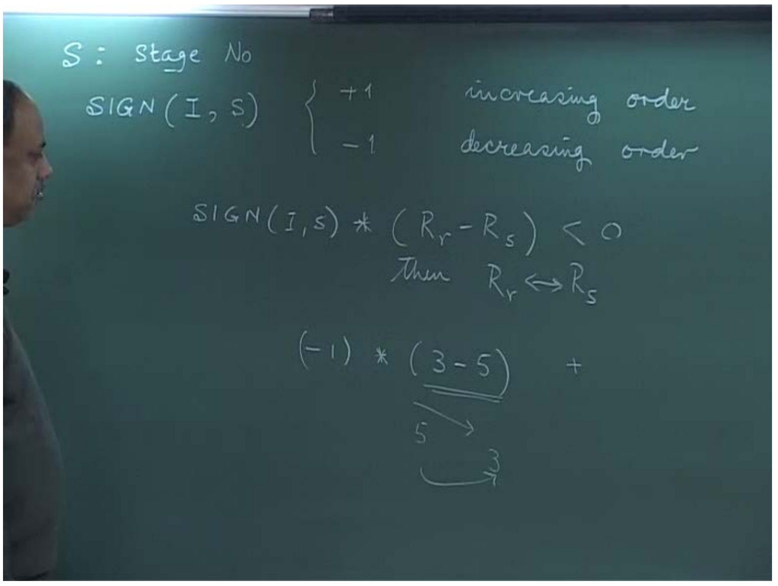
# Parallel Algorithms
Prof. Phalguni Gupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

## Lecture - 11

(Refer Slide Time: 00:50)

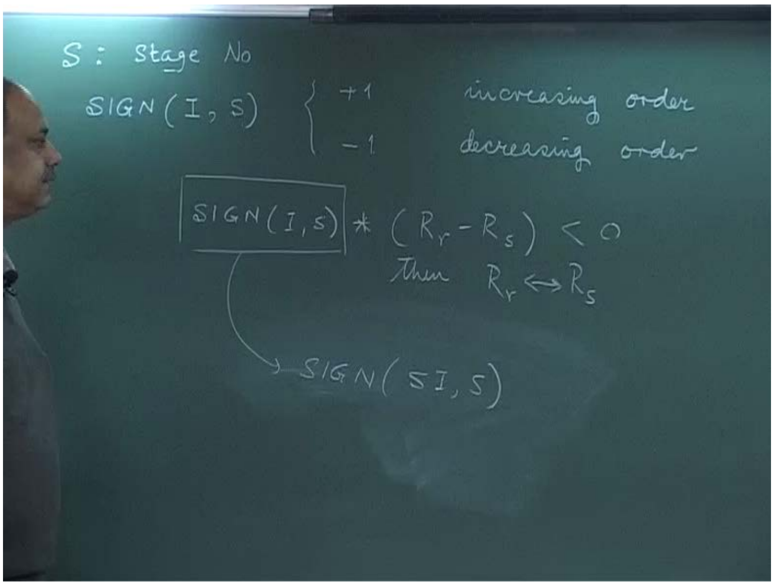



Now in the last class we discussed that how to sort an element using bitonic merge technique on machine on two-dimensional mass connected; here we yet to, we have not yet done discussion on that how to do with it in increasing order or decreasing order. If you remember otherwise, we mention s is your stage and depending upon the stage number you have to decide with it a data you want to arrange with it is. Now for that, we introduce the term sign, the index comma s sign index comma s if it is positive 1, then increasing order, negative 1 it is decreasing order.

Now you know to compute whether it is you have to arrange it in increasing order or decreasing order, what you do that is sign I comma S into your R r minus R s to find this is less than 0. Suppose sign is plus 1 and told you that rejected element will be in R r register and accept element will be r s register sign is suppose plus 1 than r r should (( )) larger element or not; plus 1 is increasing order, plus means it is in increasing order. So r r should contain the rejected element r r is less than r s then only you will get the negative value, because this is plus 1 and if you get negative value that means r r is less than r s.

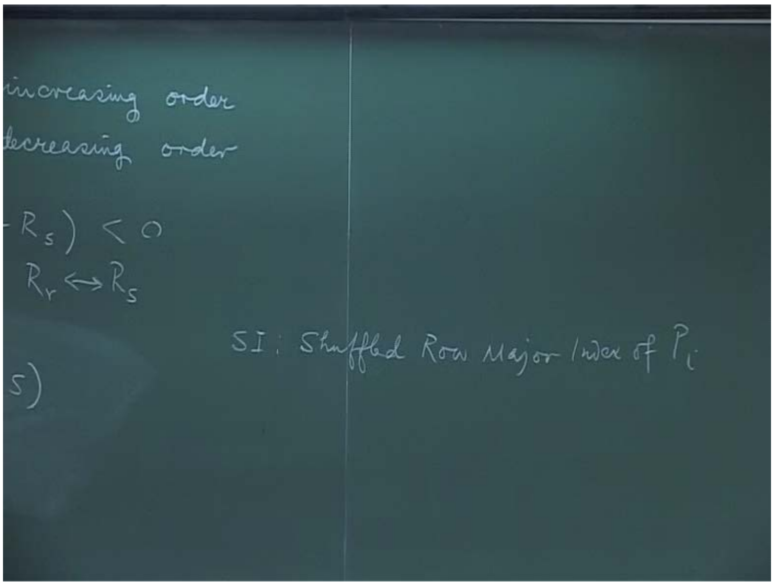
And that means r r is containing the accepted element and r s contains the rejected elements. Yes or no? Yes it is. If yes, than just you inter change r s and r r. So, what do you do after in your routine, wherever you all use in base routine (( )) you check sign of i s star r r minus r s less than 0. Then r r should be interchange with r suppose sign is minus 1, that means it should be arranged in decreasing order and so, this is minus 1 and this is 3 minus 5. Suppose r r is 3 and so, you got positive value so, r r is the rejected element yes r r will be this 1 because this is positive, right. Then this is positive so, you are going to arrange them into decreasing order. When they are arranged in decreasing order, 5 should be kept in your area and 3 should be sent back.

(Refer Slide Time: 05:00)



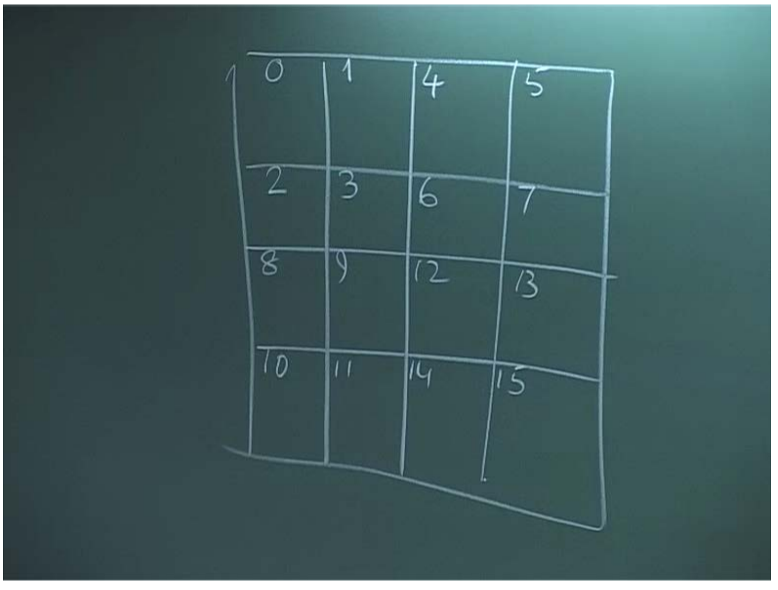

So, you find it is less than this, then you interchange. Now how to compute this 1 that is the problem yet to be solved.

(Refer Slide Time 05:11)

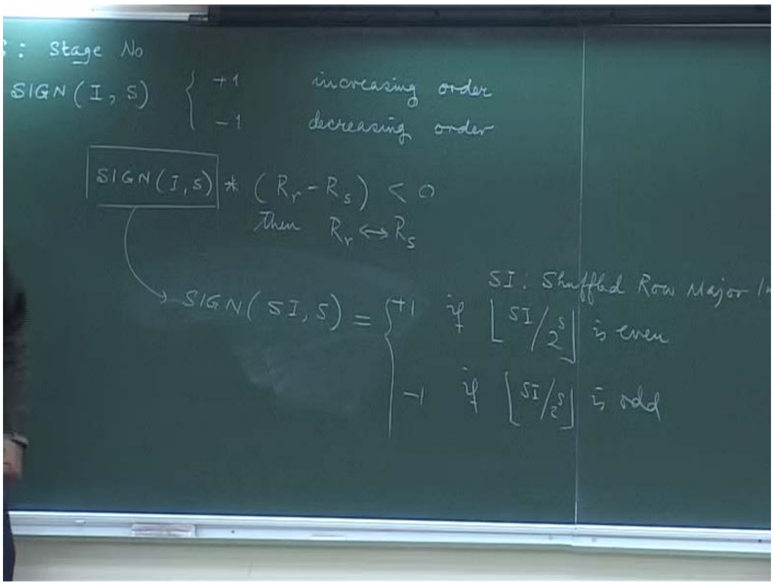



So, the definition is sign of S I is shuffled, row major index of (( )) P i; so each processes, if you remember the shuffled row major index.

(Refer Slide Time: 05:41)

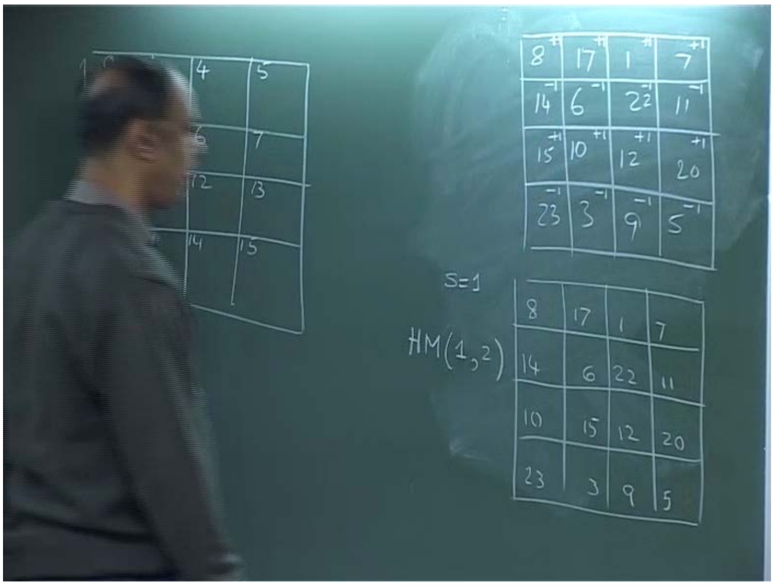


(Refer Slide Time: 07:03)




So, in shuffled row major index aim, can you tell me where will be the (( )) ? Tell me. p 0 is here, p 1 neighboring 1 p 2,p 2is here, p 3 here, p4 here, p 5, right. So, the s i is the index of the p I, the shuffled index of p i. So, sign is plus 1. If s i by 2 to the power s, it is even and minus 1 if s i by 2 to the power s is odd.

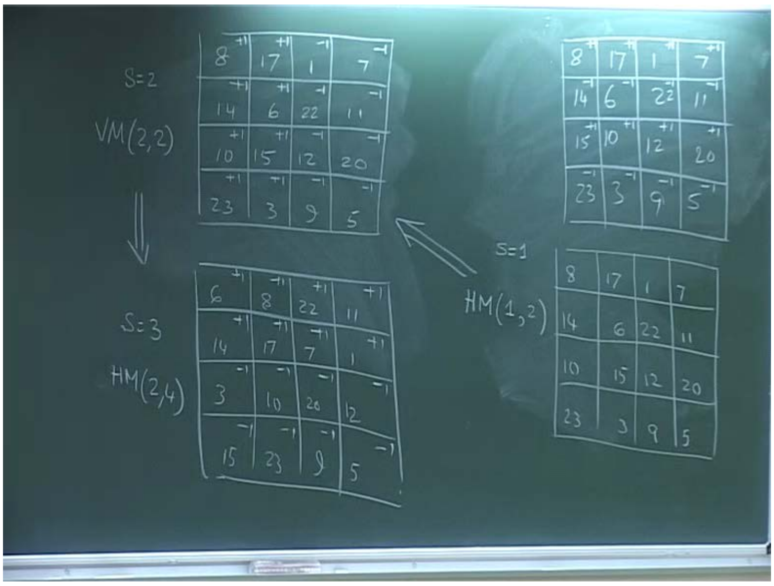
(Refer Slide Time: 10:33)




Let us see how it was, that means (( )) random number the first number 8, then 10, 1, 5, 6, 11, 20, 3, 23 (( )) 7, 9, 12, 14, 15, 17. So, let us assume these are the 16 numbers you have and they are evenly distributed. So, s equals to first k, s equals to 1 if you remember and that so, s equals to, let us do it here, s equals to 1 first you introduce 0 by 2 to the power 1, ok, which is 0.

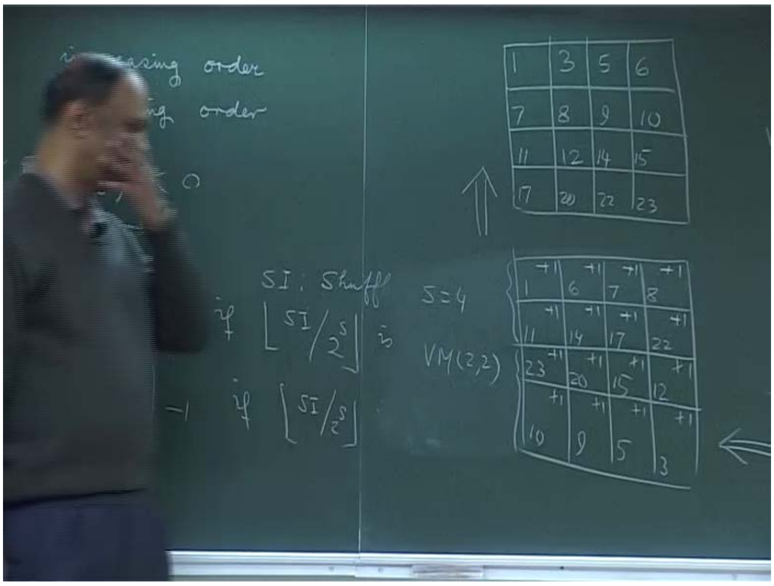
So, 0 is an even number. 1 by 2 to the power 1 is also 0. So, assume that this is also 1 plus 1 plus 1 now 2 by 2 to the power 1 is 1, which is odd. So, minus 1. This is also minus 14 by 2 to the power, 1 is even. So, this is plus 1 this is plus 1 this is 5 by 2 to the power 1, 6 by 2 to the power 1, 7 by 2 to the power 1, 8 by 2,to the power 1, 9 by 2 to the power 1, right. So, at the s equals to 1 at the first stage you get this.

(Refer Slide Time: 14:19)




Now you do that you are doing the horizontal model k equals to 1 to k. So, it is 8 17 1 7 14 6 22 11 10 15 12 20 23 3 9 5, right. So, next time you will be calling vertical merge 2 comma 2, before that we have to pick here s equals to 2 vertical merge 2 comma 2. Now 1s s equals to 22 to the power s 2 so the power s is 4, right. So, 0 by 4 is plus 1 plus 1 plus 1 plus 1. Now, this is 4 by 4, is 15 by 4, 6 by 4, 7 by 4, 8 by 4, 9 by4, 10 by4, 11 by4, 12 by4, 13 by 4, 14 15. Actually you have these data 8 7 1 7 14 6 20 2 11 20 12 15 10 23 3 9 5. So, if you do the vertical merge on this, so, you will be getting 68 14 17 20 2 11 7 1 3 10 15 23 20 12 9 5. Now you will be, you are in s equals to 3 and horizontal merge 2 comma 4, yes right. So, 2 comma 4 s equals to 3, 2 to the power 3 is 8. So, up to (( )) index project it will be 0.

(Refer Slide Time: 14:53)

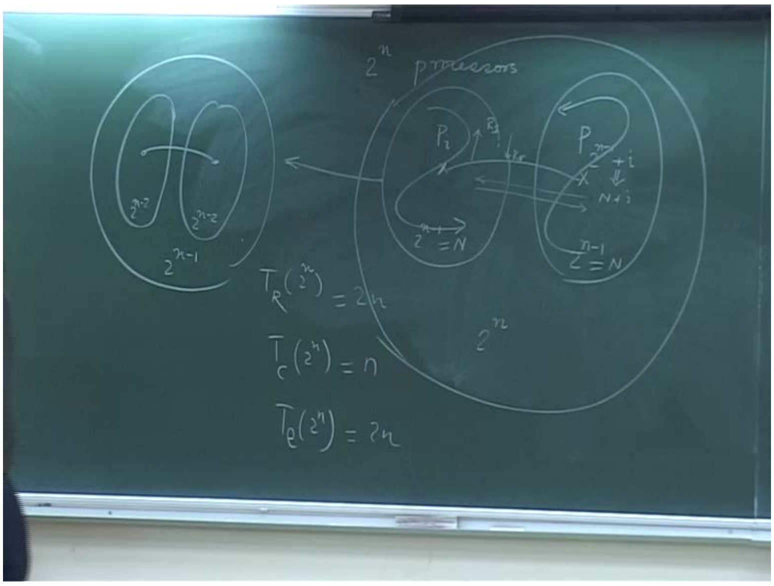



This is 0. So, it will be plus 1. So, this will be plus 1 plus, this is minus 1, minus 1, minus 1. You observe that this is increasing order, this is decreasing order. So, you can do the bitonic merge horizontal merge.

So, you get 1 6 7 8 11 14 17 20 2 3 10 20 3 20 15 12 10 9 5 3. Now, you have s equal to 4 and you will be doing vertical merging 2 commas 2. So, basically 2 to the power of s is 16 and all the (( )) will have 0 plus 1, right. This is increasing order, this is decreasing order. So, you can use the vertical merge and will be getting.

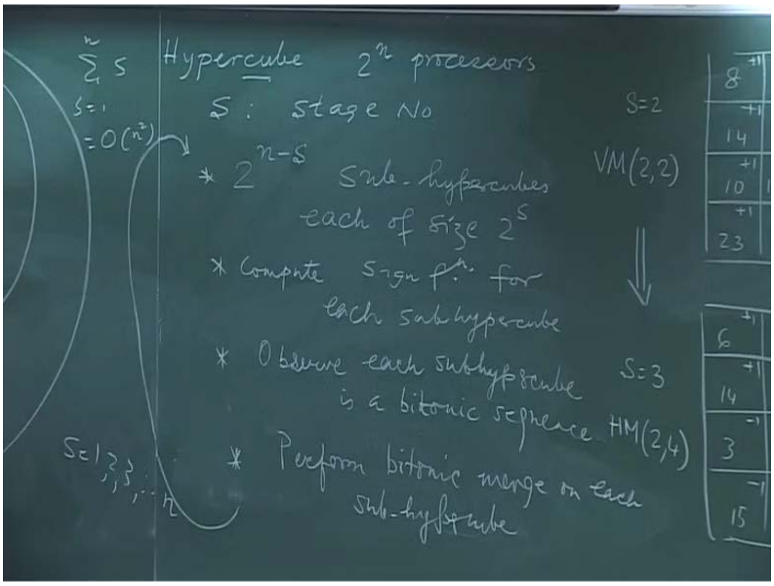
(( )) How to do that vertical merge? What is that vertical merge first palmer (( ))? So, you will get first, can you do it at, we will be getting 1 3 5 67 8 9 10 11 12 14 15 17 20 22 23. This is all about connected computes. Now can you tell me another model where bitonic merge or bitonic salt implemented (( ))? Can you give me another model where pytronic merge or bitonic salt can be implemented very easily? Let us start 1 by 1 what is the next model (( )). Let us think hypercube is what tell me? What is hypercube is.

(Refer Slide Time: 25:30)




If you have 2 to the power of n process, this is 2 to the power of n process. This contain 2 to the power on n process, this contain 2 to the power on n minus 1 processor, and each processor was connected exactly 1 processor of the other side of hypercube whose (( )) different. If it is the case the processor index is pi. Then can you tell me what is the processor index of this p 2 the power of n plus 1 plus i if in the pi this index is this (( ))? No 100 percent. So, if this is capital n, this can be written 10 as n plus I, agreed. Now suppose this is in increasing order and this is decreasing order, this is increasing order this is decreasing order, the existed is connection between these 2 processors. I can get this data here. Compare and (( )) rejected 1 here. Yes according to the bitonic sequence property in that case this will be a bitonic sequence this will be another bitonic sequence. All these elements were smaller than every element of this, yes. If it is the case now this I can think about right, and I can proceed for this. So, in order to do that what should be the time publicity for merging a bitonic sequence of size 2 to the power n on a hypercube of size 2 to the power n? Tell me, is it a difficult proposition? This is clear, now, yes or no? This is the bitonic sequence of n size 2 to the power n on a hyper cube of size 2 to the power n. pi is connected to pi plus 2 to the power of n minus 1. So, I can bring this data from here to here. In 1 shot send back the rejected element of (( )). This entire element is smaller than this and this is the bitonic process and this is the another bitonic, agreed. Now recursively you call this 1 simultaneously. You call this 1 also to merge. Now if it is the case so, can you tell me what is t r 2 to the power n, t c 2 to the power n, and t e 2 to the power n? You do not need any, what is that, at this time you do not need any module. What is that malt? Thus you bring this, right. Tell me what is c r? How many routine you need this is better especially are a register 2to the power n (( )). What is distance why distance means that where distance what is that? This is 1 up only because every process has looking connection and this is 1 of them you do not (( )). This and this and this may be wilding, this a little more compare to this sub hypercube of this and we assume that this link the speed of light. So, up (( )) that increasing the length is not that much effected. So, let us assume they are all here. Initially r r this is move to r s. We will be bringing it here right, and then rejected element will be sending back and then you will be putting it again in r r register. This is the step right, so, every time we will be getting 1 comparison and it is hobbling it hobbling it hobbling. So, how many comparisons should be given (( ))? Then this is clear. What about this 1 routine, how many routine, right. Bitonic marts on hypercube can be d1 now can I use this idea for sorting. You observe it is basically bottom up and while at the higher bottom you will be using top down. That is the idea. Every way in the stage of soughting here it is increasing order and you put outing, that this mode at the higher between the higher path there will be n by i plus 1 merging techniques of sequence will be there right.

(Refer Slide Time: 32:06)



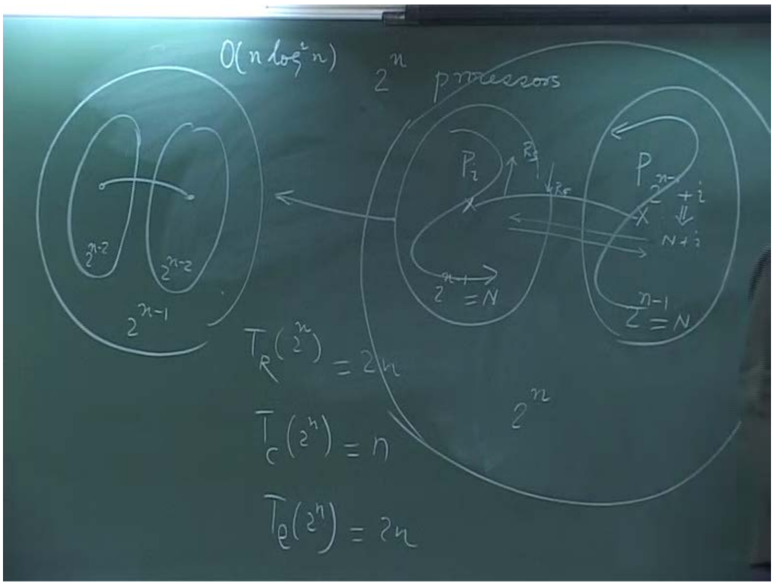

That is the idea in the hypercube 2n. Here also we have s 1 s 2 s 3 s is the stage number. This will indicate what is the r e increasing order and decreasing order and this find functions. You will be obtaining based on whatever your d1 in the case of m, is indicate s by 2 to the power of s. You observe basically this is a issue, this hypercube of size 2.

This is another hypercube of 3, this is another hypercube by 16 and so on, right. And initially you have 2 to the power of n, initially issue that to the some hypercube each of size 2 to the power of 5, right, at the s stage ratio there are to the end n minus s of cube each of size 2 to the power s and compute sign function for each sub hyper cube and each sub hyper cube as sub bitonic sequence perform this is equals to 1 2.

How many type this is equals to 1 2 3, right. In that case, type of complexity to sought element, are sought to the element type. Discuss this is the time complex merging. Now what is the time complexity to seek to them for element on hyper cube of type for element, right, quick? Did you get this (( )) border not influenced. This is stages the sum mention over here this is first second.

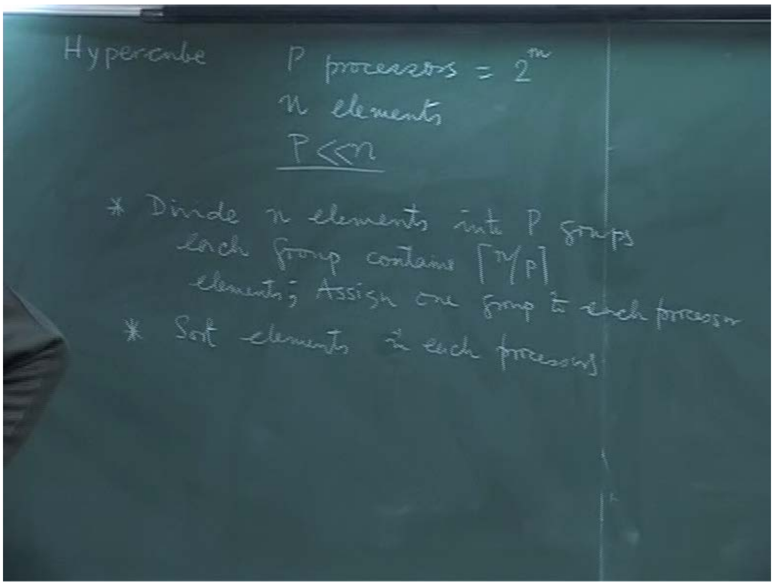
So, if you get any influence, which are bolder elements. No? This is, yes, this is some want to end. So if I take so, you will be writing sum mention over here. This is 1 2 and it is n square similarly, 2s where so, hyper cube of machine is very powerful or an element right. It is loss time complexity, loss parent but, it is a low a log on affected to the vanity right. To the log f affected because it is order and log parent and sought elements.

(Refer Slide Time: 35:30)



So, log n factor is way. Now tell me if first thing is n here, we observe that ratio it is having every 1 limit. That means the number of s is not adopted (( )). Now what happens if it is, you want to make as, that suppose you have p plus and n element. How are you going to do and what would be the complexity.

(Refer Slide Time: 35:56)

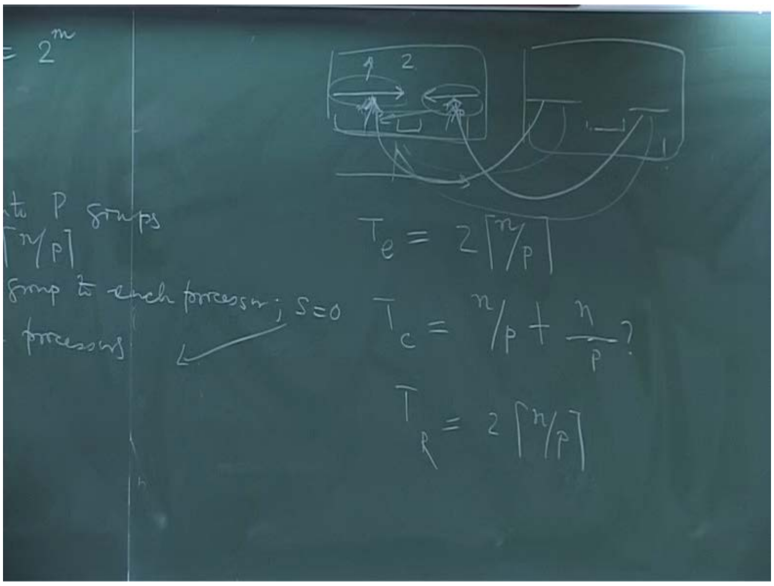



Now any idea if we have p prospective? Hyper cube p processor and n elements p equals to 2 to the power n. These lesson and that is 2n may be you think it is a last (( )) more of parent, that may be in here may be that as a, so, anyhow tells may be in problem are going to do it. Yes that is straight part divide n element into p groups, in each group contains the elements and may be the last pursuant take little less than those not sorted, what is n.

So next is each process sort elements. In each process assign 1 group each from that you told that sort elements of its in its group by pseudo. Then we have got this is sorted and so on. You remember that by giving this you have already created a problem we lost because if you remember that have to face the details becomes bitonic defines right.

So, this is increasing order, this is an increasing order. Here no longer than 2 elements but, got 2 elements. You have the 2 sequences you understand the 2 elements form bitonic sequence form, is through that, is not there right. So, aid sequence rotation to the basic 0 looking for the 0 step s equals to 0. All even index group should be decreasing order. Not only you have solved but, you are making the sort of sort operation at s equals 0. State the even index transfer arrange the data in decreasing order and index process

(Refer Slide Time: 44:09)




If the sponsor is 1 other than 0 on word, the even index in decreasing order, increasing order or index in decreasing order. So, that you may subsequence for bitonic sequence agreed. So, here you put a sequence s to 0 and here 1 it means if you write here. So, you got the 2 subsequence. They are in such a way; this is a bitonic sequence this is another bitonic sequence and so on.

Now what is next? Next is that you have to remain this 2 sequence using to process and again you brought this 1 element? You brought it here, compare and rejected element use there, agreed. Here 2 processor and you have 2n by p subsequent a size and you have to do the bitonic. Yes it is the case, what is the time you need 1 way, how are you going to do that, then how are you going to do that? The problem is clear right. This is the bitonic sequence by the p, is this increasing order, this is decreasing order you have 2 prospect.

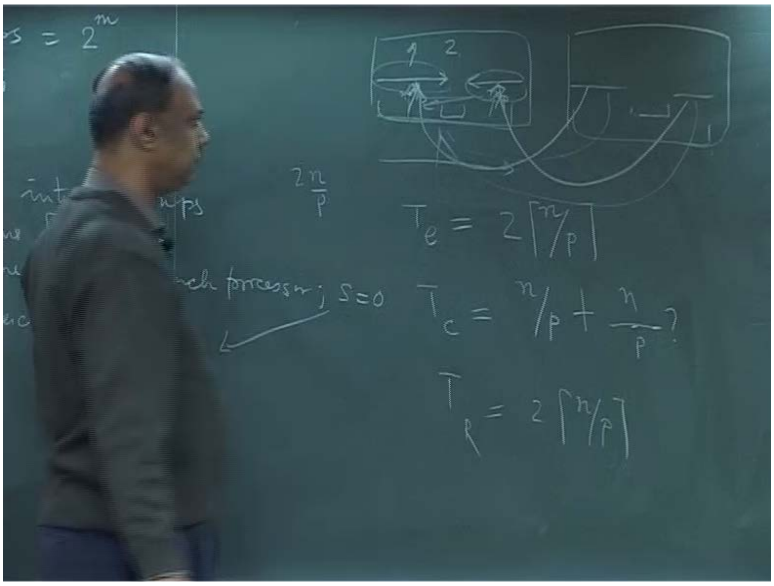
This is 2 process agreed or not? Now I want to merge it. Now what it is end by p register not in 1 register? 1 register can take only 1 element n by p register, yes. Except you use pipeline, nothing more than that, right. So, what you have to do in this whole, it bring up and you have to bring this 1 here, you want to do right. This is 1 way or another way could be 50 percent said you another it bring down then you are trying to make busy both the prescribe process. But, you may not as it is not that easy which 50 percent are going. If you bring 50 percent then after you merge it, there is no guarantee here, say the middle element here middle element place to the finally, search find out which part of this is smaller that you send. That is very complex you would not gain much with respect to time complexity. So, what based to the simplest is to bring this out this element and the reject element here and then you get this is a bitonic sequence, this is another bitonic sequence and you can merge now independently. You can do this bitonic sequence right yes or no.

So, what I am telling, bring this elements here. Compare all the elements; send back the rejected elements here, right. An external element is kept here. Each processor can be used to do the bitonic merge of n by 2 elements, right. So, you get a sorted paper.

If it is the case, what is the time you need to merge this 2 sequences each of size n by p by 2 processors. First you tell about the exchange of (( )), how many exchange this will be bringing up and again 2n by p, right. Rest I will discuss later on and then number of comparison only could you get him and (( )) of this part 2 sequence 2(( )). If it is a bitonic sequence, so, again law (( )) will be coming right. Some factor will come, what is that factor n by p of classes and it is also in term of n by p. This is also coming in the factor of n by p.

Then routine 2n by p, right. So, after doing this you get increasing order. You get decreasing order next time. What again, so, this time data movement time will be much more than 2n by p. Here the size is n by 2n by p. Now these data, this you can bring it here, this is also to bring it here, you have to pay huge amount of time. Earlier this is connected to this. See this box, this is connected, this is connected, this is connected, this is connected. You have agreed or not.

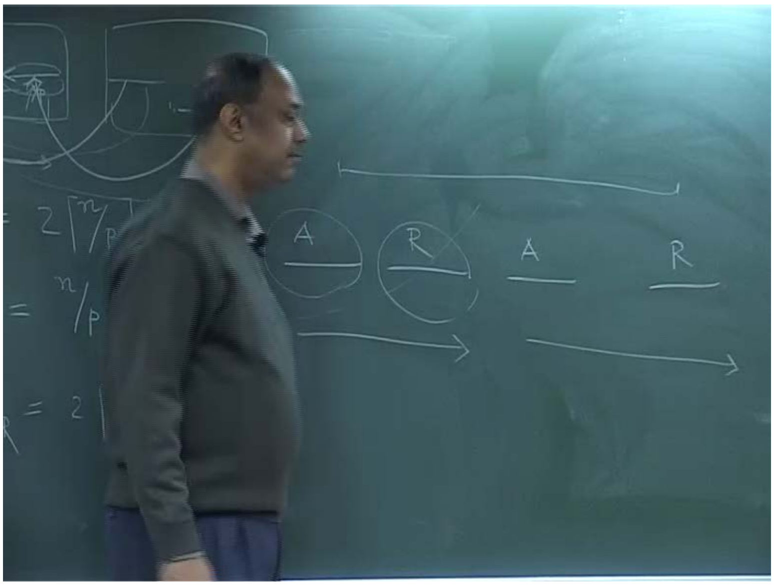
(Refer Slide Time: 45:18)




So, this context is sorted sequence. This context another sorted sequence in decreasing order. Now if you want to bring this 1, this whole thing here is in this size. This is not a small task. Next time task is increasing. If you have the other big size here and you want to bring it here, yes or no? So, can you think about this problem how to solve n element using p process? Now I am silent about the bitonic.

I am not telling their bitonic. I am increasing the complex (( )). Why I shall use them in increasing order or decreasing order because once I have to merge the 2 sequence and just I bring all these elements and merge it. Merge takes how much time, n by p plus n by p minus 1, n plus n minus 1. Simple merging, so, this is also n by p plus n by p, so, in what I am gaining, so, just I bring these elements here and merge it. Simple operation so, it is becoming 2n by p minus 1 right.

(Refer Slide Time: 48:15)

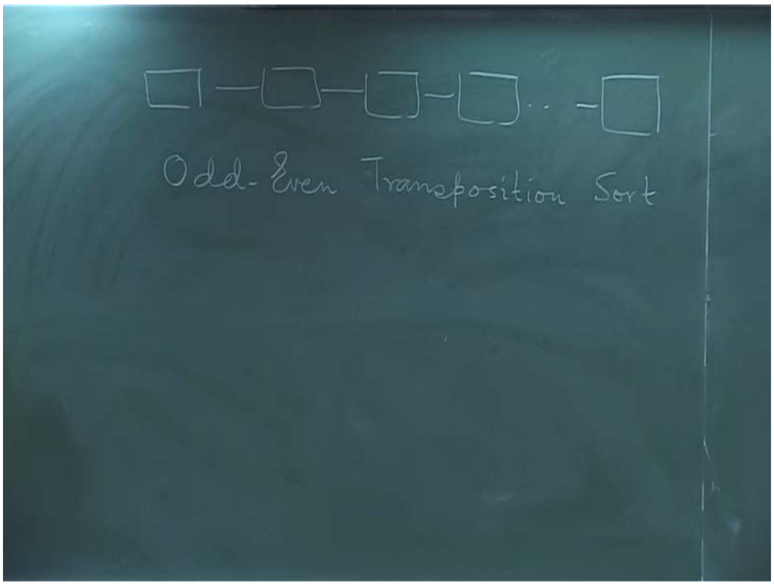



Now the problem is that you have to, this is increasing order, and this is also increasing order. Now you want to merge it. So, basically you have this sequence. This sequence forming increasing order sequence. This forms another increasing order sequence and right, next is how to merge do you have any idea.

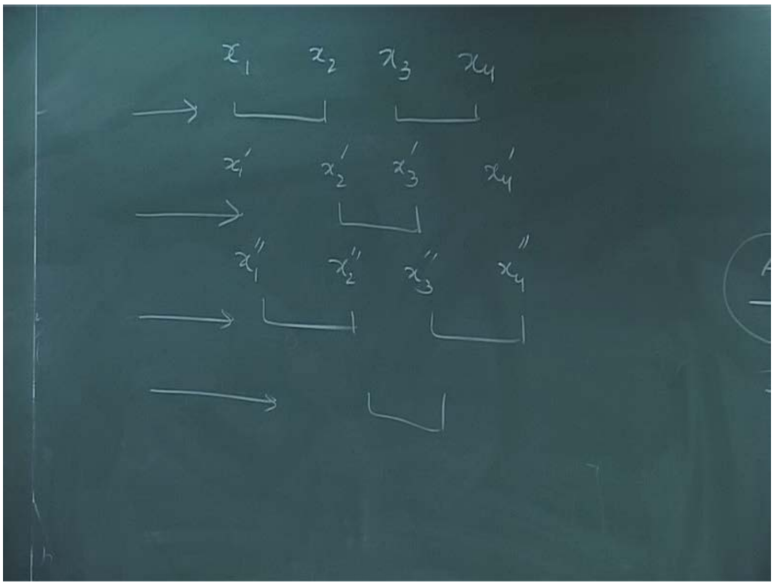
This contains all accepted elements. This contains all rejected elements. Accepted elements, rejected elements but, there is no guarantee that this accepted elements, such follows this elements right, no idea. If no idea it will stop it here. We will go to little simple problem. Then will come back again, right, because that is the way 1 should think to solve a problem. So, let us think about this. Keep it in this mind this problem to be solved. How to solve, I also do not know, you also do not know. That is the thing right yes. So, you have n elements p plus hyper cube machine. This is a thing to given for 1 element equals to n problem, is no bitonic, and is very good right. If this is low, this is low much more than p, then how to part these things that is to be said.

And you have to do that first. Let us think about the, suppose, I have a linear idea. I will come back to this again. If the linear (( )) can help me in solving this problem and this is a problem with mesh connected. Also because it is not that easy to handle the sine area if I have n cross n mesh n plus element and p cross p mesh. How to sort it is not the reason sorting but, not the efficient sorting (( )).

(Refer Slide Time: 51:09)



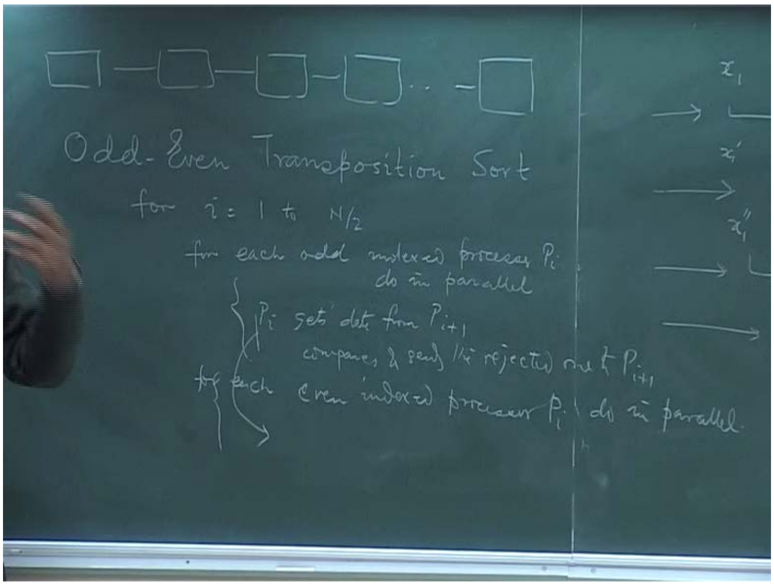

(Refer Slide Time: 54:02)



Now if in case of linear (( )) suppose I have this arrange and each of them containing 1 elements each 1 them contains 1 elements. Now tell me how can I sort it? Do you have any idea? Have you heard the term odd even transposition sort? You will tell no sir, yes or no what? Is yes (vocalized noise) we have discussed in our class odd even (( )) sort that is odd index (( )) and this is another sorting technique which is known has odd even transposition sort forget about (( )) machines. Let us come back to the sequence. Now if you have x 1 x 2x 3 x4. Suppose this mod element is there, now 1 thing you remember that in order to place x 1 indirectly or directly up to compare with the other element, right. That is there; otherwise if you cannot tell you get the product simple. So, that arrangement has to be d1 1 way is that x 1 has to be given a chance to compare with any other element.

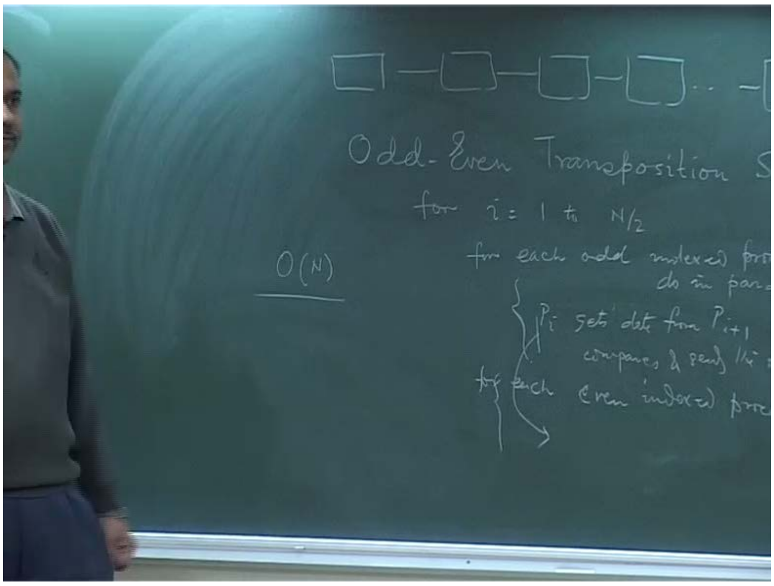
So, in odd even transposition sort first every odd is compared to is next even element. So, you get here s 1 dash x 2dash x 3 dash x4dash. That rejected element is in this part. Next stage every even is compared with next odd right. Now you get x 1 double prime x 2double prime x 3 double prime x4double prime. Next stage and the next stage, this so that will give you that x 1 will be compared with x4. Everybody gets an opportunity to compare with every other and in that case you can observe that the existence inheritate (( )) right. So, this can be d1 simultaneously, this can be d1 simultaneously, agreed. So, in the case of linear (( )) because this is very suitable for linear (( )) because pi is connected pi plus 1 right so.

(Refer Slide Time: 55:50)



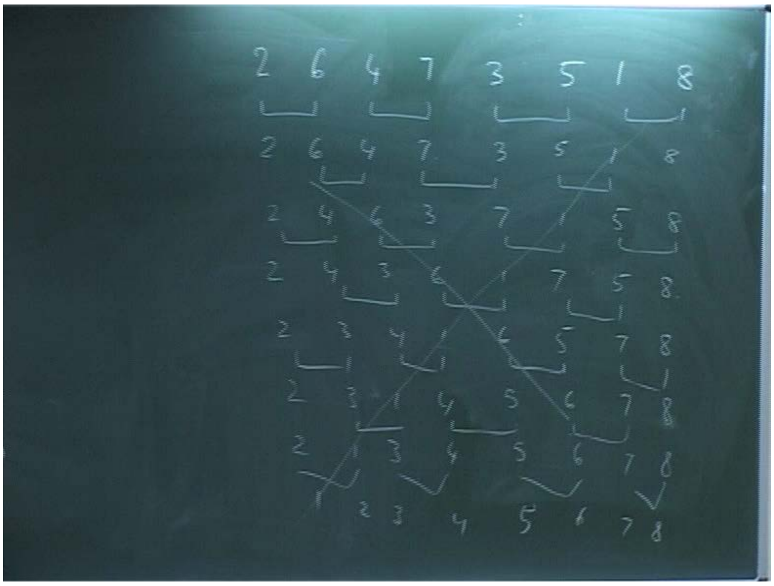

For i equals to 1 to n by 2n by 2it connects nothing n by 2for each odd index process pi do in parallel pi gets data from pi plus 1 process and then gets the rejected 1 t i plus 1. So, the time complexity for this 1 comparison and order 1 and this is also order 1. I am explaining that take for also after number of time so, this is order 1 time right.

(Refer Slide Time: 56:00)



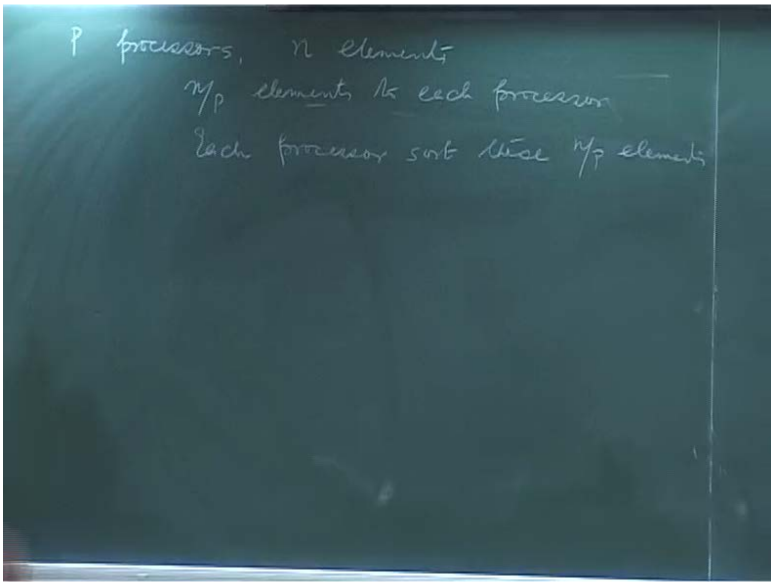

Concentrate that so you have total complexity order and right you have a 8 elements 4 1 3 6 5 7 1 and 2 ok.

(Refer Slide Time: 58:19)

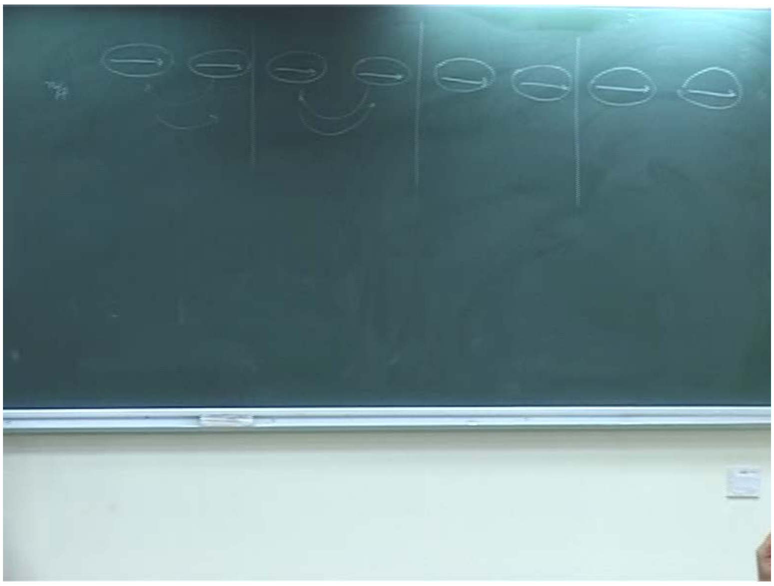



So, these are the 8 element have so you get 2 6 4 7 3 5 1 8. You get 2 4 6 3 7 1 5 8. You get 2 4 3 6 1 7 5 8. You get 2 3 4 1 6 5 7 8, here 2 3 1 4 5 6 7 8, here 2 1 3 4 5 6 7 8 you get 1 2 3 4 5 6 7 8. So, you observe that there put in the 1. So, by you get posted a number from 1 parent to another parent similarly, that 6 also got terms have these are position right, but, it is not a past optimal. There is in sequential it also it does not give me the optimal algorithm. Sequential it is order to here also it is parent right. Time of part and to the pass.

(Refer Slide Time: 59:11)

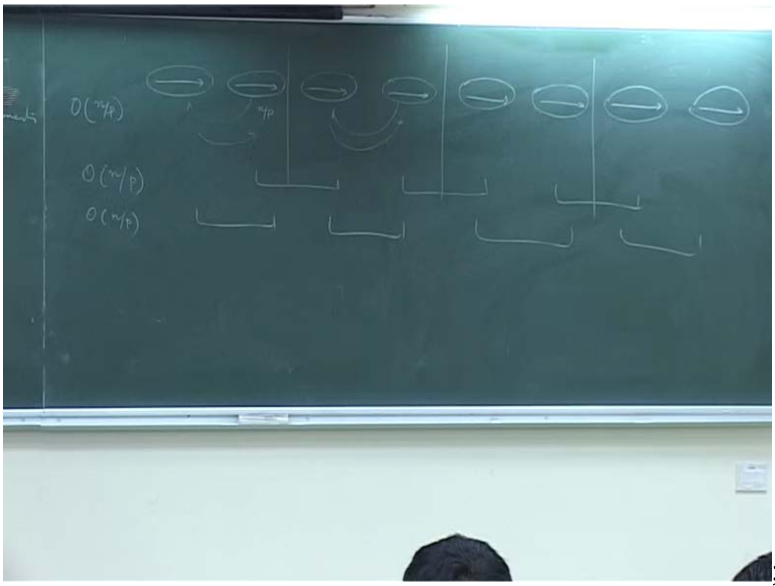



(Refer Slide Time: 1:00:55)



Now if you have p process and n element initially n by p 2 element to each processor. So, this process each process sort these elements. Now this problem is that instead of 8 of sequences f equals to parent and these are marked.

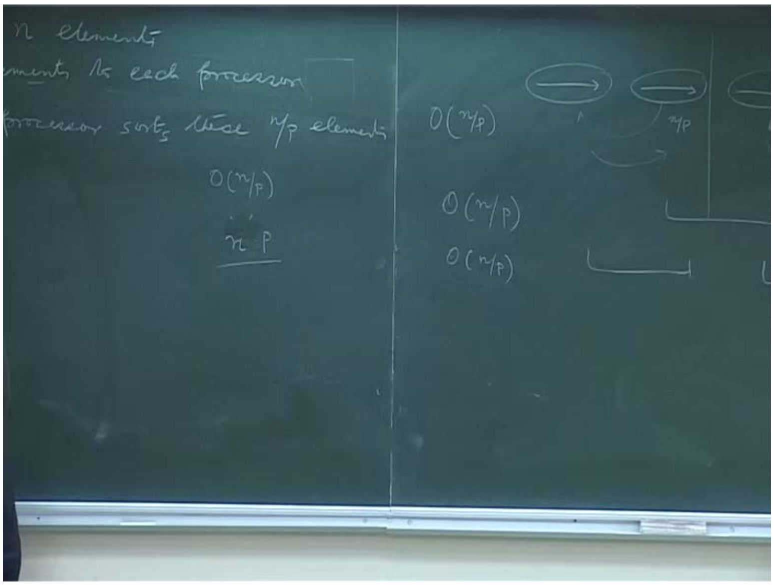
(Refer Slide Time: 1:02:00)


;

So, you observe that so, as come to next similar step, these are not so, our target is breaking the familiar. If you can break this problem then you can solve the other problem, say, the vacation of oriental position sort this have to bring it. Here rejected element will be sending back right.

Similarly is the case this and then this right. Here if it is the cases can you bring this data here and merge it and send back here right. Pearly this group I can do so, you have to bring this data. You need an n by p times and then merge it takes again n by p times. Merge it these 2 sequences are n by t. Send be back and again also p. So, it takes order and p time yes or no. Bringing data from here to here, you need at p merge it and p send back and by p. Now this data contains by p sorted is of the second type. You will be doing this which is taking again and n by p, yes.

(Refer Slide Time: 1:03:44)



Third type which takes again n by p right, so, what is the time needed? How many is time you need n by p? Why I am not increasing this size? Why is n by p? Why do you get n by p here and in between p is there? So, p by 2 time for ordering, the p time to each (( )). So, p time to do right instead you need n by again p.

So, whatever n time is sufficient for that to be in p processor. So, total time required cost is n into p. So it is cost optimally p is log n of lot of do it honor. So, if try you just think and try to get with the similar indicate be use for hyper cube honor. So, it indicate the can be d1 format.