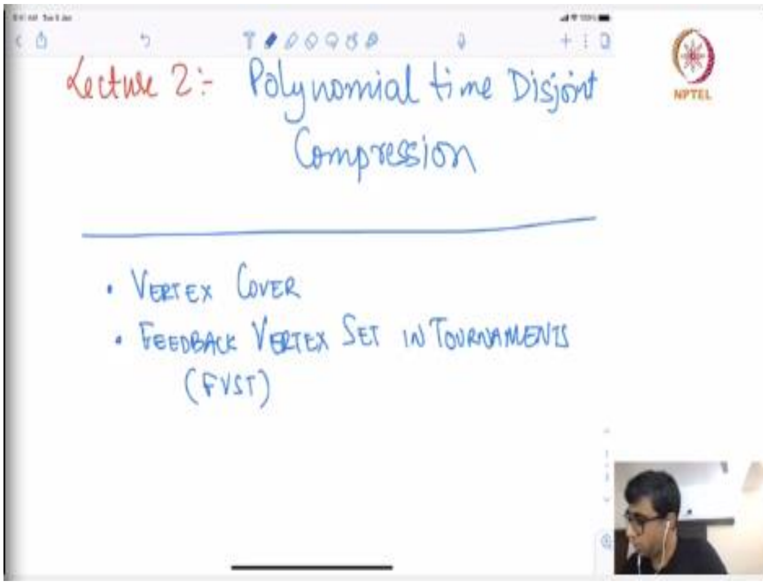
Parameterized Algorithms
Prof. Neeldhara Misra, Saket Saurabh
Department of Computer Science and Engineering
Indian Institute of Technology-Gandhinagar

Module-03
Lecture-13
Iterative Compression II: Vertex Cover and Tournament Feedback Vertex Set

(Refer Slide Time: 00:15)

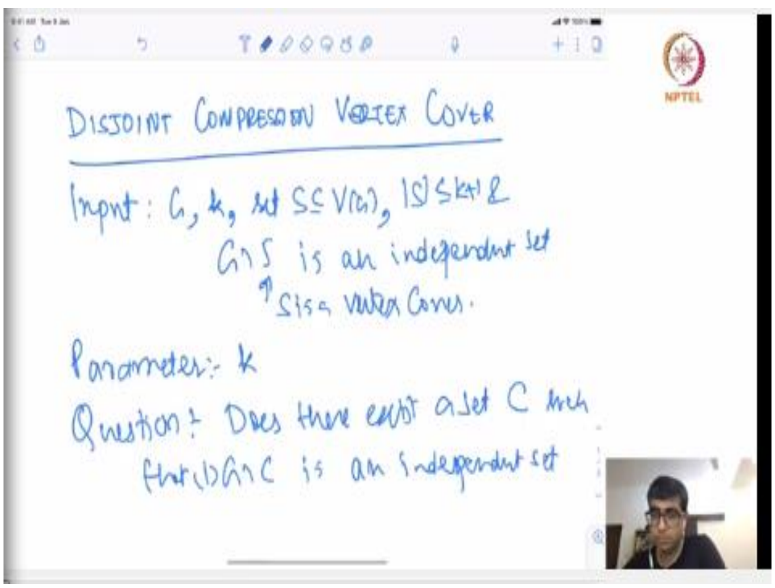
Welcome to the second lecture on iterative compression. In the first lecture we set up the whole framework. And we realized that all we need to solve to get an algorithm, we need to solve a disjoint compression of a problem. In this lecture I am going to give you for 2 problems, we are going to solve the disjoint compression problem in polynomial time.

(Refer Slide Time: 00:46)



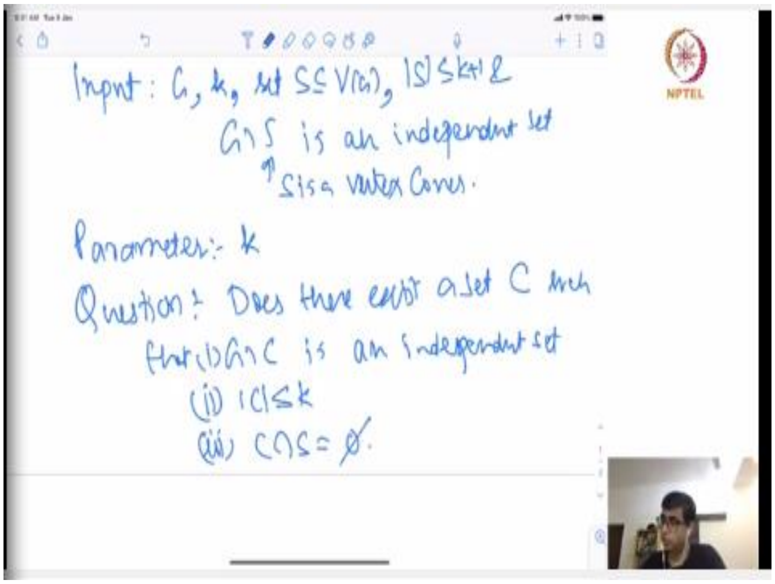
So, the 2 examples which will be constitute today's lecture is example of vertex cover and then we will look at what is called feedback vertex set in tournament. That means this will be called FVST. So, first let us look at vertex cover, so what is a disjoint compression problem for this?

(Refer Slide Time: 01:16)



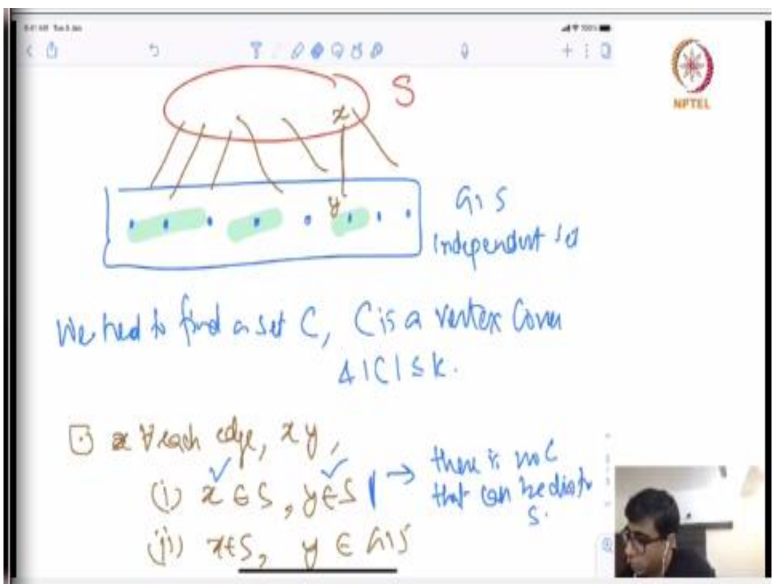
So, disjoint compression vertex cover, so what is in the disjoint? Input is G and integer k, a set S subset of V G mod s is less than equal to k and G - S is an independent set or in other word k + 1 S is a vertex cover, what is a parameter if k and what is my question. Does there exist a set C such that first of all G - C is an independent set.

(Refer Slide Time: 02:35)



Second, cardinality of C is at most k, and third C intersection S is empty, so we are given the vertex cover of size at most k + 1. And we would like to have a vertex cover of size at most k whose intersection with s is an empty set. So, look at what is given to us. So, let us draw a picture to us.

(Refer Slide Time: 03:01)

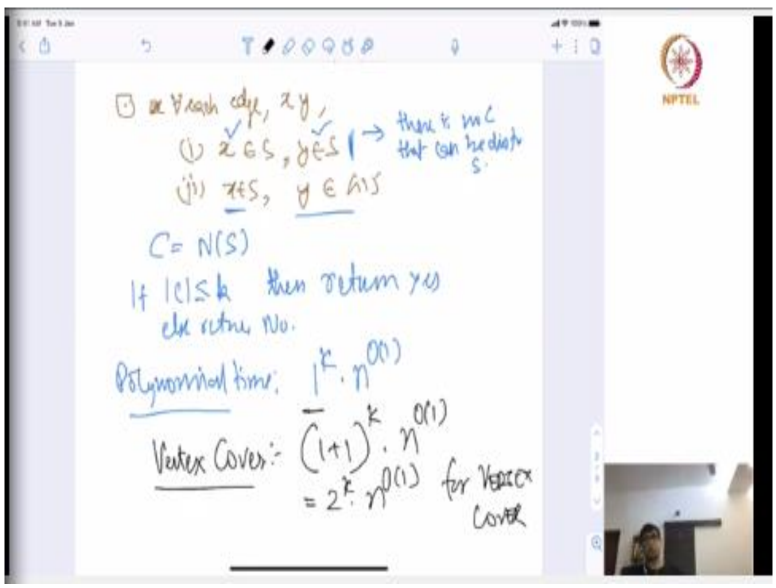


Here is an S and here is G - S which is an independent set. And we need to find a set C here. Such that C is a vertex cover and mod C is at most k. Now notice what was the role of S. S is a vertex cover. So, hence every edge is like this, every edge is adjacent to a vertex in S. Now which implies that look at any edge here, so this is a very simple thing, look at any edge here there are 2 ends points to it, one is x other is y, so what we know.

For each edge xy, there are 2 cases, first of all x belongs to S, y belongs to S, second x belongs to S, y belongs to G - S that is all that can happen, there could be an edge here. But imagine if this is the case, if this case occurs but we are looking for a vertex cover C which is disjoint from S. So, if both x belongs to S and y belongs to S for an edge, then in this case you say there is no C that can be disjoint from S.

So, now what we know such edges do not exist. So, only thing which we have is that for each edge x is in the S and y is going across. Then for such edge, because x cannot be selected in the solution you need to pick y. So, basically what is your C?

(Refer Slide Time: 05:38)



C is nothing but N of S that is it. So, in this case if mod C is less than k then return S else return no, so that is it. So, we have solved the problem in polynomial time, which implies we have solved the problem in 1 to the power k times n to the power O of 1. And hence by the whole methodology this gives us an algorithm for vertex cover in time, what was that 1 plus the running time of the compression algorithm, so 1 + 1 to the power k n to the power O of 1.

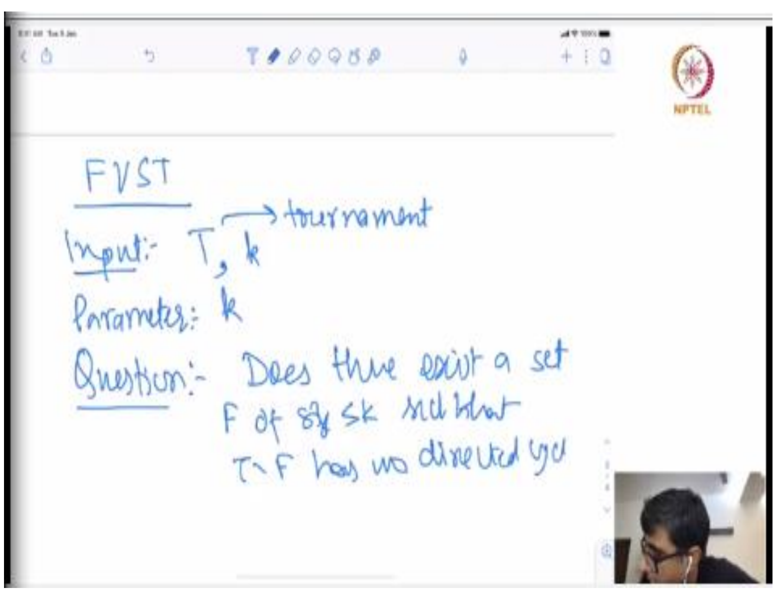
So, we get 2 power k n to the power O of 1 algorithm for vertex cover. So, because we can solve the compression version of vertex cover in polynomial time, that algorithm is very simple basically what you are saying that, look I gave you a vertex covered and we have also seen this idea already in our first lecture when we introduce the field of parameterized complexity, is that. Look, you are looking for a vertex cover which is disjoined from S then better be that the graph induced an S does not contain an edge.

Because then there cannot be a vertex cover that is disjoined from S. And then you know that for every edge one of the end points must be selected and one end point is already part of S which are not supposed to pick in, it means you have to pick up all the other endpoints into your prospective solution C. So, then you pick them up, check it is size, if it is size constraint is satisfied, you written yes else you return no.

And you can solve this problem in polynomial time which is like 1 to the power k n to the power O of 1. I wrote this 1 to the power k just to illustrate that if you fit into this the first lecture algorithm the outline we made it will lead to 1 + 1 to the power k n to the power O of 1 time algorithm for vertex cover.

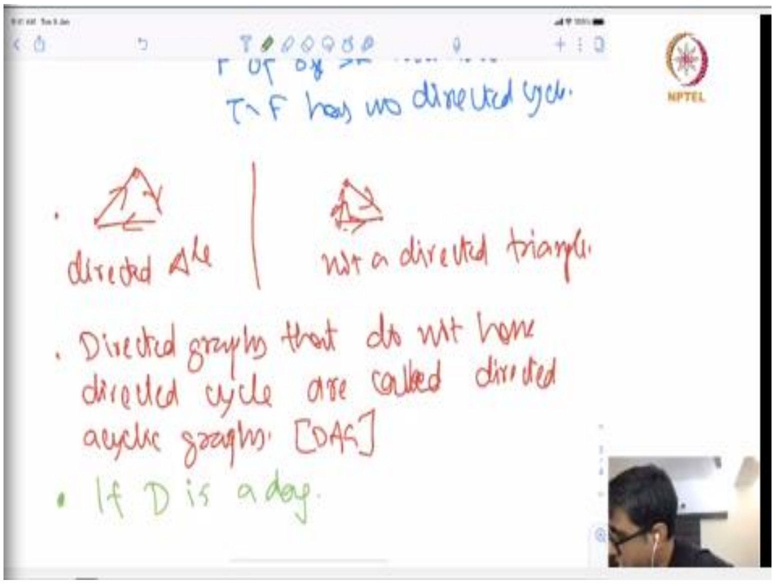
So, it is very simple see, so once we set up the whole iterative compression setup look at the kind of structure it already provided that we can solve the problem in polynomial time by just following some simple set of strategies or methods. So, the second problem that we are going to consider is slightly more non-trivial, and so this is called feedback vertex set in tournaments.

(Refer Slide Time: 08:22)



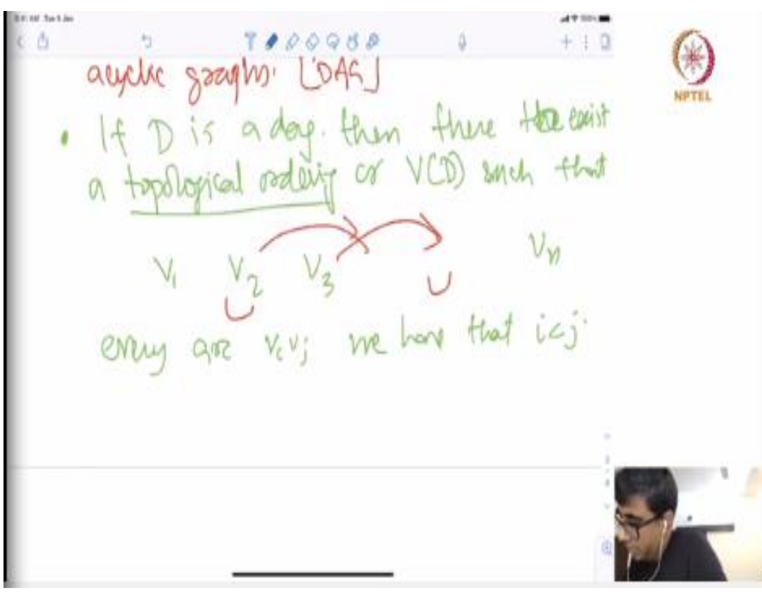
I will define each of this term. So, what is an input? Input is a tournament and integer k, so tournament, parameter is k. And the question is does there exist a set f of size? At most k such that T - F has no directed cycle**.**

(Refer Slide Time: 09:11)



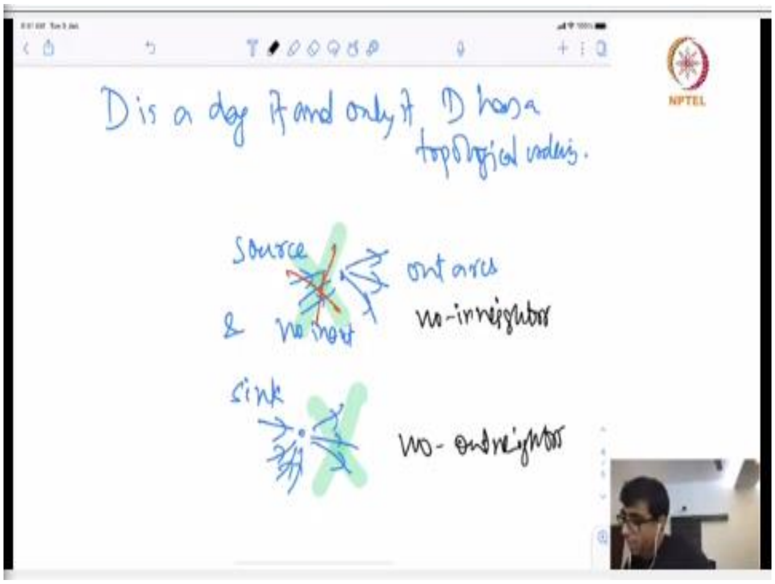
So, for example this is a directed triangle, but look at this, this is if you forget this direction then this is an undirected triangle. But if I start from here I cannot go back to the same vertex, so it is not a directed triangle. And directed graphs that do not have directed cycles are called directed acyclic graphs. So, this is or which you must have heard DAGs and there is a property of them. What is the basic property of them?

(Refer Slide Time: 10:27)



If D is DAG then there exist a topological ordering of vertex set of D such that, so what is the topological ordering? So, there exist some ordering like V 1, V 2, V 3 dot, dot, dot V n such that every arc V i V j, we have that i is less than j. So, basically then you can come up with an ordering. So, with every arc goes from a smaller index vertex to a larger index vertex.

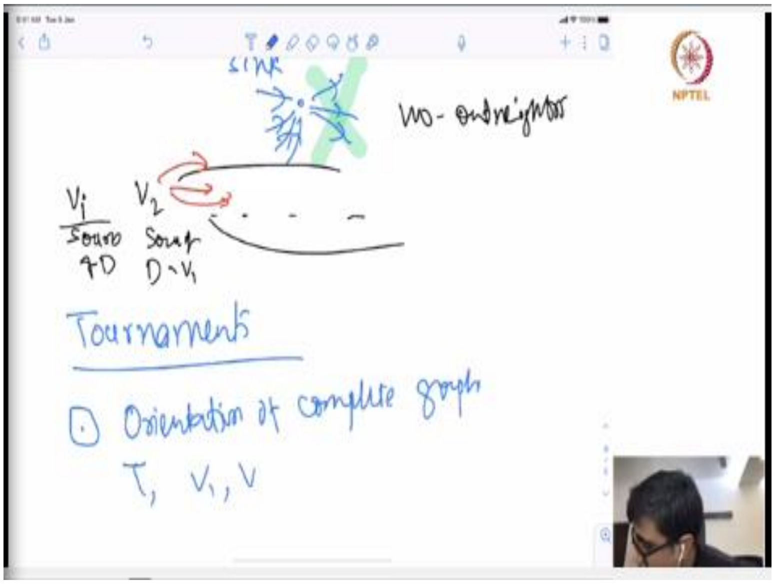
(Refer Slide Time: 11:21)



And in fact what is known is that D is a dag, if and only if D has a topological ordering. And how is this topological ordering is find is that you first find a vertex of, with what is called? Source, what is your source vertex? Source is a vertex which has only out arcs and no in arcs. So, these kinds of arcs are not allowed, then this is called source. Similarly you define sink, what are sink?

All those vertices which has only in neighbours but no out arcs or no out neighbours this is forbidden, this is forbidden sink. So, source means, what is the meaning of source, no in neighbours. What is sink? No out neighbours. So, look at the first vertex, like so in the topological ordering first word say you can always show that it is a source. And if you delete this vertex the next vertex will be form a source of the other guy so on and so forth.

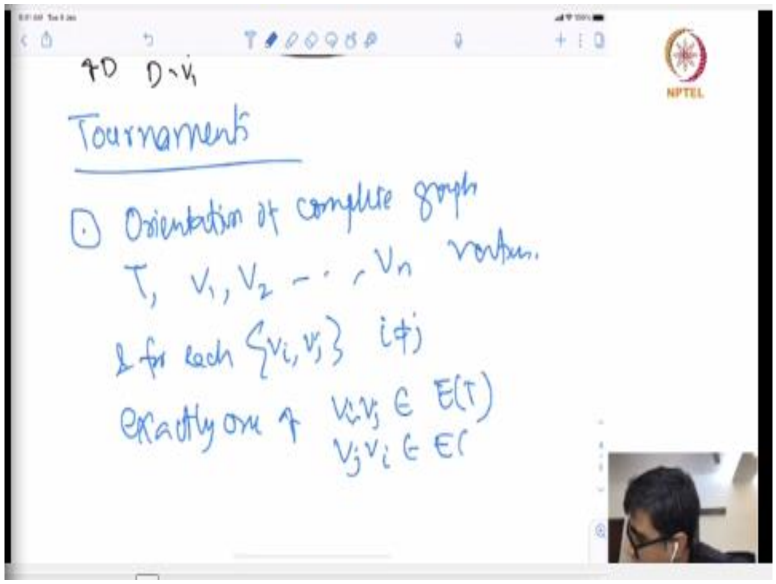
(Refer Slide Time: 13:13)



So, the way to get a topological ordering is start with first V 1 source of D sum V 2 source of D - V 1 so on and so forth. So, every vertex is a source in the remaining vertex, which immediately implies that the only arc which is incident on them towards this is going after them, like all the arcs appears after V 2. And so this is why you can orient them, so this is a very good property of a dag which is used quite extensively in the algorithm.

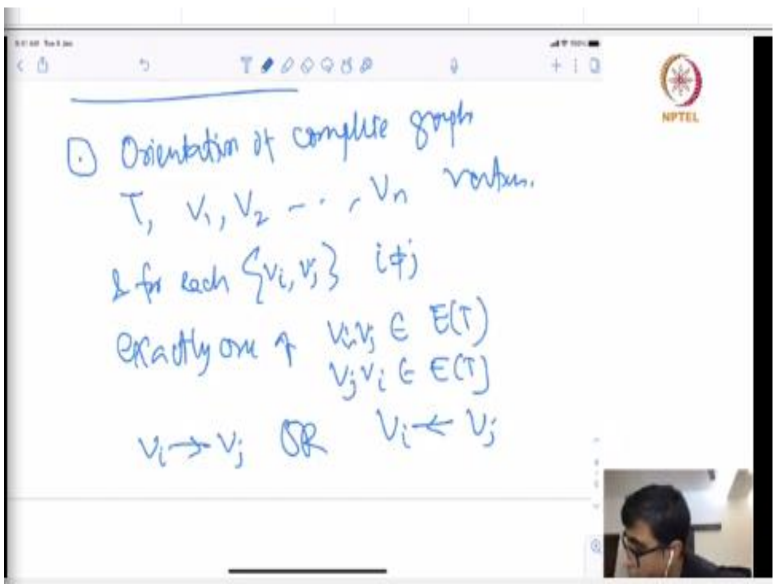
And you must have seen this how to find topological ordering also using the algorithm which I told you or using some doing DFS twice. so now let us go back to the problem which we were dealing with tournaments. So, what is tournaments? Tournaments is an what is also called orientation of complete graph or in other words tournament has V 1 to V 2 n vertices and for each V i V j i not equal to j.

(Refer Slide Time: 14:26)



Exactly one of V i V j is in arc set of T or let us call it x otherwise just to avoid confusion. At set of T or arc set of T or V j V i is in the edge set of T. So, what is the meaning of this?

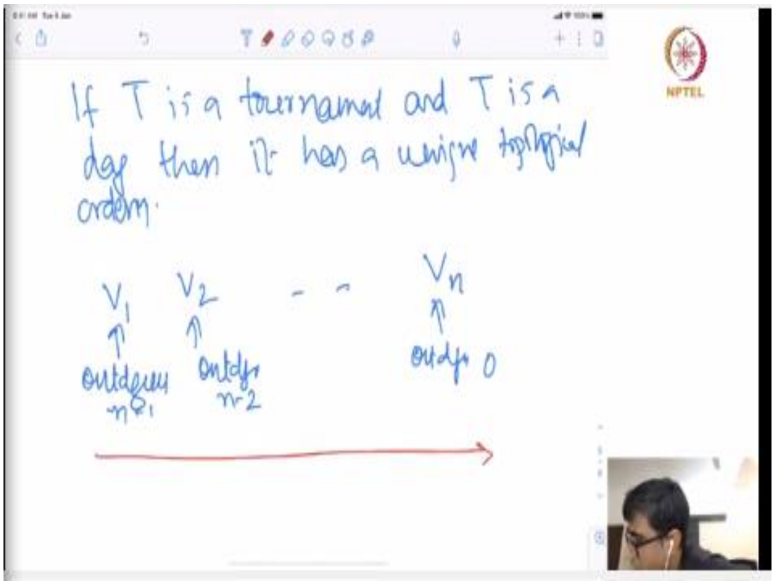
(Refer Slide Time: 15:00)



So, given a vertex V i and V j either you have an arc edge like this or arc like this. So, at least you have arc in one of the directions. So, this is why what is the complete graph? A graph is called complete if for every pair of vertex there is an edge. So, this is basically an orientation, so for each edge you have decided a direction to go from one vertex to other. So, this is why this is a tournament is a complete set of vertices.

And between every pair of vertex there is exactly one arc either going from V i to V j or going from V j to V i.

(Refer Slide Time: 15:42)

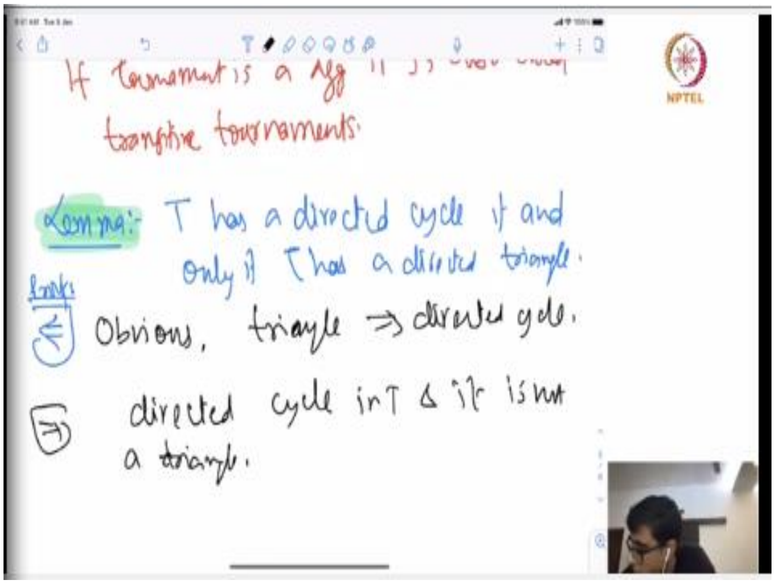


Notice that if a T is a tournament and T is a dag, then it has a unique topological ordering. Why? Because the first vertex has to be the source vertex of T, it means we know that starting from this vertex has arc to every one of them. And that arc direction is fixed because this guy has no one coming in, so everyone only leaves him. So, the first vertex of the property is that it is a unique vertex of out degree n – 1.

What is the property of V 2? It is a unique vertex of out degree n - 2 so on and so forth. And what will be the vertex V n? It is out degree is 0, so there is an unique topological ordering, so there is like no more topological ordering. There is one fixed unique topological ordering of so this is an important fact to remember.

So, we have taken a short break from the our problem and so I have told you little bit about what feedback would like whatever directed graphs are, what is directed acyclic graph, what is topological ordering, what a tournaments, what is a unique topological ordering are?

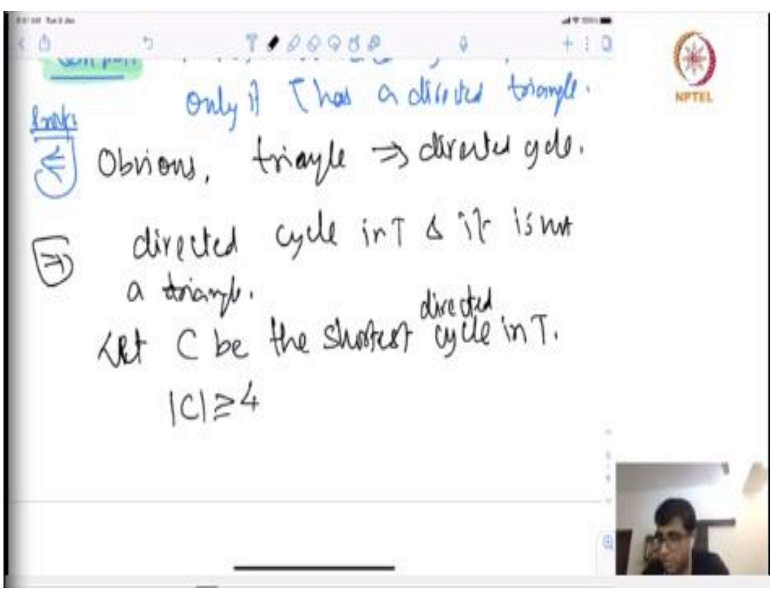
(Refer Slide Time: 17:40)



If T is a dag, if tournament is a dag it is also called transitive tournaments. So, now the one very special property which we will use for tournament is actually and this is a key lemma for the first what is the lemma? T has a directed cycle if and only if T has a directed triangle proof, this is a key lemma. What it tells us? That if you wish to hit all directed cycles in a tournament then it is sufficient to hit a directed triangle, and why it is this statement true? Let us prove this.

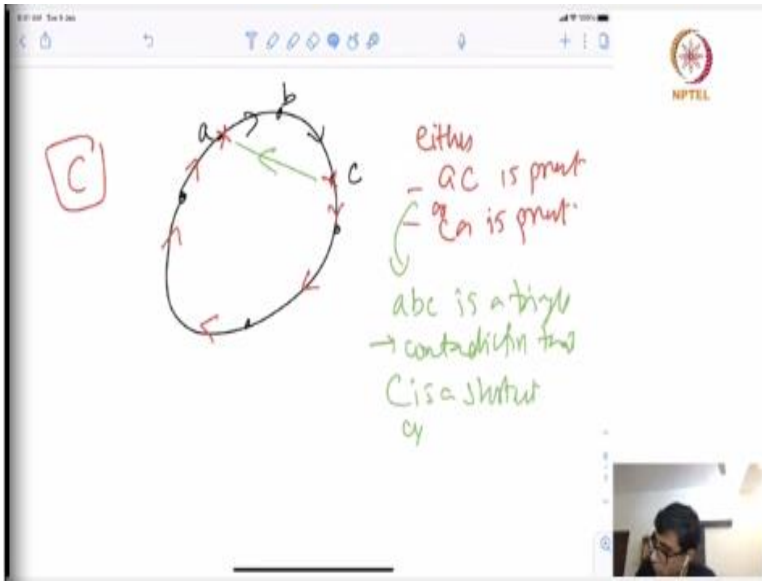
This statement is true, reverse direction is obvious, why obvious why? Triangle implies directed cycle, but let us look at a directed cycle in T and it is not a triangle. Then I will show that it has a T has a directed triangle.

(Refer Slide Time: 19:40)



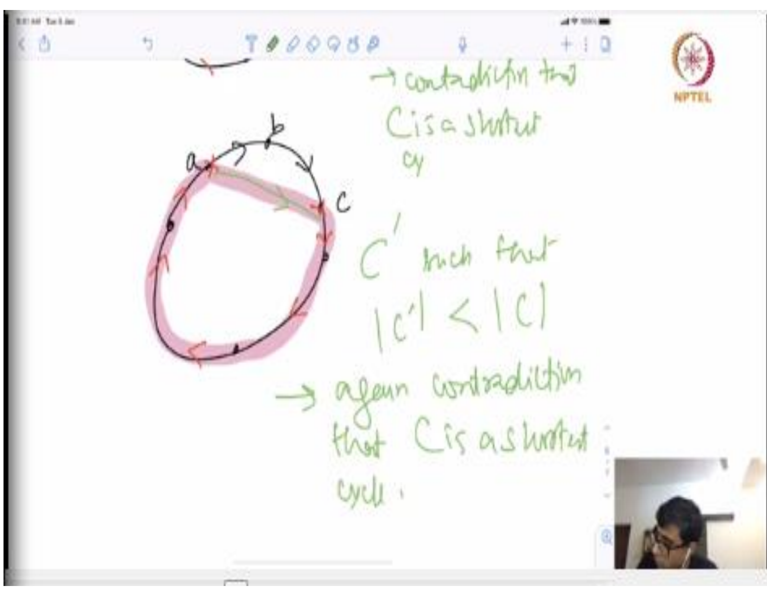
So, let C be the shortest directed cycle in T, and we know C is greater than equal to 4 otherwise we are happy.

(Refer Slide Time: 20:02)



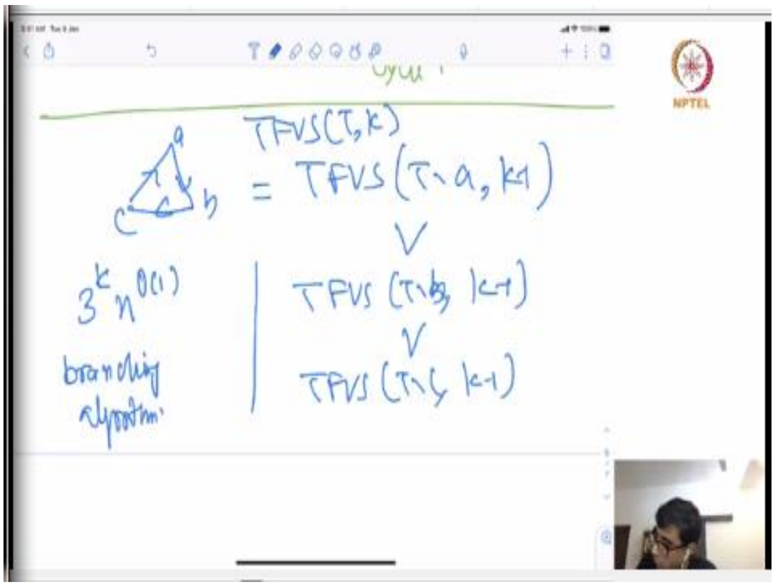
So, let us draw this. Let us look it is a directed cycle, so let us announce length a, b, c and there are more vertices here. Now let us ask ourselves a and c and this is an orientation of this cycle, this is c each. So, we know that either ac is present or ca is present. Let us ask ourselves what happens if ca is present? If ca is present then in this case a, b, c is a triangle, contradiction, that c is a shortest cycle which implies the orientation is.

(Refer Slide Time: 21:39)



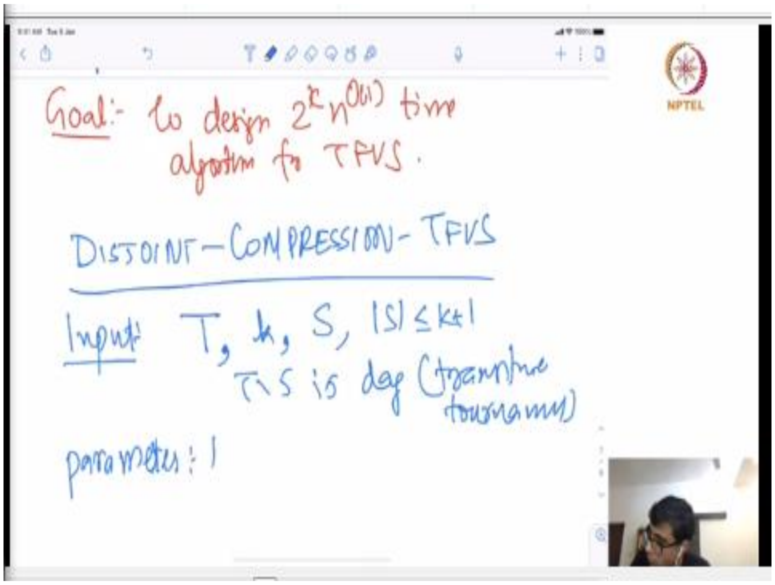
So, now let us look at. Now, so ca is not the present, now suppose ac is present what happens? But now notice that we have got a cycle, we have got another cycle C prime such that cardinality of C prime is less than cardinality of C. Again contradiction that C is a shortest cycle. So, what we have been able to show? We have been able to show the following lemma is that, T has a directed cycle if and only if T has a directed triangle. This lemma immediately gives us a simple branching algorithm.

(Refer Slide Time: 23:04)



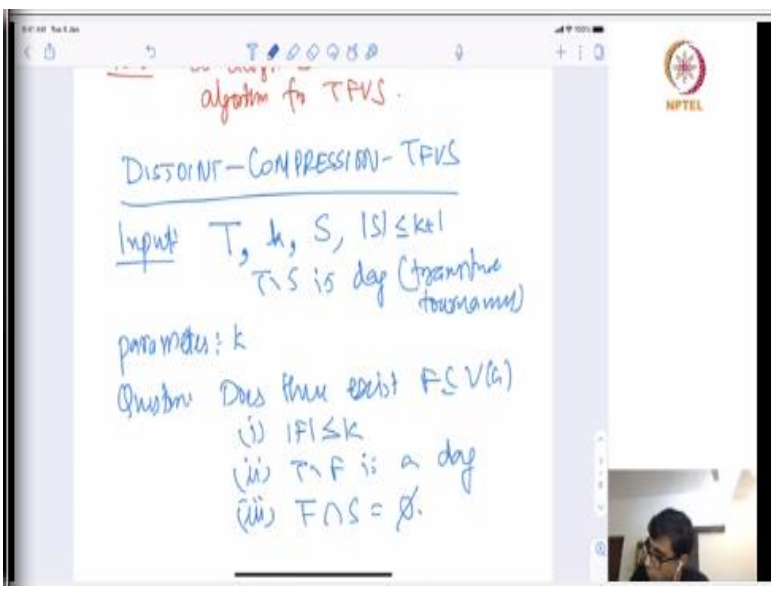
That you find a directed triangle a, b, c and recursively solve the problem on TFVS T - a, k - 1. So, tournament feedback vertex set T, k is equal to either this is true or TFVS T – b, k - 1 is true or TFVS T - c, k - 1 is true. So, this will give you a 3 power k n to the power 1 branching algorithm. What we are going to do is to design a goal.

(Refer Slide Time: 24:04)



Goal to design 2 power k n to the power O of 1 time algorithm for TFVS and how we will do? We will see. We are going to solve, let us not go up, so what we are going to solve? We are going to solve disjoint compression tournament feedback vertex in poly time. And once we are able to do this, we will solve, so what is my input? It is T, an integer k, and S mod S is k + 1, T - S is dag or transitive tournament k.

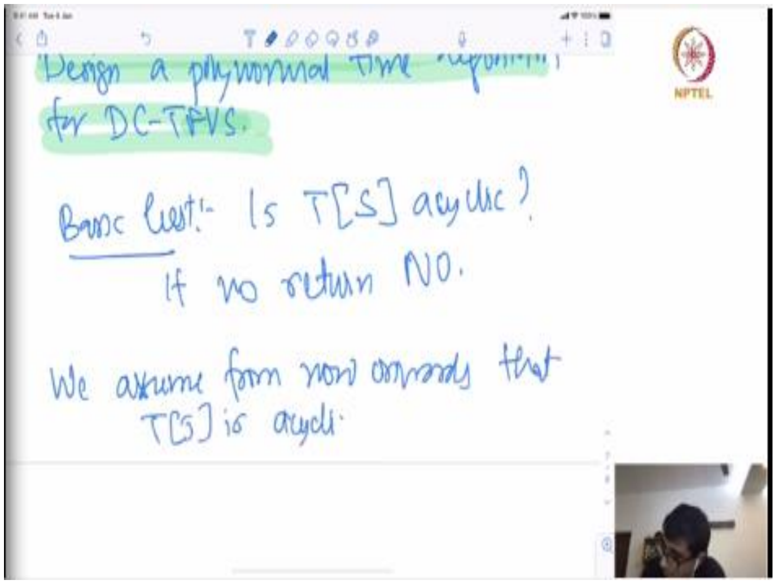
(Refer Slide Time: 25:35)



Question, does there exist F subset of V G, so that first mod F is at most k, second T - F is a dag or transitive tournament, it is a dag. Third F intersection S is empty. So, same like vertex cover we are given a tournament feedback vertex set of size k + 1 and we have some set of properties.

So, the first thing as we did in vertex cover is that, let us check, look. So, now what I am going to do?

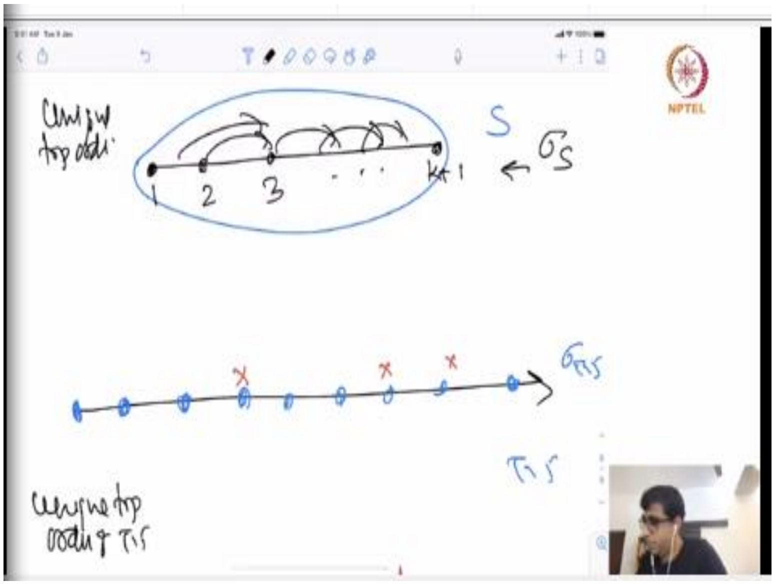
(Refer Slide Time: 26:36)



So, we are going to design a polynomial time algorithm for let us call it disjoint compression TFVS, disjoint compression tournament feedback vertex, this is our goal. And how are we going to achieve this? So, first basic test, what is our basic test? And this is what? It is tournament induced on S acyclic, because if tournament induced and S itself is not acyclic, it means there is a directed triangle or a directed cycle which is contained inside this where all the vertices are forbidden to be part of F.

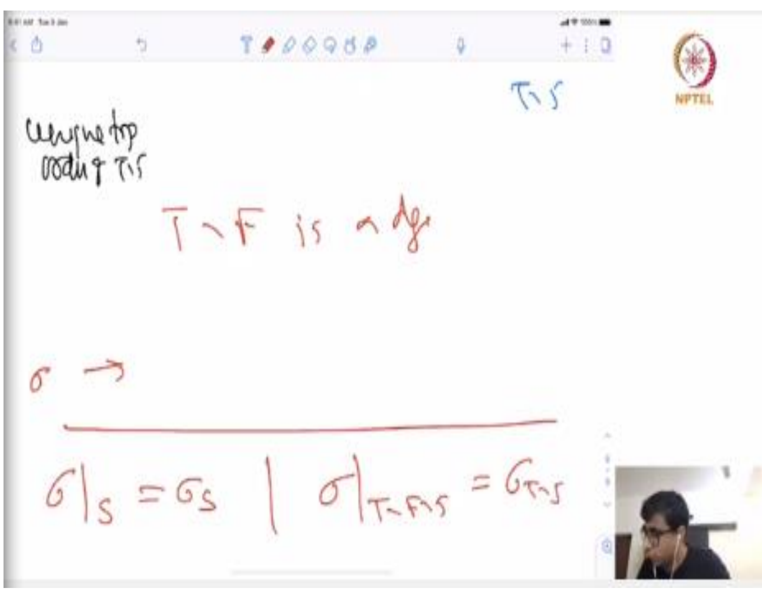
Then there is absolutely no hope to find a solution which will be disjoint from S. So, basic test, if no return, no. So, we assume from now onwards that T of S is acyclic. Then what we know?

(Refer Slide Time: 28:20)



Here is my S and here is my say T – S, what do you know? That S is acyclic; it means it has a unique topological order, let us fix that topological order. And I am going to write this vertices 1, 2, 3 dot, dot, dot k + 1, I am named that. And T - S like unique topological order, I will call this ordering sigma S. Similarly you have a unique topological ordering of T - S. Let us try to understand, let us call this sigma T – S, this ordering and this ordering.

(Refer Slide Time: 29:44)

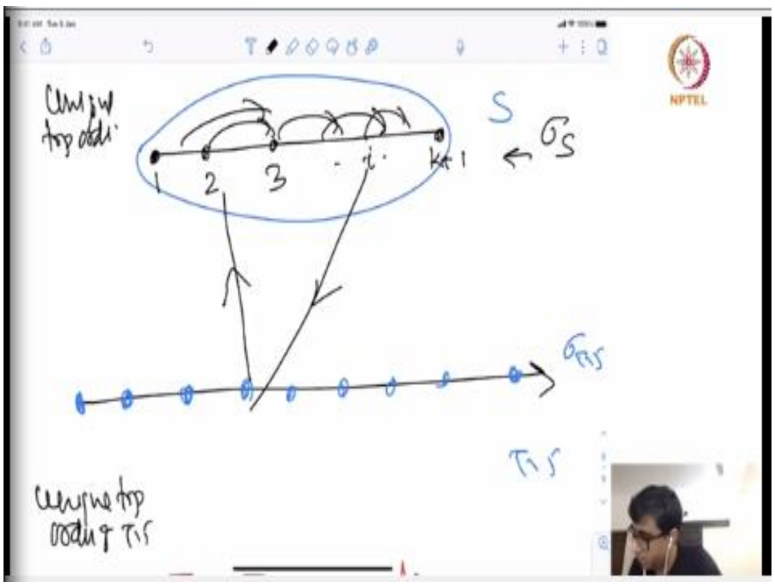


Just notice one thing that our goal is to take some vertices away, such that S union T - S or rather T - F is a dag. Now let us look at an ordering sigma of T – F. Now I claim to you that look if you look at this sigma and look at the sigma restricted to vertices of S then this is exactly equal to S. If you look at sigma restricted to T - F - S this is exactly equal to sigma of T - S.

So, basically what happens is that the vertices which are going to remain from this ordering, and the vertices which are going to from here? So, if you look at the topological ordering of these vertices after deleting the solution vertices then that is also a tournament, so that also the unique topological ordering.

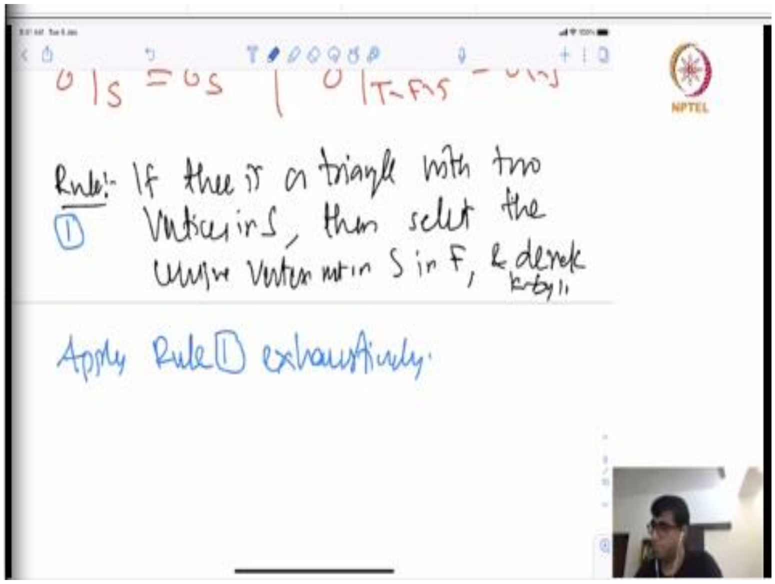
And in fact the good great property of the topological ordering is that maintains the topological ordering already given on S and on T - S, that is a property. So, now some reduction rules. What happens, so now what we know about this? All the arcs are going like this here.

(Refer Slide Time: 31:28)



Now let us fix a vertex such that it has an arc like**.** Then what does it imply suppose it is some vertex 2 i? So, if basically what is a reduction rule?

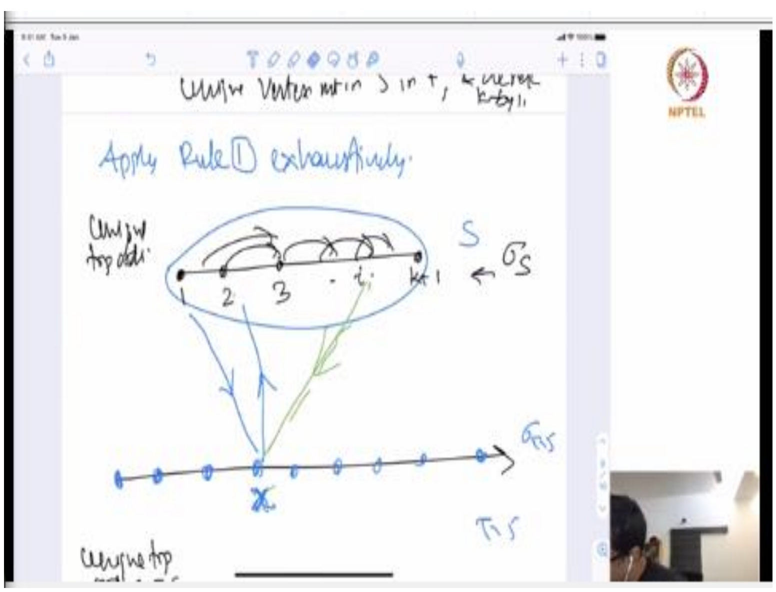
(Refer Slide Time: 32:01)



Rule is if there is a triangle with 2 vertices in S, then select the unique vertex not in S in F and decrease k by 1. Which is same as, so let us now apply rule let us say 1 exhaustively, so we have applied this rule exhaustively. So, what will happen? So, when you apply this reduction rule exhaustively either k have become 0 and there is still triangle then you can say no.

Or you have applied this rule 1 exhaustively and now you have graph where you cannot apply this reduction rule and there is some value of k. Now let us try to understand how does that world look like? So, this is an important.

(Refer Slide Time: 33:34)

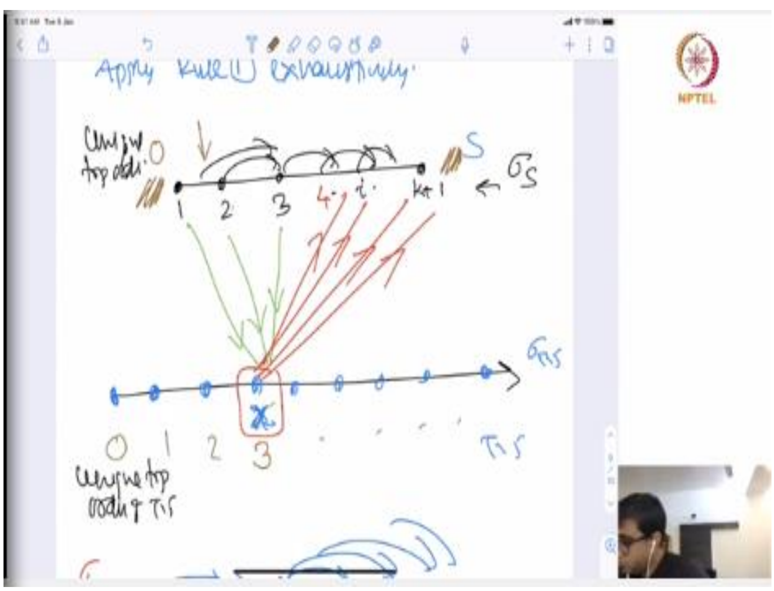


Now let us fix a vertex, now let us fix this vertex and let us ask how is the orientation of arc 1 x? So, notice the orientation of arc let us call this vertex x. If it is going from 1 to x or it could be going from x to 1. So, suppose it went from 1 to x then I claim to you that look at 2, what is an arc? Suppose it goes from x to 2, then I claim to you that you cannot have any other vertex from 3 to k + 1.

Such that you have an arc of this nature, this cannot happen because otherwise x 2 and i and x will form a triangle. So, what happens is that you have in arc coming in and after that once you have outer coming going out of x you cannot have any other in arc coming. So, like you could have in, in, in and then out, out, out, out but you cannot have in out in, this is not going to happen, you cannot have out, out, out and then in this is it.

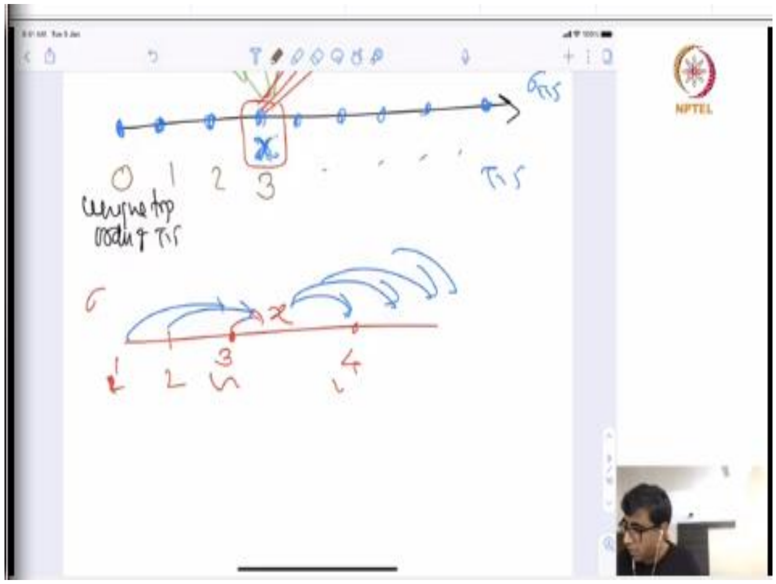
So, the moment you get out like the first vertex you get x to some vertex which is out, then to every other vertex you will get out only otherwise reduction rule 1 would have applied.

(Refer Slide Time: 35:35)



So, basically what happens is that after this is that you could have like this, you could have like this but then once you start to have this then to everybody else you will have this, so this is. And this is a property which holds for every vertex like there is only one change from in to out but there is no change from out to in for a particular vertex. Now what is that means? Look at this, so if a vertex x is not part of F then where this vertex x will?

(Refer Slide Time: 36:17)

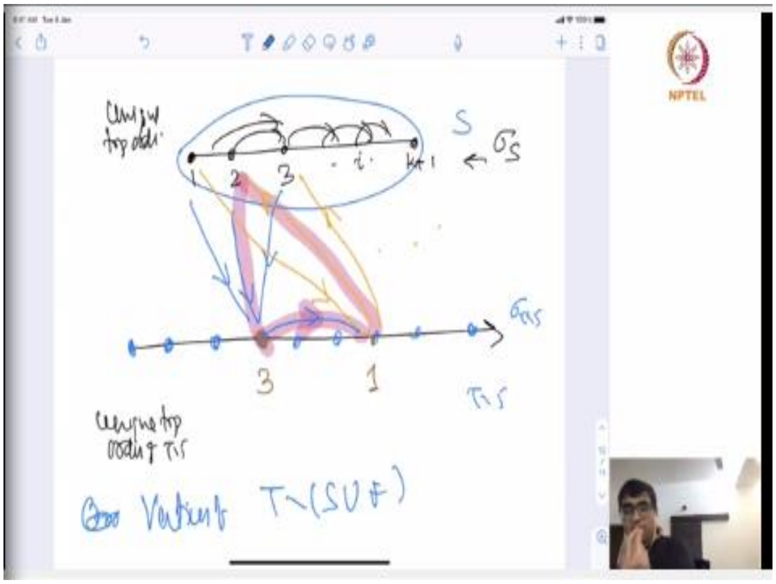


Then if I look at sigma then my vertex x is going to occur after 3 and it has to occur, so suppose this is 4 that is starts to go out like this, so suppose this is a vertex, so where x will? X is going into the location between 3 and 4 because then only all the arcs from 1. Because you know that 1, 2, 3, k is going to be the order in which they are because they are not going to be deleted.

Only then this and this will have arcs like this. So, either x is part of my tournament feedback vertex set or x is going to be in the location 3 to 4. So, for notice that for every vertex there is a one particular location where it can belong to. So, I am going to write for this particular vertex location, let us write it with brown, location 3. So, similarly for there could be, so a vertex could come here, why?

Because every vertex goes in or a vertex could come here, so I am going to call a vertex which can come here location 0, a vertex which can come here location 1 so on and so forth. So, I could have 0, 1, 2 so on and so forth. Now let us look at sigma T - S, look at, so this is a location. Let us ask ourselves this is the vertices which will go into the location, now let us ask ourselves that what happens? If I have 2 vertex.

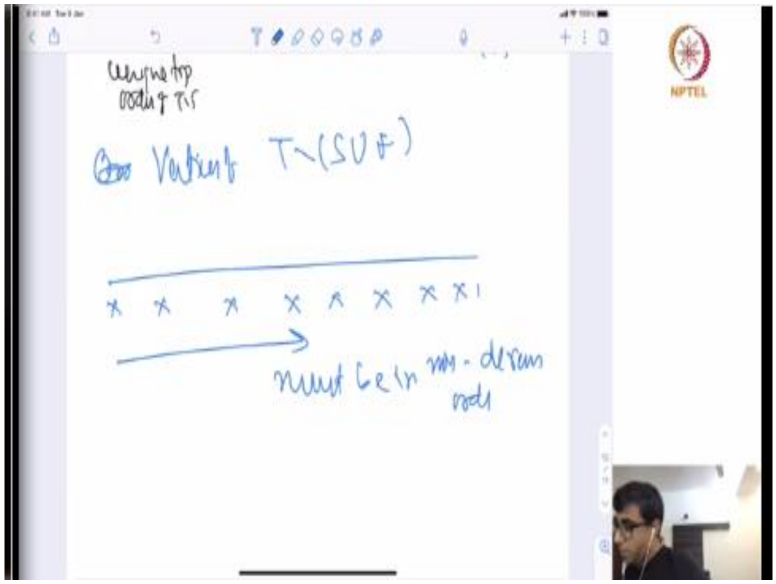
(Refer Slide Time: 38:18)



Let us ask ourselves what happens if I have 2 vertex he says I need to go to location 5 and this vertex says well I need to go to location, or say this vertex says I need to go to location 3 and this vertex says that I need to go to a location 1, what is the meaning of that? The meaning is that 1 has what is the property? That x says that I have to go to location 3, meaning x this particular vertex has an arc of the following nature; up to this let us draw of this nature.

And what does he say? Look, I need to go to location 1, it means he have arc and after that he has an arc of this nature, but we also know that we have an arc from 3 to 1. So, let us just try to see what happens? Now I can go 3 to 1, then I can go 1 to 2 and from 2 to 3 I can come back. So, if there are 2 vertex in this ordering, such that a vertex has a higher number and a vertex which has a lower number. If they are in a different, this thing then they will form a triangle. So, what does it mean?

(Refer Slide Time: 40:37)

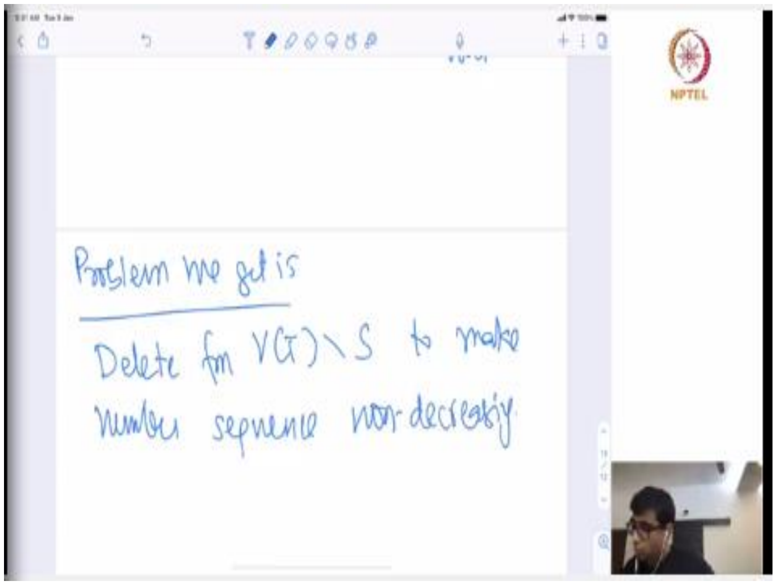


Like it means if we look at the vertices of T - S union F and look at the ordering of these vertices, look at the numbers of these vertices. They must be in non-decreasing order. Because if they are not in non-decreasing order then it will form a cycle, so what is this problem? Then our problem is nothing but so what we know and if you give me any non-decreasing order sequence here.

Like if you look at this sigma S - 1, if you give me a any sequence which is non-decreasing I can fit this into this without getting a triangle. Because look at all this non-decreasing number like put them where their location, because then what happens? Then everyone goes to their location and every arc goes from left to right, left to right.

So, notice that if I delete a subset of vertices which is my tournament feedback vertex set, what I get is a sequence of numbers which are in non-decreasing order. And if you give me any non-decreasing sequence and then complement forms a tournament feedback vertex set. So, what is this problem?

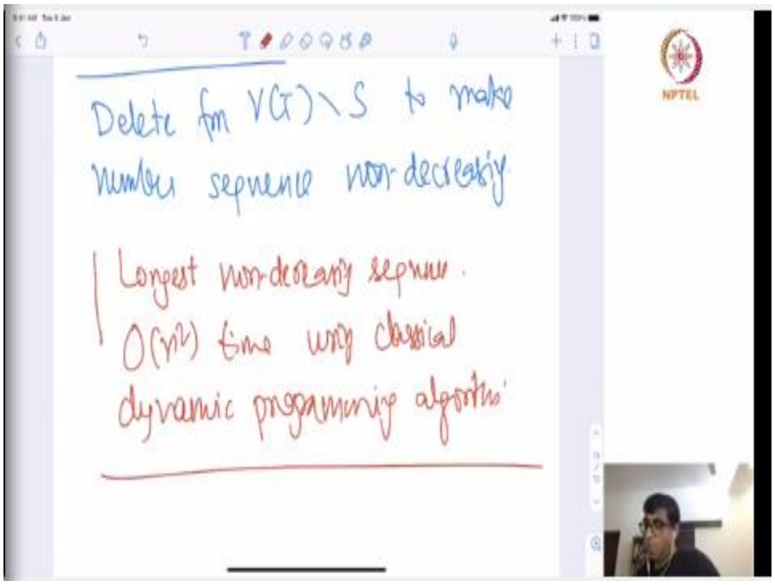
(Refer Slide Time: 41:58)



So, the problem we get is, what is the problem we get? Delete from vertex set of T - S to make number sequence non-decreasing. So, what did we say? Look if from this number sequence if 2 numbers survived, such that one has like higher number appears before lower number, then we get a triangle, this cannot happen. So, it is always the case that my number sequence is like monotonically like number sequences non decreasing.

And if you give me any non decreasing sequence, then I can fit this and get a tournament. So, basically my problem has become we get is delete from V T - S to make number sequence non-decreasing. But what is this problem? This is a very famous problem which you have done in your class.

(Refer Slide Time: 43:12)



This is nothing but longest non-decreasing sequence, and this can be done in order n square time I think, using classical dynamic programming. So, let us just go back and recap what we did, so we had to solve this disjoint compression problem. So, first we made sure that tournament induced an S did not form a cycle, then we said ok, fine it has a unique topological ordering, then we put this.

We found the unique topological ordering of T - S and our main observation was that after we delete vertices we basically are merging these 2 we are merging sigma S and sigma T - S in a way. Such that they are like the relative orderings are preserved and every arc goes from left to right, this is what we observe from here. That if we look at that combined if you after deleting F if we look at the combined what you call topological ordering.

Then if I restrict myself to S the topological ordering is preserved of, if I restrict myself to T - F - S it is preserved. And then we first applied one simple reduction rule that look if there exist a cycle that contains 2 vertices from S, if there is a cycle that contains 2 vertices from here. Then I know if there is a triangle that contains 2 vertices from S, then there are only 3 vertices.

And 2 you are forbidden to be picked into your solution then you have to pick the last one. So, this is what this reduction rule implied. After we apply this reduction rule exhaustively we get into a problem with the following property. So, what we did? I said look at a vertex x if you are either you are deleted, if you are not deleted what is the place that you have what is the location with respect to sigma S that you can belong to.

So, we saw that look the great property happened after reduction rule 1, that for any vertex like if I start to follow from 1 to k + 1. Then there exactly one place unique place where it shifts from being in neighbours to out neighbours and this series continues. So, it could be that everything x is just out neighbour 2, then what is the location he can belong to? Then this guy can belong to the only place which is 0 because x cannot be placed anywhere else without violating the topological ordering constraint given by sigma S.

So, there is a unique place where x could belong to and notice that like and I write 0, 1, 2, 3 based on which location can you belong to. Now notice one thing because even here they have to maintain the constraint. So, notice that if I could if all these number 0, 0, 1, 2, 3 all like if this was an increasing sequence, then if you would have placed these vertices in the order in which they appear in sigma T - S in these locations.

Then together they form a sequence where every arc goes from left to right but this may not be true. But we notice that hey, hey look, look, look if we delete the tournament feedback vertex set indeed these numbers must be in increasing sequence why, because if they are not in increasing sequence then we can form a triangle, great.

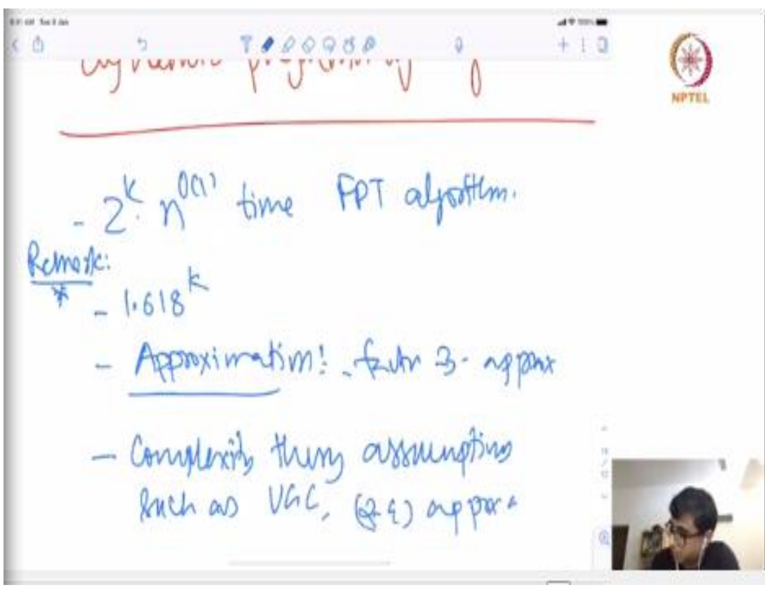
So, basically we have reduced our problem to find a number sequence as large as possible or to find a number sequence that contains all the vertices of T - S but k of them. And that problem is nothing but a longest non-decreasing sequence of numbers, and that we can do it in order n square time using classical dynamic programming or maybe n cube n cube for sure, I think n square is possible.

So, that should be all for this. So, you notice that the disjointness constraint actually gives us so much structure that several times we can actually solve the problem in poly time. For vertex cover it was pretty easy actually, so vertex cover was even more easier because all we did for vertex cover is that hey all we did is the graph induced an S independent if not say no otherwise take all the neighbours outside.

And then for 2 number and feedback vertex set we reduce the problem to finding longest non-decreasing numbers. So, towards that we first define what is the notion of directed graphs? What is the notion of dag? What is a notion of topological ordering? And we also prove this very nice property that tournament has a directed cycle, if and only if it has a directed triangle.

And we utilize this property to constrain ourselves to look at only directed triangle and not worry about any directed cycle. Because the moment we are able to hit all directed cycles or obstruct all directed triangles we are able to hit or obstruct all directed cycles. And that immediately gave us 3 power k simple branching algorithm. And but using iterative compression we are able to design for disjoint version a poly time algorithm.

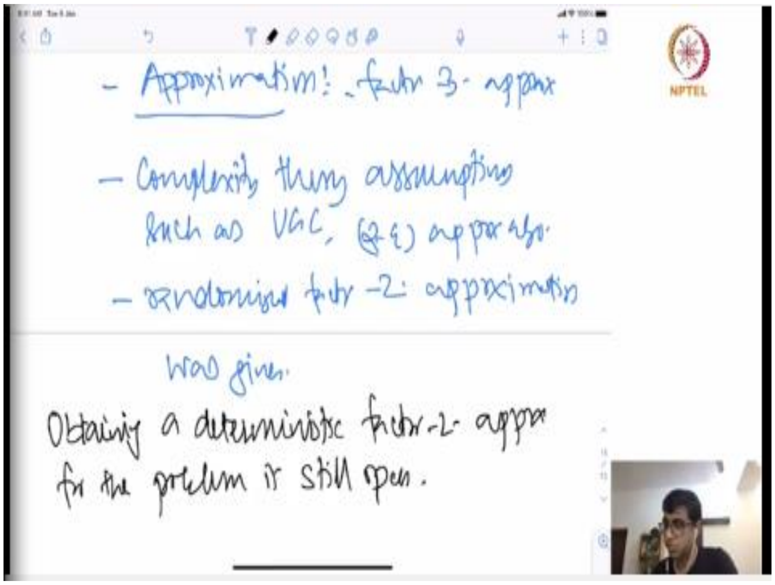
(Refer Slide Time: 49:09)



Which immediately implies 2 power k n to the power O of 1 time FPT algorithm. Some remark, not very important but it is an important remark. There is an algorithm faster than 2 power k 1 I think 1.61 to 8 to the power k which was obtained using some very nice clever trick. And for approximation a simple approximation algorithm for this problem is that you take the triangles and delete it, take fine and the triangles delete it and that gives you a factor 3 approximation.

And but it is also can prove that you cannot have under some complexity theory assumptions such as unique game conjecture, that you cannot have 2 - epsilon approx algorithm.

(Refer Slide Time: 50:31)



And only recently in 2020 a randomized factor 2 approximation was given. Obtaining a deterministic factor 2 approximation for the problem is still open. And other algorithm is this factor 2 approximation is quite interesting and simple and it is like just divide and conquer or merge short. So, if someone is interested can write to me and I can give you a pointer to.

But these are just small set of remarks that has nothing to do with the lecture but just to give you a fulfillment of what is known about these problems. With this let us conclude this lecture and we will continue our foray into iterative compression a little bit more in 2 more lectures, see you then bye, bye.