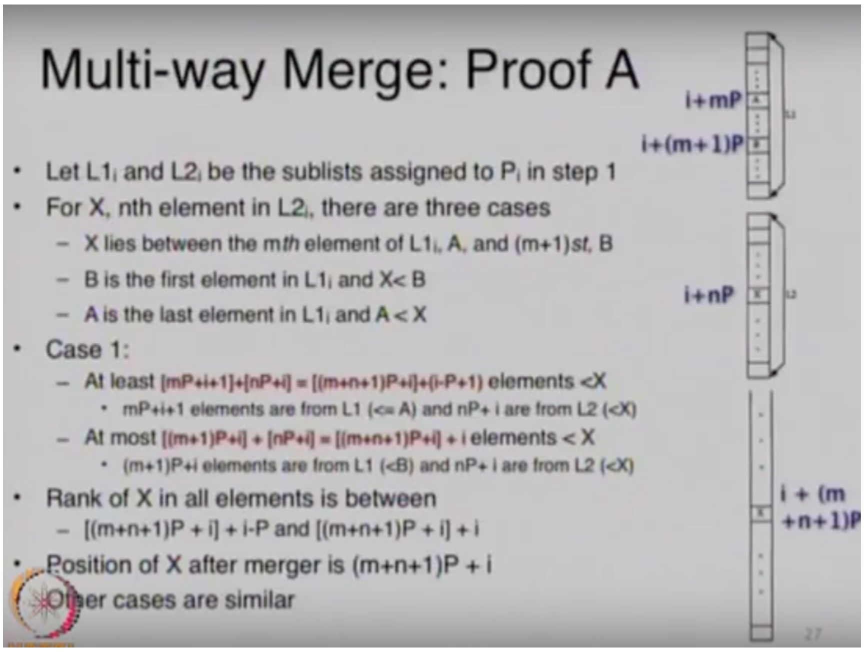
Parallel Computing
Prof. Subodh Kumar
Department of Computer Science & Engineering
Indian Institute of Technology – Delhi

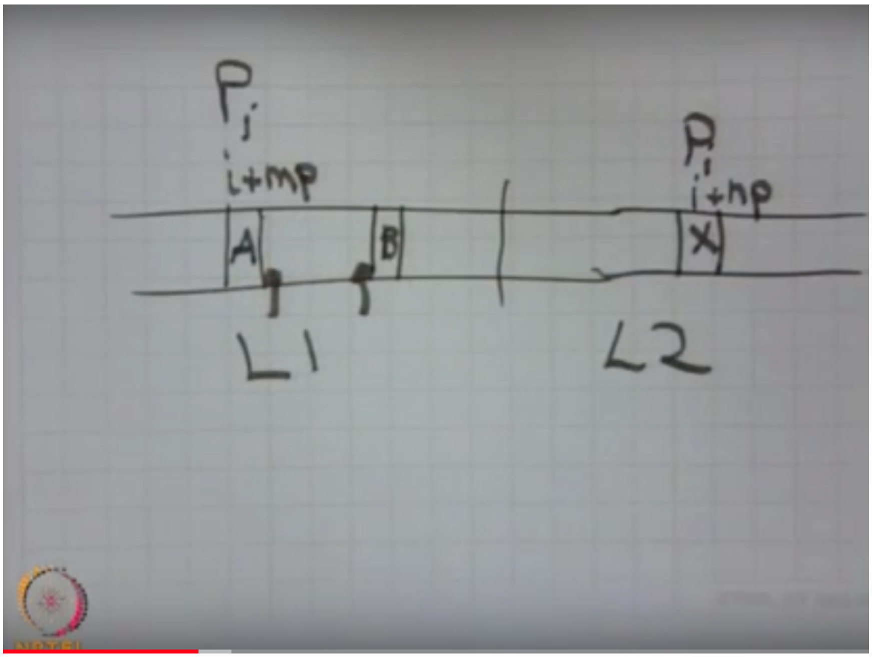
Module No # 07
Lecture No # 31
Algorithms, Merging & Sorting Continued

(Refer Slide Time 00:28)



Multiway merge and we will prove a few things along the way. So what the diagram you see on the top right corner has two halves are there as if you will. L1 at the top and L2 at the bottom. And I will recreate the same diagram.

(Refer Slide Time 00:44)



This is L1. This is L2 ok. And these are two elements assigned to processor Pi. This is A. This is B. And this is also assigned to processor Pi from the second list. It is X ok. And X is what we are considering as a candidate like every other candidate without loss of generality. We pick one element of Pi from the second list and figure out where it would have gone on the first list ok.

And so its rank is between A and B ok. It also says that this is position I + M times P. And this is nth thing in its, I am running out of space here now so i plus n times p alright. So X is the nth element. Or the element we have chosen in the second list is the nth element. And it happens to be after the nth element in the first list. Nth element being A. So can we say a few things?

If we consider this element X, how many elements knowing that it goes between A and B right. Its rank is between A and B. Or its rank is the number of elements Pi has. “Professor - student conversation starts” What is its rank exactly, knowing that it goes between A and B? So rank of X in L1; N. A and B adjacent? A and B are adjacent in its striated pattern. So it goes between A and B. That is the main thing. Sorry? Somewhere between A and B so its rank is + I + however how high is the one beneath.

Exactly so in within P1, its rank is m right. Well it is, is it? M + 1 actually because PM + 1 Sorry? PM + 1. Yeah P is its actual position but we do not know its comparison with everything else. The rank we are talking about or? No. Only that P? We cannot right because we do not know its comparative value with respect with everything. Professor – student conversation ends.

We only have compared it to Pi has compared it to A and B and other things which were P apart. And we have found that it comes after it ok, and A is the nth element within Pi. That is why it is I + M times P. Pi starts at I and Nth element of i will be I + M times p right. So rank of this element X within the P is section is M + 1 right. Because there are exactly m plus one elements less than it.

For the time being let us assume that everyone is unique right. So X and A and B are different. "Professor – student conversation starts" You want to take an example? Suppose A is the first element; M is 0, i. First element of Pi, whatever i may be. "Professor – student conversation ends" Its M is 0 right. So that means there is one element less than X. Its m is 0 means there are 0 elements to its left right.

If there are 0 elements to the left of A then there are 0 + 1 elements to the left of X because X has everything to the left of A and A. Fine ok? Ok so the statement that we are going to prove is that what we are trying to do immediately as a sub step of that proof is to ask what is the minimum number of elements that have to be before X in the entire L1. Not just for P1s. Professor – student conversation starts Pi + M. Pis; I + M? I + MP. All the elements before A right? But there are more elements going to come. "Professor – student conversation ends"

So first we are not going to do the entire proof. We are just going to do the main parts of the proof. And then and there are three cases to consider. One is that there is an A and a B right. X goes between a given A and a given B. The other boundary cases are is X is the first element. Which means X is, there is no A. Or X is the last element. With that is there is no B. Which are again special cases of the same argument.

So we will only consider the case one which is X lies between a value A and a value B. And so we want to know how many elements are going to be less than X. Or its turned out to be too small. This thing you never can tell because that automatically keeps shrinking that text until it fits. And so you do not get to read the actual value actual font size of the text.

Professor – student conversation starts we can see. It is readable right, good. “Professor – student conversation ends” So why does it say that it has MP + I + 1 + MP + I elements less than X? We know that every element that is less than A is definitely less than X. “Professor – student conversation starts” And so how many elements are there? Sorry? What is the position of A? Position of A which is; i plus one. One raised to the power A. One raised to the power A.

Oh one sorry. I got confused. Yes + 1 is for A. “Professor – student conversation ends” To know how many elements are to the left of A to that you add one. Those are definitely less than X. Are there more that are definitely less than X? Some lies in L2 right. You know where X belongs in L2 in nth position.

So we know NP + I elements are less than it. So just little bit of massaging and it turns to the form M + N + 1 times P + I + 1 - P + 1 ok. It is probably easier to read than for me to read that out. Now we are going to ask how many numbers is at most how many numbers are going to be less than X. “Professor – student conversation starts” X maybe just before B right?

All the elements but B. All those extra elements. “Professor – student conversation ends” And so you add that. You say M + 1 because M + 1 was the rank of B in its Pi times P + I + NP + I. And we will and the idea of putting the top thing in a given form was to make its form similar to the bottom thing.

And what do you notice? One has I + P as its second term. The other has just i as its second term ok. So there are at least that many - P elements less than it and at most that many elements less than it. Where is it actually going to be? If we are going to sort it we know that is when we merge it in its original position merge for the positions of Pi then N elements are going to be less than it.

M elements are going to be less than it right. N elements of the second side. M elements of the first side. So N +M + 1 is its position in Pi. So N + M + 1 times P + i will be its position when Pi merges it. That is where its going to place it. And so it is going to place it at a positon that is at most P away from its final position ok. Its not only at most P right. Its actually i or I - 1 or I - 2 or up to I - P, but never I - 1.

So wherever you place X it its final resting place will be that or any element up to P element before it. “Professor – student conversation starts” But then why did I say. If you see the other one. If you see the other way right. “Professor – student conversation ends” If you take an X on the other side. Then you are going to come up to the same conclusion. And so an element in general will either be within P of its left side or within P of its right side.

And so that is one process solved. This second diagram does not, it should have actually come earlier. Does not add too much to the discussion. It is basically saying that is where it would have gone in. It would have been placed after merging ok. Now I want to say that each block is sorted after merging right. I just said that each element is within P of each other. Professor – student conversation starts Ok so maybe will show you.

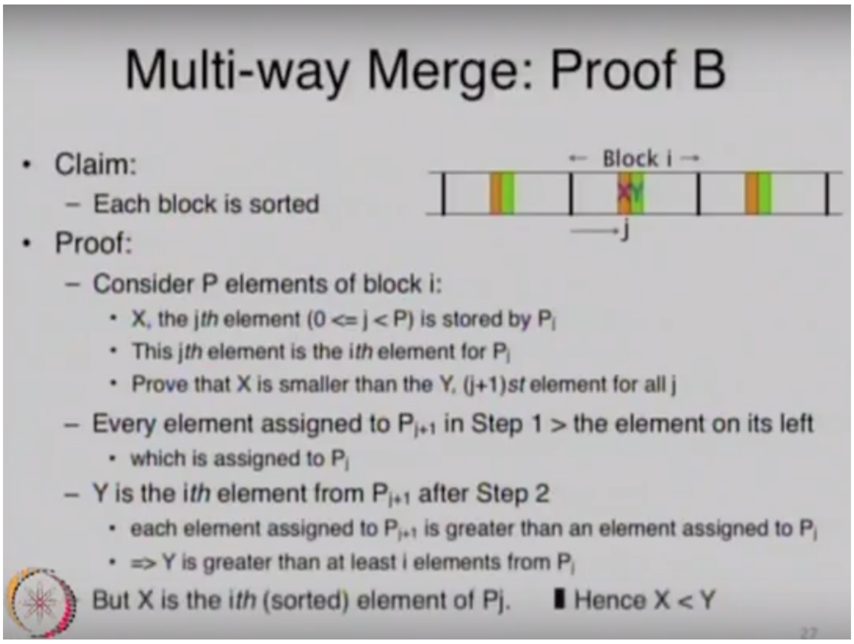
But how we are concluding after we have found that at least these many elements are less than this and at most P. So how do we know its final position. “Professor – student conversation ends” By knowing its rank right? So we need to how many elements are less than X right. If you tell me that thirty-nine elements are less than X in the total list. Then I know its going to position number thirty-nine. That is its final resting place.

I just do not know the rank right. But if tell you that the rank is between these two numbers. Then you know its final resting place. Somewhere in between those two right. Now when I am trying to find its rank what am I trying to do? How many elements are less than it right. So that is that is all I am trying to do. Trying to figure out how many elements are possibly less than X. And so let me ask first.

How many elements? What is the smallest number of elements that is less than X right? “Professor – student conversation starts” There are some elements we know definitely less than X. Those elements less than A. Less than A. Less than X. “Professor – student conversation ends” Less than A in L1; less than X in L2. Maybe no other element is greater than X but that many elements are definitely less than it. That is the first number. Second number is we know.

In L2 we definitely know X is in its sorted position right. So in L2 only elements to the left of X are less than it. But in L1 everything to the left of B is possibly less than X right. So X comes somewhere here. X would be at this position or at that position. That is it. So there are P elements from here to there. Is that clear to everybody?

(Refer Slide Time 14:59)

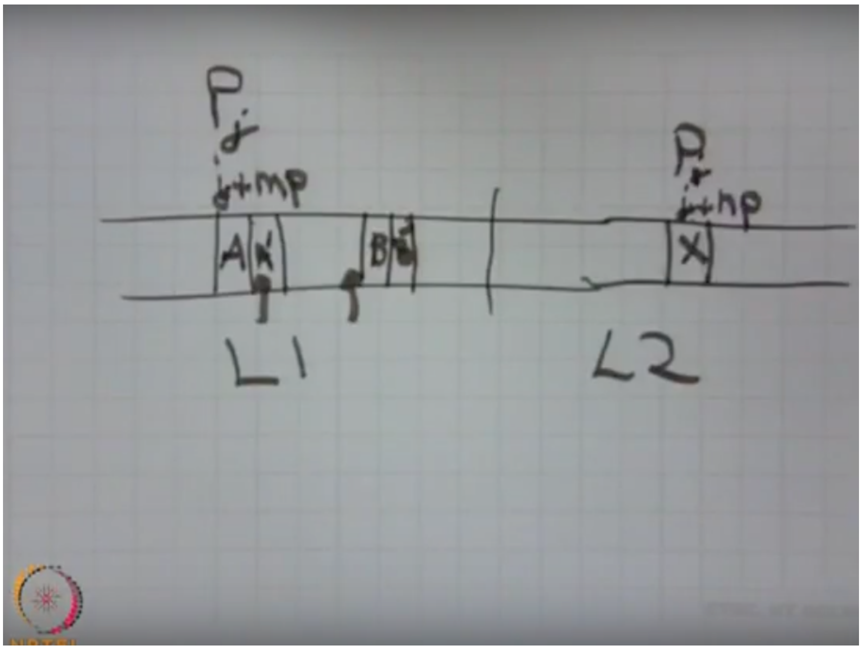


After we get this merge, claim is that each block is sorted. And block is again every P. So we are looking at some arbitrary block I ok. And we are looking at two elements, consecutive elements in this block. And for this block to be sorted, X must be less than one. And let us just say that the number X we have picked is Jth from the left in its block.

Its at jth position within its block. So ith block, jth position within the block is the number we have picked. And Y is just to the right of it. And we are going to have to prove for us to say that block I is sorted. That is less than Y ok. Now there are few things that you can immediately see. “Professor – student conversation starts” If X is the jth position then who put X there? Jth block. Jth block right, Pj. “Professor – student conversation ends” If Y is at the next position then it is put there as PJ + 1 right.

Can you now claim that every element that Pj plus one had has enough numbers to its left no. The numbers of PJ for example X has enough numbers to its left. That is the thing I want to prove. And so this goes without saying that every element that was given to PJ + 1 had a number less than it, PJ. PJ had a number that was less than PJ + 1 right. So PJ + 1 had picked in here.

(Refer Slide Time 16:59)



So just to go back to this picture. Is if this A was Pi let us turn it to j. Alright so A was assigned to PJ and A prime was assigned to PJ + 1. B prime was also assigned to PJ + 1 ok. X prime was also assigned to PJ + 1. That is the way cyclic assignment works. So if you look at the elements of PJ + 1, every element of PJ + 1 has something less than it. That is all it is. This is no magic. So when you pick an element of PJ, you are guaranteed to have some element less than that one ok.

And that less than number is definitely going to Pj right. So now the question is does Pj have enough less numbers um for this to be true ok.

(Refer Slide Time 18:14)

Now because it is block i or yes it is block i, which number? Is it basically a transposition, which number of PJ + 1 was Y? Ith number right. Block zero gave you the gave everybody their zeroth number. Block one gave everybody their first number. Block i gave everybody their ith number.

So X is ith number of block of PJ. And Y is the ith number of block PJ + 1 fine. Nothing fancy, ok. And every element we just said this every element that was assigned to PJ + 1 is greater than an element assigned to PJ. And Y was greater than the ith element assigned to PJ. And how many other elements were assigned to Pj that was less than Y? Everything before X right?

Everything that before X that Pj had. How many things did Pj have before X? Pj had how many things before X? This is ith block; i things, i things right. Because Pj where were we. So let us say this is since we are using X and Y let us just stay there. No? Yeah X and Y let us stay there. This X it comes from Pj. This Y comes from PJ + 1. And how many elements are less than this Y in X? X is definitely less than Y.

But everything less than X is also less than Y. "Professor – student conversation starts" How many elements less than X? I elements. Right? That this X was in L2 now so does not matter. But hold on X is less than Y? No X was less than good question. "Professor – student conversation ends" Good question. We are not proving that X itself is less than Y. I mean that is what we are trying to prove.

But we are not saying we are not assuming that X itself is less than Y. We are saying that there are, I was overloading X. And let me explain. Y is the ith element of J + 1 right fine. Now how many elements of j are less than 1? The ith element of j and every other element of j before ith of j ok. It just so happens that X is over there now. But whatever that element used to be ok.

So i elements of Pi are less than Y fine. Ok that is all we are trying to say; i element of Pi are less than Y. But what do you think X is? X is now after sorting the ith element that Pj was creating right. Since there are at least i elements, or no. There are exactly i elements less than Y and X is that ith element. X has to be less than Y ok. Simple enough argument. Alright again I will repeat it here.

We are trying to prove that X is less than Y. We have just instead proved that there are sufficient number of elements less than Y in this block right. How how many is sufficient? I; f there are i elements that Pj had that was less than Y and X happens to be the ith element after sorting, then X was less than Y. And to make sure that there are enough elements that Pj got that was less than Y, we simply used the fact that every element like Y that Pj got had a corresponding X to its left ok.

Again I am overloading Y but corresponding assigned assignment to its left ok. Think about it. We are out of time so we cannot complete the proof left. We will do that on Thursday. "Professor – student conversation starts" (())(24:05) L1 and L2 (())(24:06) This is for any block; any block coming in L1 or L2. "Professor – student conversation starts" Because the same are the same argument holds right? That just picking ever ith of L1 or L2. So this has nothing to do with N or M. So a block is sorted and an element is within P of its position.

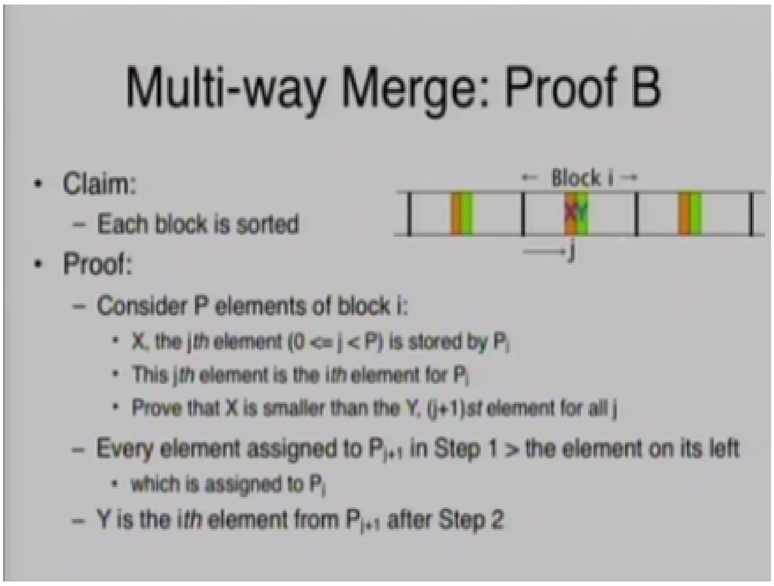
Which means an element can be going to the next block or the previous block. Now where else ok. So we will there is a little bit of more stuff to tie up our loose ends before we finish. Are there any questions?

We were trying to merge two sorted lists. In this example it is L1 and L2. And the algorithm was to take every odd element from L1, no this is three-way merge so every Pth element from L1. Every Pth element from L2. And one processor is going to merge them. You have another set of Pths from L1 and another set of Pths from L2 and so on. So P processors each are going to be merging it in parallel. And at the end and they are going to do this in place right.

So 0 and P - 1 and 2 P - 1 will now be sorted across from between L1 and as well as L2. And then we do two steps of sorting between groups of P right. So first block of P and second block of P gets merged. And then third fourth in parallel is getting merged and so on. And then we do one more step of leaving the zeroth alone.

Merge first and second, third and fourth, fifth and sixth and all of those (())(26:08) ok. And one of the things we already saw was that elements will be within P after the first step. When you have taken every Pth element has been taken by processors and merged.

(Refer Slide Time 26:39)



We also saw that every block is going to be sorted right. So after the first step is done, when you go back and look at the chunks of P each. They are all going to be sorted. So we are going to skip thru those proofs. “Professor – student conversation starts” Does that complete the proof? This is where we were at is it? Yes we do. Yes we have; no sorry. We do have to continue it from here. But the question is are we done ok. “Professor – student conversation ends”

We mean the two things we have proved is that within each block. One is that elements are within P of each other. And the second is that within each block they are going to be sorted. “Professor – student conversation starts” So now the question is after step two are we done? Sir, initially also the lists were sorted.

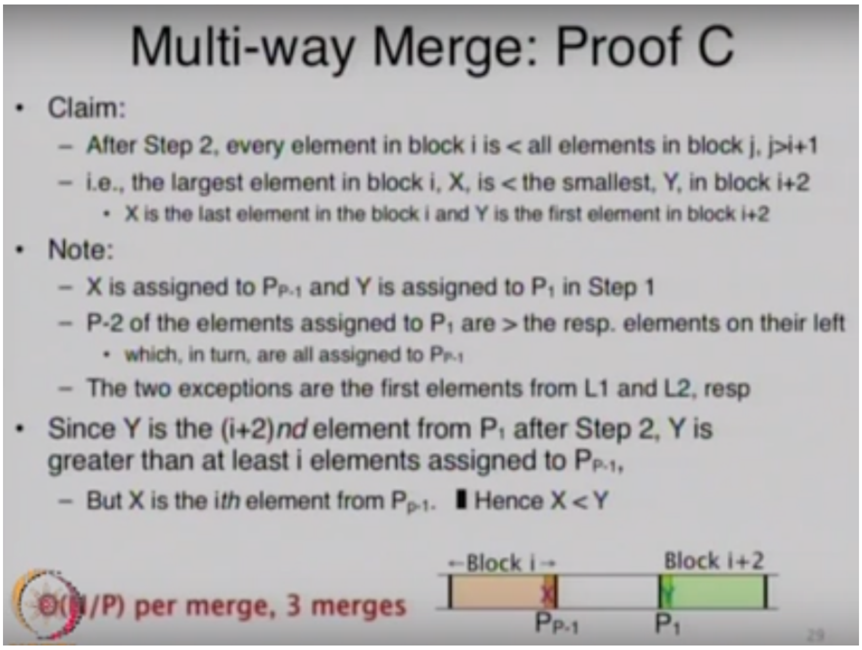
Yes, yes. So now this placing is in continuation with the sorted list, final sorted list. But the left list and the right list was not sorted with respect with each other. That is why you have. The left list was sorted. The right list was sorted. And then you have merged the even ones. Merged the odd ones. Yes sir. Or no I keep saying even and odd because it comes from batch Ls. But we have merged every Pfs right.

So the blocks that we have generated after the merger, every element in the block came from a different merger right. And so there is no guarantee that it is or there is of course but we have to realize it. We have to prove it. Sir ok fine I can understand but the second elements were always more than the first elements. The first had all the first chunk of P elements, they were always smaller than the second chunk of P elements.

They may not be. No. Those elements they come from the other is they may not be the holding same properties. So we need one scan thru of adjacent pairs. Not just adjacent. You mean adjacent blocks. But adjacent in both ways. Adjacent in both ways.

Adjacent in left and right both ways. Right so that is what the second step was doing. “Professor – student conversation starts” Second step said so we have established that each of these blocks of P. It is better to talk in the context of a picture and let us go thru that.

(Refer Slide Time 29:25)



We had generated that proof B slide said that if we look at these blocks of P we know that each is sorted. Each block of P is sorted. We do not know that all of this is less than all of that right. Or for that matter less than all of the remaining ones. Professor – student conversation starts.

But what does that mean in the claim of number one? Claim number one in this slide? Yeah. So its after the two steps are done every, yeah the two steps meaning the first was the division and the first merging step. It says that the block i cannot, any element in block i cannot be greater than elements in block i plus two. So that is the same claim as each elements actual target position is not P away more farther than.

Right basically that that is how you prove it. It is the same claim. It is effectively the same claim. Yes. "Professor – student conversation ends" So we will just quickly rush through this. We know that these two are mergers. These two are going to be merged just like any other merge that we have talked about right. Because this is sorted.

That is sorted. Individually each block is sorted. I just do not know if everybody here is less than everybody there. Simply merge these two like any other two sorted lists being merged. And then we will be done right. Except that the other side may also have less than P or be in a position which is greater than some P position, up to P positions after where it was supposed to be right.

So we have to look at the left side also. So how would you prove it? And this this is now kind of trivial after you realize that it is within P of each other. But let us go thru the slide anyway. You essentially want to prove that the last element in a given block is less than the first element two blocks from now. This step is a bit more involved than just knowing that you are.

If every element is within P of each other, that does not mean that every element. So here the statement of this (())(32:25) so if you wish. If you look at elements in block i then they are guaranteed to be less than elements in block I + 1, I + 2. "Professor – student conversation starts" That is not implied directly. Yes sir because this element i can come in i plus one block and so can X. Yes we can be sure that it will not lie in that index range.

So we only know that X maybe within this place to that place. We do not know about other elements right; some other element that is in. May come to this thing. Block B or should have been in block I + 2 is actually here right. "Professor – student conversation ends" Or I should say should have been in block I + 3 has reached block I + 2. So you cannot claim that just because you are within P of it your final position that anything that is greater than P from away from you is guaranteed to be greater than.

Those two are different statements. "Professor – student conversation starts" We do have to prove so in this example. Sir why do we need to prove that? You have to just say what is the rank of X. Rank of X. So we can be sure we are very sure that rank of X will not lie in block i plus two. We know that. But we do not know about the current distribution of elements right. If everybody else were at their rightful place, then we could definitely say that X cannot be greater than any element in P +. In fact now then we could say P + 1.

Sir my question is that why should I worry about the values in the blocks. I have identified where X is currently. And I have identified how can be how much it can be varied, P left and P right. So I can talk in terms of index that it will not be lying on this index. It will be surely less than on this index. Definitely. So if I am able to figure that then why should I worry about whether.

No the question is for an element in block i do we have to compare it against elements in block I + 2 right. We know that X is going to lie somewhere here. But if we have to pick one element from here X has to shift one element one position to the right. It will still stay in the same block right. It may have to shift one position to the right. Cannot we say that block i lays in block I + 3. That was three is definitely true; i plus two is also true.

Yeah but if we just use i plus three then is it not only. Because you only need you are only merging one to the left, one to the right. Yeah i mean we can change that step and guarantee that it is just less than (())(35:32) are less than are less than. No that with the given algorithm will not. Yeah or you maybe not do 2 but 6 more merges but. Yes, yes. I understand but why do it when you can actually without anything else you can. You already know its going to not be needed. Why would we do it.

"Professor – student conversation ends" Alright so again we are working in block i. Or we have looked at an element X in block i and we are trying to prove that X, any element in block i is less than any element in block I + 2 alright. Now we say if we prove it for the last element being less than the first element. Last element of block i if we show that it is less than the first element of block I + 2, then we can say this for everybody else.

Because we know block i is internally sorted. Block I + 2 is internally sorted. So the smallest of block I + 2 is in the first position. The largest of i is in its last position. So if you can say something about these two we said the same thing about everybody else ok. So now let us look what where these things came from. X was assigned to all those elements that were assigned one in a striated pattern right.

Processor 1, processor 0, processor 1, processor 2, processor last, right, processor P + 1. And block I + 2 was, the first element of block I + 2 was given to processor P0 right. Just like every other first element of entire block or zeroth I should say. I hope I sometimes switch between zeroth and first. I think though the slides are consistent but I may not always be. Alright so now if you look at the elements that were assigned. So if you look at elements are assigned to P0 right. They would be all of these first elements.

How about the elements that are assigned to PP minus one the last one. All the last elements. Every one of those last elements is greater than enough number of other elements right. Because they are last in their block and their block was sorted in when we started. L1 was sorted. L2 was sorted. So if you look at those elements that were picked out by the last processor PP - 1, every one of them had something to its left. It was given to P0.

All the all the elements to the left were given to P0 except the bottom of the list right? The first element that P0 got was not to the left of a last element of another block right. The L1 when the P0 got the first element of L1 was assigned to P0 was not after a last element in PP-1. Similarly in L2 also there was one such right in the beginning. It was assigned to P zero but there was no PP - 1 to its left ok.

So there are two such instances. One from L1. One from L2. So there are then remaining P - 2 things that P0 had which were less than what PP - 1 had ok. No? Why do not you try to say what I said. Yeah maybe drawn figure is easier. If you go back and look at the assignment right which was just alternating assignment, P0, P2, P3, P4, P let us in fact just so I can say it easier. Let us say there are 15, 16 processors. So P0 to P 15 ok.

And there are every element has then 16 block, 16 element. Every block has 16 elements. And P0 gets element 0 and P 15 gets element fifteen. P0 gets element 15 which is less than element 15. Because it was sorted. L1 was sorted L2 was sorted. So whenever you gave element 16 to P0 you definitely gave element 15 to P 15. For every such P0 assignment bar two there is a P 15 assignment to its left ok.

So that is all this is saying. That because of the 16 assignments that you made to P 15 two of them did not have this pattern. That P 0 was followed by P 15 right, sorry. P 15 was followed by P0. Which two are those? The very first element. The very first did not have anything to its left so there is nothing that was assigned. So there are now fourteen such pair where what you get in P0, whatever you get in PP - 1, P 15 has a greater element in P0.

Because every time you assign something to P15, you assign something lesser to P0. I said the opposite thing. Every time you assign something to P0, you assign something smaller or maybe I said the same thing just negated it twice. Every time you assign something to P0, there is a P 15 to its left and there are 14 of them. There is one pair missing. One of such P zeros does not have a P 15 to its left in L1.

Similarly there is one also in L2 ok. So the other fourteen of the sixteen P zero assignments have a P15 to their left ok. So every P0 has sufficient number of P15 to the left. And what is the Ys position in its processor right? So again coming back to the 16 processor example. So which element is Y for that processor? Y is given to processor P0; I + second right? Its block number I + 2. Block 0 contributes the zeroth element to everybody. Block one contributes the first element to everybody. You can see it right?

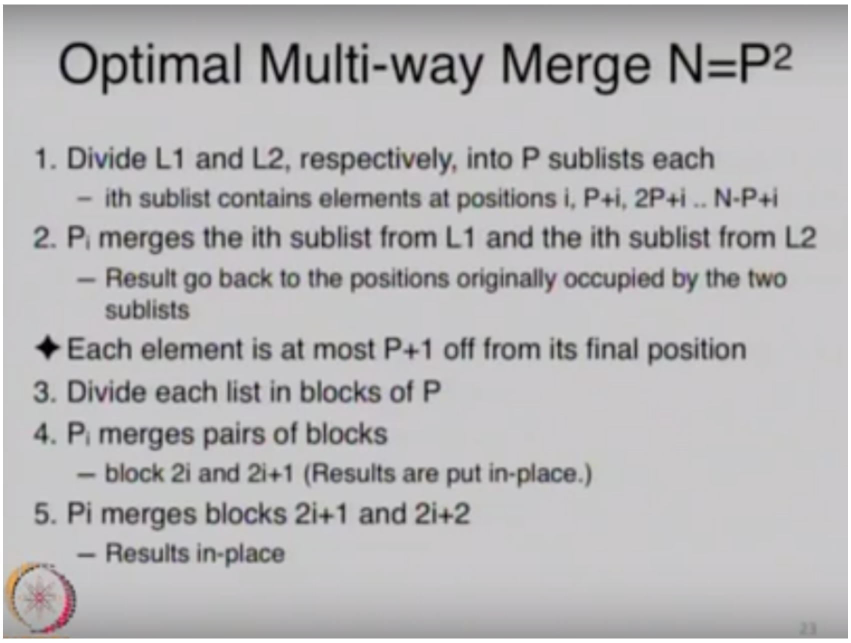
Two contributes the second element to everybody right. So block i contributes the ith element to every processor. And so now this element Y we are talking about is the I + second element that was allocated to P0 ok. "Professor – student conversation starts" And so these I + 0 elements that are less than Y on processor number 0 most of them right; bar two will have an element less than it on the last processor. Y would have at least i numbers in P fifteen less than it.

Yes (())(43:59). That is correct right. Which is precisely what this slide says. "Professor – student conversation ends" Which number is the position Y in its block in its processor? I + second right? And so within the processor, things that are assigned to a given processor are in sorted order. So if I + second element is Y then i plus one elements are. No its rank is I + 2, so I + 2 elements are definitely less than Y within its processor ok.

And everyone has an element less than it right, which was exactly assigned to P0. All those less than elements that we are talking about were assigned to P0 right. And so the ith element of P0 sorry. I yeah sorry right, ith element of P fifteen is less than Y, but ith element of P15 is X right. Because it is in block i and it must be the ith element of P 15 ok. Ok I do not want to (())(45:55) this too much.

The logic is not involved at all but I think if I had drawn a picture that was more detailed than what I can do on the slides. It would have helped a little. But what i can think about it if you are still not sure. So this ith element is the element X. So X is less than Y. Let us look at the three steps ok.

(Refer Slide Time 46:23)



Ok so does division take time? That is implicit right. Basically every processor with its processor id will know where to begin. And it knows how many processors there are so it knows what its stride size is. Certain stride is all needed, is all that is needed for it to logically construct its list ok. Now it has to merge two sorted lists right. It does the things it got from L1 were sorted.

Things that it got from L2 were sorted. But within them it has to do something. So for the time being let us say it uses some third space to do some merging but brings it back to this purpose location. How long? N by P right. Each processor gets N by P from here N by P from there and it sequentially merges it. At the same time another processor is got its own N by P. Everybody has got its own N by P. You sequentially merge it after N by P time, you are done with that step.

Then what remains? "Professor – student conversation starts" Now again these blocks are size? N by P, N by P. "Professor – student conversation ends" And you merge 0, 1. Where as somebody else is merging two three. Somebody else is merging 4, 5; P of them P pairs, precisely P pairs. N by P again same thing for the last one. It maybe one fewer but who is counting.

So total time being spent is N by P, which is optimum work wise right. Now this is I also wanted to go thru this specific algorithm because it is an example of something that does not strictly use the work time scheduling that we have seen before. It actually works very well.

The merging works well. It is optimum right. If you think just in bare terms. If you have N things to do.

Order and work was needed sequentially; P processors to do them with. N by P is as well as you can do asymptotically right. Asymptotically N by P is the best you can do. So this is optimum. “Professor – student conversation starts” Is it not it? So sir if you look at so I have 16 processors. And it is taking me hundred things to do in one processor. Then we say 100 / 16 you take (())(50:02) so that I cannot do greater than 16 times 3 right?

Actually, it will better than 16 times 3. Moreover, if you have thousand processors it is not going to look as bad. So but I am saying P more processors. Will the speed be P / 3. P / 3 ? It in practice it will be a little bit better than P / 3. “Professor – student conversation ends” Why? Assuming certain architecture things because now instead of having one big list for the entire processor you have smaller lists (())(50:43) right.

So known locality between each one of them and better memory performance. But again yes instead of merging everything once you are merging it three times. So in that sense it is not embarrassing, embarrassingly parallel. But that is kind of a side track issue. The main issue that I wanted to focus on was that it is optimum. The work wise right it is taking linear amount of time; linear amount of work. But here we have not really done the time analysis. We said N = P squared.

But then we are kind of hiding it. We will talk about what happens if N is greater than P squared. But if you just look at the time in the work time sense, how long does it take? P is equal to root N. So it is root N right. So time wise it is not doing that well that well. So here is an example of something that it is work optimal; not very well in terms of time.

But because it is work optimal and we have established a much more equitable distribution of work which is very local in nature. Very sequential in nature. It works very well. “Professor – student conversation starts” So for given that is sequential will do it in order N in time. And then when we do it in parallel we are getting order N / P. So is it not it I mean the best we can do apart from the factor 3?

Yes that is why we are calling it optimal. So why do you want to call it the time is not very good; that is the best that. I did not say that precisely. “Professor – student conversation end” I said if you look at the work time behavior then you would come to the conclusion that it is N by root N. So it is square root N right. Where as you would like it to be something in the neighborhood of log N. In fact we have seen before. Let us stop for just a minute.