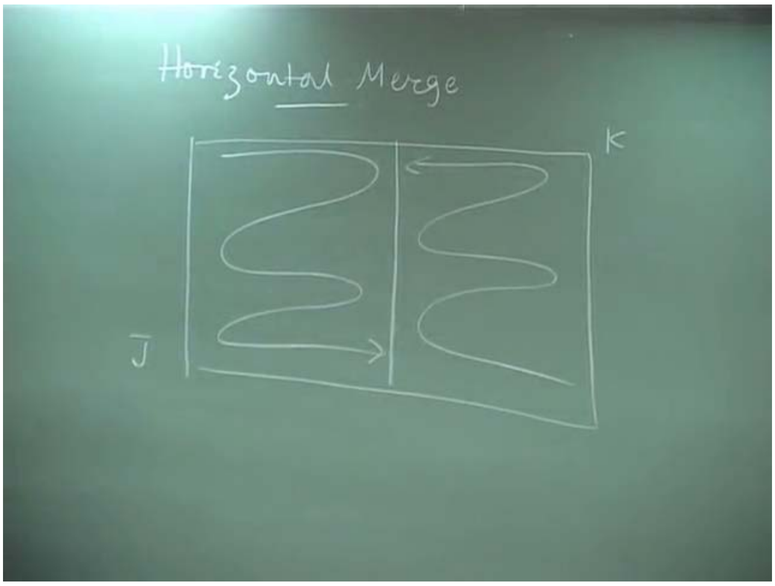
Parallel Algorithms
Prof. Phalguni Gupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 10

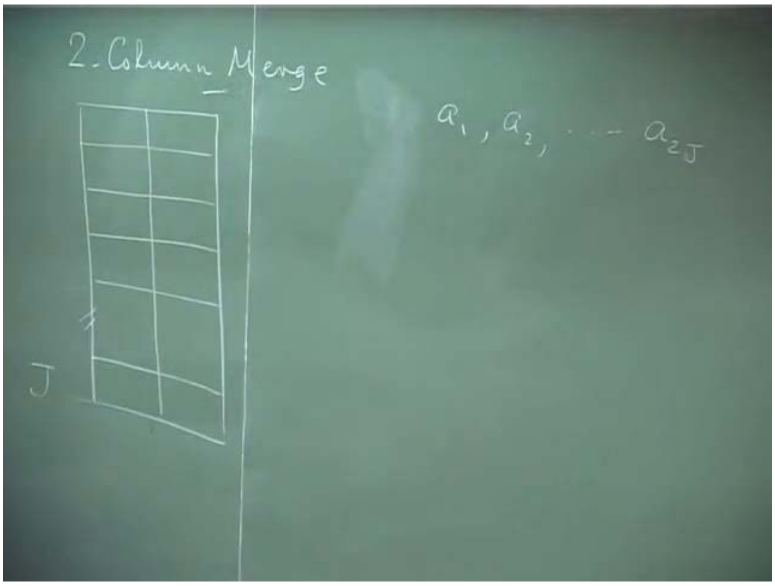
So, in the last class, we were discussing about what we discuss, we discuss about the horizontal merge. Now, of vertical merge vertical merge, now today first we will be discussing horizontal merge and then we will discuss the main sorting algorithm. Finally, we will try to consider an example of which because the through which we can say how the algorithm wants.

(Refer Slide Time: 00:49)



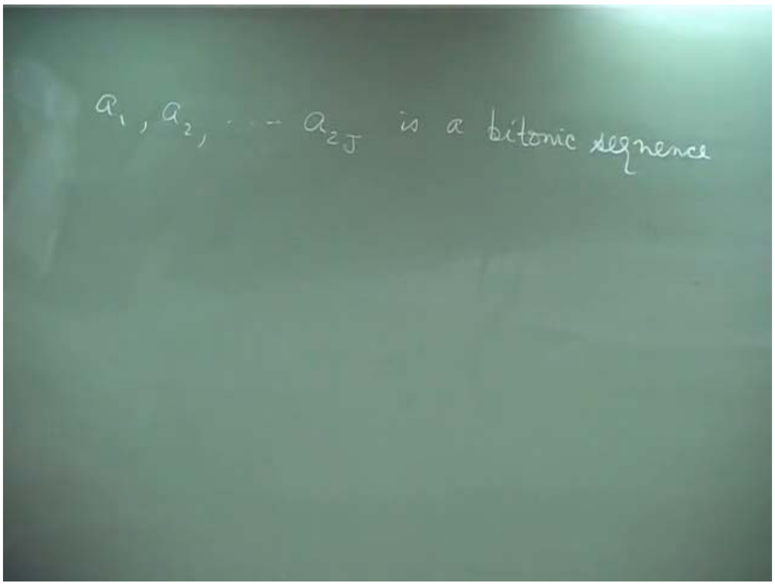
The horizontal merge first part, the problem is that suppose you have J cos k the problem is this is increasing order, this is decreasing order and you want to merge it. In order to do that, first we need to define say two column merge that problem is that you have one column J rows and 2 J elements.

(Refer Slide Time: 01:27)



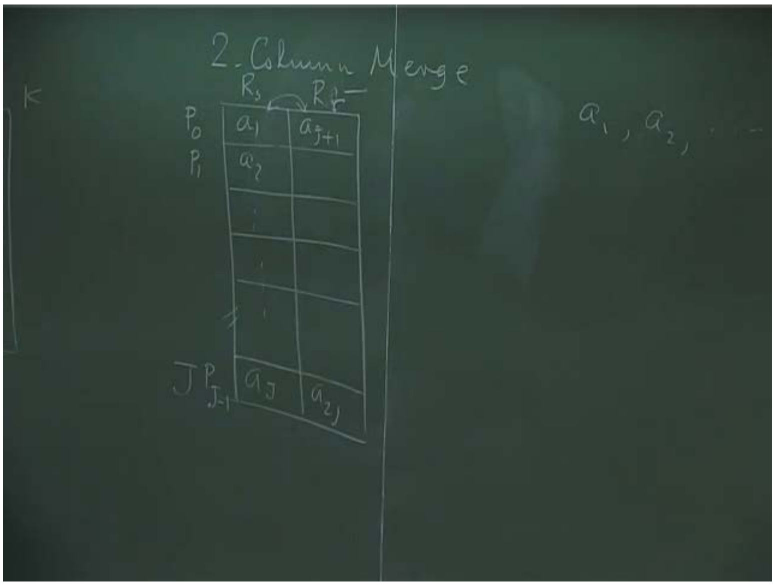
You have a 0, a 1, a 2, a 2 simplicity down there is 3 1, a 2, a 2, J this two J elements you have which is a bitonic sequence in the second line absent one is with us.

(Refer Slide Time: 01:57)



The other one is 18, 17, 4 plus 4, 8, 9, 7, 17, so a 1, a 2, a 2 J is a bitonic sequence and the elements are stored in a column of J rows in such a way that you have a 1, a 2, a J and then a J plus 1 to a to w J.

(Refer Slide Time: 03:05)

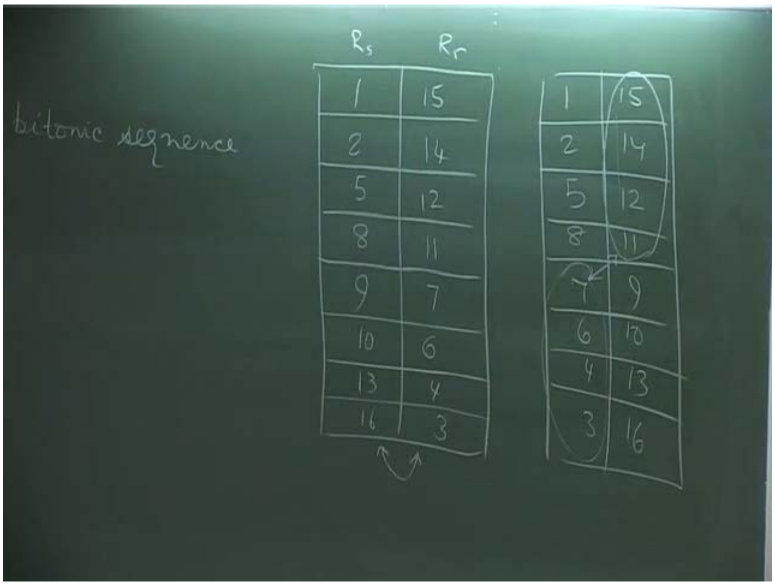


Let us assume that this delta is stored like this P 0, P 1, P J minus 1, this P 0, P 1, P J minus 1 are the processors and this is in R s resister this is in R k resister or R r resister. So, you have a 1, a 2 a 2 J elements in form a bitonic sequence and we have a column of J rows r, n J rows. Each row is assign to a processors p 0 to p J minus 1 and i th row contains the a i and a i plus J, a i is in R s resister and a i plus J is in R r resister. So, what you are going to do it see the simple thing is there in the bitonic sequence is based on the only one property that a i is compared with a i plus 8.

So, this is ready for you that this has to be compared this has to be compared, so if you compare and rejected element will be in this side and accepted element will be this side. Now, after that I want to do it recursively, so what happens here after comparing these elements will be these elements will be a bitonic sequence. These elements will form a bitonic sequence and all these elements will be less than or equals to this one.

Then, recursively will be solving these two, so same idea that you have this one compared a change compared a change. Now, you have to bring these elements in this zone and these elements should go in this zone. If I then this will be comes a bitonic sequence, this will be another bitonic sequence and all these elements should be smaller than these elements, what I am doing this is compared with this the rejected elements are lying here accepted elements are lying here. Now, I want to make it these two halves, you know to make these two halves and recursively I want to use it. So, these halves should be in touch with this half so that this forms a bitonic sequence this form.

(Refer Slide Time: 06:53)



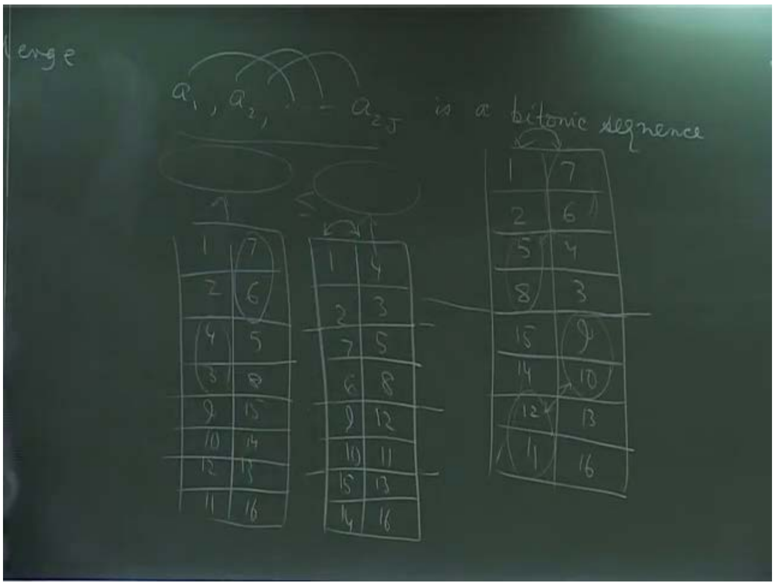
R_s	R_r
1	15
2	14
5	12
8	11
9	7
10	6
13	4
16	3

1	15
2	14
5	12
8	11
7	9
6	10
4	13
3	16

Another bitonic sequence and all these elements should be smaller than these elements and then you do it recursively say for example, tell me bitonic sequence of 16 elements 1, 2, 3, 4, 5 for a 1 2. So, this is a bitonic sequence you have and this is a R s resister and this is a R r resister, what we are doing that a i is to be compared with a i plus n and the accepted elements in R s area and rejected elements in R r resister.

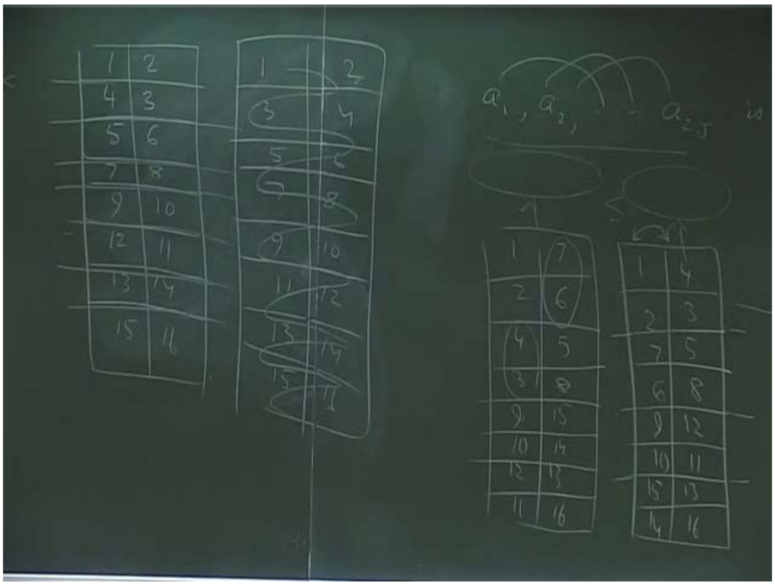
So, if it is the case you get the result of this like, now you observe you observe this becomes a bitonic sequence and this becomes another bitonic sequence and is a by the definition of the by the carom of bitonic sequence. Now, you want to use the same algorithm you want to same algorithm for what you have. So, what I want that these elements should be here so that I get the bitonic sequence and these elements should be here so that I get another bitonic sequence.

(Refer Slide Time: 08:15)



So, what I will do it, I want to inter change this thing, so if I inter change it, I get 1, 2, 5, 8, 7, 6, 4, 3, you get 15, 14, 12, 11, 9, 10, 13, 16. So, you observe that you get two bitonic sequence and all these elements will be smaller than these elements. Now, again you perform two halves compare a change and then inter change these two elements, so you get here 1, 1, 7, 2, 6, 4, 5, 3, 8, 9, 15, 10, 14, 12, 11, 13, 16. If I inter change these two, I get 1, 2, 4, 3, 7, 6, 5, 8, 9, 10, 12, 11, 15, 14, 13, 16.b. Now, you observe that this is a cluster of four element, this is a cluster another four elements and each cluster is a bitonic sequence and again you do compare a change.

(Refer Slide Time: 12:12)



 So, compare a change and then inter change, so you will get the and 5, 6, 7, 8, 9, 10, 12, 11, 13, 14, 15, 16 and then you get basically eight such bitonic sequence you just do the compare a change you get 1, 2, 3, 4, 5, 6, 7, 8. So, you get the sorted sequence in this form right here. Now, if you observed that I could have done after doing the compare exchange, we observe that this sequence is the bitonic sequence this sequence is the bitonic sequence and all these elements should be smaller than these elements.

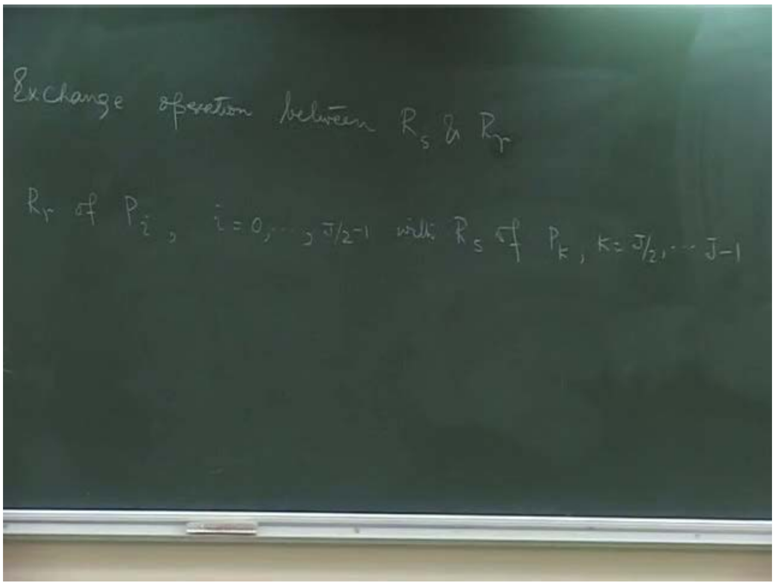
Now, I could have done the column merge, I could have done the column merge and I could have could have got this sorted this is also sorted sequence. It is based on the bitonic sequence, but is the problem is that I do not achieve the goal like that 1, 2, 3, 4, 5, 6, 7, 8 like that which I need for horizontal merge because is the horizontal merge it should be like this not this way.

(Refer Slide Time: 14:44)

So, that is the reason why instead of doing the column merge we are doing that interchange first and then we are a repeating that two column merge so that data is to final data is told in this form A 2 column merge. First step is that perform compare exchange operation between R s and R r registers, we have already assume the data is in the form that a 1, a 2, a 3 a 2 J in such way that a i and a i plus J is in the i th row and R r R s and R r registers respectively.

Now, if J is greater than 1, if J is equals to 1 that means there is only one row and that means that you have the only two elements which form the bitonic sequence, just one compare exchange is sufficient to make it sorted form, now if J is greater than 1.

(Refer Slide Time: 17:07)

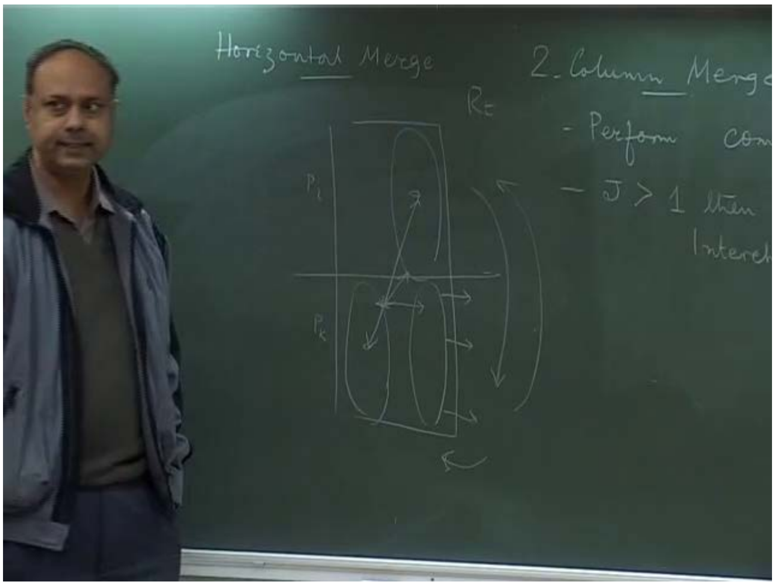


Exchange operation between R_s & R_r

R_r of P_i, $i = 0, \dots, J/2-1$ with R_s of P_k, $k = J/2, \dots J-1$

What I am going to do that rejected element has to be compare has to be inter change with accepted element of that say inter change content of R r of P i, i is zero to J by 2 minus 1 with R s of P k, k is J by 2 to J minus 1. Now, how we will going to do this see what we told this has to be inter change how are we going to do that x equals to inter change, what is x.

(Refer Slide Time: 17:30)

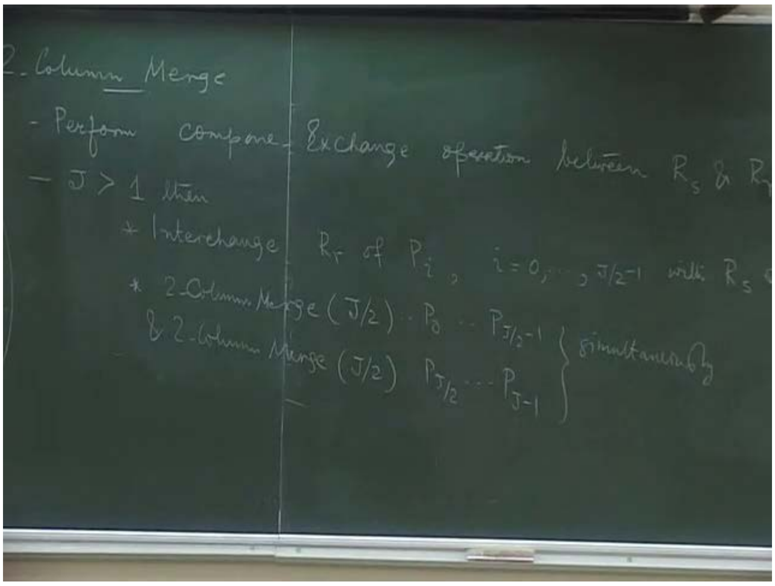


How are you going to get these two values, this is in P i and this is an say P k and it is in the form of mesh P i connected to four neighbor only.

Student: Sir, more contents of a R r are there in R t.

Yes, that is the reason why we need R t, so the element here you moved to R t, then bring down next step you bring down all these elements here then you inter change this then you bring that, so this is the step and then finally, you move again these elements.

(Refer Slide Time: 19:04)



2-Column Merge

- Perform compare-Exchange operation between R_s & R_r

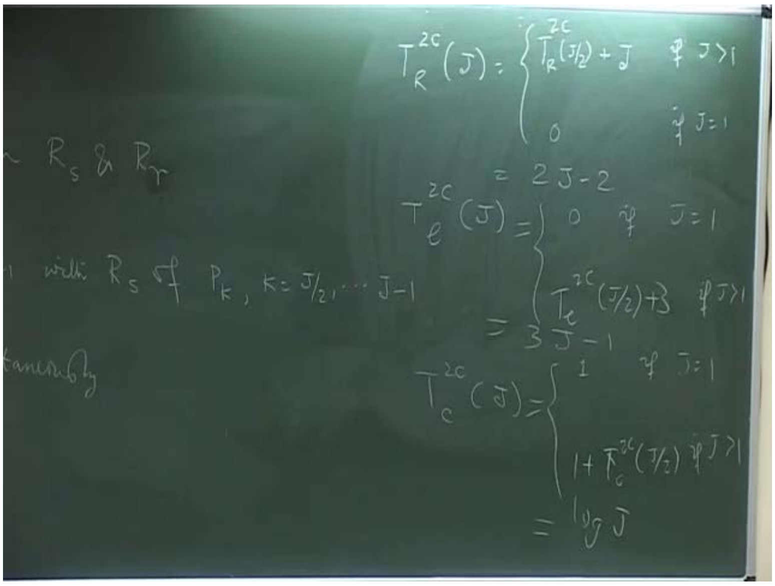
— $J > 1$ then

* Interchange R_r of P_i, $i = 0, \ldots, J/2 - 1$ with R_s

* 2-Column Merge $(J/2)$ $P_0 \ldots P_{J/2-1}$ & 2-Column Merge $(J/2)$ $P_{J/2} \ldots P_{J-1}$ } simultaneously

Now, after inter changing you will be doing two column merge J by 2 with J P 0 P J by 2 minus 1 and two column merge J by 2 p J by 2 to p J minus 1. Simultaneously, both of them can be done in paragraph, now if it is the case, then what is the time complexity what if notation are you what are complexity t 1.

(Refer Slide Time: 20:04)


$T_R^{2c}(J) = \begin{cases} T_R^{2c}(J/2) + J & \text{if } J>1 \\ 0 & \text{if } J=1 \end{cases}$
$= 2J - 2$
$T_e^{2c}(J) = \begin{cases} 0 & \text{if } J=1 \\ T_e^{2c}(J/2)+3 & \text{if } J>1 \end{cases}$
$= 3J - 1$
$T_c^{2c}(J) = \begin{cases} 1 & \text{if } J=1 \\ 1+T_c^{2c}(J/2) & \text{if } J>1 \end{cases}$
$= \log J$
R_s & R_r
with R_s of P_k, $k = J/2, \dots J-1$
taneously

I do not R 2 c and you have J, so routing if J is equals to 1, then it is 0, there is no routing, now if J equals to more the if J is greater than 1, you obviously have t R J by 2 2 c plus what is the routing required. This data has to be moved here and then again you will be moving back, so what is the routing step total number.

Student: Sir can we bring this data down and take this data up.

How?

Student: Say for this half.

This this data you have bringing here.

And then next one take up.

How are you going to because i told you the routing and R r register is the only register through which.

We wants the above R r and R t. How R r will sent direct to R r only routing resister also.

Student: Sir, the initial value.

This one you have to do here.

And R t.

R t now you bring.

Similarly, in the left above.

Student: Sir, the left part it is left.

This R r contents your original data R s contains you cannot touch R s r s this R s is all accepted data this you want to retain here this contains all rejected data, but you want to keep them here. Only this is the this part this part has to be inter change with this part in order to do that and your rotating is done through R r resister only. So, you move this data here, you bring this data here, inter change these two, then you send back the data here, then this you bring back what reverse way this you do one of method.

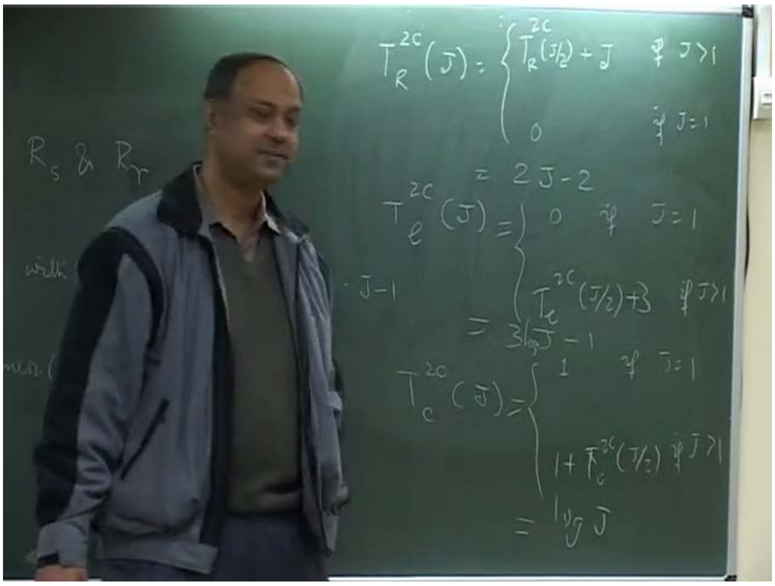
So, you need r routing R by 2, this side again R by 2 going back, you need R rotating R this way r is J rotating. Now, what is the solution of this you assume J is to be equal 1, 5 up to 2, tell me what is the solution J plus J by 2 plus J by 4, 2 J or 2 J minus 2, what time we did it entering this 2 J minus 2. Next one is number of exchange operations and another one is number of transferring this and compare these things.

First, let us do the number of compare exchange it is one if J is equals to 1, otherwise it is one plus P c 2 c J by 2 if J is greater than 1. There is no comparing exchange operation, so I write 1 plus P c 2 c J by 2 and this solution is logged here this solution is logged here what about here there is no exchange operations.

So, J O if J equals to 1, this side I am writing t e 2 c J by 2 plus if J greater than 1 how many exchange operations you have perform here, first is one is you are bringing data here, then second is you have bringing data here inter exchange operations. Then, you are sending back through routine then again you will be bringing the data here 3. So, it is 3, so it is got answer is what said un this is one and this is 3, 3 log low impact on 0 minus 1 are sure.

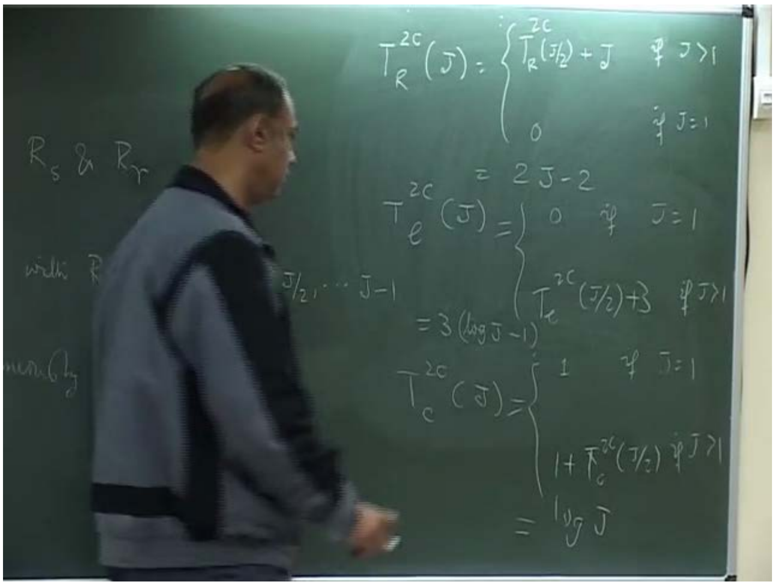
So, if I put J equals 1, then it is become 2, tell me what 3 log J minus 1, suppose J. Let us assume the thing J equals to 2, answer should be J equals to 2, answer should be 3 J equals to 3 answer should J equals to 4 answer should be not know, so what should be the log J.

(Refer Slide Time: 27:50)


$T_R^{2C}(J) = \begin{cases} T_R^{2C}(J/2) + J & \text{if } J > 1 \\ 0 & \text{if } J = 1 \end{cases}$
$= 2J - 2$
$T_\ell^{2C}(J) = \begin{cases} 0 & \text{if } J = 1 \\ T_\ell^{2C}(J/2) + 3 & \text{if } J > 1 \end{cases}$
$= 3\log J - 1$
$T_c^{2C}(J) = \begin{cases} 1 & \text{if } J = 1 \\ 1 + T_c^{2C}(J/2) & \text{if } J > 1 \end{cases}$
$= \log J$
R_s & R_r
with
J-1

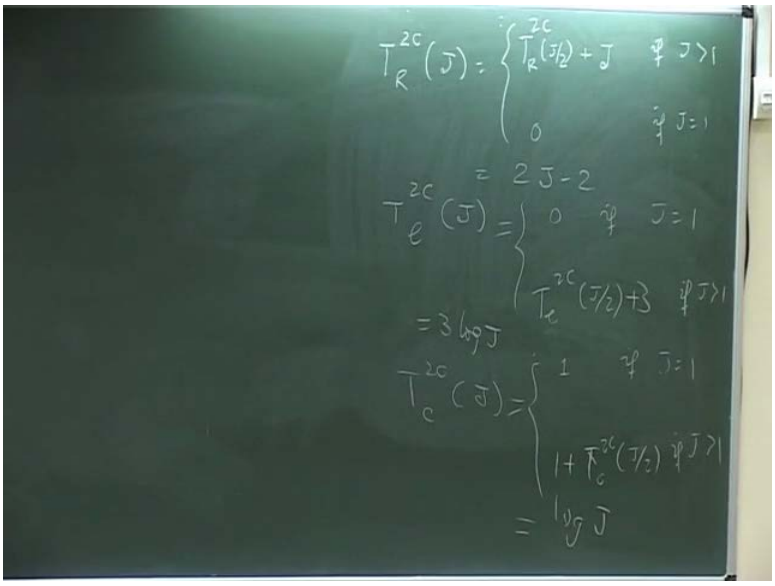
This is the thing on written, so it is 3 log J if it is J equals to 1 it is 0.

(Refer Slide Time: 27:55)


$T_R^{2C}(J) = \begin{cases} T_R^{2C}(J/2) + J & \text{if } J > 1 \\ 0 & \text{if } J = 1 \end{cases}$
$= 2J - 2$
$T_\ell^{2C}(J) = \begin{cases} 0 & \text{if } J = 1 \\ T_\ell^{2C}(J/2) + 3 & \text{if } J > 1 \end{cases}$
$= 3(\log J - 1)$
$T_c^{2C}(J) = \begin{cases} 1 & \text{if } J = 1 \\ 1 + T_c^{2C}(J/2) & \text{if } J > 1 \end{cases}$
$= \log J$
R_s & R_r
with R
J/2, ... J-1

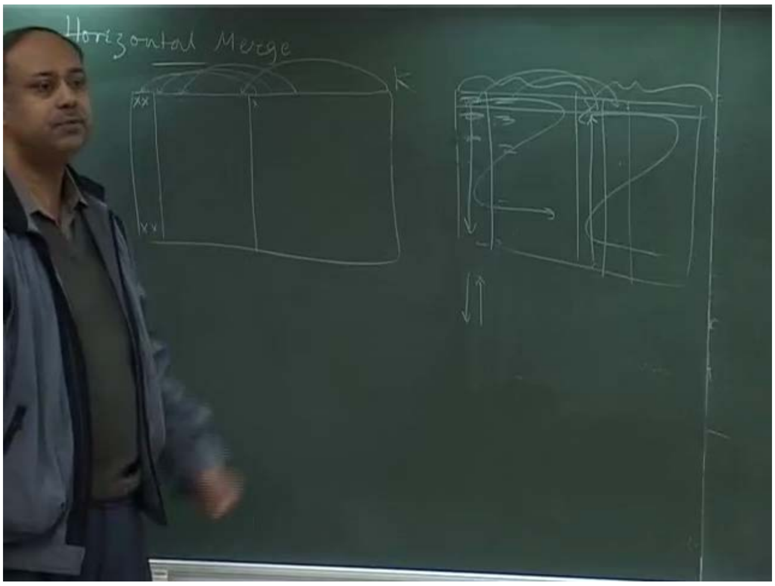
And 2 it is 3, 4 it is 6 and so on.

(Refer Slide Time: 27:59)


$T_R^{2c}(J) = \begin{cases} T_R^{2c}(J/2) + J & \text{if } J>1 \\ 0 & \text{if } J=1 \end{cases}$
$= 2J-2$
$T_\ell^{2c}(J) = \begin{cases} 0 & \text{if } J=1 \\ T_\ell^{2c}(J/2)+3 & \text{if } J>1 \end{cases}$
$= 3\log J$
$T_c^{2c}(J) = \begin{cases} 1 & \text{if } J=1 \\ 1+T_c^{2c}(J/2) & \text{if } J>1 \end{cases}$
$= \log J$

It is there, so it is 3 log, so the two column merge is done because you want data should be in such way that that all the elements in the first row smaller than the all the elements in the second row and so on, so this idea is used in the horizontal merge.

(Refer Slide Time: 28:28)



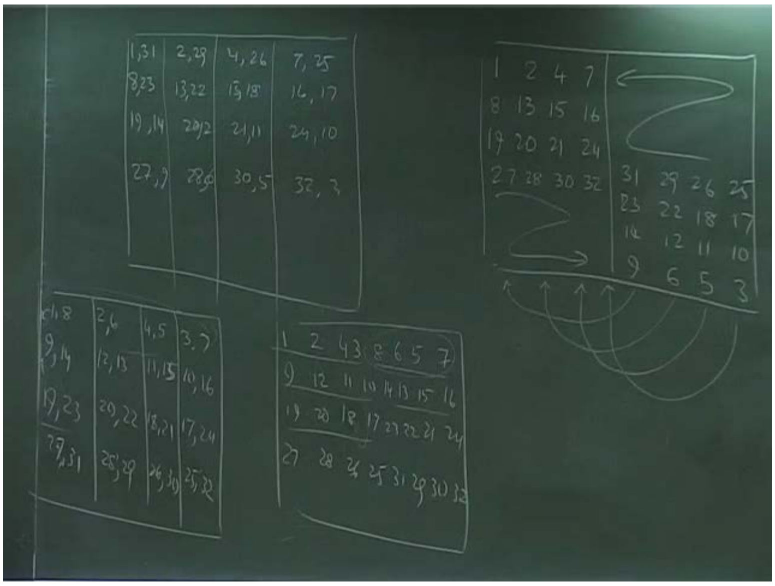
So, the elements you have in this forms all there it forms a bitonic sequence and it bring this data here. So, initial if you remember this forms a bitonic sequence, now you will observe that these elements, this is come here this also will form a bitonic sequence, you have two bitonic sequence.

Now, if this is an increasing order then this column is also in increasing order, if this is decreasing order, and then this column is always decreasing order. Now, if I bring this data here, this is forms a bitonic sequence because this was increasing order, this was decreasing order.

Similarly, if I bring this data here, I get another bitonic sequence here and so on, we also and also if I merge the two column merge here, and then all the smaller elements should be here next column element should be here and so on. Similarly, is the case, so what I do, I rejected element; I send back again these rejected elements are rejected elements in the first row.

So, then these elements are smaller than these elements, so all these elements will be smaller than these elements similar all these elements 0, 0, 2, 0 smaller than these elements. So, even if I am sending the rejected elements, these elements are smaller than these elements and also each of these will form a bitonic sequence. These will form a bitonic sequence and then you do the simple room merge each of them, we will find the sorted sequence.

(Refer Slide Time: 31:18)



1,31	2,29	4,26	7,25
8,23	13,22	15,18	16,17
19,14	20,12	21,11	24,10
27,9	28,6	30,5	32,3

1	2	4	7				
8	13	15	16				
19	20	21	24				
27	28	30	32	31	29	26	25
				23	22	18	17
				14	12	11	10
				9	6	5	3

1,8	2,6	4,5	3,7
9,14	12,13	11,15	10,16
19,23	20,22	18,21	17,24
27,31	28,29	26,30	25,32

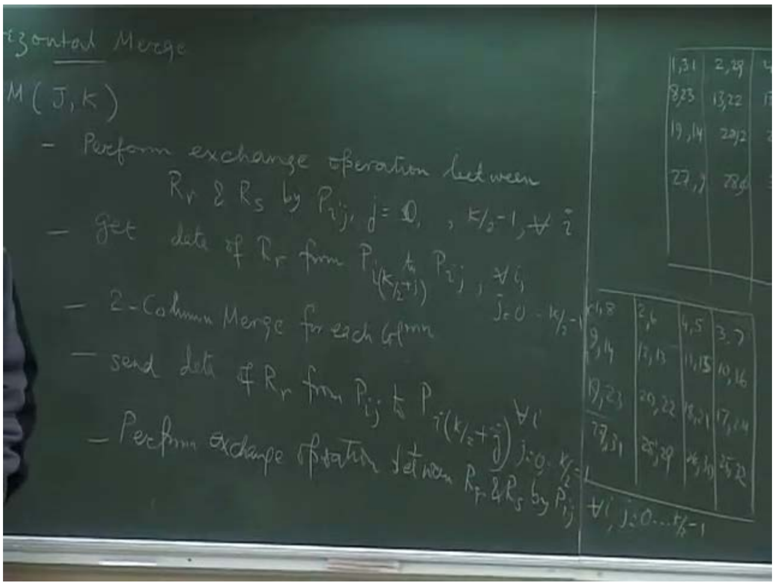
1	2	4	3	8	6	5	7
9	12	11	10	14	13	15	16
19	20	18	17	23	22	21	24
27	28	26	25	31	29	30	32

So, let us consider first one example before we discuss the algorithm 16 plus 16, 1, 2, 4, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 30, 21, 32, 33, 34, 30, 5, 30, 6, 30, 7, 30, 8, 39, 30. So, you have suppose this thirty two elements and this form increasing order, this form decreasing order and you observe this whole thing is a bitonic sequence. What we are doing, I am bringing these elements in the first column, these elements in the second column, these elements in the third column, these elements in the fourth column. So, you get for I am not writing, now this part I get 1, 31, 2, 39, 4, 30, 6, 7, 35, 8, 33, 13, 32, 15, 18, 16, 17, 19, 14, 30, 12, 31, 11, 34, 10, 30, 7, 9, 38, 6, 35, 32, 3.

Then, I do the two column merge for each column, what I will get 1, 8, 9, 14, 19, 32, 34, 37, 31, what about here, 2, 6, 12, 13, 30, 28, 4, 5, 11, 13, 13, 11, 15, this 15, 11, 15, 18, then you get 3, 7, 10, 16, 17, 34, 35, 22. Then, we send back the rejected elements to each counterpart, so I get 1, 2, 4, 3, 8, 6, 5, 7, 9, 12, 11, 10, 14, 13, 14, 13, 15, 16, 19, 30, 21, 19, 30, 18, 17, 33, 32, 31, 34, 37, 38, 36, 35, 31, 39, 30, 32, this is rejected element, I send back.

Now, first thing you observe here that all these elements are the smaller than these elements, all these elements smaller than these elements, second thing you observed. Now, after sending make each of them is bitonic sequence and all these elements such smaller than these elements, sorry all these elements smaller than these elements all these elements smaller than these elements and so on. So, you can you introduce the room merge here, room merge here, room merge here and so on and finally, you will get the sorted sequence.

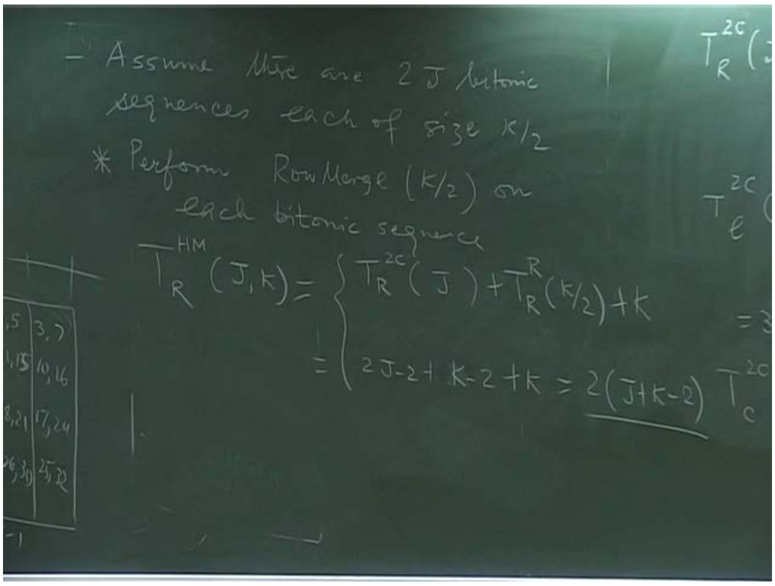
(Refer Slide Time: 38:31)


Horizontal Merge
M (J, k)
– Perform exchange operation between R_r & R_s by P_ij, j = 0, .., k/2 − 1, ∀ i
– get data of R_r from P_i(k/2+j) to P_ij, ∀ i, j = 0 .. k/2 − 1
– 2-Column Merge for each Column
– send data of R_r from P_ij to P_i(k/2+j) ∀ i, j = 0 .. k/2 − 1
– Perform exchange operation between R_r & R_s by P_ij ∀ i, j = 0 ... k/2 − 1
1,31 2,39
8,23 13,22
19,14 20,12
27,7 28,6
1,8 2,6 4,5 3,7
9,14 12,13 11,15 10,16
19,23 20,22 18,21 17,24
27,31 28,29 26,30 25,32

So, you have further to merge J comma k now tell me what I will write initial data is a pertain this side whether it is an R s register or R r register data issues let us assume data is in R r register in that case you assume a data is in R r register. Then, what you have to do what exchange first thing is that compare how many compare, how can you shift it because they are all in a R r register that you have decided. So, perform exchange operation in the first half, so first is perform exchange operation between R r and R s of P i or by p i i is what 1, 2, 1 or 0, 0 to k by 2 minus 1.

So, actually P i J i if you write p i J, but for all J for all J, so you have perform that then get the data of R r from p i k to p i J for a not i J i is what, i refers to what i refers to what row. So, I made mistake here this should be J this should be i, this should be k by 2 plus J and for all i J is 0 to k by 2 minus 1. Then, you perform two column merge for each column for each column right send data of R r from p i J to p i k by 2 plus J for all i J is 0 to k by 2 minus 1. Next perform change operation between R r and R s by P i J for all i J i k is 0 to k by 2 minus 1, so I get back the accepted elements in R r of each.

(Refer Slide Time: 42:48)



- Assume there are 2 J bitonic sequences each of size $K/2$

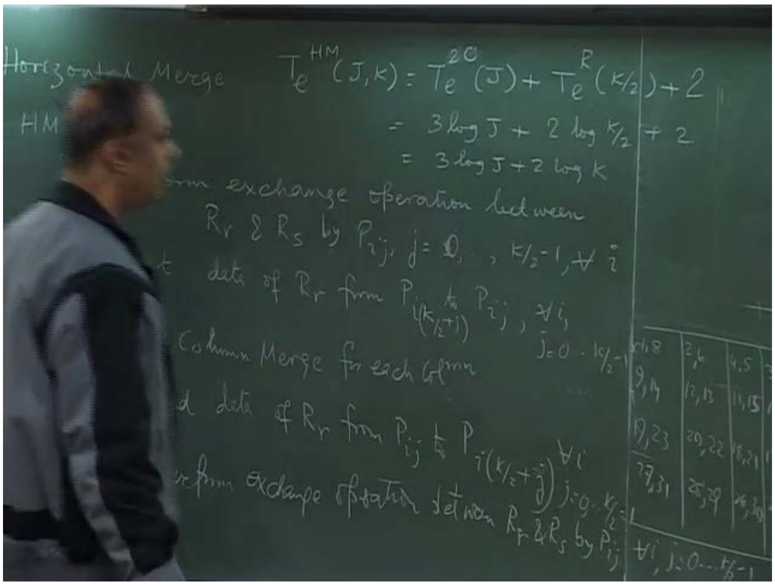
* Perform RowMerge $(K/2)$ on each bitonic sequence

$$T_R^{HM}(J,K) = \begin{cases} T_R^{2C}(J) + T_R^{R}(K/2) + K \\ = 2J-2+K-2+K = \underline{2(J+K-2)} \end{cases}$$

So, I have done up to this, now I have to do the row merge here, you write assume there are 2 J bitonic sequences sequences each of size k by 2 sum are perform row merge k by 2 on each bitonic sequence, so this is the steps involve P in further row merge. Now, a question is coming to obtain the time complexity R t horizontal merge J comma k tell me what is the rotine. So, you have you have R 2 column merge J plus t r row merge k by 2 plus this is whatever you needed that I have added whatever you needed here that I have added and what is the number. So, this one routine step and this is another routine step, so this is k.

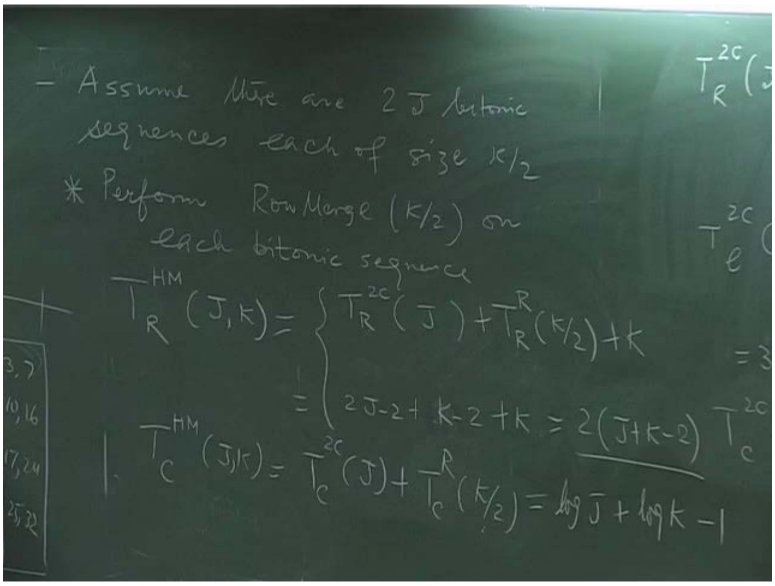
So, can you tell me what is the solution for this? What is the solution for this I have now you obtain which is 2 J minus 2. What about this one row merge k by 2 row merge k answer is what 2 k minus 2 k r it is k minus 2 k shifted by k by 2 minus this constant actually plus k. So, this gives you 2 times J plus k minus 2, now what about exchange horizontal merge J comma k, what about time you need exchange horizontal merge J.

(Refer Slide Time: 46:15)


Horizontal Merge
HM
$T_e^{HM}(J,k) = T_e^{2C}(J) + T_e^{R}(k/2) + 2$
$= 3\log J + 2\log k/2 + 2$
$= 3\log J + 2\log k$
exchange operation between
R_r & R_s by P_{ij}, $j = 0, \ldots, k/2 - 1, \forall i$
data of R_r from $P_{i(k/2+j)}$ to P_{ij}, $\forall i$, $j = 0 \ldots k/2 - 1$
Column Merge for each Column
data of R_r from P_{ij} to $P_{i(k/2+j)}$ $\forall i$ $j = 0 \ldots k/2 - 1$
form exchange operation between R_r & R_s by P_{ij} $\forall i$, $j = 0 \ldots k/2 - 1$
1,8
2,6
4,5
9,14
12,13
11,15
19,23
20,22
18,21
27,31
25,29
26,30

So, here one exchange operations here another exchange operations, so two exchange operations we are doing, so what is t e this result is one k log J, what about this one, what is the result? 2 log k by 2 plus J plus 2. So, it is 3 log J plus 2 log k, now there is no compare operations here except this here.

(Refer Slide Time: 47:45)



- Assume there are 2 J bitonic sequences each of size $K/2$

* Perform RowMerge ($K/2$) on each bitonic sequence

$$T_R^{HM}(J,K) = T_R^{2C}(J) + T_R^{R}(K/2) + K$$

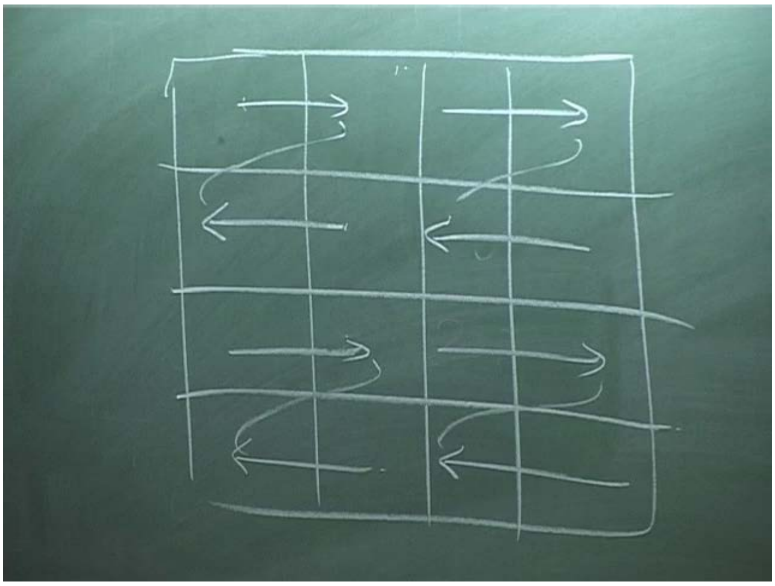
$$= 2J-2+K-2+K = \underline{2(J+K-2)}$$

$$T_C^{HM}(J,K) = T_C^{2C}(J) + T_C^{R}(K/2) = \log J + \log K - 1$$

3,7
10,16
17,24
25,32
$T_R^{2C}($
$T_e^{2C}($
$=3$
T_c^{2C}

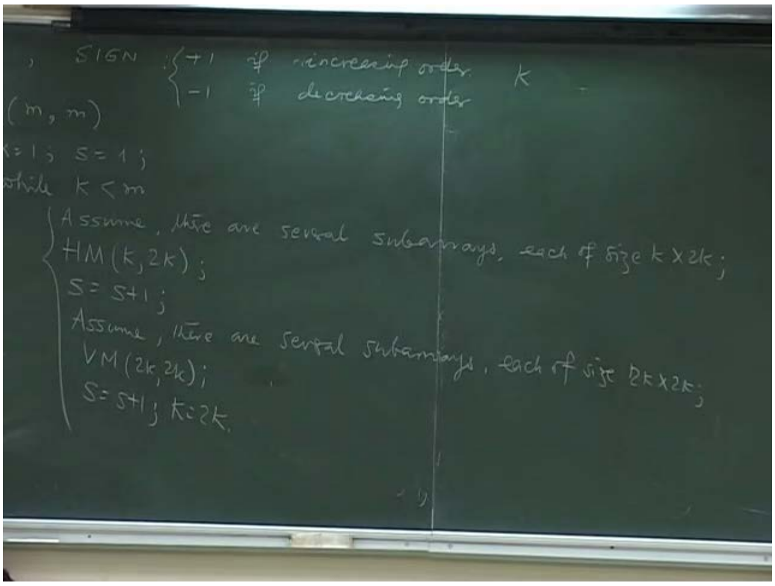
So, I can write t c h m J k is what about time, you need compare you need 2 C J plus T C r k by 2, this one belong to J k log J what about row merge log k by 2. So, I can write log k minus 1, now you know what is what is vertical merge and what is horizontal merge you are ready to write the sorting algorithm. So, when writing the sorting algorithm you have to keep it in mind is the bitonic it is a form of bitonic sequence, one sequence should be keeping order another sequence has to be in decreasing order first.

(Refer Slide Time: 48:47)



If you have the four plus 4, 4 plus 4, this is a bitonic sequence this is another bitonic sequence you will be a, you will be a arranging the in this box. So, that a this forms a bitonic sequence then you will be merging this one, so first it is horizontal merge you have to do then vertical merge horizontal merge vertical merge and so on. Every time, you have to keep track whether I have to keep it, there are increasing order or decreasing order that has to be.

(Refer Slide Time: 49:41)



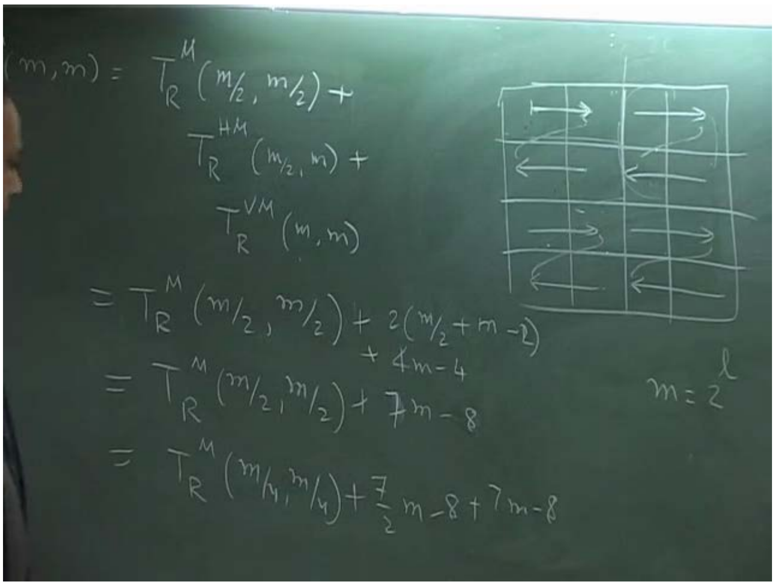
```
,   SIGN : { +1  if  increasing order.   k
           { -1  if  decreasing order
(m, m)
k = 1;  s = 1;
while k < m
    { Assume, there are several subarrays, each of size k x 2k;
      HM(k, 2k);
      s = s+1;
      Assume, there are several subarrays, each of size 2k x 2k;
      VM(2k, 2k);
      s = s+1; k = 2k.
```

So, I need to know what is the sage in number, I need to know a sign function to tell if class one if it is in increasing order increasing order, how I will compute the sign function, we will decide later one if it is decreasing order. So, I need to know the stage first stage second stage and so on and a sign function and second thing I need to know a variable k to tell say your horizontal merge k plus k plus 2 k vertical merge 2 k plus 2 k. What it means that first time you will be arranging this way consisting of two elements, in the second stage you will get another one that instead of horizontal merge in vertical merge afterwards to size becomes 2 plus 2 k.

In second phase, you will get this side to knows, but four columns and then four rows four columns and so on. So, k will be that point initially we assumes that k equals to 1, s equals to 1 and you are interesting to solve n cross m elements by m cross m processors in two dimensional mesh. Every processor contains one element, so you have n plus m processors every processors contains one element you are interested to sort this m square elements k equals to one first stage s equals to 1 while k is less than m.

Assume that there are several sub arrays each of size k plus k 2 k perform horizontal merge k plus 2 k, then stage is increase by 1 assume there are several sub arrays each of size 2 k plus 2 k use vertical merge 2 k plus 2 k 2 increase s by 1. Again, next stage n k by 2 k, so this is you main algorithms, so you observe this and this t's is several sub arrays bitonic sequence at this stage is sub array of size 2 k plus 2 k is a bitonic sequence. Here, I have not told about the sign function whether increasing order or decreasing order, then I will discuss separately.

(Refer Slide Time: 54:24)



$$(m,m) = T_R^{M}(m/2, m/2) + T_R^{HM}(m/2, m) + T_R^{VM}(m,m)$$

$$= T_R^{M}(m/2, m/2) + 2(m/2 + m - 2) + 4m - 4$$

$$= T_R^{M}(m/2, m/2) + 7m - 8$$

$$= T_R^{M}(m/4, m/4) + \frac{7}{2}m - 8 + 7m - 8$$

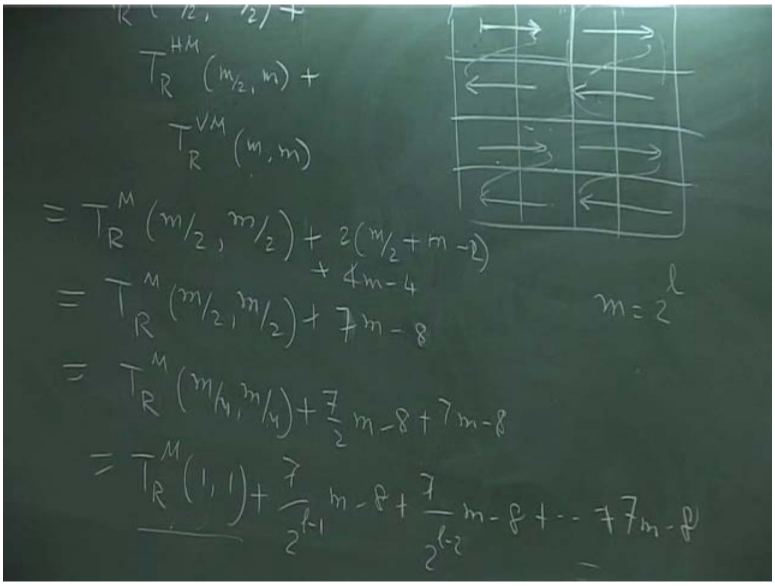
$$m = 2^{l}$$

If it is the case, then what the time complexity t main algorithm size m cross m number is of retains. Now, if I start whether 2 a sequence compute you can start 1 plus 2, then 2 plus 2, then 2 plus 4, then 4 plus 4 different somehow and so on. Instead of doing that, let us seeing this way, suppose I get suppose this is a sorted sequence, this get a sorted sequence and so on. So, that this is a bitonic sequence this is a, another bitonic sequence and then you have to use horizontal merge and vertical merge. You will be getting the total of sorted sequence, suppose I assume that this is sorted sequence, this is sorted sequence.

So, this is a bitonic sequence this is sorted sequence this is sorted sequence this is a bitonic sequence you call you call horizontal merge to get a sorted sequence for horizontal merge to get a sorted sequence point of view. You get whole bitonic sequence and vertical merges, so if this is the case, so can I write here the time complexity for this is nothing but T R Main algorithms m by 2 cross m by 2 plus t r horizontal merge m by 2 m plus t r vertical merge m comma m. Instead of looking for bottom line approach, I am thinking about the top down approach, I am by some recursive method, I have sorted this sum sorted this sorted this and sorted this.

Now, this is a bitonic sequence, so make it a sorted one which needs horizontal merge algorithm, so this is the time complexity simultaneously you do this. Then, you combined these two using the vertical merge which is the time needed for doing this is this. So, this is 8 this is 7 and assume m equals to 2 to the power l, what is the solution now T R M by 4 cos m by 4 plus 7 by 2 m minus 8 plus 7 m minus 8 m equals to 1 answer is 0, so up to m equals 2, we have to go that means l.

(Refer Slide Time: 59:34)



$$T_R^{HM}(m/2, m) +$$

$$T_R^{VM}(m, m)$$

$$= T_R^{M}(m/2, m/2) + 2(m/2 + m - 2) + 4m - 4$$

$$= T_R^{M}(m/2, m/2) + 7m - 8$$

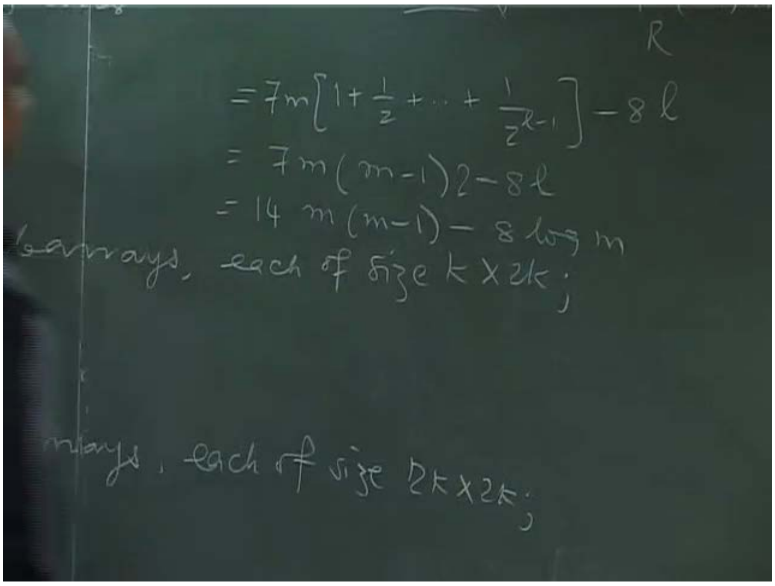
$$= T_R^{M}(m/4, m/4) + \frac{7}{2}m - 8 + 7m - 8$$

$$= T_R^{M}(1, 1) + \frac{7}{2^{l-1}}m - 8 + \frac{7}{2^{l-2}}m - 8 + \cdots + 7m - 8$$

$$m = 2^{l}$$

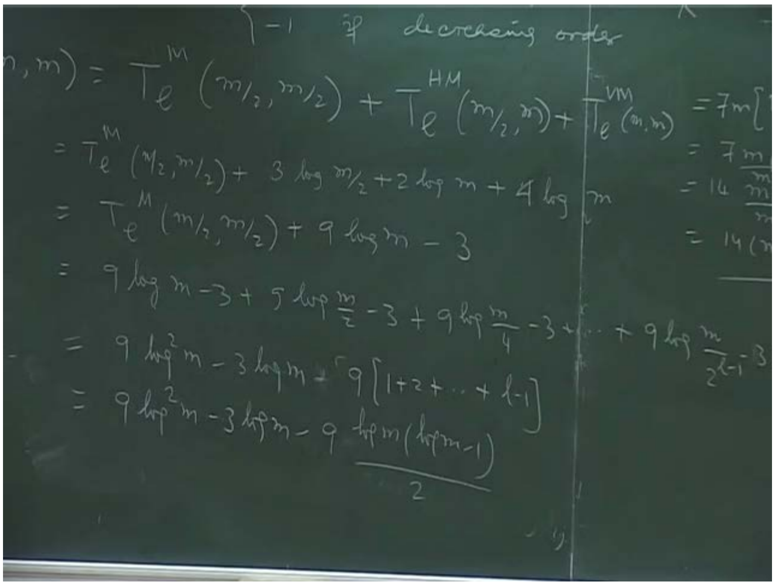
What is the answer this is 2 square T R M 1 comma 1, 7, 2 to the power I will write is it sure, this is 0, this is 0, tell me the solution now.

(Refer Slide Time: 01:00:03)


R
= 7m[1 + 1/2 + .. + 1/2^{l-1}] − 8 l
= 7 m (m−1) 2 − 8 l
= 14 m (m−1) − 8 log m
arrays, each of size k × 2k;
arrays, each of size 2k × 2k;

Here, 7 m 1 plus minus 8, l and this is 2 to the power l minus m 2 to the power l minus 1 is 7 m, m minus 1 2 to the power l minus 1 divided by half into 2 minus 8, l, so this is nothing but 14 m into m minus 1 minus 8 log m.

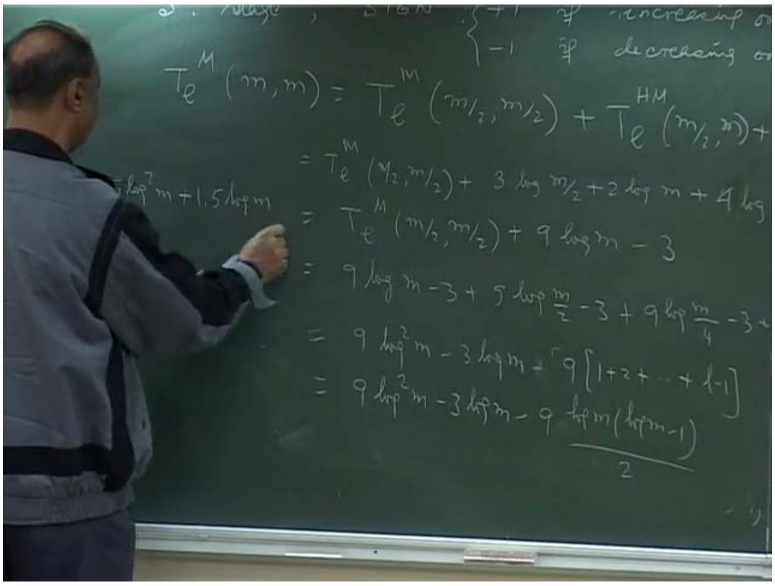
(Refer Slide Time: 01:01:04)


−1 if decreasing order
m, m) = T_l^M (m/2, m/2) + T_l^HM (m/2, m) + T_l^VM (m, m)
= T_l^M (m/2, m/2) + 3 log m/2 + 2 log m + 4 log m
= T_l^M (m/2, m/2) + 9 log m − 3
= 9 log m − 3 + 9 log m/2 − 3 + 9 log m/4 − 3 + .. + 9 log m/2^{l-1} − 3
= 9 log² m − 3 log m − 9 [1 + 2 + .. + l−1]
= 9 log² m − 3 log m − 9 log m (log m − 1)/2
= 7m[
= 7 m
= 14 m
= 14 (

Now, you have t exchange m main t exchange main m by 2 m by 2 plus t exchange horizontal merge m by 2 plus t exchange vertical merge m comma m. Now, you obtain 3, 3, 3 log m by 2 plus 2 log m what about this one, 2 log J k 2 log 2 log l square that is 4 log m this is this gives you what three log m three log m plus two 5 plus 9 log m t e m m by 2 m by 2 9 log m minus 3. So, what is the solution for this 9 log m, tell me minus 3 plus 9 log m by 2 minus 3 plus 9 log m 4 minus 3, 9 log m.

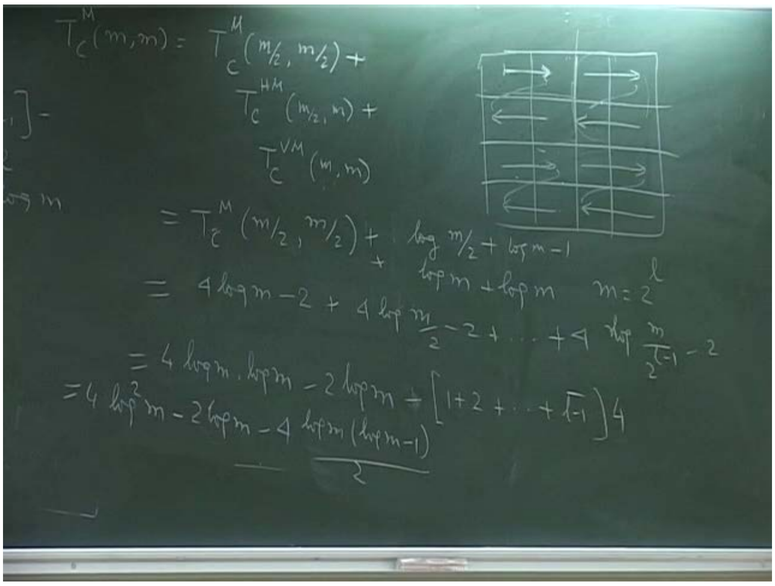
So, it is 14 m minus 1 minus 8 log m, what about this one, what is time still it is away, tell me this is this expansion. Then, what is the answer many log m 9 log m. So, log squared m minus 3 times log m minus first one is minus 9, second one is what 18, so I should take 9 out 1 plus 2 plus l minus 1, so I get 9 log square m minus 3 log m minus 9 log m log m minus 1 divided by 2.

(Refer Slide Time: 01:03:06)


-1 iff decreasing or
$T_e^M(m,m) = T_e^M(m/2, m/2) + T_e^{HM}(m/2, m) +$
$= T_e^M(m/2, m/2) + 3 \log m/2 + 2 \log m + 4 \log$
$\log^2 m + 1.5 \log m$
$= T_e^M(m/2, m/2) + 9 \log m - 3$
$= 9 \log m - 3 + 9 \log \frac{m}{2} - 3 + 9 \log \frac{m}{4} - 3 +$
$= 9 \log^2 m - 3 \log m + 9 [1 + 2 + \dots + l - 1]$
$= 9 \log^2 m - 3 \log m - 9 \frac{\log m (\log m - 1)}{2}$

So, you get four point 5 log square m plus 7.5 minus is there no 4.5 plus 1.5 log m.

(Refer Slide Time: 01:03:20)



$$T_C^M(m,m) = T_C^M(m/2, m/2) + T_C^{HM}(m/2, m) + T_C^{VM}(m,m)$$

$$= T_C^M(m/2, m/2) + \log m/2 + \log m - 1 + \log m + \log m \qquad m = 2^l$$

$$= 4\log m - 2 + 4\log \frac{m}{2} - 2 + \dots + 4\log \frac{m}{2^{l-1}} - 2$$

$$= 4\log m . \log m - 2\log m + [1 + 2 + \dots + l-1]4$$

$$= 4\log^2 m - 2\log m - 4\frac{\log m(\log m - 1)}{2}$$

Now, tell what is the number of comparisons one is first one is log J right this perform is log J, log m plus first one, Log m by 2 minus log m, Log m by 2 minus log m. What is the formula, Log J minus log k, Log J minus Log k minus log k minus log J minus log k minus this is minus J plus log k minus 1 and what about vertical merge log J log J plus log k. So, this keeps you 2, 3, 3.5, 3, 4, 1 , 2 , 3, 4, 4 log m minus 2 minus 2 plus 4 times log m by 2 minus 2, 4 times log m by 2 to the power n minus 1 minus 2. How many log m s 4 into log m into log m because there are log m 2 log m minus 2 times log m plus or minus 1 plus 2 plus l minus 1 into 4.

So, 4 times log square m minus 2 log m minus 4 times log m log m minus 1 divided by 2, so 2 log square m only right such simple one good. So, this is about the number of comparisons, you need now the sign thing only because you have can you give 2, 3 minutes. So, I am yet to cover in the next class, I will cover the sign function.