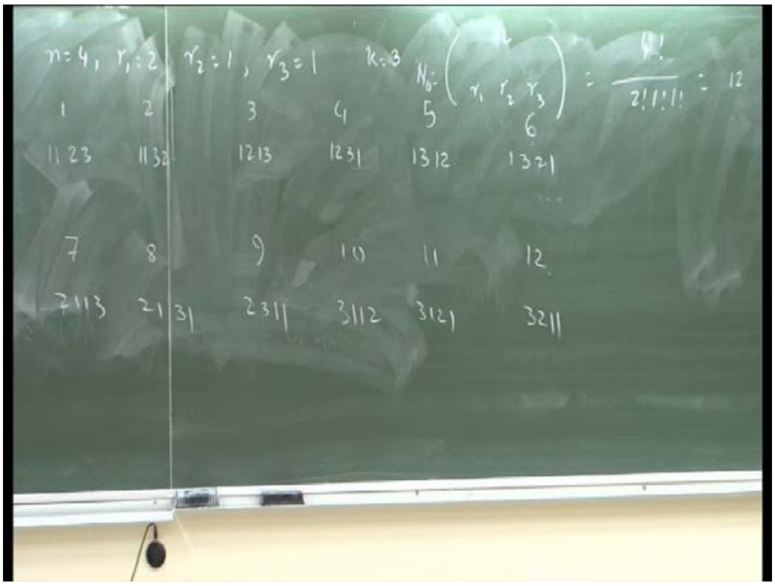
So, that I discussed in earlier class in all the earlier classes that you can have cost optimal results of this to find the sum of n numbers. So, same idea suppose I use simple thing, I use k processors of sets to make there are k elements, k things you have to compute S m 0, S m 1, S m 2, S m k, I employ k processors. So, log k time is sufficient log k time is sufficient to give you the sum of k elements, so if I employ if I employ P k processors sets, P k processors further and for each permutation I keep k processors, right?

(Refer Slide Time: 28:48)



In that case in that case the time complexity becomes N 0 by P into r log k time, because at by k processors I generate one permutation. So, if I employ k p processors, then I will be generating p permutations at time, so total number of iterations require are N 0, P is the total number of iterations and at each iterations you need r log k. Obviously, this is not cost optimal, but it is very simple to use it not that much difficult like is it ok… So this is about your permutation algorithms.

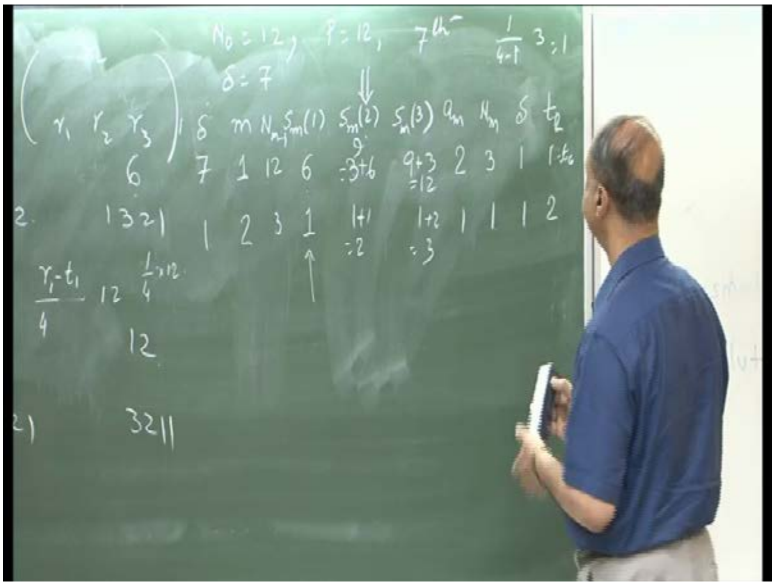
(Refer Slide Time: 29:51)



Let us see an example, suppose I have n equals to 4, r 1 equals to 2, r 2 equals to 1 and r 3 equals to 1. That means there are four objects, three kinds of objects and two of first kind, one of second kind and three is a third one is of third kind, so k is 3 what is your N 0, r 1, r 2, r 3, which is 4 factorial, 2 factorial, 1 factorial, 1 factorial that is 12.

What are those twelve elements you have? First let us see 1, 2, 3, 4, 5, 6, so this form onwards; let us understand lexical permutation is 1 1 2 3 is the first combination, then to 1 1 3 2, then 1 2 1 3, then 1 2 3 1, then 1 3 1 2 and then 1 3 2 1 you have 2 1 1 3, 2 1 3 1, 2 3 1 1, 3 1 1 2, 3 1 2 1 and 3 2 1 1, so these are the twelve permutations you have.

(Refer Slide Time: 31:49)



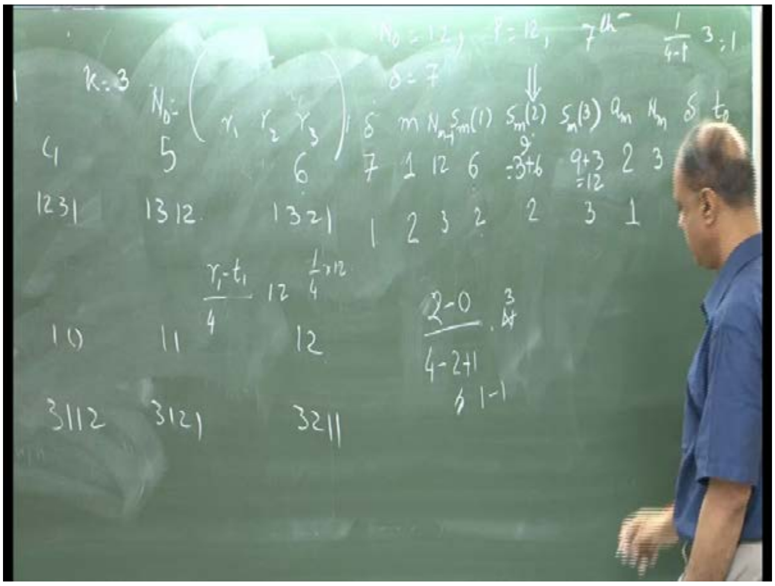
Now, let us assume that there are twelve processes you have, so you have a N 0 is 12 and number of processor is 12 say what happens seventh permutation, what it generates? So, you have delta is equals to 7, now so you have delta, you have m, you need to generate S. S m 1, S m 2, S m 3 because three kinds you need generate that, then you need to generate what is your a m, then your N m. Then delta value and then t value, delta is your N m is here somewhere else N m here, delta is 7, m is 1 here I have to write N m minus 1 and this is 12. What is S m value S m 1? S m 1 is r 1 minus t 1, r 1 minus t 1 t 1 initially is 0 and n is 4 minus n is 1 so 3.

N m minus 1, N m minus one is 12, tell me what should the value r 1 is 2, this 6, r 1 is 2, 3 and 4 of 14 fix rate, n minus m, m is 1. No, this is not right, this should not be n minus 1 that should start with 0, n minus, because t i is, I have to get 6 because the first element is 1, 6 of that. So, this can be m minus 1 this can be m minus 1 and here I have to write t is r 0, n minus 1. So, it is r 1, 2 and 6 this is 6, now what about r 2, r 2 is 1, 1 by 4 into 12. So, this is 3 plus 6 so it is 3 plus 6 is 9 and here it is 9 plus 3 because this is also again 1 or 3 is 1 so it is 12.

So, you observe here there is six ones first element is 1 that is the 6, then 3 first element is 2 that is your 3 and 3 first element is 3. Now, you have to find out i set gets several lines in that so 6 and 9 so 7 lies here, so a m is 2. Now, what is your N m, N m is that this minus this is S 2 minus S 1 which is 3, delta is 1, 7 minus 6 is 1, t 1 value is increased by 1, which is 1 t base two increased by one so it is one. So, delta is 1 m is 2 what is N m 1, N m 1 is 3 what is now S m 1, S m 1 is r 1, r 1 is now r 1 minus t 1 that means 1, 1 by n is 4 m is m minus 1 is 1 and this is 3 and that is 3, so it is1 so this is 1, right? Now, what is S m 2? S m 2 is also 1, S m 3 is also 1.

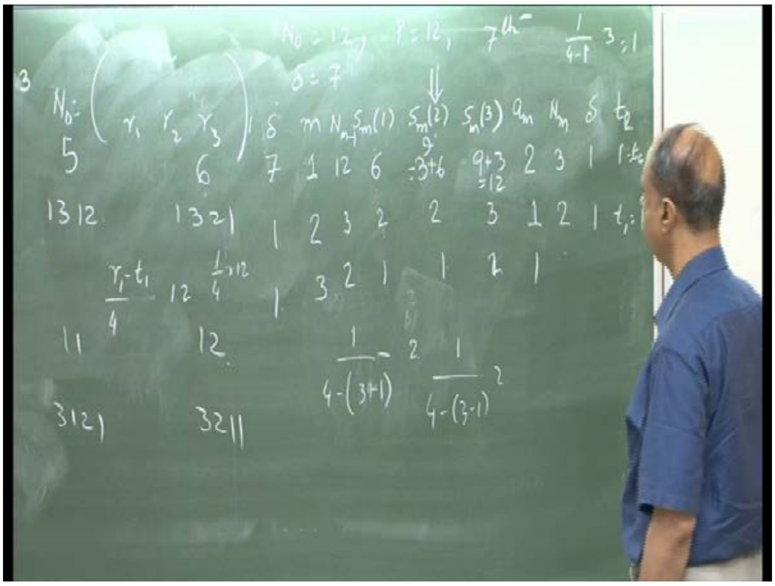
So, 1 plus 1 it is 2 and 2 plus 1 it is 3 and if you compare that it is lying here, so a m is 1 and N m is 1 delta is 1, t 1 is now 2, deduct a m is 2. So, it is t 2 is increased by 1 I made mistake here t 2 is t 2 is increased t 2 is increased by 1. Now, to solve 1 here so this is this calculation will be wrong, r 1 is 2 minus 0 n is 4 n is 2 minus 1 into N m, N m is what N m minus 1 is 3, then this is come in this so this is 2.

(Refer Slide Time: 39:50)



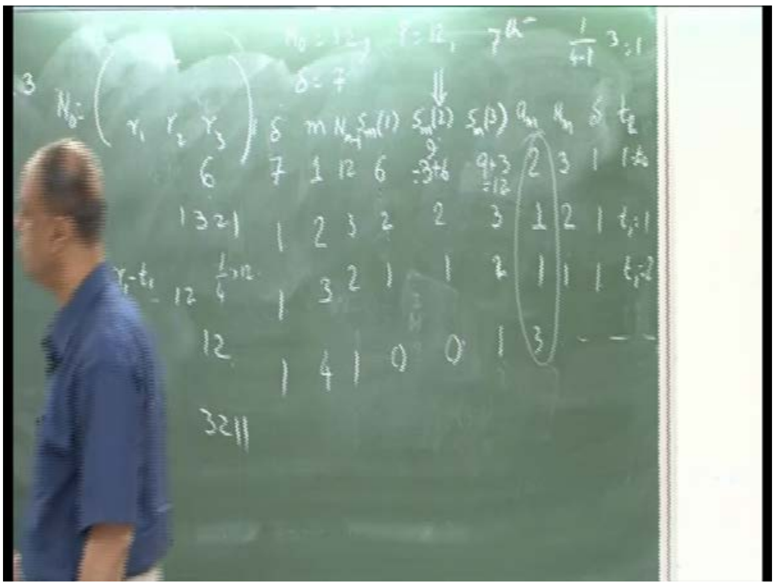
So, this is 2 and this become 2 again because r 2 is 1, 1minus 1, so it is 0 so r 2 plus this is 0 and this will become 3, a m is now 1, 1 is this a m is 1 N m is what is the value of N m?

(Refer Slide Time: 40:19)



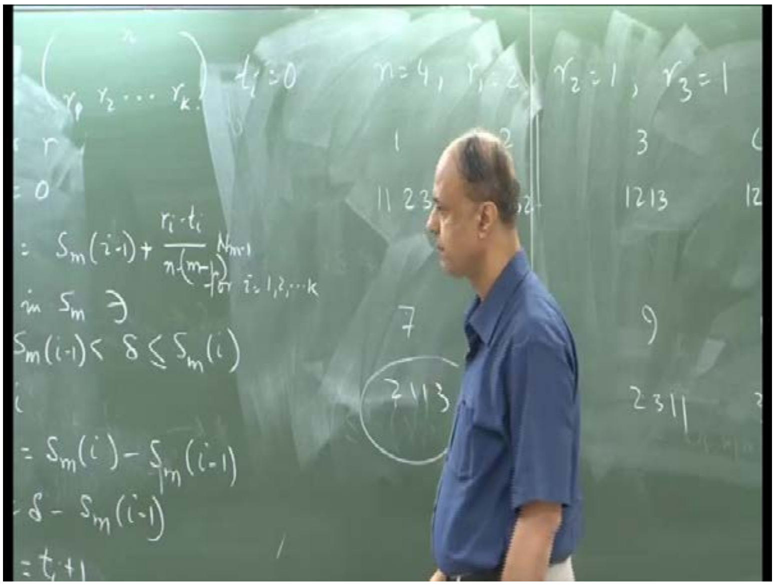
Now, N m is this minus so N m is 2, delta is 1 and here t 1 is 1 t 1 is 1, delta is 1, N m is 3, N m minus 1 is 2, what is S m 1, S m 1 is r 1, r 1 is 2 minus 1 is 1 and n is 4 minus m is 3 plus 1 3 minus 1 sorry 3 minus 1. And here it is 2 so 4 and here 2, 2 cancel 1, so this is 1. There is nothing in S m 2 is 1 and S m 3, 1 plus 1 is 2 this 1 is better of an elements. Let us consider r 3 is 1 4 minus 3 minus 1 into 2 so this is also 1, 1 plus 1 is 2.

(Refer Slide Time: 41:51)



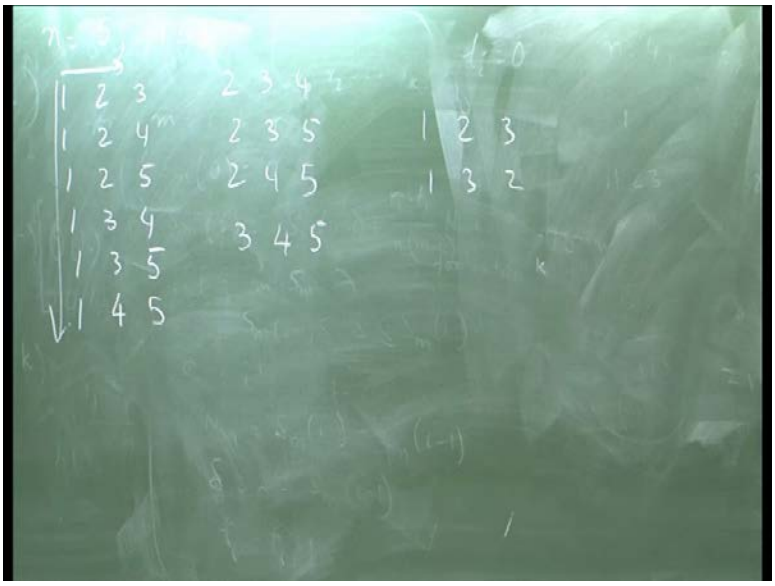
So, now 1 will be comparing so this will be in the first elements 1, N m is what N m becomes again 1 because this minus this is 1 delta is 1 and t 1 is increased, how t 1 is h is the t 1? Then this 1 m is 4, r is r is 4, N m minus 1 is 1 and this is 0, so this is 0, this is 1 and the resultant will be this 1 is 3 and the rest is not required because four elements we generated, so result will be 2 1 1 3.

(Refer Slide Time: 42:45)



So, that is your 2 1 1 3, the ratio of combination in lexographic form and you know what is combination, combination is nothing but the selection of all the elements (( )).

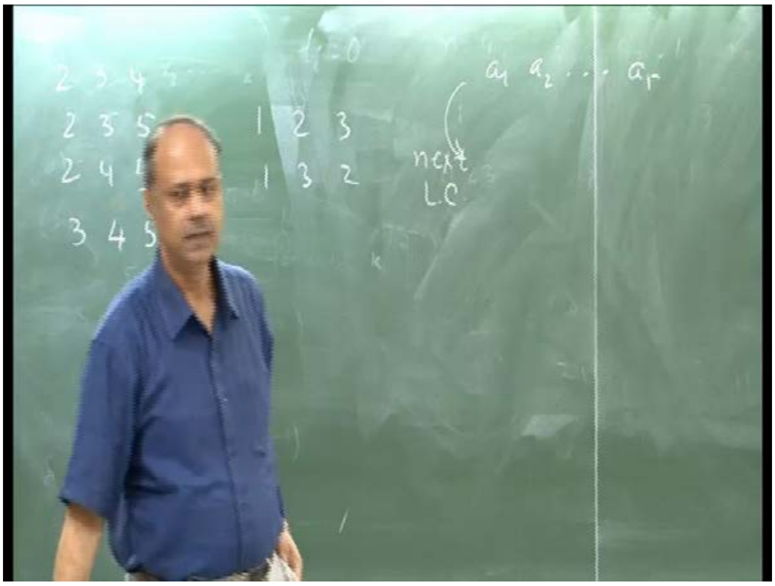
(Refer Slide Time: 43:00)



So, suppose I have n equals to 5 and r equals to 3, then there are how many combinations you have n combinations you have. Suppose, this 5 numbers are 1, 2, 3, 4, 5 then the combinations are 1 2 3, 1 2 4, 1 2 5, 1 3 4, 1 3 5, 1 4 5, then you have 2 3 4, 2 3 5, 2 4 5, 3 4 5 right this is the lexographic order.

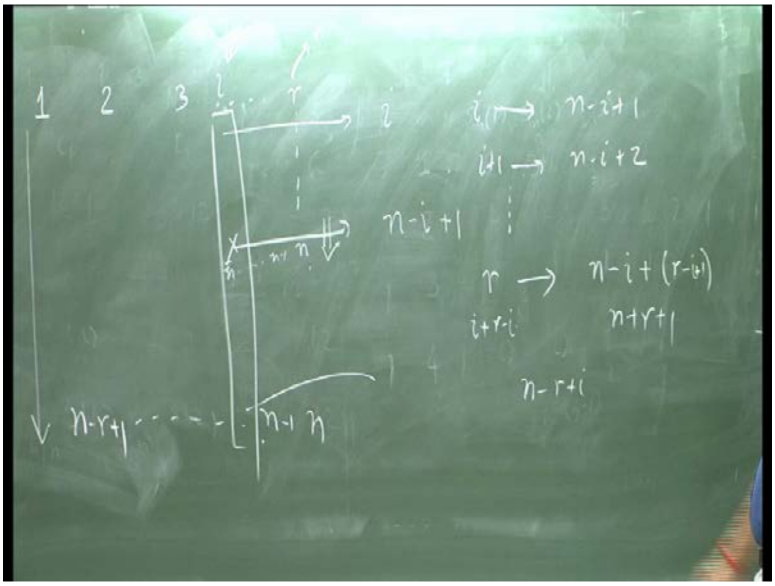
So, why lexographic order I mean you see that this is in increasing order and this is also in increasing order, 1 2 3 4, 1 2 4 1, this is in increasing order and these are another. Also that highest permutation highest, sorry highest combination is lexographic precedes the i plus format combination, because in the case of combination one thing is that 1 2 3 and 1 3 2 they are all same. Occurrence of element does not make sense in the case of combinations. So, here what we take in the case of lexographic that we take that for arrangement where the elements are in increasing order that is the idea.

(Refer Slide Time: 45:30)



Now, in the case of combination (( )) what you do given i th combination I want to generate i plus 1 th lexographic combination. So, if I have i th combination like a 1, a 2, a r, can I generate my next next lexographic combination from this that is the problem, if I can do it the problem is solved. Now, one thing you remember that first combination of r elements first lexographic combination of r elements, right?

(Refer Slide Time: 45:50)

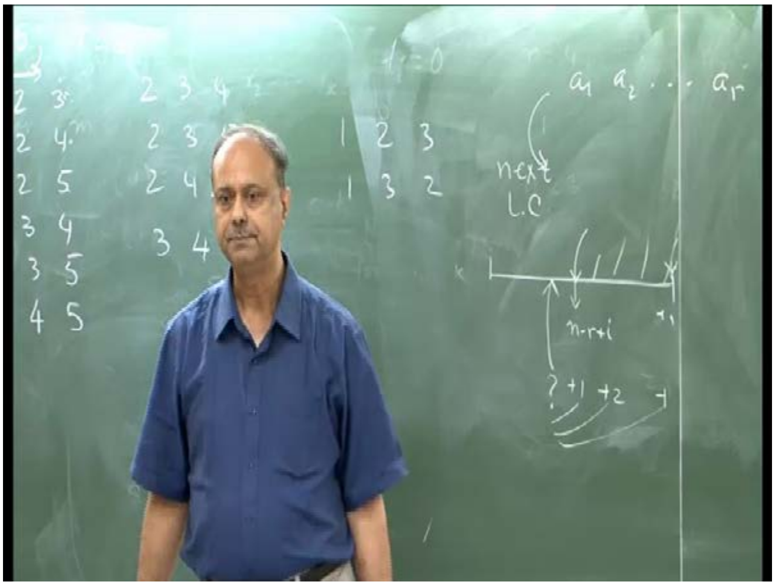


This is the first form that 1 2 3 were 1 up to r and can I find out what is my last combination, last combination must be right, this is my last combination. So, this is the first combination this is the last combination, last lexographic combination, that is 3, 4, 5 here it is n is 5 minus 3 is 2, 2 plus 1 is 3, 3 4 5. Now, can I find out what is the minimum suppose this is the i th element, i th element, now this i th element what is the minimum value, minimum value is i what is the maximum value it can retain current finder.

What is the maximum value it can take, i th position. In a position other position is n, maximum value it can take n i th position, n minus i plus 1 right is it? Now if what is the maximum value of r can take this i th position it can take n, i th position. See one thing you remember, if this is i next element will be i plus 1 onwards it is always increasing order. Like suppose, this is n minus i plus 1, then this will be plus 2, this will be. If i th position is n minus i plus 1, i plus one th position it is n minus i plus 2, r th position it will be r i plus r minus i and this is equals to n i, i is minus i plus r i minus i plus 1.

So, what is the value it is n plus r plus 1 it can take maximum i n, yes or no? This can take maximum n, so what will be the value here this is n minus 1 here it is r minus i plus r minus 1 minus, r it should be in terms of n because it is n, n minus 1. So, when i equals to r then it is n, n minus r plus i n plus i minus 1. Why minus 1 is coming when i equals to r? Next time there is n minus r plus i, so n minus r plus i. So, the maximum value of an ith location is n minus r plus i. Minimum value is i maximum value is n minus r plus i this is known now.

(Refer Slide Time: 51:06)



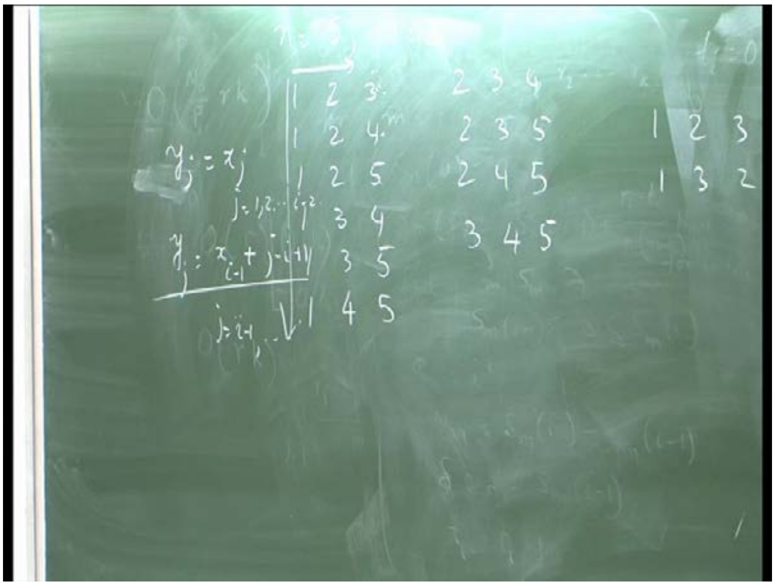
Now, the thing is that suppose I have a combination of r elements, if this last element is not n I can get easily the next lexographic 1 just by adding 1 here. Suppose, if you want to 1, 2, 3, 4 up to 3 and this is not attained the maximum, it is maximum 5. If it is not maximum just I had 1 I get the next combination, this is not maximum I had 1, I get the next 1. So, first one is clear that if the i th element does not leads to the maximum one, I can get the next combination by adding one.

Now, suppose that i th element has attained it is maximum, what it means that this value is n minus r plus I, it has attained it is maximum i th position. Remember that if i th position has attained it is maximum then next all elements also attained their maximum. Then this is maximum this is maximum, this is maximum, this is maximum, if this is maximum then all of them are maximum because this is in increasing order and one shot of the previous one.

So, if it has attained maximum, then this will be attaining maximum, this will be attaining maximum and so on. What it means that this has not attained it is maximum, just previous element i minus one th element has not attained it is maximum am I right? But i th element has attained maximum, this onwards everybody got the maximum, but this is not maximum.

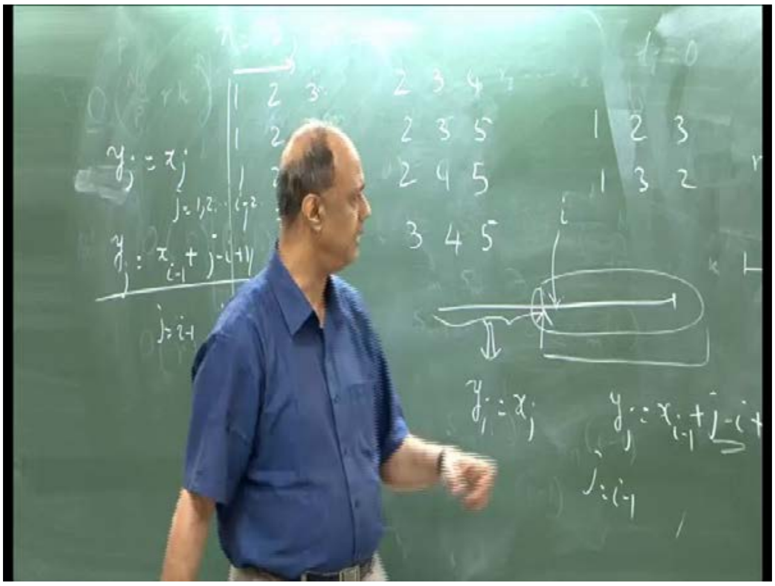
So, if this is not maximum I can generate the next combination just by adding 1 here, by adding 2 here, by adding remaining so this plus 1 this plus 2 this plus 3 and so on. So, that will give you the increasing order of combination, so if I know i minus 1 eth location where low I can add all of them to get the next one.

(Refer Slide Time: 53:47)



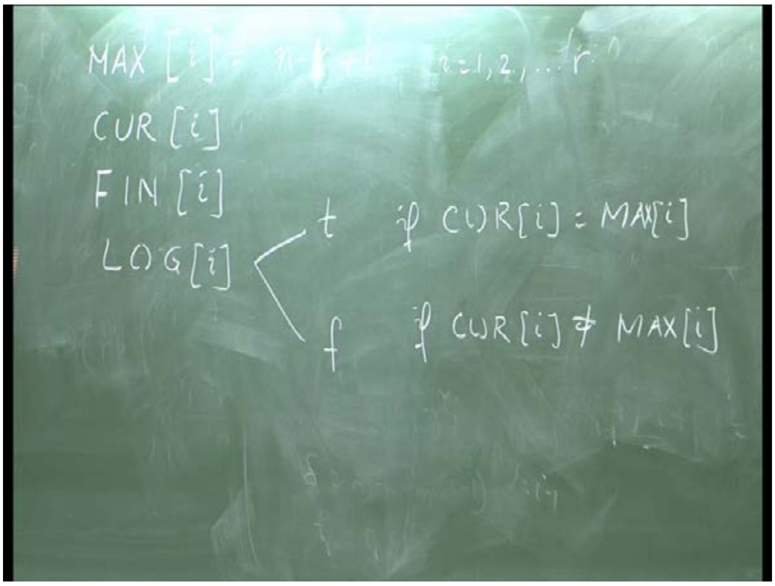
The formula is x i is equals to or let us put y j is equals to x j, x j and j is equals to 1 2 same thing I will be retaining i minus 2. y j is equals to x of i minus 1 plus j minus i plus 1, for j i minus 1 onwards, x what is the formula you will be writing that suppose this has attained it is maximum, this has not attained it is maximum.

(Refer Slide Time: 54:44)



Then this value up to this I will be retaining same, this one onwards I have to change it, that is I am writing a same that is nothing but y j is equals to x j same for this region, for this region y j is equals to x of i minus 1 plus j minus i plus 1. So, in the case of i minus 1, it is not plus 1 plus 2, because in case of i minus 1 this gives you j is i minus 1, j is equals to i minus 1, so this gives you minus 1. So, just adding 1 to the i minus one th position when j equals to i then whatever number I got plus 1, plus 2, plus 3, plus 4.

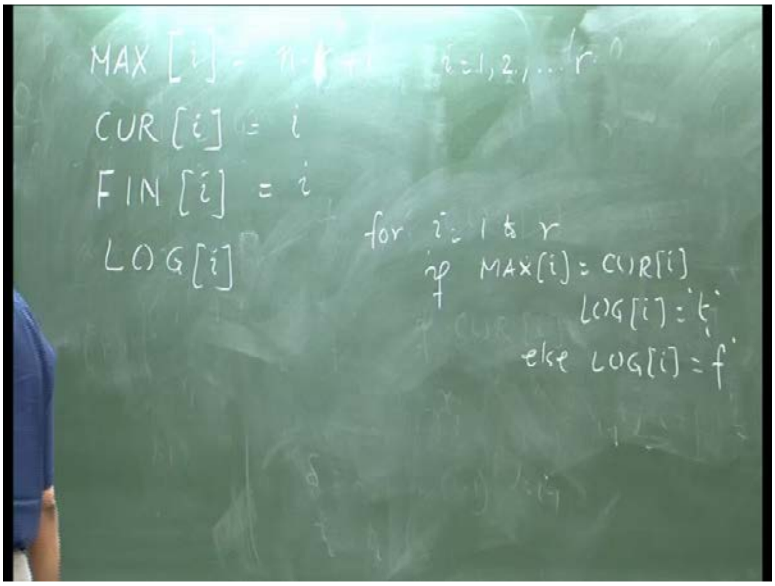
(Refer Slide Time: 55:50)



Now, let us define few arrays first otherwise it will be difficult to write alone, I know 1 n a is max, that means maximum way it can attain, so what is your max i, n minus r plus I, sorry right and i equals to am I right? 1 to r, when i equals to 1 i equals one it can take n minus just I have obtained n minus r plus 1, when i equals to r then it can take. Then I need to define a current status of my combination, so from there I will be understanding whether any position whether it has attained it maximum or not? So, current i is a current combination, final i is the array to print the combination to print a combination and then you have logical i, logical i.

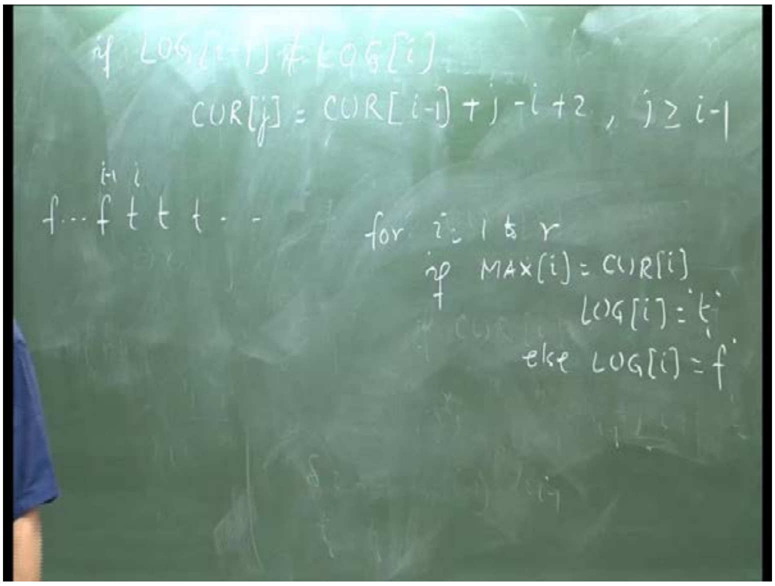
It is the i th element in logical array I will be introducing i th element whether it is attained it is maximum or not. If it is maximum, then you write false otherwise it is true, that you write true otherwise it is false, it is true if current i is equals to max i. That means current position has attained it is maximum and if current i is not equals to max i, so this four arrays I need to design my algorithm. So, initially initially max i equals to n minus r plus i, current i is i, so it is 1, 2, 3, 4, 5, 6, 7, 8 up to r.

(Refer Slide Time: 58:55)



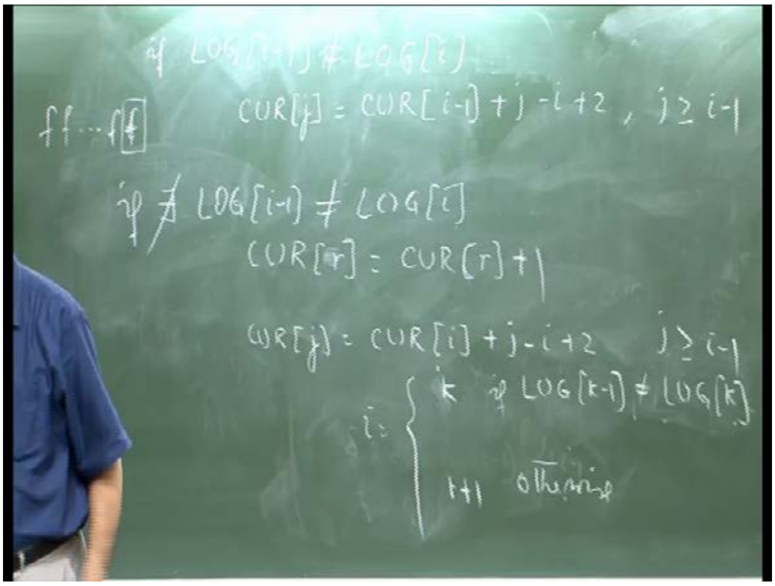
This were also will be a final result will become a (( )) 1, so i and logical i you have to check, so logical i in order to check it. For i equals to 1 to r if max i is equals to current i, then logical i is true else logical i is false so logical i is also differ.

(Refer Slide Time: 59:39)



Now, if logical i minus 1 is not equals to logical of i what it means, if logical i minus 1 is not equals to logical i. You remember one thing; that 2 comes from the right to the left all since available at the left, because I am always adjusting at the end of the combination. So, logical i minus 1 is not equals to logical i it means equals to several false, then here it is true true true and so on. So, this is i minus 1 and this is i, that means i th position onwards you got your maximum, so if logical i minus 1 is not equals to logical i. Then you can define your current i or not current i current j let me write current j is equals to current of i minus 1 plus j minus i plus 2. For all j greater than equals to i minus 1 so current j i will be defining now by just i have obtained this formula.

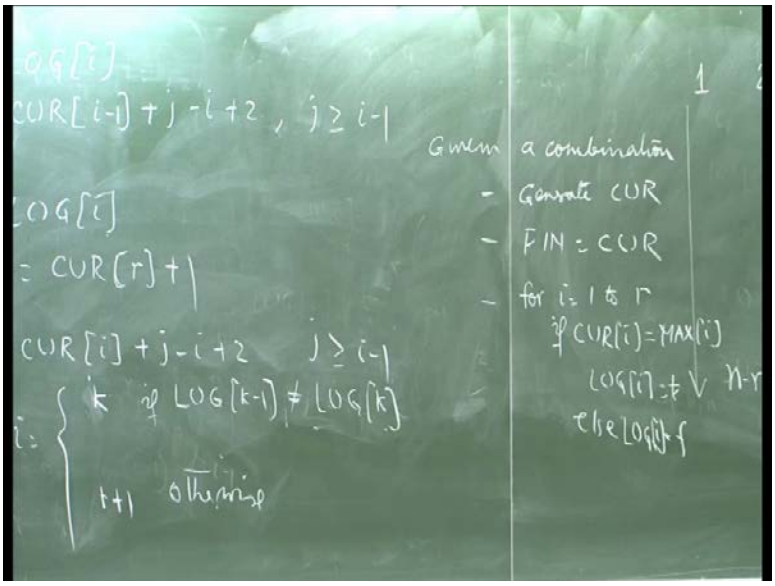
(Refer Slide Time: 01:01:25)



Next one is if does not exist logical i minus 1 knot equals to logical of i, if they does not exist in the search then what does it mean, that means you did not get a condition where you have false false false false and then t, you did not get such a condition. That means this is also false, that means the last element has also not attained it is maximum, so in that case, I can always write current of r is equivalent by current of r plus 1 just i increased.

So, in that case combining this two can I write current of i is equals to or current of j is equals to current of i plus j minus i plus 2. For j greater than equals to i minus 1 I have to define. Now, what is j? What is i? i is just, I am combining the result j. If logical k minus 1 is not equals to logical of k and r plus 1. Otherwise, this I have used this two statements to define the new one. So, given a sequence or given a combination I just check based on the biological statement and current statement I can get the next one which will be my final.

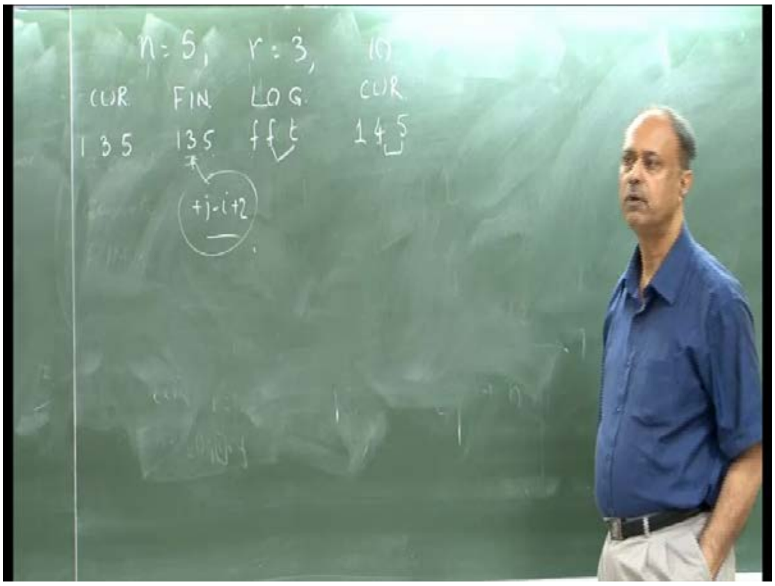
(Refer Slide Time: 01:04:02)



So, which I am tell given a combination, generate current based on this and then your final is your current or print current or print final and then you have to update your logical. So, if for i equals to 1 to r if current i is equals to max of i then logical i is true else false. So, this will give you whole level to do of that given a combination you can easily obtain what is my next lexographic combination. Now, if I have r cross sets so for each of this is if I know one cross set for allocated to one element r elements, so r processors for generate r elements, right?

So, everybody looks about this i th position checks is position of i th element so constant time is required to generate next combination. Since, they are n c r combination so you need n c r time to generate n c r combination using r processors it is not fast optimal. I think this is very simple, but there also for completeness sake let us consider one example.

(Refer Slide Time: 01:06:32)



Suppose, I have n equals to 5 and r equals to 3, n equals to 5 and r equals to 3 we have n such combination. Now, let us assume that I have current, I have final, I have logical, now say let us assume that 1, 3, 5, is my current (( )), so this is printed 1, 3, 5. Now, 5 has attained it is maximum, so it is false, false? True. Now, what I will do generating my current that is nothing, but once this is differ this is same. So, no change here this is different, then this will be incremented by 1 and the next element will be added by 1 that is 4, 5 this plus 1, instead of added by 1 what you did this element will be read.

This will be read by each processor, this will be added with j minus i plus 2, so everybody can compute j minus i plus 2 and add with this value. So, then this takes order one time, but only issue is here that each processors must be able to read a location simultaneously. So, concurrent read should be have, so this is all about your combinations.

Thank you.