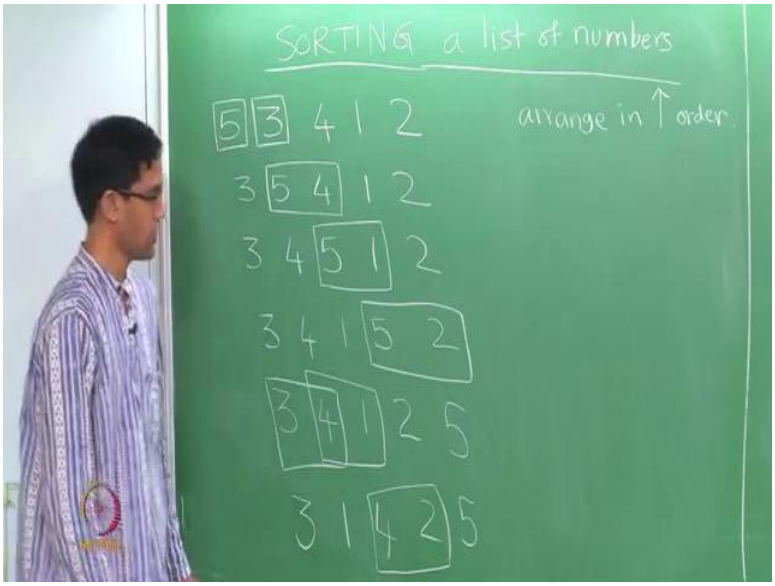
An Invitation to Mathematics
Prof. Sankaran Viswanath
Institute of Mathematical Sciences, Chennai

Unit
Combinatorics
Lecture - 17
Sorting lists of numbers and crossings in Tangle diagrams

Welcome back, so what we will talk about today is the following problem, that of sorting a list of numbers.

(Refer Slide Time: 00:20)



So, sort a typical example, I write down the numbers from 1 to 5, but not in increasing order, I write them in some permuted orders. So, I write 5 3 4 1 2 for instance. And now the problem is to try and sort these in increasing order. So, it is means, you must try and change interchange positions of these numbers, such that it now reads 1 2 3 4 and 5. So, of course, this is the standard problems, so what we want to do is to arrange these numbers in increasing order.

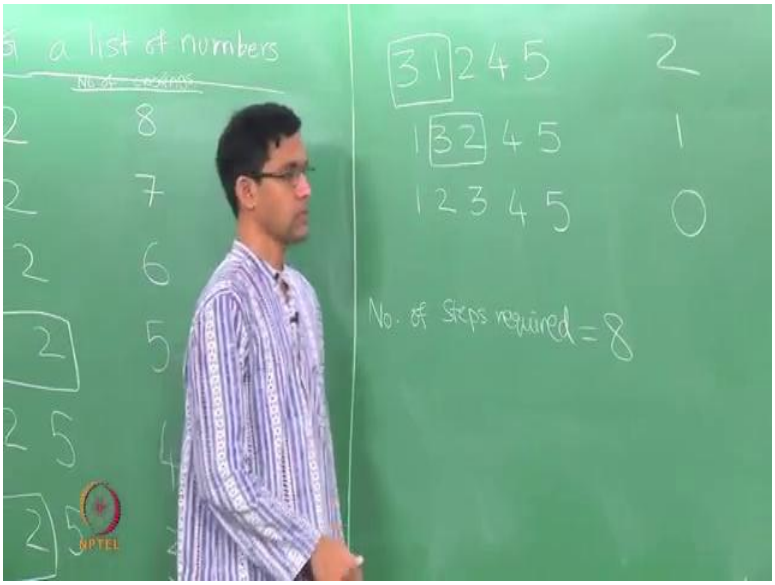
So, many different algorithms for sorting list, one of them is the following, you do a sequence of comparisons, so what we do is the following adjacent comparisons. So, you take 5 and 3, you take the first two numbers in the list. If they are in the wrong order, which means that the first number is larger than the second, then you swap them, because you want them in ascending order. See, you swap them, provided they are in the wrong order, there are in descending order, so the first two have now been taken care of.

So, now, you move on and compare the next two numbers in the new list. So, again 5 and 4, they are in the wrong order, so you interchange them, you swap them, you make this 3 4 5 1 2. So, now, the first two numbers and the second and third numbers are in correct order, so now you move on to the next pair. And again, you swap them if they are in the wrong order, so in this case again you need to swap.

So, you interchange them, make them 1 5 2, and again you move on to the last two numbers which in this case you again need to swap. So, this is the sequence of numbers that you generate of a list that you generate. Now, observe at this point, you know what you have with this is that the number 5 is in the correct place. So, what this is really managed to do is that the number 5 is in the appropriate place which is in this case the very last.

But of course, the rest of the numbers are not necessarily in the correct order, so you need to keep doing this. So, once 5 has reached it is correct place, you start again from the beginning, you are again look at 3 4 and you again swap them if they are in the wrong order. So, in this case there is no need to swap, so you do not do anything, so then you focus your attention on the next two numbers. This case 4 1, they are in the wrong order, so of course, you need to swap those two now, the next 3 1 4 2 5. So, the window now moves from 4 1 to the set of two numbers, so it moves to 4 2. So, again you need to swap, so let we write this out here.

(Refer Slide Time: 03:20)



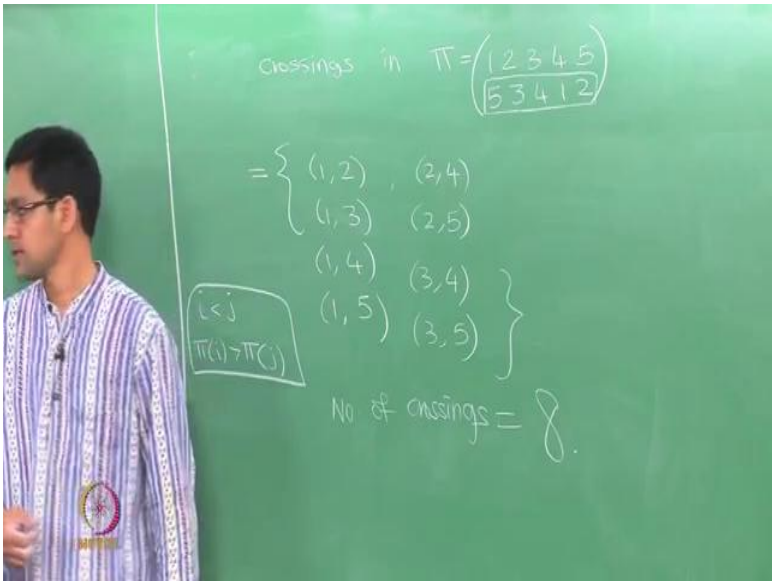
It now becomes 3 1 2 4 5, and now you compare the last two numbers and here you do

not really need to swap, because they are in the correct order. So, now, at this point you have sort of scan the list again and what happens is that, both 4 and 5 are in their correct positions, but the first three may not be. So, you again begin from the first, you start at 3 1, if there in the wrong order you swap, so it becomes 1 3 2 4 5. Now, you move your window one step to the right, you compare these.

If they are in the wrong order, then you swap and now you are done. So, what you get at the very end of this process is a list in ascending order as required. So, now, here is an interesting observation regarding this, if you look at the total number of steps that it took us to get to the correct answer. So, let see what happened, you started with the given list. Now, step 1, step 2, step 3, step 4, step 5 and then you have step 6, step 7 and step 8. So, when you perform this algorithm 8 times, you get to the list in ascending order, so let us write that down.

So, the number of steps required for this particular algorithm where you sort of compare adjacent elements. So, the number of steps required turns out to be 8 and here is an interesting thing. If you look at crossings, so remember all of this is somehow related to permutations, so let us also do the following.

(Refer Slide Time: 05:08)



Let us look at the number of crossings of the original permutation, number of crossings in the original permutations. So, what do you mean by the original permutation pi? I think of it as, well let us write it in two line notation, think of the original list of numbers as just being, it just tells you 5 3 4 1 2. So, this is the original list of numbers that is given

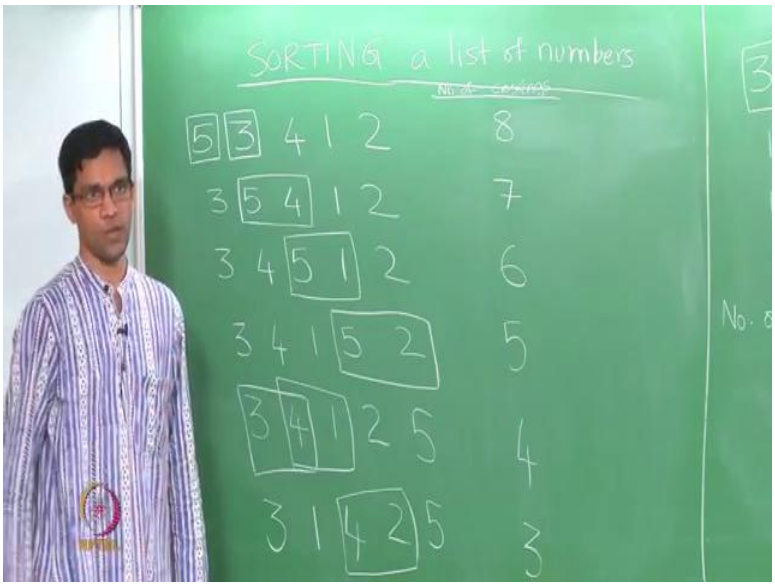
there, I think of it as a permutation, it is a permutation of the numbers 1 through 5, so I have written that permutation down in two line notation.

So, let us count the total number of crossings in pi. So, where I could draw the crossing diagram, but for one's let us just use the formal definition of crossings, which is I need to look for all pairs i comma j, where i is smaller than j, but pi of i is bigger than pi of j, this is the crossing condition. So, let me look for all pairs of numbers like that, so from the top row. So, for example, i is smaller than j, so I look at 1 and 2, pi of i 5 is greater than 3.

So, surely 1 and 2 are in the crossing list, similarly 1 3, 1 4, 1 5 each of these pairs counts as a crossings, because I satisfy the definition of crossing. Similarly, if you look at let us see, so for 2, so I have 2 4 and 2 5, for 3 I get 3 4, 3 5 and that is it. So, this is the full list of crossings for the given permutation pi and observe there are in fact 8 crossings. So, the total number of crossings of the original permutation pi also happens to be 8. So, this is the list of crossings, so this is the set of crossings and so the number of crossings is of course 8.

So, here is another, so this turns out that it is not an accident, it is true. In general, the number of crossings of a given permutation, you can think of as being another way of… Well, what is it do, it tells you the number of steps you will need in this adjacent swapping algorithm, before you get the list in ascending order. It is a sort of a measure of how many steps you will need to keep doing this adjacent swappings, before you finally get the correct, the sequence in ascending order.

(Refer Slide Time: 07:51)

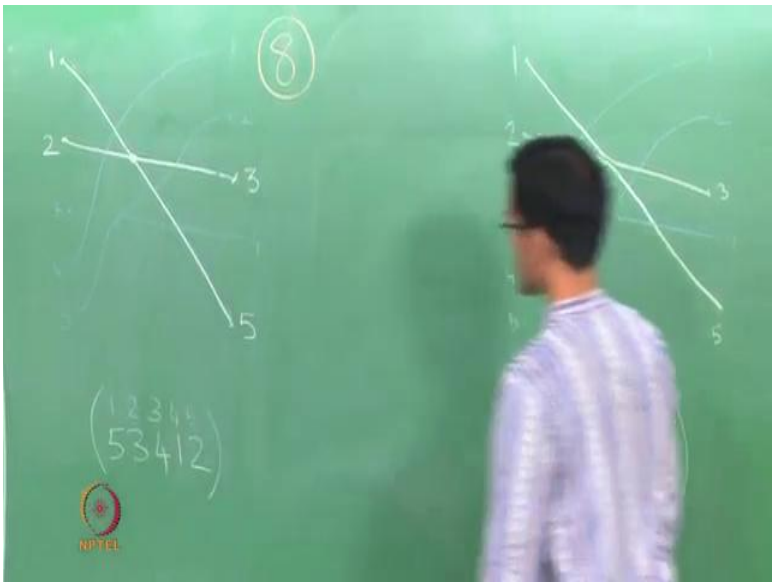


And it is a sort of instructive to, actually count the total number of crossings here, so let us count the number of crossings. For each of these permutations, here the first permutation that we started out, I said well we just counted the number of crossings is 8. Now, of course, then I have done one step to the algorithm and it is a sort of interesting to ask well, how many crossings are there here. So, here is the interesting answer, it turns out to be exactly 1 less than the original.

Now, once more you do this and you wonder how many crossings just this have, well it has 1 less. This has 1 less crossing, this has 1 less crossing, this has 3 crossings, ((Refer Time: 08:33)) this guy has 2, this has 1 and of course, the permutation which is in increasing order has no crossings at all, because it is crossing diagram we just mapped 1 to 1, 2 to 2, 3 to 3 and so on. So, here is the sort of another interesting aspect of crossings, when you perform this algorithm to sort a list, at each step of the algorithm what you end of doing is decreasing the number of crossings by 1.

So, I am sort of going to leave this checking as an exercise, but let just draw the crossing diagram to see, why when you go one step in this process, you will decrease from 8 to 7. So, let just do that and the rest I will sort of leave for you.

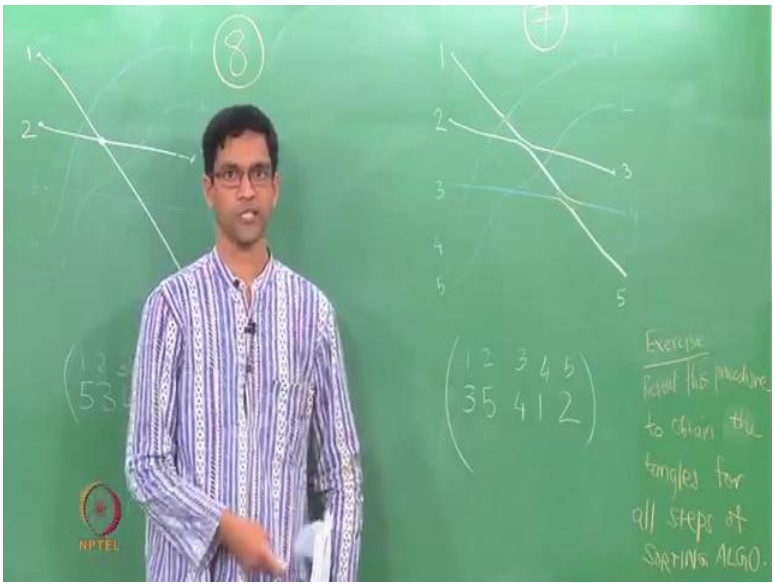
(Refer Slide Time: 09:15)



So, here is the first permutation 5 3 4 1 2. So, what is it do it, maps 1 to 5, so I am drawing the picture of the permutation 5 3 4 1 2. Can you think of it as a list? So, I am writing it in a one line notation, so remember in two line notation this means 1 maps to 5, 2 maps to 3. So, this is basically the permutation, so 1 goes to 5 and 2 goes to 3. So, 2 maps to 3, so that is the diagram. I am not drawing the rest of the lines right now or maybe I will just draw them, it will like 3 maps to 4.

So, I have 3 maps to 4 or 4 maps to 1 and 5 maps to 2, so that is really the diagram of this permutation. Now, let just see what happens when you do the first swapping alone. So, the first swapping what did you do, it interchange the positions of the 5 and the 3, because it compare just the first two numbers. So, one step of this algorithm does the following.

(Refer Slide Time: 10:41)



It interchanges 5 and 3, I get 3 5 and then I get the rest are unchanged 4 1 2 are unchanged, so this is 1 2. So, here is the permutation which you get after one step of the sorting algorithm. So, again it is really only the action is really here 1 and 2, so 1 and 2, so they are really do not participate in the action, so it is 3 4 and 5. So, let us see what does it do, well earlier you had ((Refer Time: 11:16)) 1 mapping to 5 and 2 mapping to 3. But, now the roles are interchanged, what you have is 1 maps to 3 and 2 maps to 5.

So, here is how we can construct this new diagram, so I will do exactly what we had earlier. So, 1 maps to 5 and 2 maps to 3 was how the original permutation looked. Now, what we want really do is, sort of at the place where they cross, you sort of want the 1 to send them along the other sort of the other directions. So, here is what I will do. So, remember I have a lot of flexibility in drawing these tangle diagrams.

I do not need to draw them as straight lines. So, what I will do is, I just slightly move this point of intersection here. So, just where they intersect, let us erase this little bit and curve these lines out in this way. So, what we have now done is, 2 was initially being mapped to 3, but sort of at the point of intersection we just twig it the little bit and sent it off to 5 instead. Similarly, 1 which was mapping to 5, at the crossing point we sort of make it change it is mind and go off to 3.

So, what you really have is a diagram which looks like this, the tangle diagram you are sort of split these two that point of intersection has now become two non intersecting lines. So, now, what you have done really is got a grid of 1 crossing, the earlier crossing

that you had where ((Refer Time: 12:46)) 1 the line joining 1 5 and 2 3 crossed each other at that point. You have now resolved that crossing; you have now removed that crossing and instead have two non crossing lines.

Now, observe the rest of the picture really does not change. So, earlier I had 3 goes to 4, so I will just write the same way draw the same picture 3 to 4, 4 going to 1, we draw the same picture, 5 going to 2 we draw whatever you do earlier like this. And so as far as the blue lines are concerned, the number of times the blue lines intersect the white lines that does not change, because for all practical purposes the white lines are really the same as the old white lines.

The only change is taking place in a small neighborhood around the intersection, where that is just a slight modification in the two lines. So, the blue lines are not really going to see this, you know see the fact that there is something change at the point of intersections, because they are going to sufficiently far away. So, you should really think of, this is the sort of correct way of thinking about what happens when you perform the sorting algorithm.

You start with some crossing diagram and ((Refer Time: 13:57)) the lines for 1 and 2, the two adjacent fellows which you say swap. If that is a crossing, which means that you know the number in the first place in this case of 5 is bigger than the number in the second place which would lead to a crossing. If there is a crossing, what you do is you just remove that crossing by sort of doing the procedure that we just talked about.

You just erase that point of crossing and then let these two lines become non intersecting. When you do that, what you have really achieved is swapping those two numbers, because now you are going to map 1 to 3 and you are going to map 2 to 5. So, this is what happens at the level of crossing diagrams when you perform an adjacent swap and the rest of the diagrams unchanged. So, it is clear enough that if the original number of crossings is 8 or whatever number it is, so here is the total number of crossings in the original diagram.

After performing a one step swap, you have of course, reduce the number of crossings by 1 and which crossings did you get rid of precisely the 1 where you did this manual and so on. So, again you what you do in the second step the crossing algorithm is to look at 2 and 3 for instance. So, the line joining, you know the 1 which emanates from 2 thus in fact, cross the line which emanates from 3. So, what you want to do is, in the next step of

the crossing algorithm is, in the sorting algorithm is to resolve this crossings.

So, at this point you will erase it and sort of make the top line go along the top and the bottom line go along the bottom, you make them non intersecting and so on. So, here is a nice exercise, you know just purely pictorial, so all these are really nice, because you can do them, you know just graphically, visually they are rather appealing. So, exercise do this for every one of the 8 steps, exercise draw all these diagrams, so repeat this procedure. So, repeat this procedure to obtain all the crossing diagrams of all the steps.

So, all to obtain the tangles for all steps of the algorithm and at every step, doing the same thing sort of, erasing this and sort of making it go up and so on. So, let me just do one more step to just make this go this way, this go this way. So, I am keep doing this, so when we keep doing this at every step finally, it sort of magical to see that, when you keep doing this again and again, at the very end what is it, you are going to get.

You are going to get a tangle which sends 1 to 1, which sends 2 to 2, which sends 3 to 3 and so on, so you should sort of see how the various pieces arrive. So, for instance 1 to 1, what probably happens is you go down like this and you, at some point you will slightly resolve this. So, that the 1 now goes up to 1 like that and the 2 goes to 2 and so on and so for. So, it sort of very interesting to see how this original diagram will sort of change step by step, so that at the very end it lands up sending every number to itself.

So, it is a rather nice pictorial exercises and I will sort of verge you to work this out entirely yourself. So, that you get a nice pictorial feel for what sorting really does at the level of crossings.