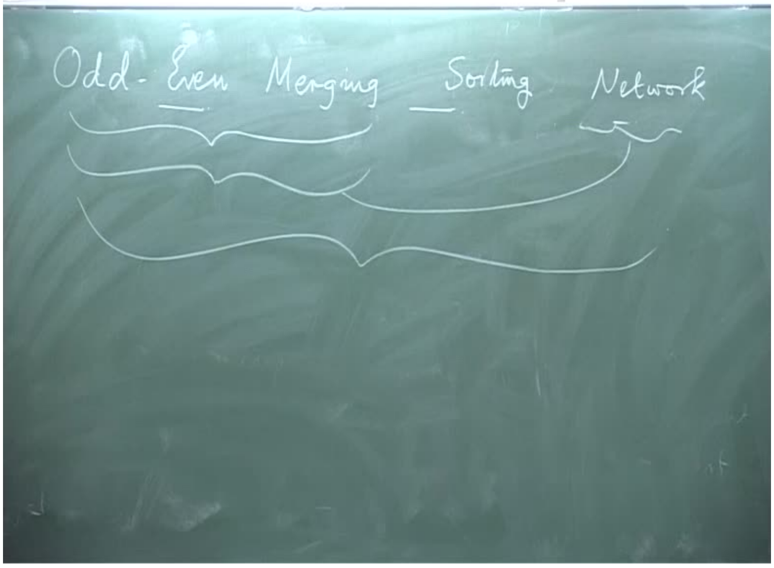
Parallel Algorithms

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

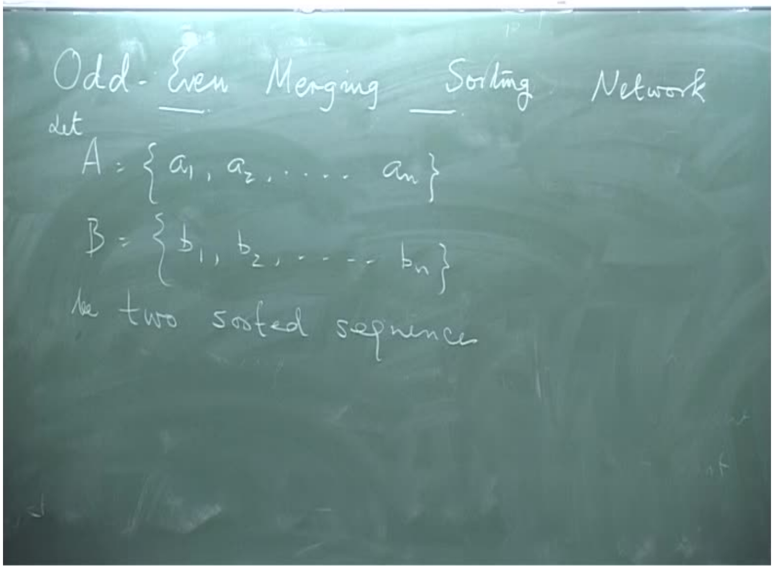
Lecture - 7

(Refer Slide Time: 00:18)

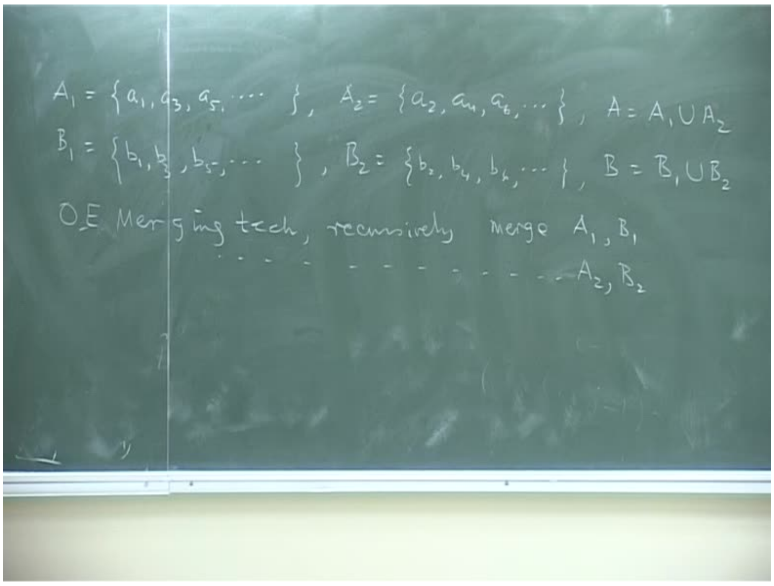


So, today we will be discussing only odd or even merge sorting network. So, and since I assume that you do not know anything about odd even merging. So, first we will be discussing odd even merging technique, sequential algorithm. Just then we will be discussing the odd even merging network and then finally, would be discussing about odd even merging and sorting network. First we will discuss about sequential odd even merge technique, second will be discussing odd even merge network and then finally, sorting network.

(Refer Slide Time: 01:47)

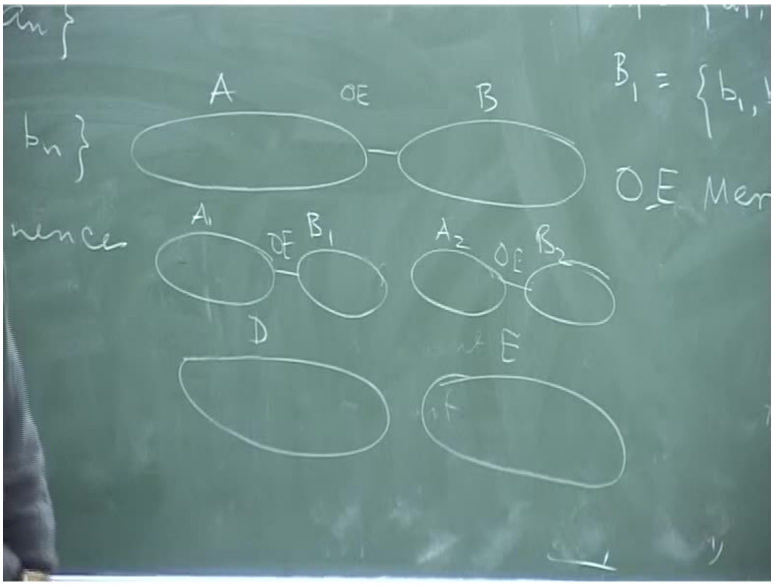


(Refer Slide Time: 02:33)

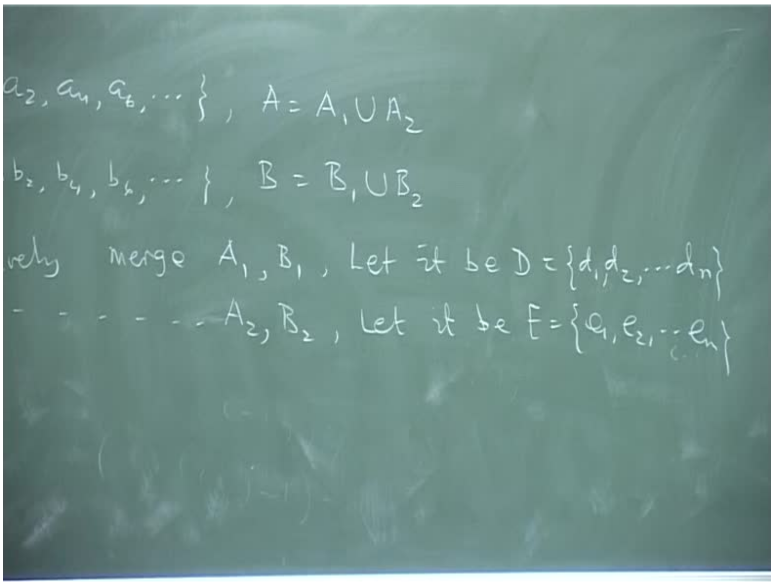


So, this is special purpose sorting network. Let us assume that a and b are the 2 sorting sequence and the problem is to merge this 2 sorted sequence by odd even merging technique. What it does that it forms a1 a2 are the 2 different sets. a1 contains the elements which are added next and a2 contains the elements which are given index. Similarly, is the case of b1 and b2. Now by odd even merging technique recursively merge a1 b1 and similarly, a2 b2, right.

(Refer Slide Time: 04:22)

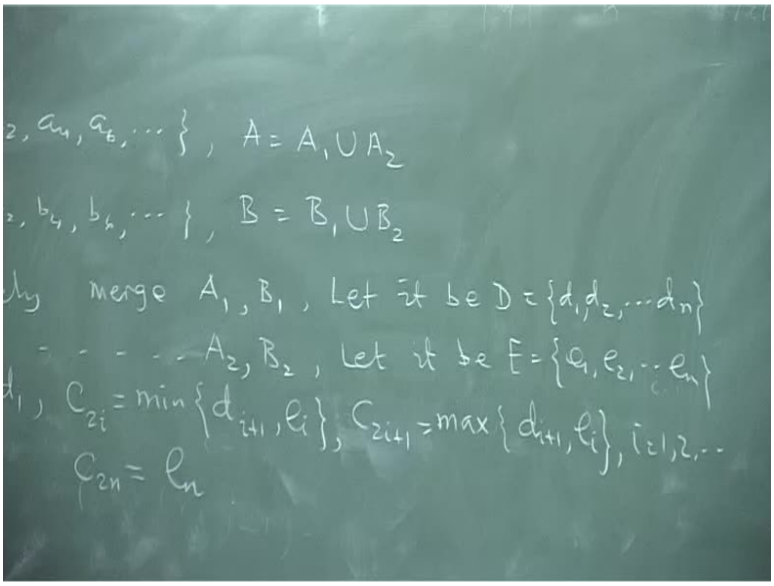


(Refer Slide Time: 05:06)



Say we have sequence a sequence b. Now you get a1 b1 a2 b2, right, you have the odd even merging technique. You are defining same thing you are defining here odd even merging technique recursively, you get d sequence you get e sequence recursively. You have d1 the odd even merge of this you get d sequence you get e sequence again. So, it’s d now, if it is a case then can I draw the conclusion b1 is the minimum? Sure, can I draw the conclusion because minimum is either a1 or b1 right.

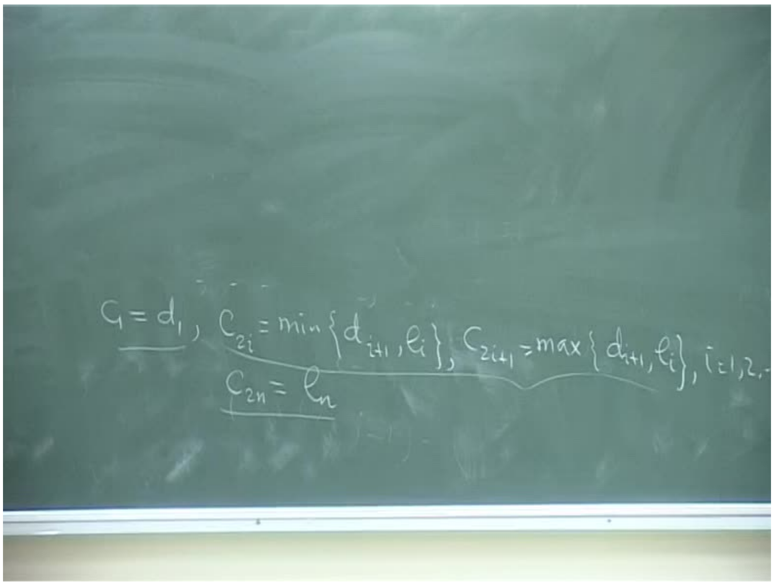
(Refer Slide Time: 06:27)



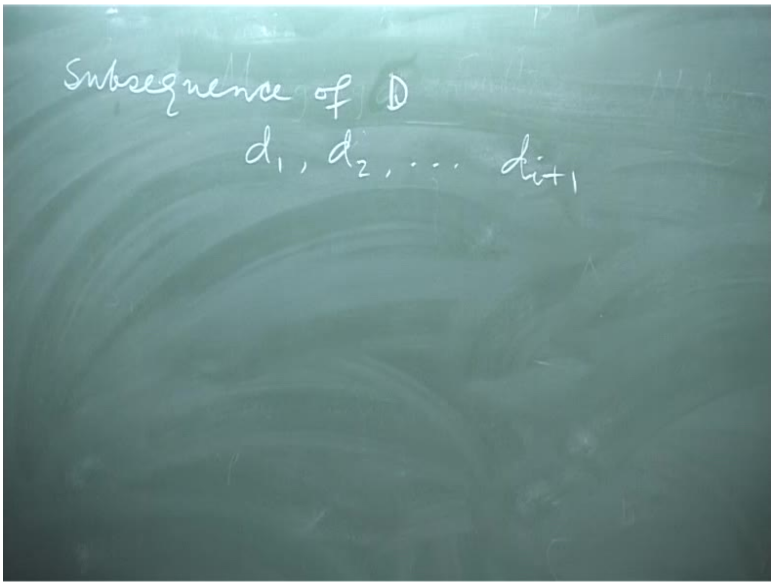
So, d1 is the minimum that conclusion I can draw. Similarly, can I do the conclusion that en is the maximum? Right, this too is clear to you. Now second thing is that which is the second element second small? Any idea? d2 of u r right so this is the thing this idea has been taken so let us define c 1is d 1c 2i is minimum of d i plus 1e i i is equal to 12on words and c 2n is your here c 2i plus 1actually i have to write somewhere zero so thus i am tell the who is the next form here c 2i plus 1is maximum of d i plus 1e i i is 12x

So that is the things say what is the stating what is the common problem is that given a and given b 2sorted sequences i pick up the odd sequences which form a 1odd sequence of b gives you b 1even that index elements form a 2and b 2recursively you are merging a 1b 1you get the sequence b 1to d n similarly, you get the sequence e e 1to e n now s 1is given c 2n is e n and c 2i is define by the minimum of d i plus 1e n e i and c 2i plus 1is maximum of e i plus 1here

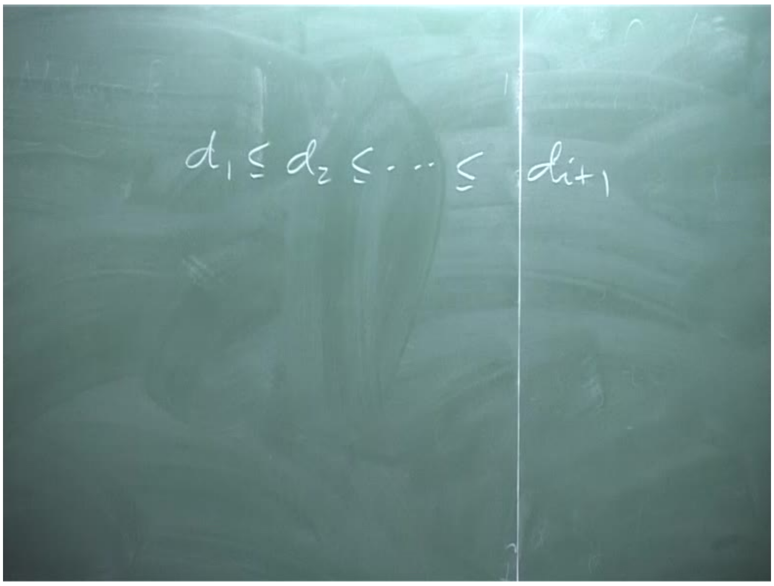
(Refer Slide Time: 08:33)



(Refer Slide Time: 08:46)

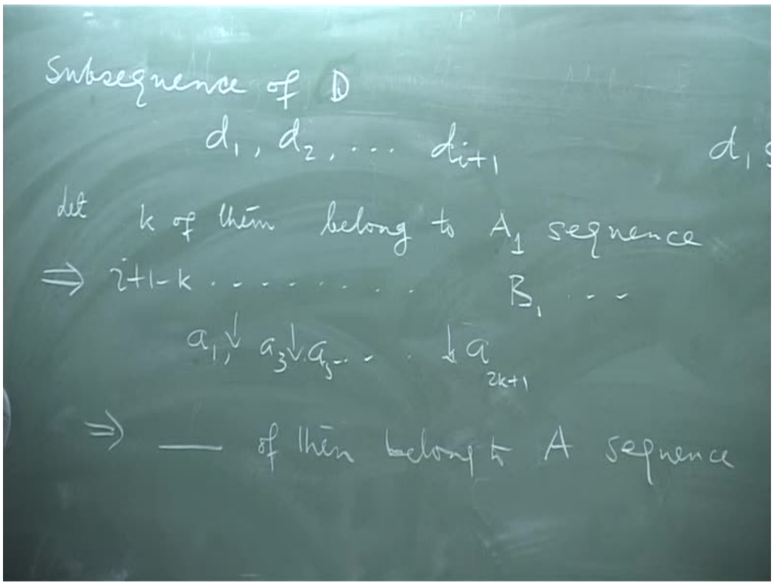


(Refer Slide Time: 09:11)



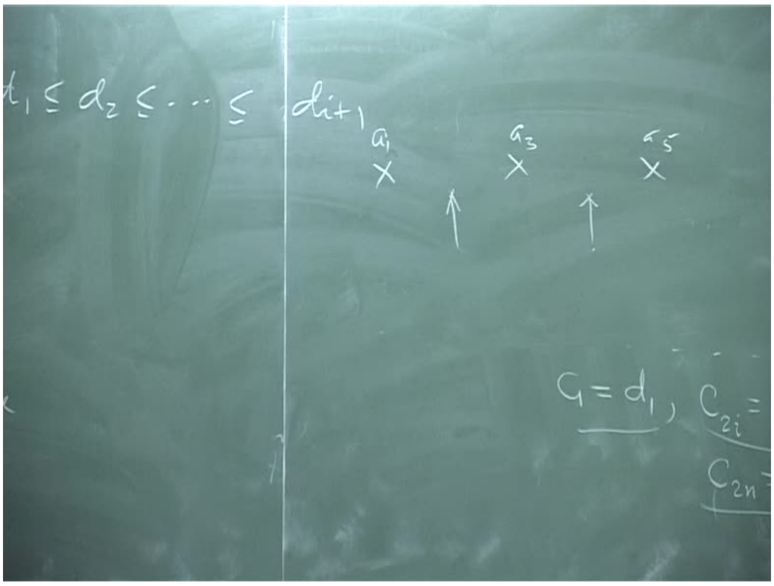
Right now we have to prove that this give the sorted sequence that is the only thing. If we can prove this you are happy. So, this obviously we go up to prove it, this is obvious you do not prove this part, only thing is that you have to prove this part. Now, let us consider the subsequence of D that is d1 d2 d i plus 1. Let us consider this subsequence and you will find it said this twice the criteria because, this is recursively sorted is satisfy this criteria.

(Refer Slide Time: 09:26)

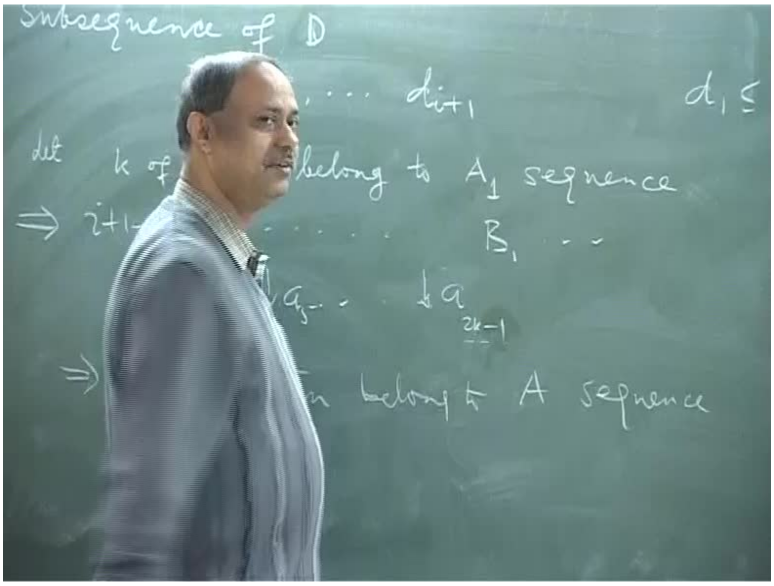


Now, let us assume k of them belong to A sequence, because d1 d2 …di plus 1. This D sequences obtained from odd the index subsequence of a and odd the index subsequence of b. So, either we can assume the k of them belong to s a sequence. What is a, a means a 1sequence k 1after belonging to a 1sequence then, does it imply that i plus 1minus k of them belongs to b ones of b 1sequence to this total number elements? Take it better listed please tell me. k of this is are option k of them belong to this it indicates. Then i plus 1minus k of them belong to b sequence now, what does it mean that k of them belong to s a 1sequence means that a 1 a 3 a 5 and so on. And you got what is the index of this 2k plus 1is it, if it is the case then can i write how many of them what would be the highlight belong to original a sequence original, what should be the number here 2, see how many elements are here k elements.

(Refer Slide Time: 12:04)

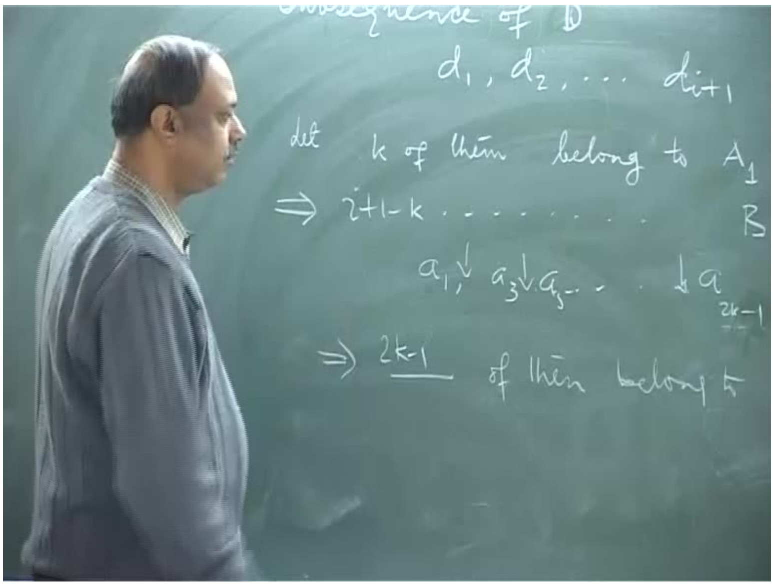


(Refer Slide Time: 12:56)

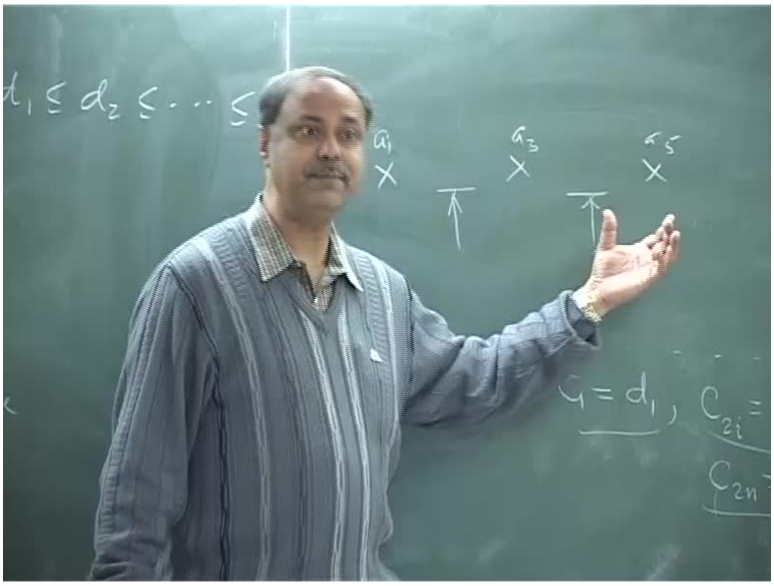


So, k elements are you sure hundred percent, i have 3 elements here. How many elements are here a 1a 3 and a 5. No this index may be wrong I do not know, sure hundred percent. You have 3 elements in between how many elements are there, how you are telling k if this 1may be 2k minus 1may be 2k minus i. I am not talking about this part I am taking that I have odd index elements and between the odd index how many even index elements are there. That is k minus one

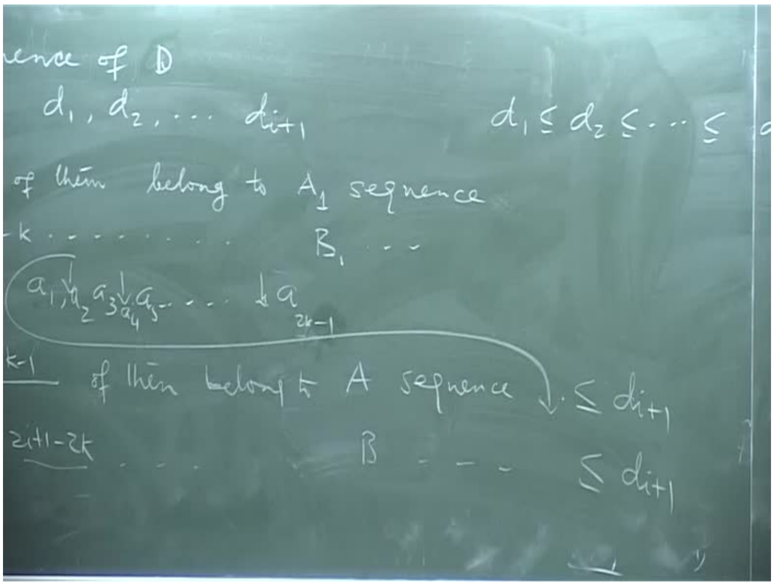
(Refer Slide Time: 13:12)



(Refer Slide Time: 13:38)

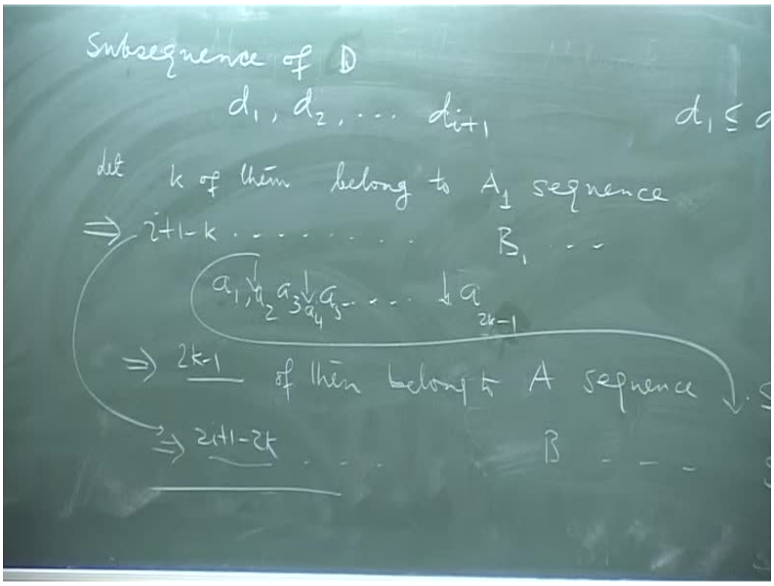


(Refer Slide Time: 13:55)



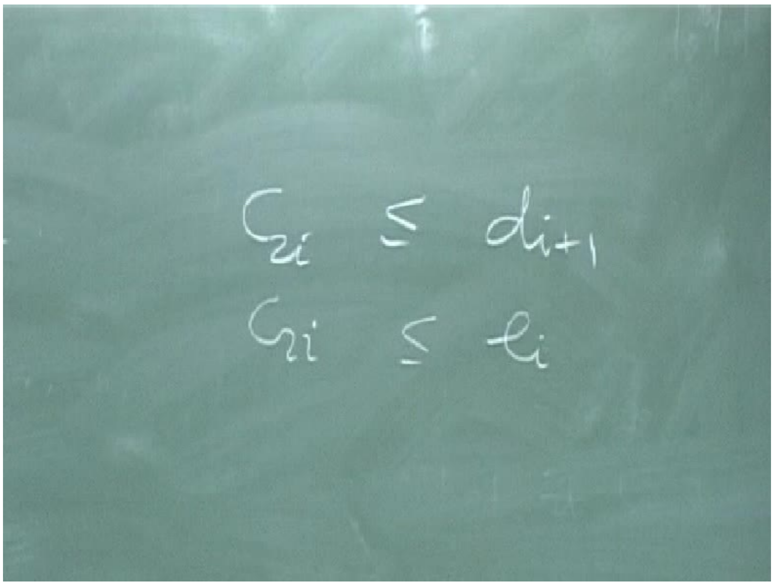
So, I can write it is k minus 1agree. So, what is that 2k minus 1because, k minus 1even index elements k elements are odd index elements. So, total 2k minus 1elements were belonging to a sequence and they are because this is sorted degrade. Now, if it is a case so, can you tell me from here how many elements of that belonging to b sequence? How many odd index elements i plus 1minus k and how many even index elements i, i minus k then, what is the total number of elements why 2i plus 1minus 2k that may elements belonging to b sequence. What does it mean 2k belonging to a sequence which is less than and equals to d i plus 1that many elements. Which is less than equals to d i plus one?

(Refer Slide Time: 15:46)

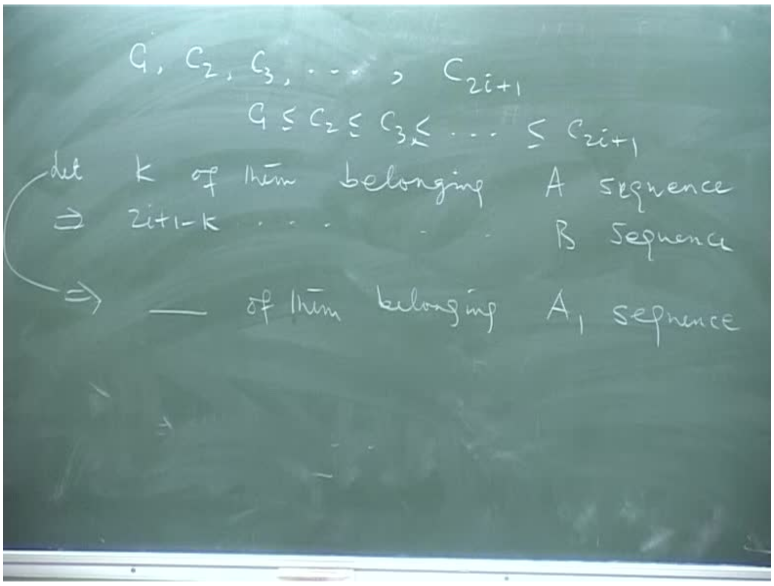


Is it what we did first, we do not that a element of b subsequence belonging to a 1sequence it indicate that i plus 1minus 1elements belonging to b 1sequence it indicates that 2k mi it indicates the 2k minus 1elements belonging to a sequence less than equals to b i plus 1. and similarly, 2i plus 1minus 2k elements belonging to b sequences less than b i plus 1 and if I add this 2 what i get 2i elements belonging to a and b less than equals to b i plus 1.

(Refer Slide Time: 16:06)

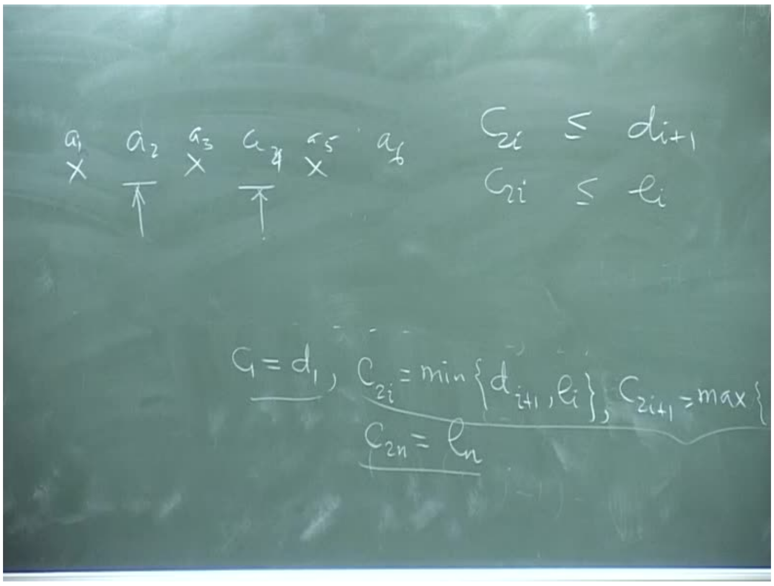


(Refer Slide Time: 16:56)

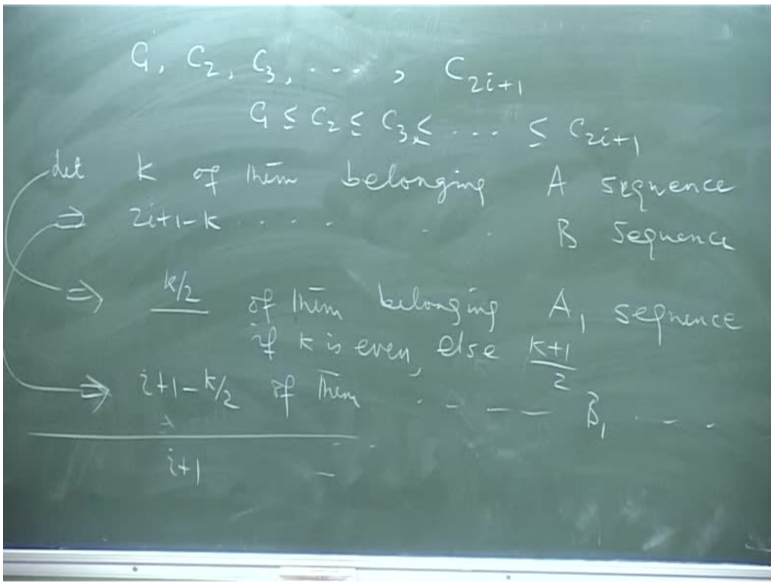


Now, if see the sorted sequence so, can i write c 2i is less that equals to d i plus 1 because by combining this 2 either 2i elements belonging to the combined a and b is less than equals to d i plus 1. And since c is the sorted sequence that is the assumption so, c 2i is less than equals to di plus 1similarly, you can show c 2i is less than equal to e i just start with d i 1and you get. So, let us considers the subsequence c 1c 2c 3 c 2i plus 1now, this is sorted sequence. So, i can write this is equals to this.

(Refer Slide Time: 18:26)

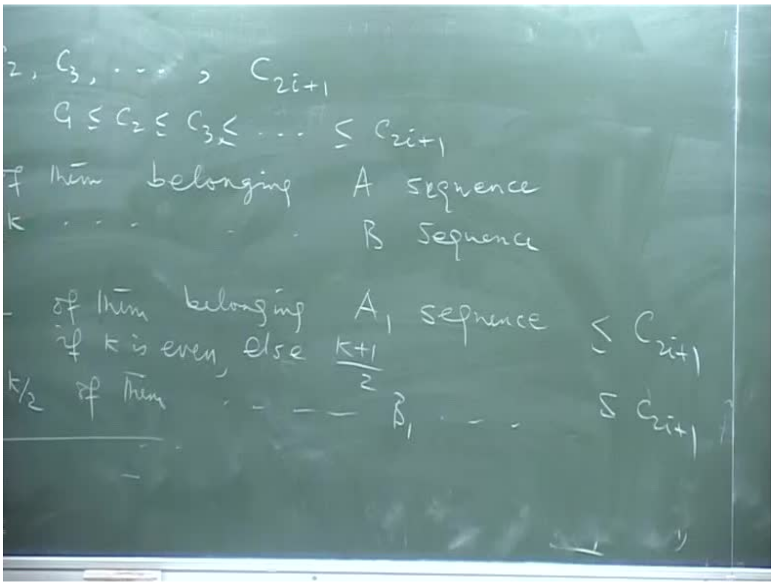


(Refer Slide Time: 18:51)



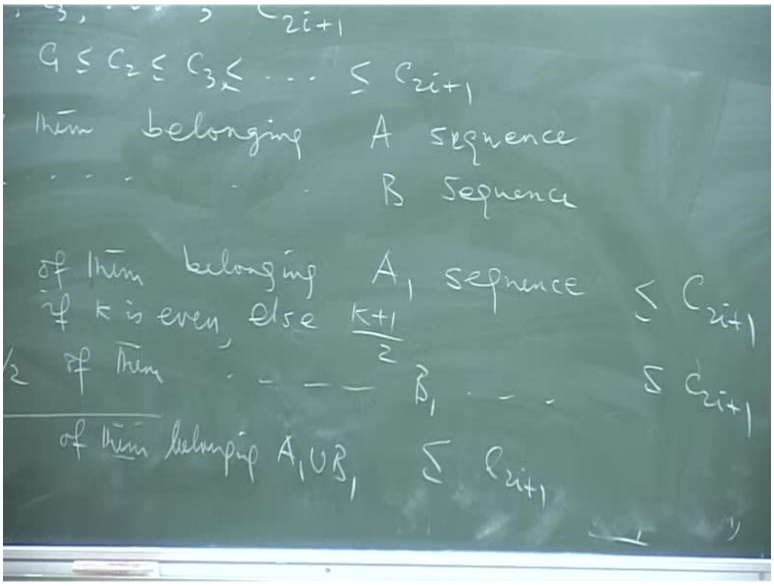
Now, let us assume that k of them belonging to a sequence so, i can write this is assumption i can write 2i plus 1minus k of them belonging to b sequence because obviously, that can c has been obtain from a and b only. Now can you tell me how many of them in that case from here of them belonging to a 1 sequence say this is a subsequence belonging to c? How many of them belonging to a 1 sequence. Now, the k should either given order if it is odd, what is the value and you can remain? So you would be writing k of i 2 of them belonging to a 1if k is even else k plus 1by 2of them belonging to you.

(Refer Slide Time: 20:36)

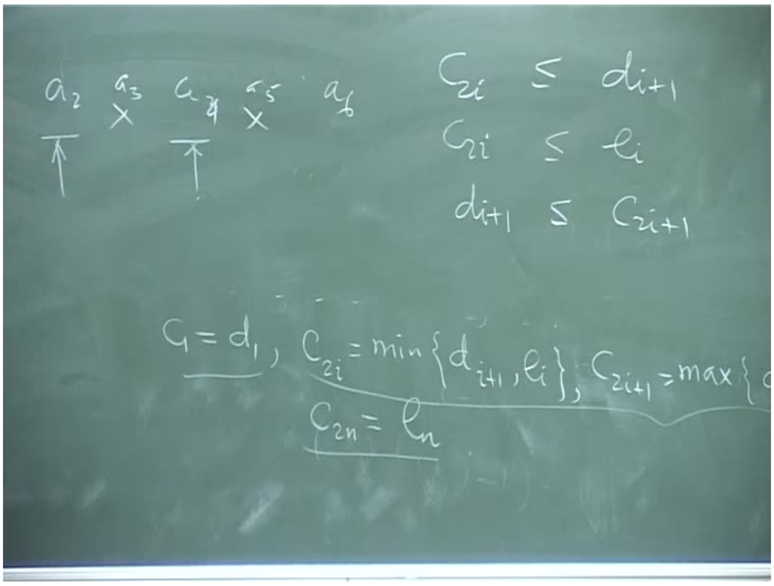


Read it now if it is a case from here, can i write something how many of them belonging to b 1 sequence. Now, assume that k is even assume first k is even otherwise if k is odd you have to 2 little manipulation that i 2i plus 1minus k divided by 2. Some manipulation apart k, k equals to k s or what should be the changing design of in both the case if i get it. Tell me if add this 2i get i plus 1of them so, belonging to k by 2of them belonging to a 1subsequence which is less than equals to c 2i plus 1 k by 2of them belonging to a 1subsequence to be less than that is the thing i consider up to this. So, I can write this is less than to c 2i plus 1.

(Refer Slide Time: 21:00)

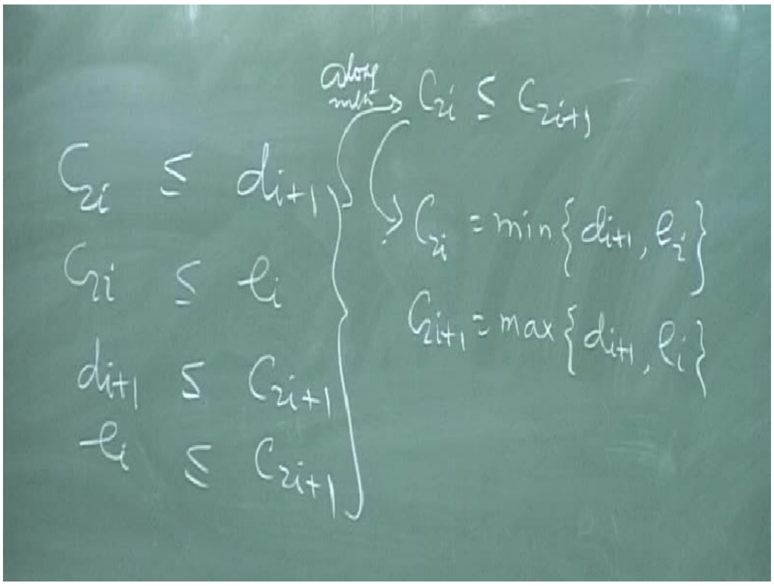


(Refer Slide Time: 21:26)

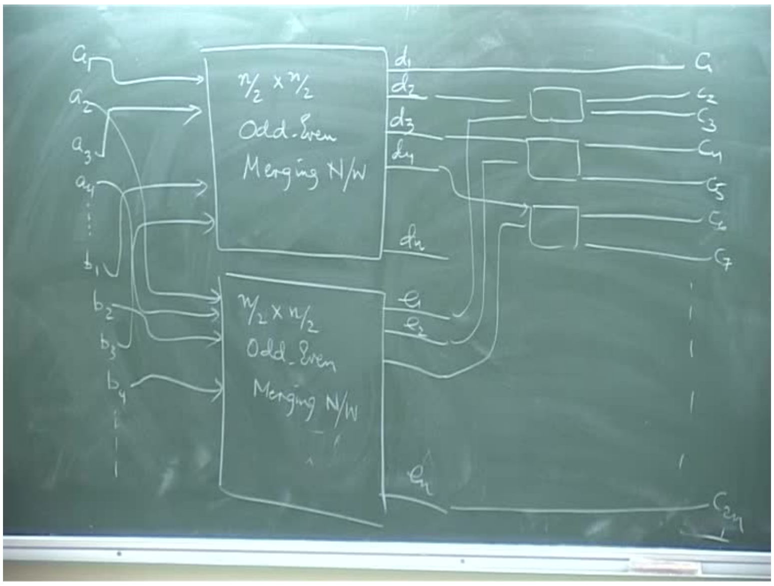


Similarly, I can write this is less than equals to c 2 i plus 1that means i plus 1of them belonging to a 1union b 1or the merge of a 1a 2a 1b 1is less than equals to c 2i plus 1 and, this gives you the condition that d i plus 1is less than equals to c 2i plus 1because, i plus 1of them d i plus 1is 1 d is the sequence of a 1union b 1. So, b i plus 1is the first i plus 1elements of this 1so, i can write d i plus 1is less than equals to c 2i plus one.

(Refer Slide Time: 21:59)

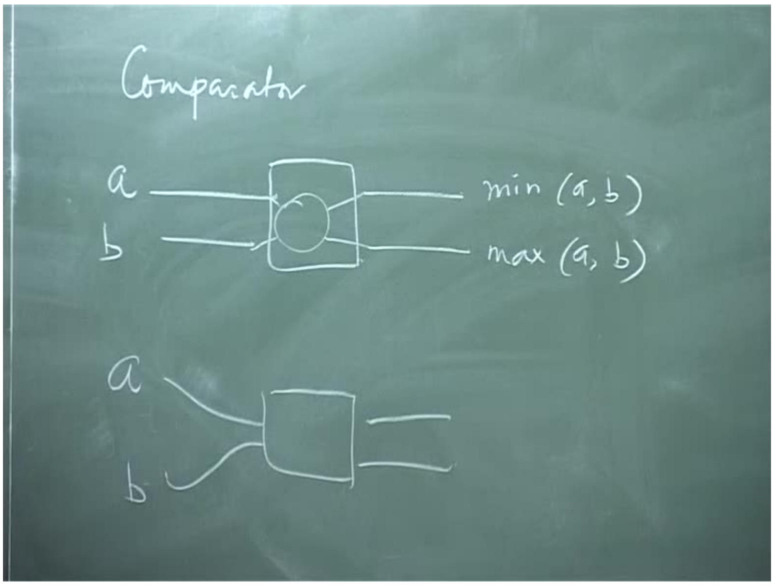


(Refer Slide Time: 23:24)



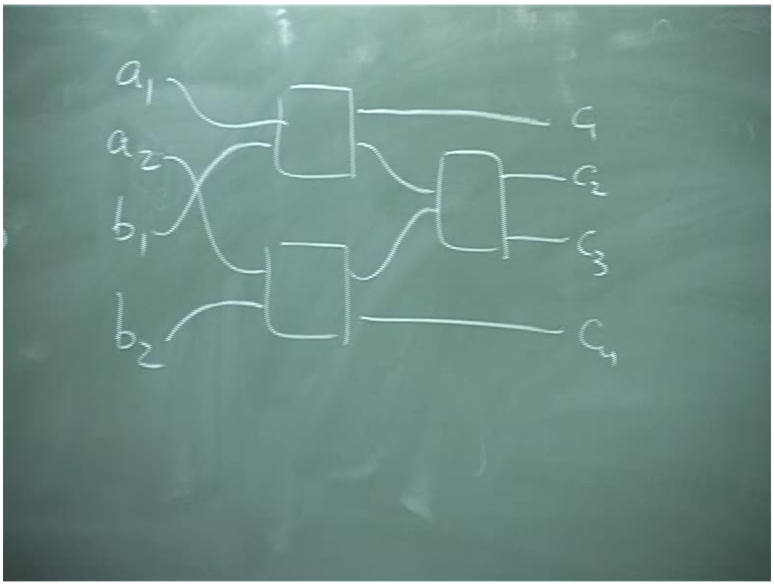
Similarly, you can write e i is less than equals to c 2 i plus 1 so, combining this four equations you can write that d i plus 1is minimum of not d i c i is c 2 i because you knows that c 2 i is less than equals to c 2i plus 1.So, from here you can write d i plus 1e i and c 2 i plus 1is maximum of d i plus 1, combining this four equations along with this you can easily prove this so, the odd even merge technique correctly merges 2 sequence a and b as stores at a sequence c this idea can be use to define the merging network, you have suppose n elements these are a 1a 2 a 3 a 4 b 1b 2b 3 b 4 and you have a record name is n by 2cross n by 2network odd even.

(Refer Slide Time: 26:28)

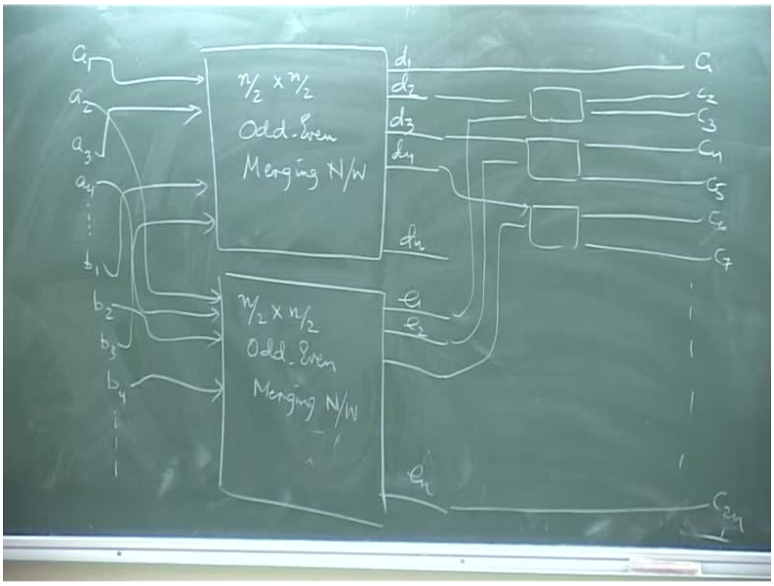


Here input is odd indexed and here input is all even in that limit. So, some recursive technique you get the sequence d 1d 2 d 3 d 4 d n you get e 1 e 2 e n so, d 1is your c 1 e c 2 f and then d 2 is compared with d 1to get c 2 c 3 d 3 is compared with e 2 to get c 4 c 5 you get d 4 is compared with c 6 c 7 and so on this is the simple network in that case if it is the thing so, what I need in this simple comparator I need a cross which is nothing but, a comparator now, the comparator is define like this you have a input a, u have a b and output is minimum of a comma b maximum of a comma b.

(Refer Slide Time: 27:27)



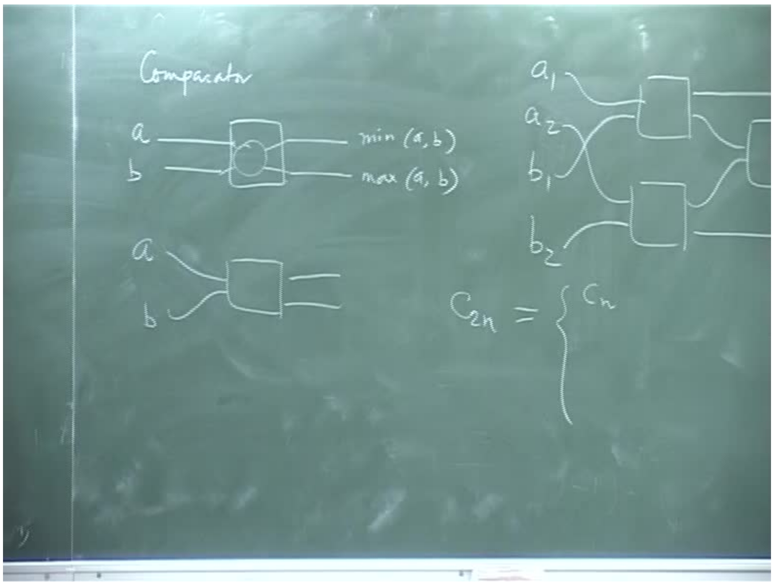
(Refer Slide Time: 23:21)



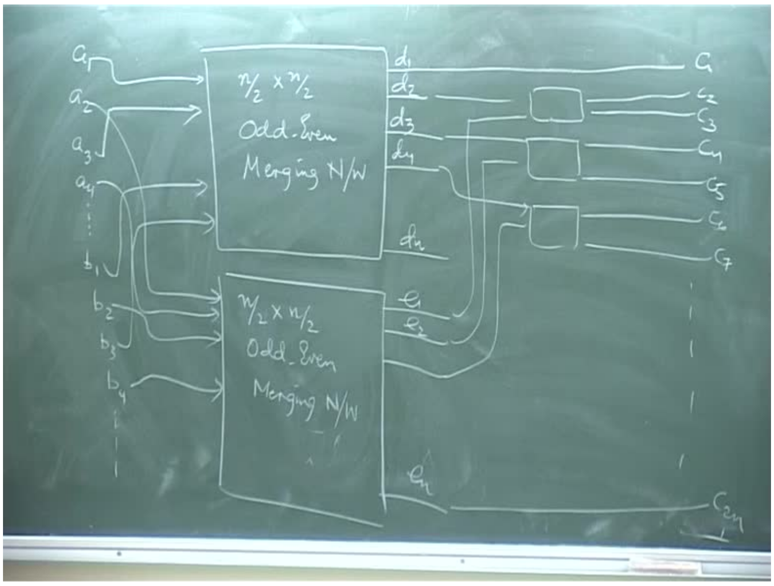
So, you need a comparator here which take the input and give to the output like this. Now, if I have the 2 elements a and b how many comparator unit to ((slov)) mark this 2 sequence 1comparator and the structure is very simple. Now, if I have I need this later on, now we have a 1 a 2 b 1 b to these are the 2 sequences each of size to your need. What should be the (( )) now, first problem is then I want to find out the number of comparators, you need to merge the 2 sorted sequence each of size n the number of comparators you need to merge 2 sequences each of size n.

Tell me n minus 1 are you sure? Tell me what is the number first, how do you do that you obtain a recurrence relation what is the recurrence relation. So, tell me the recurrence relation. Tell me what is the recurrence relation 2 n minus 1this is the number of process you need, how did you get that now how did you get it is 2n minus 1something has computer mind based on some base. Basis what is that basis 2 n prepare how to how do you because my feeling is that this type of problems, you can easily solve through recurrence relation I do not know other way how you can do it.

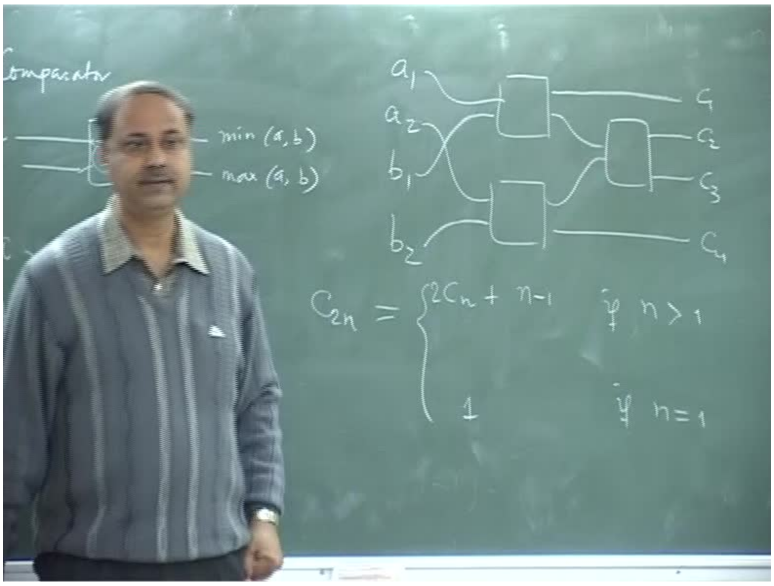
(Refer Slide Time: 30:59)



(Refer Slide Time: 23:21)

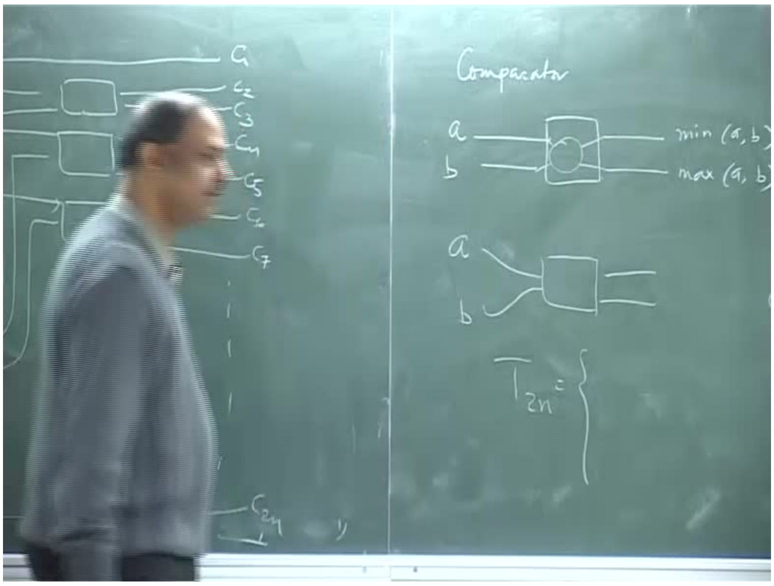


(Refer Slide Time: 31:27)

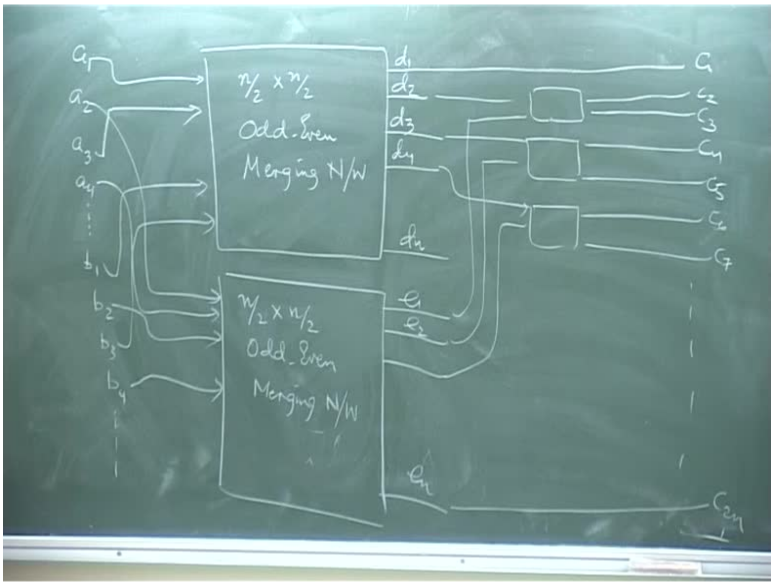


If you have any new idea tell me that would it be use full for me. So, if I consider c 2 n because, do not 2n in (( )) c 2 n is my comparator is the number of comparators you need then c 2 n can be written as c n for this block and for this block also unit again see there. So, 2 c n plus how many comparators unit then minus 1plus n minus 1 and you know a recourse relation is vary if you put boundary quotation also. So, 1way is in that case 1if n equals to 1if n is greater than one.

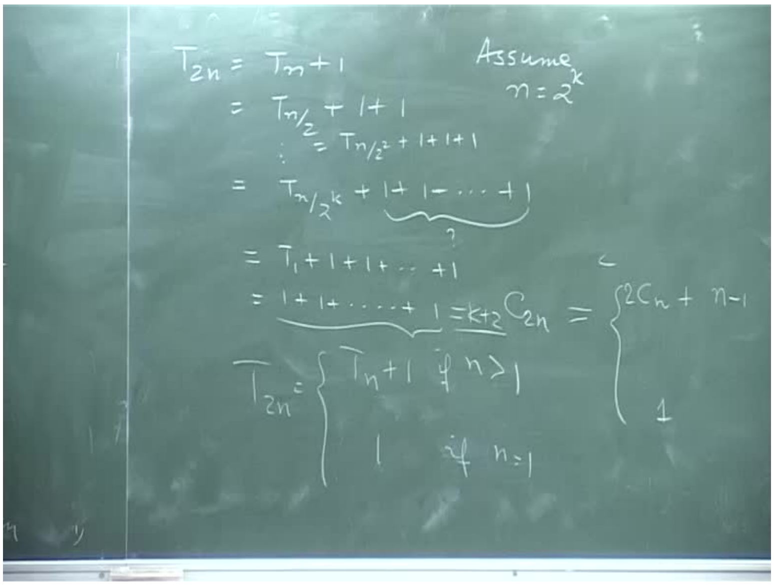
(Refer Slide Time: 33:09)



(Refer Slide Time: 23:21)

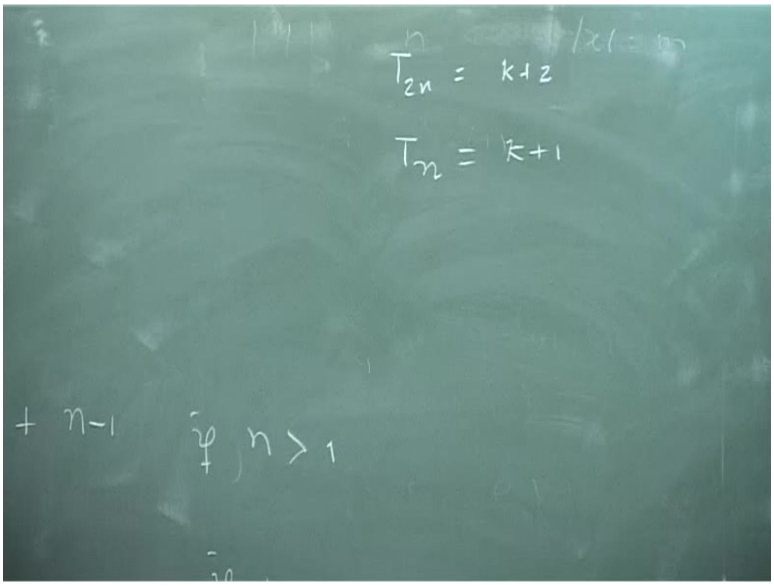


(Refer Slide Time: 34:20)

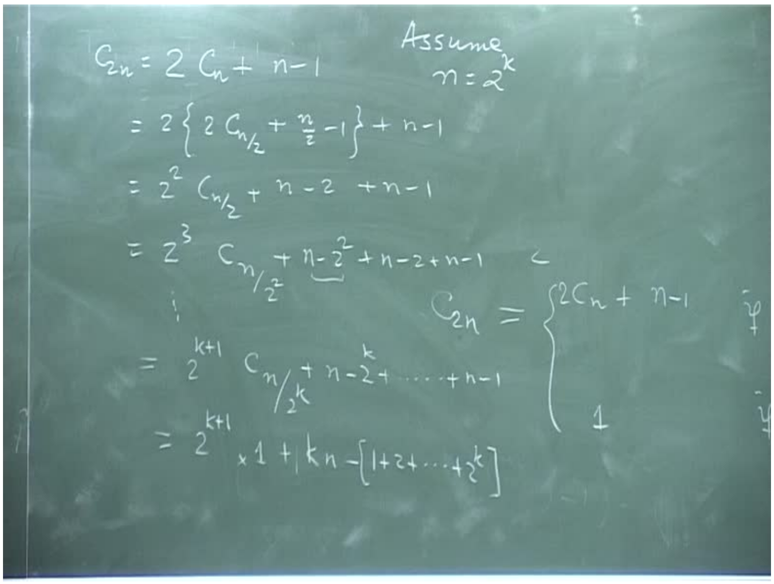


Now, if it is a case you check whether you recirculation varying for this or not that is n equals to 22 into c 1c 1is 12 into 1is 2plus, 1see unit the c comparator agree so, this is the number of comparators you need will solve it later on now next 1is the number of parallel steps you need how many steps you need or what is the time of to merge 2 sorted sequences can you tell me. Now, t 2n is equals to what tell me the recurrence (( )) function why to create this 2hubs but, they can do it parallel analysis so, I can write t n plus 1if n is greater than 11 if n equals to 1 anything else, I have to find out 1is number of comparators another 1is number of now I need to know the solution of this 2 so, I can write this 1as t n by 2to the power of k plus 11, 11 how many 1k minus 1next time I will write t n by 2 to the power 2 is equals to 1 plus, 1 plus 1sure k plus 1k minus 1is not there now, k is them still or not sure so if it is k plus 1then it is k plus 2 because this is t 1t 1is 1so, k plus one.

(Refer Slide Time: 36:43)

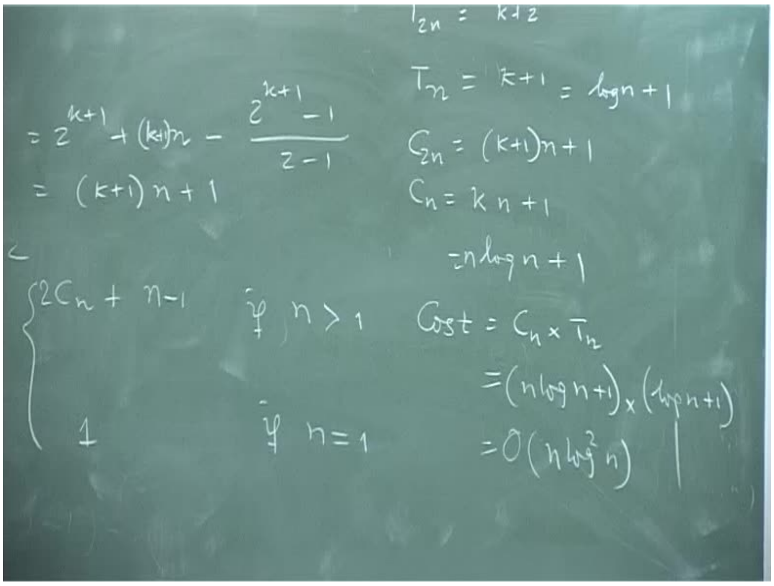


(Refer Slide Time: 36:56)



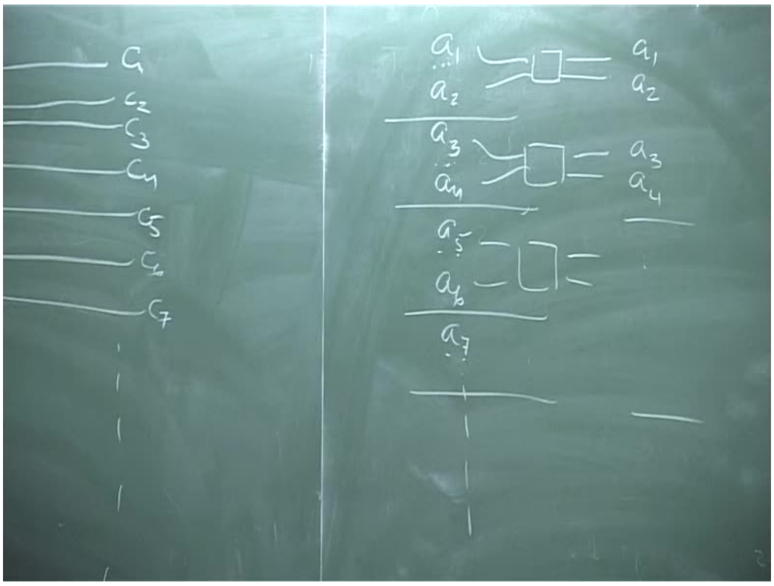
So, t 2n t 2n is your k plus 2 so, basically if I have to write t times of n it is k plus 1, if I have to write in terms of n then it is t n is equal to k plus 1, while writing the time(()) which and not can 2n now, if it is n this k plus 1now what about this solution 2 times c n plus n minus 1now, tell me what I write is it can, I write this can I get this yes or no if it is that then I can write 2 square c n by 2 plus n minus 2 plus n minus one.

(Refer Slide Time: 39:47)



So, next time what I will write tell me, what I will write here 22cube here c n by c n by 4 here n minus is it so, what happens to k time 2 to the power k plus 1c n by 2 to the power k n minus 2 to the power k. So, how many ns k ns tell me k plus 1where this 1k plus 12 to the power zero 2 to the power 1 and so, it is gives you k plus 1n plus 1 so, you get c 2n k plus 1n plus 1so, you can what is the value of c n k n plus 1k is log n k is long n. So, n log n plus 1 now, first is c n into t n c as in n log n plus 1and t n is log n plus 1which is ordered then (( )) you got it and belongs specification. Now, what is the sequential lower bound to mark the 2 sorted sequence of each of size n. What is the lower bound 2n are you sure to merge the 2 sorted sequence each of size n, what is the lower bound 2n minus 1it is n plus n minus 1. So, it is 2n minus 1 so, it is order n and it is not this is logs square factor in a way from the lower bound.

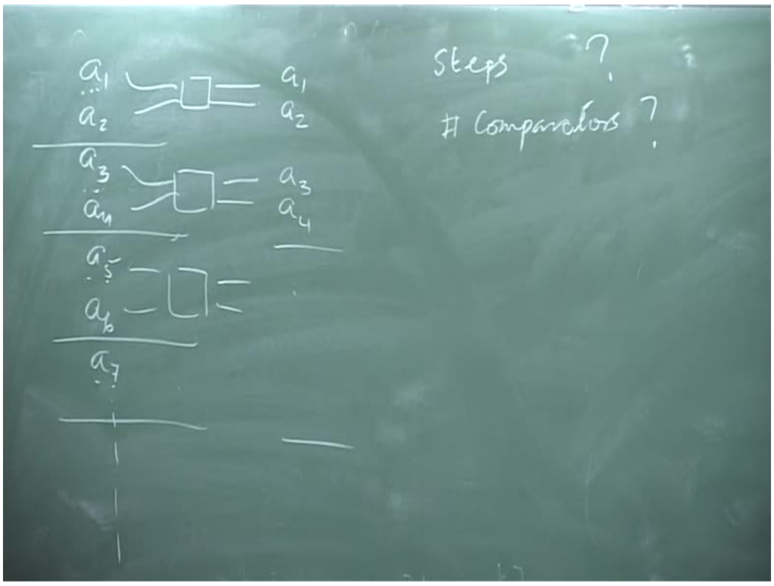
(Refer Slide Time: 43:26)



So, the this is your parallel odd even merging network. Now, can I use this merging network to sort the energumens? let us assume that a1 a2 a3 a4 a5 a6 a7 and so on this other an elements there admit really recruited and you want to sorted using odd even merging technique the strategy is very simple. You assume that there are n by 2 groups each of size and these 2 side group is a 2 subsequences. Each of them sorted that is that I have defined that this is a group, this is another group, this is another group and this is another group and so on. And we assume that the 2 sequences each of size 1.

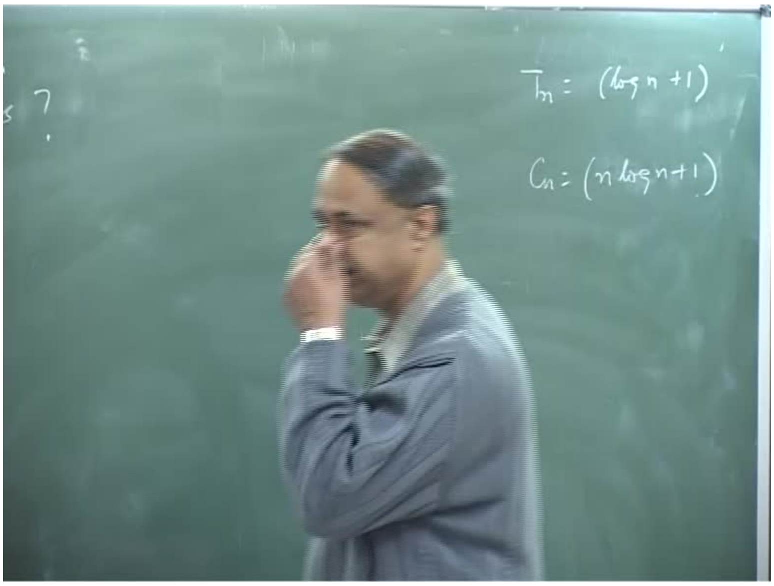
Now, this is sorted this sequence is sorted, this sequence is sorted, this sequence is sorted and this sequence is sorted if it is the case I can have 1comparator to give me the sorted sequence. I can have another comparator so, I get a 1 new a 1 new a 2 new a 3 new a 4 and so on. Now, you have n elements these n elements property in that the other n by 2 sequences n by 2 sequences is sequences of size 2 and they are sorted.

(Refer Slide Time: 46:16)

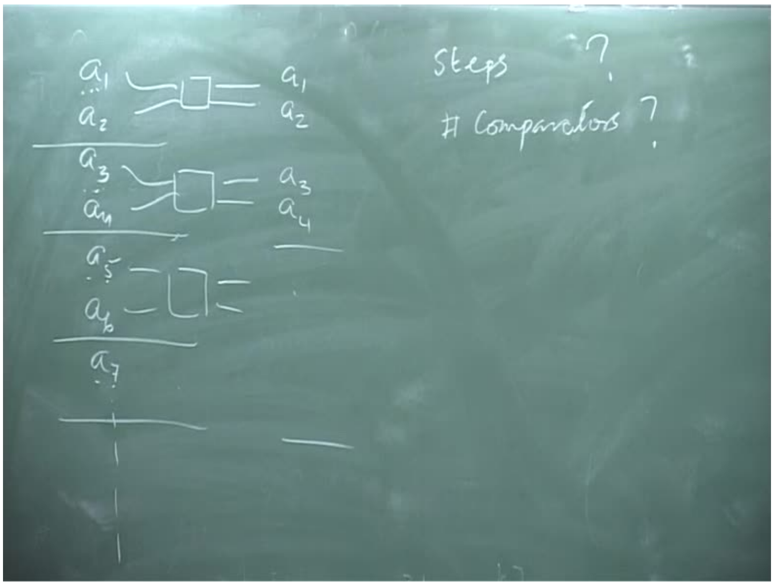


Now, you dividing in based on 4 elements, 4 elements, 4 elements these 2 elements sorted these 2 element sorted you can use odd even merging of merging network to sort this you, can use the odd even merging network sort this and so on agreed,. So, after some step you will find then you have n by 2 sorted sequence n by 2 sorted sequence and you use a merging sorting network of size n by 2 plus n by 2. So, give you a sorted sequence of size n so, if it is the case now you tell me what is the number of steps you need and then number of comparator x unit.

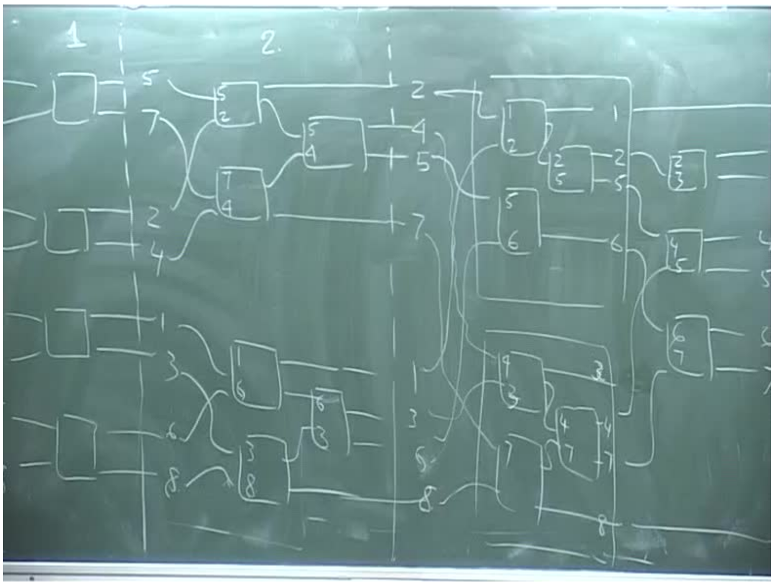
(Refer Slide Time: 46:52)



(Refer Slide Time: 46:16)



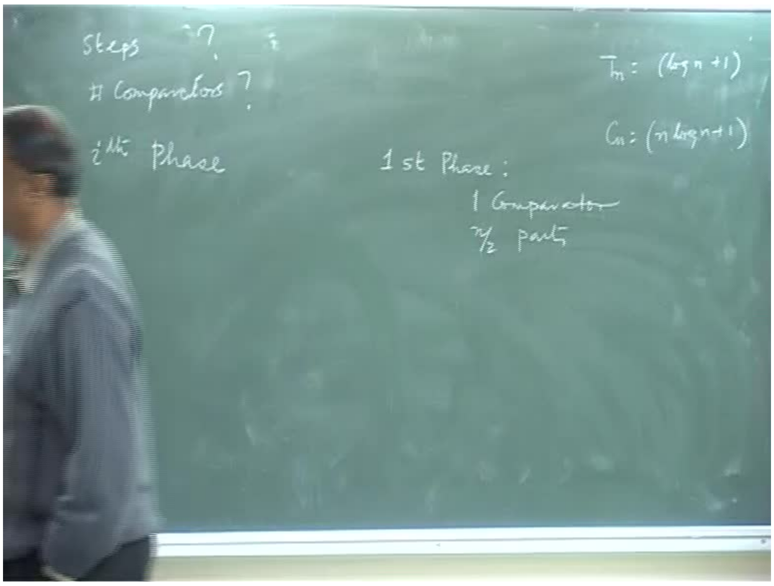
(Refer Slide Time: 52:02)



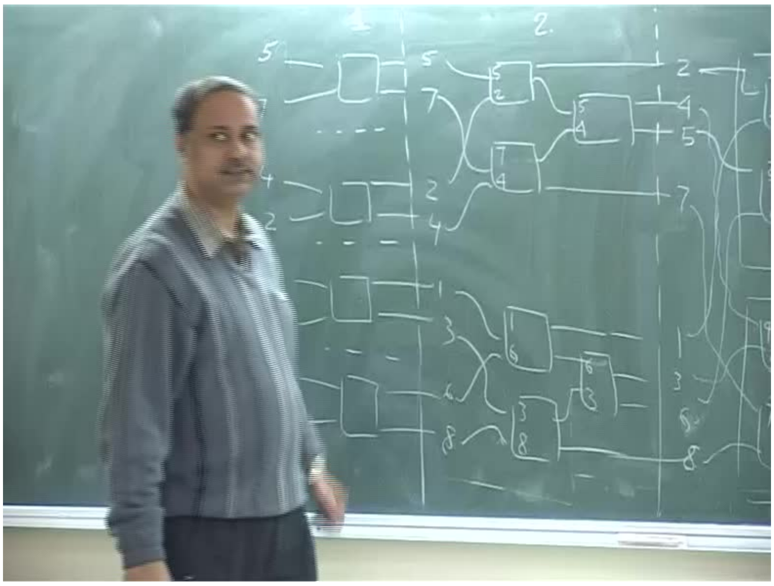
So, these are the two things you have to find out that number of steps you need then number of comparators you need here. We obtain t n c n t n is what t n is log n plus 1and this is n log n plus 1. Then we obtain already now, this strategy let us first for simplicity. Let, us consider 1example then you realize tell me 8 random numbers 1of them you can 5 the next number 7 4 said 2 8 t this is the 8 numbers you have.

So, use comparators here output is 5 7 24 13 6 8 now use 5 is come 2is come here 7 4 16 3 8. So, 2and here 5 will come here 7 4 4 5 here 16 3 3 6 8 now, this 1 basically like all repair. What is this same 2 has come this is 3 so, you will be getting this sorted (( )) so, you can think that is page 1 page 2 and this is page 3. Now, if you have n elements how many changes will be there login failures?

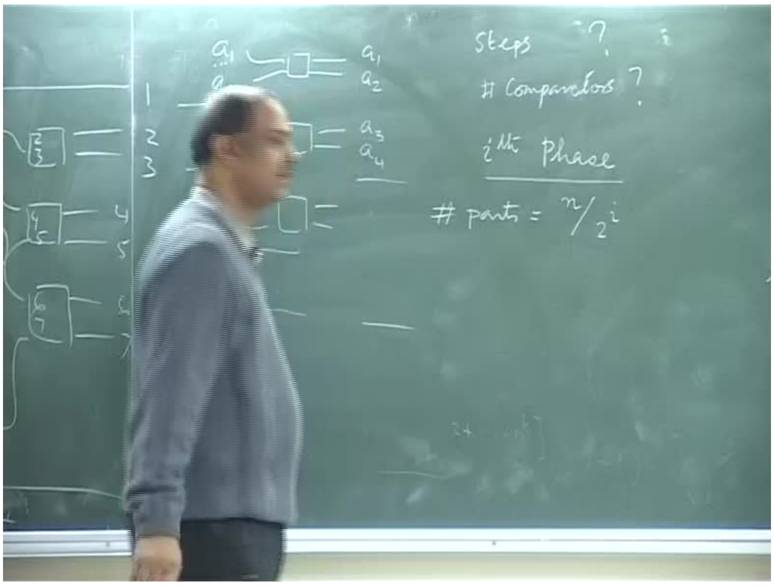
(Refer Slide Time: 52:47)



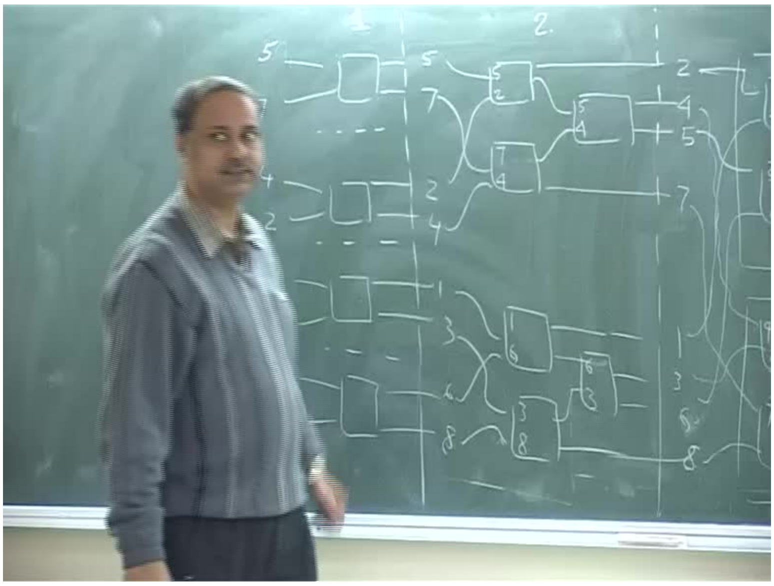
(Refer Slide Time: 52:56)



(Refer Slide Time: 53:48)

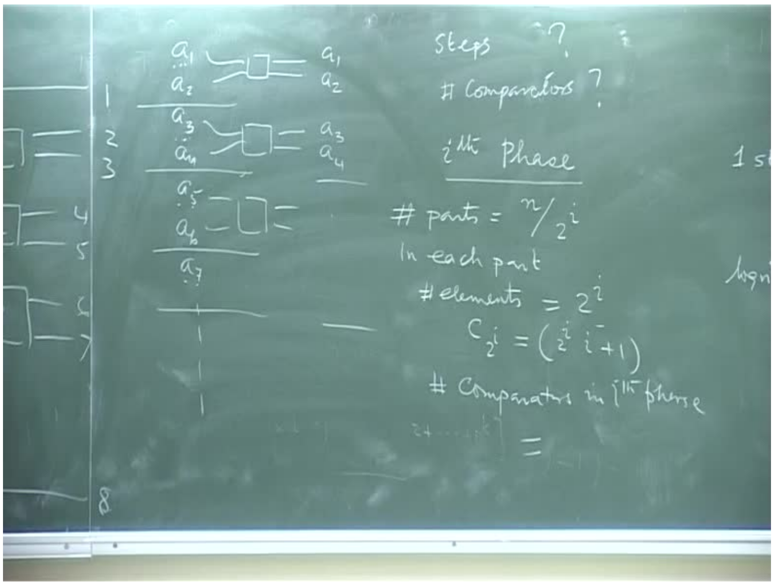


(Refer Slide Time: 52:49)



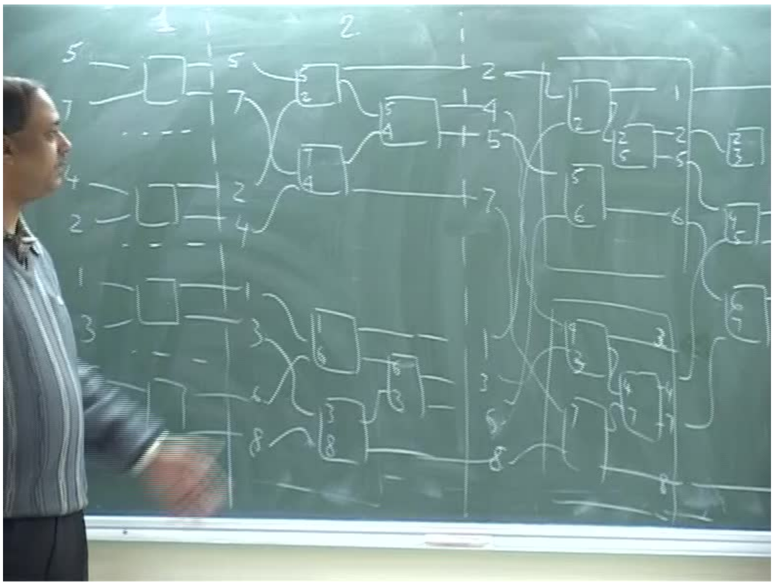
Now, in the i-th phase we are looking for in the first phase you need 1comparator and n by 2 path this is 1comparator 1there are n by 2 parts in log in phase log n-th phase, you need n log n plus 1comparators and 1path. Now, you need to find out I need to find out what happens for the i-th phase can you tell me in the i-th phase. How many parts are there number of parts in the i-th phase in the first phase, How many parts in the second phase in the i-th phase n upon 2 to the power I. And now, in the i-th phase and in a particular path so, k-th path or a first path because they are chimerically distributed if you have 3 here can also here 3.

(Refer Slide Time: 56:16)

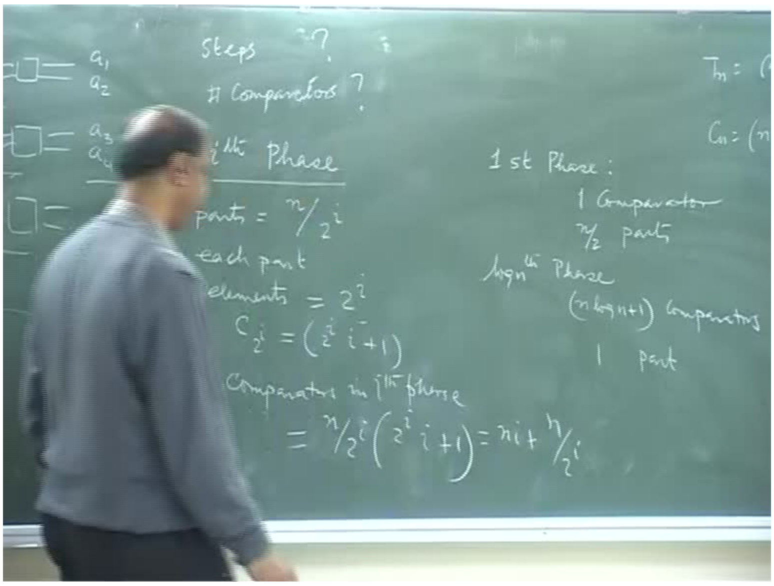


So, I need to find out any 1 path how many comparators are there, that will solve your problem in the i-th or in each part how many elements are there in each part, in each part how many comparator elements are there number of elements 2 to the power i. Very good if I know this 2 to the power i. What is the number of comparators you need so, the number of comparators c 2 the power i is 2 to the power i into i plus 1. So, in each part you need that many comparators agreed. So, you need total number of comparators total number of comparators in i-th phase is one.

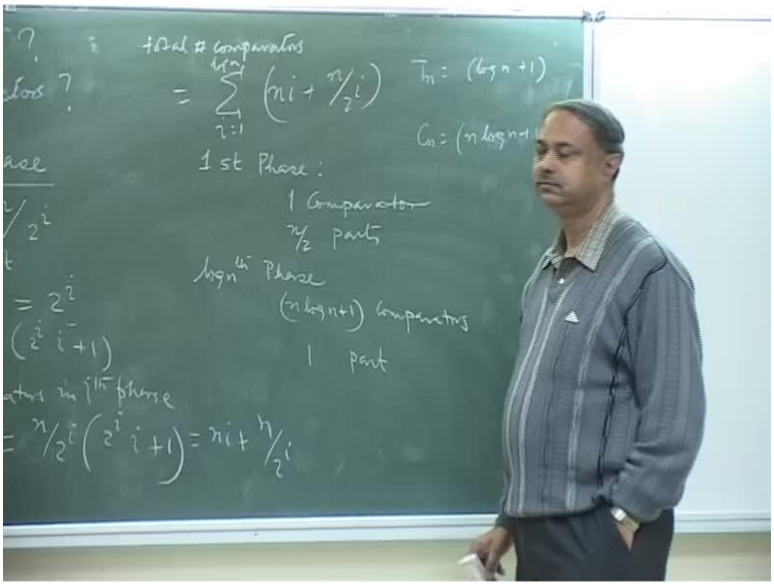
(Refer Slide Time: 52:49)



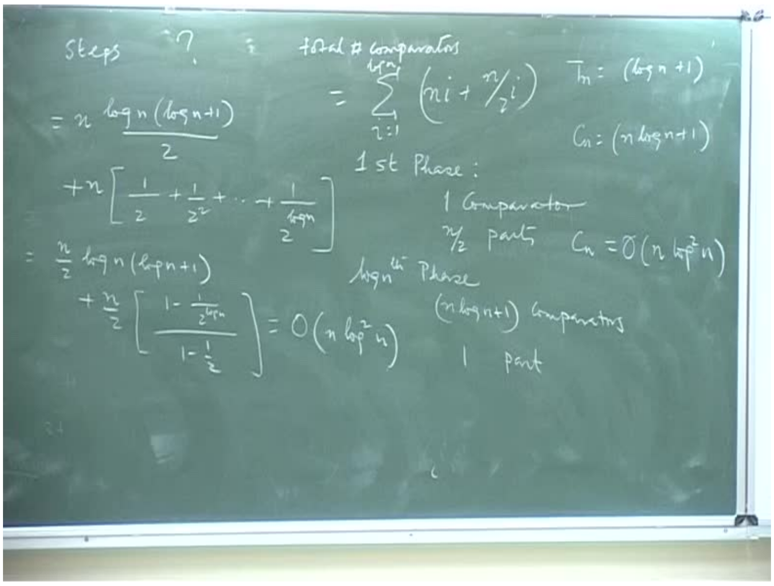
(Refer Slide Time: 57:05)



(Refer Slide Time: 57:55)

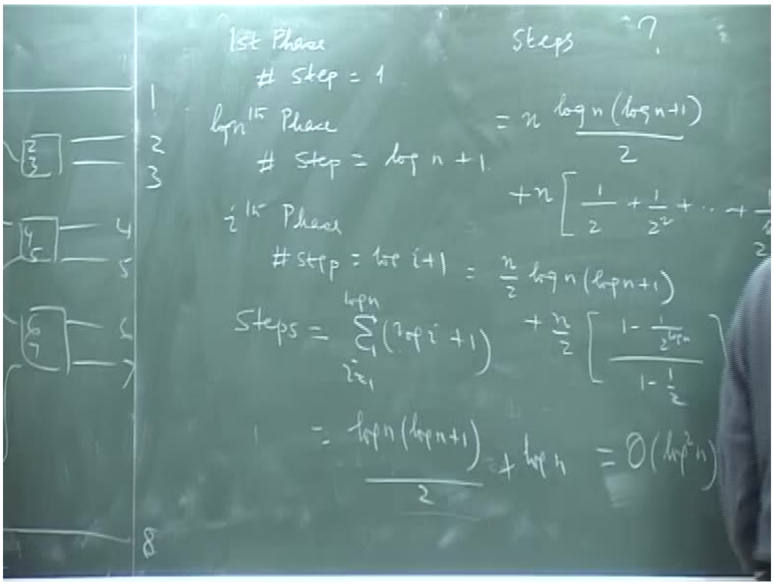


(Refer Slide Time: 1:00:09)



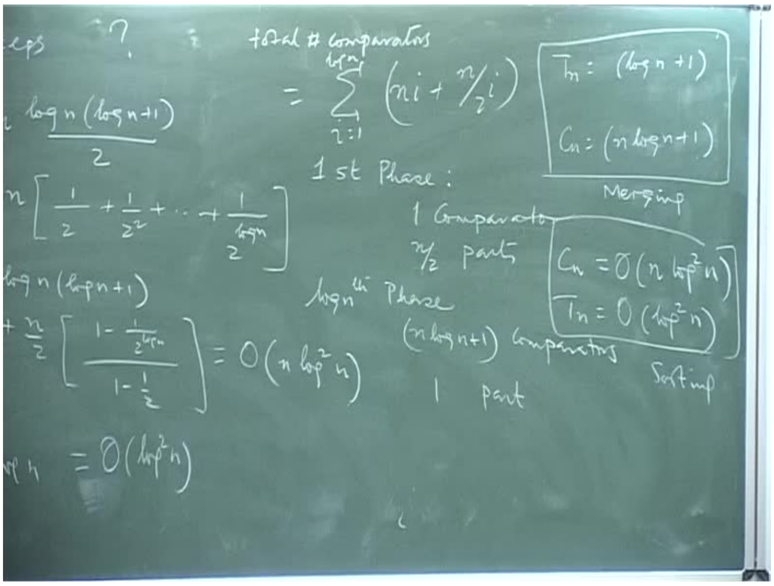
So, clip pick in this part I need 3 comparators, compare what is the total number this plus this so, tell you same thing you tell n upon very good so, you have n upon 2 to the power i 2 to the power I, i plus 1that is your n i plus n upon 2 to the power i again so, the total number of comparators you need total number of comparators equals to summation n into i plus n by 2 to the power I, i is 1to log n. Now tell me the sum, what is the sum n log n plus tell me the value of this, this is order n because this is 1 by n this becomes 11n by half.

(Refer Slide Time: 1:01:46)

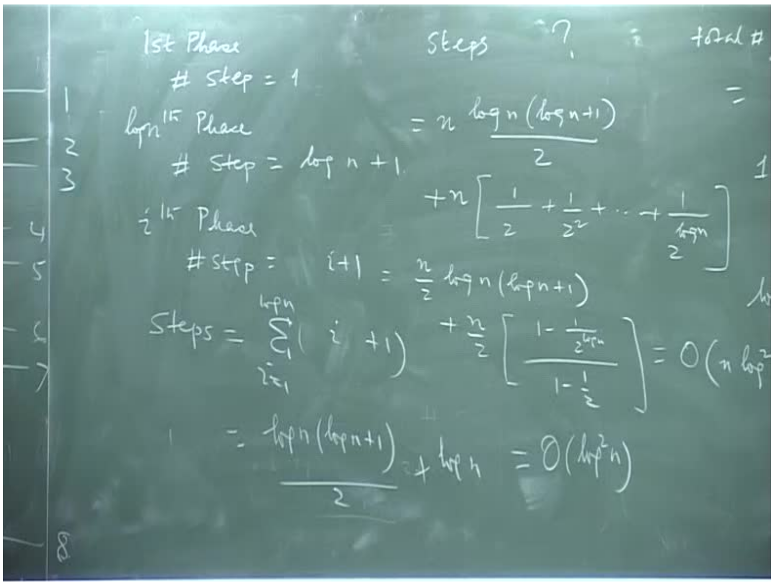


So, it is this is order n and this is so, which i can write order n log 2n so, you obtain your cost by comparators number of comparators is ordered n log squared n. Now, what is the total number of steps use 2log n, how did you get in the i-th step in the first step first phase, first phase number of step is 1n-th log n-th phase number of step is log n plus 1i-th phase number of parallel steps is log i plus 1.

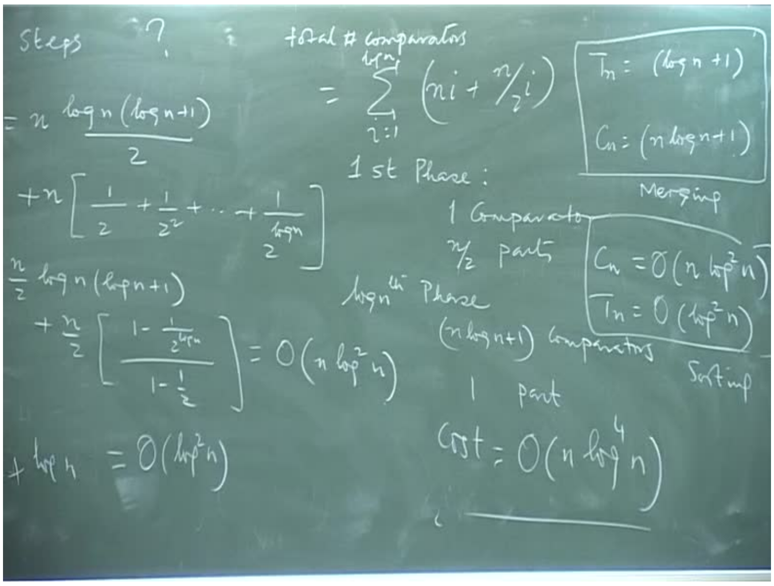
(Refer Slide Time: 1:02:05)



(Refer Slide Time: 1:02:30)

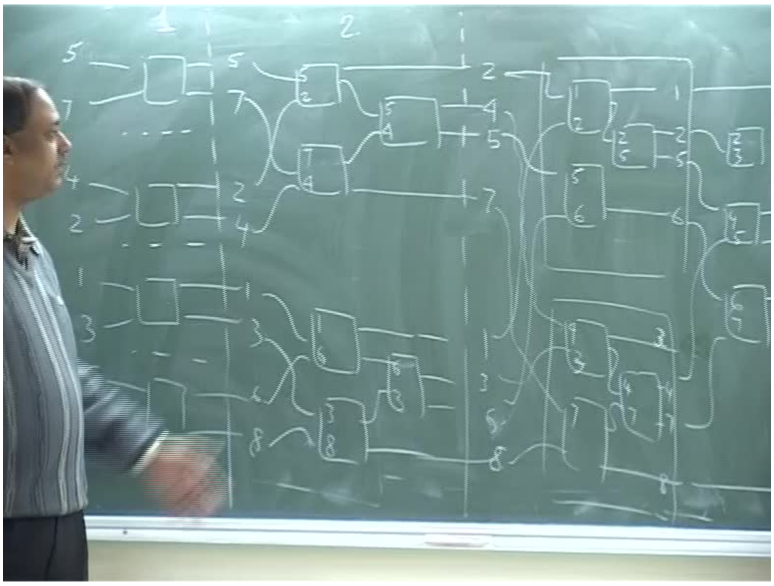


(Refer Slide Time: 1:02:46)



So, the total number of steps is equal to summation over i equals to 1 to log n log i plus 1. Which is log n plus 1 divided by 2 plus log n by need not 2 log so, this gives you ordered log square n so, this is that for merging this is for sorting. So, the cost of sorting network is ordered n log to the power 4 n log lower bound of sorting algorithm is ordered n log n here it is n log term 4 n.

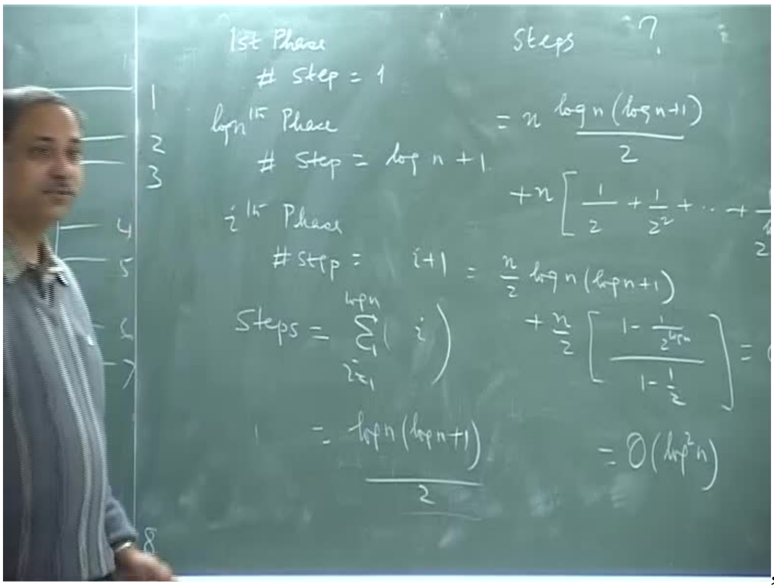
(Refer Slide Time: 1:04:10)



So, log cube n act as aware form the cost of cumulative now, can you read that what is the cost for algorithms are or in (( )) sorting network in last class i covered n square log n and that was a more complex 1. You observe that it forms the binarity connection. So, who is going were now, how many connection you can have, also many problems are there but, here the it is a uniform structure comparators are uniform in nature and this a clean diagram known he has some problem. But, the number of comparators to many n log squared. See why line here you observe these part is not be used when the data has been moved to this phase this comparators are either.

Now, what my point is that can you do some manipulation or some swathing technique so, that these can be used for this also that you know the only thing is that this is the structure, this structure and this structure is same this structure and this structure is same. I could have only this link has to be say that so, that if we can manipulate little with that, we can accept the code in that case that number of comparator should be use and number of comparator should, you can reduce that cost should reduce what you are increasing, see everything you converted you are increasing the complexity of the connection. By using say zero the connection structure is this that data will come be coming from this side and if it is 1the data will be coming from this side.

(Refer Slide Time: 66:17)



,

Now, those type of manipulation you put so, you have to work when (( )) and you see if it is possible. Can we reduce this by say log square factor or even if it is a log factor. Let, us do it let, us see how it looks arises what first phase it is 1 so, it is i he still thinking about that 1 be happy is it now, if it is this is everything is you right yes it is i but, it's not a matching first phase why is come matching but, these is a original somewhere has been comparator , that 1why it is log n here i do not k or k n plus 1there is another you are not attractive at this time lets, prepare that should be clear in that case that the simple (( )) from 2n, 2n conversion here only you have to wait and see you have to see that is should be cared. So, make the correction from the conversion t 2n to t n here it should be t n instead of k n plus one. So, you can consider this as you have to 1of the stamp paper problem that can you use less number of comparator to have this odd even merging number networks. You can define your operators also will not that you define now, for that you can take the help of the art of computer programming Boolean shortage such type by (( )) there is a sorting network (( )) minimum sort number of comparisons so you can use some of them or 1of them to find the possible why 2elements for merging. Why 2 why not 3 I have the very good sorting network for 5 elements which take 7 7 comparisons why not 5 5 times so but, you have to do the little study on that and then you can design.