

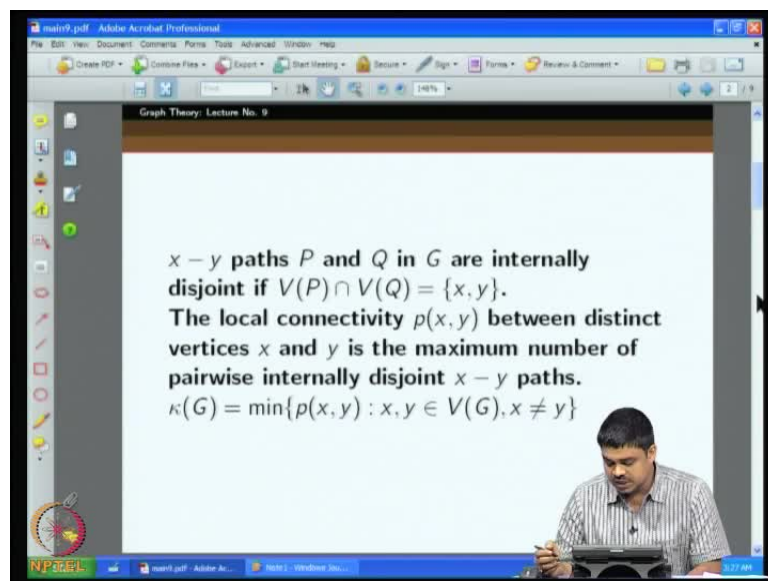
Graph Theory
Prof. L. Sunil Chandran
Computer Science and Automation
Indian Institute of Science Bangalore

Module No. # 02

Lecture No. # 09

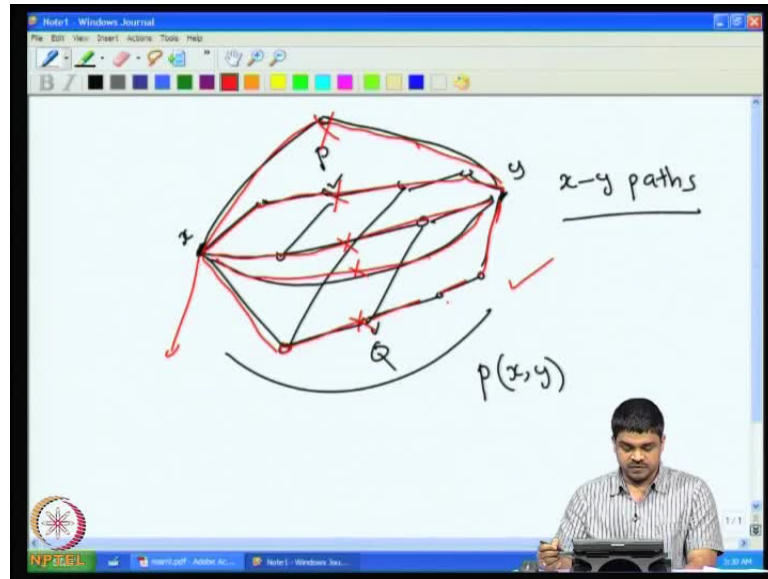
Connectivity: 2-connected and 3- connected graphs

(Refer Slide Time: 00:20)



Welcome to the ninth lecture of graph theory. Today, we will look at the concept of connectivity in graphs. There are two kinds of connectivity, we are going to discuss both of them. First, it is about vertex connectivity, the second, edge connectivity. So, intuitively, connectivity is something about **whether the given graph** how well the given graph is connected. So, we need to understand what it means. Say, the most natural way to define it, probably is this. Consider 2 vertices x and y and then, we can say 2 paths from x to y , I call it x - y paths, P and Q in G are internally disjoint, if they share only the end vertices.

(Refer Slide Time: 01:33)



For instance, if these are the 2 vertices x and y and suppose, this is one path P between x and y ; this is P and here is another path Q between the same 2 vertices x and y , say that these two x - y paths are internally vertex disjoint, if they share only the beginning and then vertices. That means this and this; also, this will not be in this path and any other vertices will not be in this path.

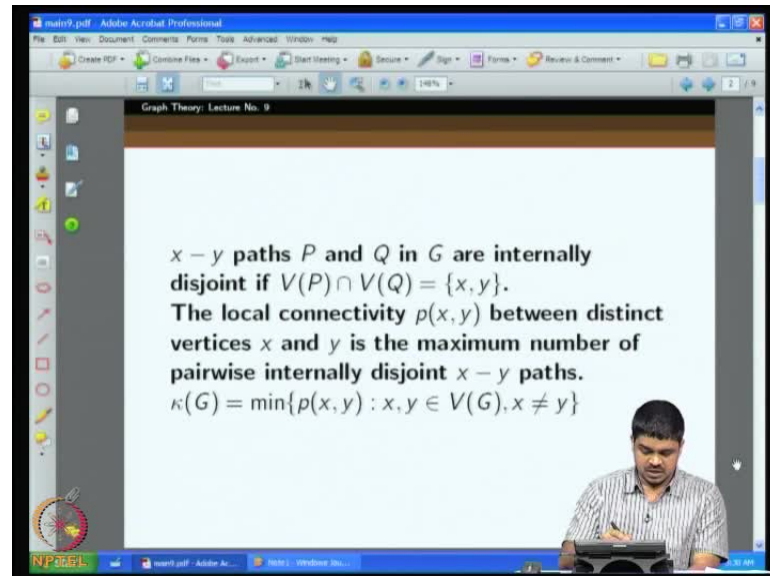
The local connectivity between x and y . So, we can call it P of x comma y is defined as the maximum possible number of internally vertex disjoint paths between x and y . For instance, here in this graph suppose, I have some more paths; suppose, I have several connections here, may be even some connection like this. So, how many internally vertex disjoint paths can we see?

It is possible there are several paths, but then if you are looking about internally vertex disjoint paths, I can take this one, then I can take say this one and I can take this one, I can take this edge itself; **it is a internally vertex** it is internally vertex disjoint with other paths I have already selected in this one and this is all internally vertex disjoint. I got 1, 2, 3, 4, 5. I cannot take more than this.

So, this is the local connectivity between x and y . The graph can be much complicated. There can be other vertices, but it may so happen that **if you cut all these paths so then** if you take all these paths, then any other path has to share some vertices with none of

these paths; that is what it means. **So, there would not be** If you take one more path then it the entire collection together will not be internally with vertex disjoint.

(Refer Slide Time: 04:03)

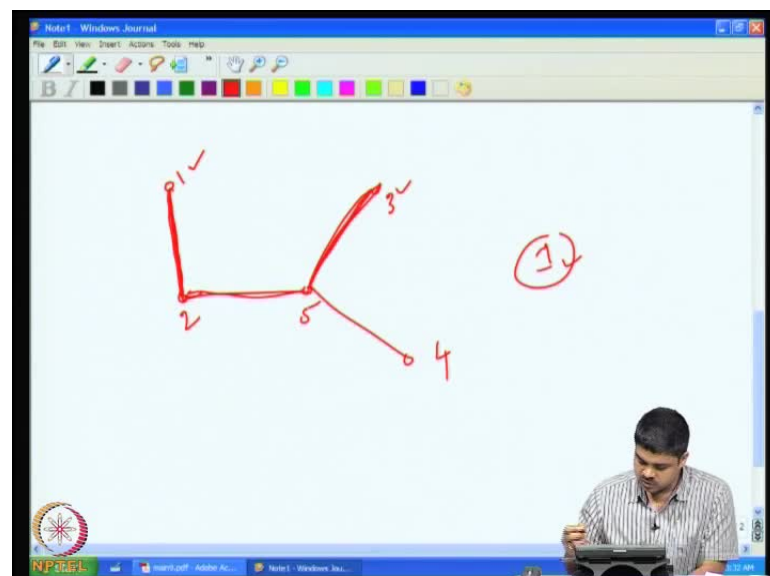


The screenshot shows a presentation slide with the following text:

$x - y$ paths P and Q in G are internally disjoint if $V(P) \cap V(Q) = \{x, y\}$.
The local connectivity $\rho(x, y)$ between distinct vertices x and y is the maximum number of pairwise internally disjoint $x - y$ paths.
 $\kappa(G) = \min\{\rho(x, y) : x, y \in V(G), x \neq y\}$

Now, what is the connectivity of a graph? You can have several pairs of vertices x, y in the graph and then for every pair, you have a so called local connectivity. The local connectivity between x and y is the maximum number of pair wise internally disjoint $x - y$ paths. Say among all the pairs, you look for the minimum local connectivity; that is going to be the connectivity of the graph; this makes sense.

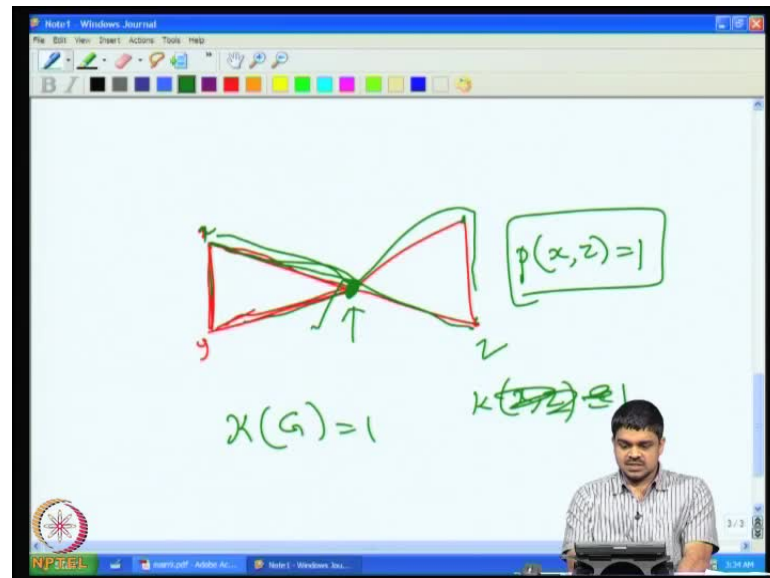
(Refer Slide Time: 04:45)



The screenshot shows a drawing application with a graph. The graph has five vertices labeled 1, 2, 3, 4, and 5. Vertex 1 is at the top left, vertex 2 is below it, vertex 5 is to the right of vertex 2, vertex 3 is above vertex 5, and vertex 4 is below vertex 5. There are edges connecting (1,2), (2,5), (5,3), and (5,4). A circled '1' is drawn to the right of the graph.

You can take some examples. For instance, let us look at this graph example. Here I can take this is 1, 2, 3, 4; between 1 and 2, there is only one internally distributed disjoint path, nothing more you can take. Like that 2 and this is 5, 2 and 5, 3 and 5, 4 and 5, everything is 1 and between 1 and 3, you can get only one internally vertex disjoint path - namely this one.

(Refer Slide Time: 05:44)



So, any pair, if you consider there is only one internally vertex disjoint path and therefore, the minimum overall pairs, if the local connectivity of that x - y pair, if you minimize over all pairs x - y, then you will get only one here for this graph.

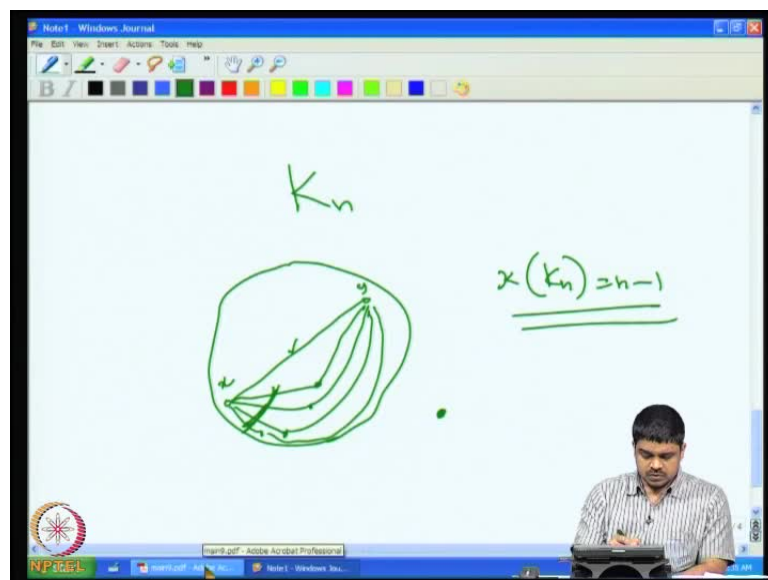
Here, the connectivity of this graph is going to be one. Now, let us take a little more complicated example, more complicated example than the earlier one. It is still simple, but here we can see that if I want between x and y, what is the local connectivity; there is one path here, there is another path I can take. Any other path I try to take, for instance, if I try to come, use this thing, I will have to go via this and then I have already used this one; so, then they will not be internally vertex disjoint. So, I cannot add anything more these are the only two things.

So, like that between this pair also, I have 2 internally vertex disjoint paths, but on the other hand, if I take this x and this z, then I have this path I can get one and any other one because that one has to go through this middle one or the any other one will have to cut through, go through the same vertex. So, it will share an internal vertex. So, it will not be

internally vertex disjoint. $\kappa(P, z)$, the local connectivity of P, z is going to be one. So, when you minimize over all pairs, then the local connectivity has to be less than or equal to 1, but then we know that it cannot be 0 because between any two pairs, there is at least one vertex.

So, we use κ to denote the connectivity of the graph. Not this, this one. This is κ , the local connectivity; κ is the connectivity of the graph, when you minimize over all pairs.

(Refer Slide Time: 07:47)



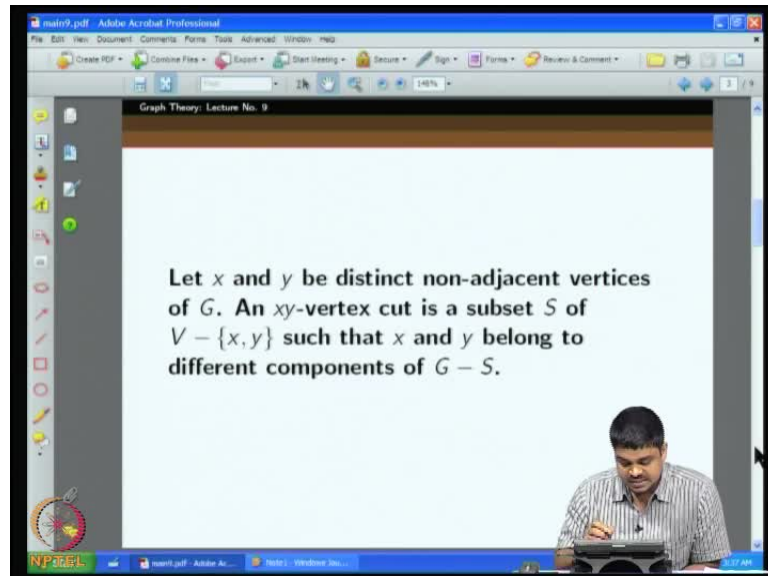
So, this is the most natural notion of this thing. Now, you should understand that when I considered 2 vertices x and x , same vertex then you cannot define this thing. So, we do not worry about x and y being same.

So, consider a complete graph K_n . Now, what will be the connectivity of K_n by our definition here? You can take any 2 vertices, x and y , here. You know there is one path between them and also you have total n minus two other edges going out of it. These edges reach several different vertices. From any of them, I can come back to here with this. Is not it?

So, therefore, the total is 1 here and this n minus 2, total n minus 1. So, this is the thing. If you have a trivial vertex, just one vertex, you can take it as 1 connected and

connectivity as 1. It is by convention **in fact there is** because anyway here, our definition that minimize over all pairs does not make sense. So, there is only one vertex here.

(Refer Slide Time: 08:48)

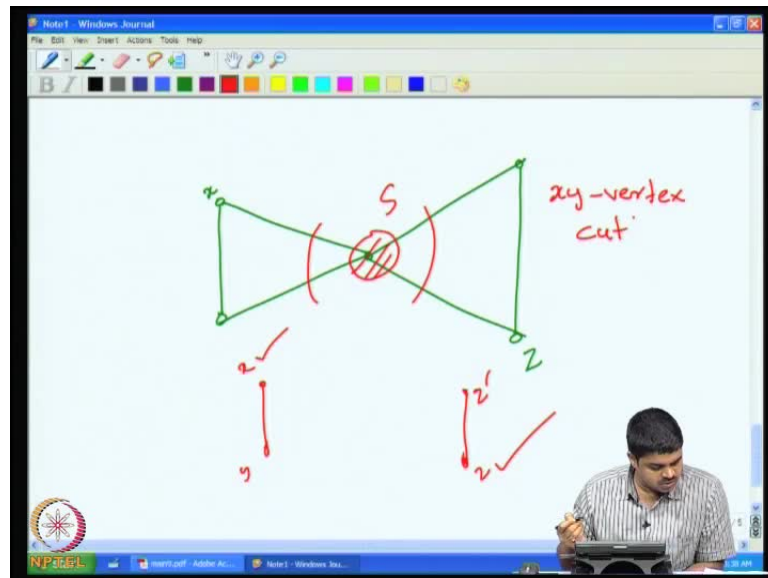


So, now, that is one intuitive definition of connectivity, which is like what we will think, when we hear the word connectivity. It is trying to capture our intuitive notion of what connectivity should mean.

Now, here is a slightly non intuitive way of doing it. Here, it says let x and y be distinct non adjacent vertices of G . So, here in the graph, I pick up 2 vertices with the condition that they are non-adjacent; there is no edge between them. Now, you try to disconnect x from y in the graph by removing some vertices other than x and y . Outside x and y we remove some vertices and try to make sure that in the remaining graph, there is no path from x to y anymore.

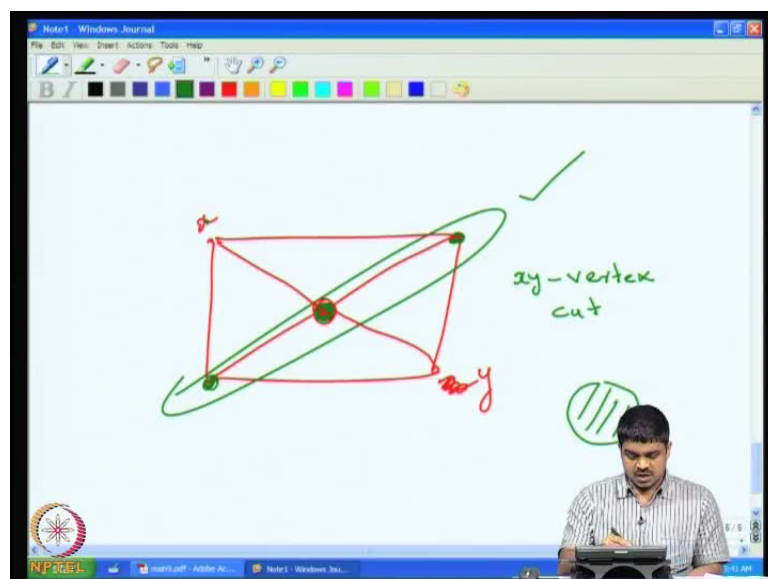
So, then I ask what is the minimum number of vertices, you have to remove to do that. Such a collection of vertices by removing which, you can make sure **there are at** that in the remaining graph, we have x and one of the components of G minus x , y and y in another component. Here xy vertex cut is called such a vertex, say such a subset of V minus x, y . So, we are not allowed to take x and y in the subset S . Such a subset **x, y** S is called an xy vertex cut. So, an xy vertex cut is a subset S of V minus x, y such that x and y belong to different components of G minus S .

(Refer Slide Time: 10:49)



We can look at some examples. For instance, here, let us take our graph which we clearly earlier. Here, suppose, if you want to they take two non adjacent pair that means you should not take suppose if I pick up x , I should take say, z here. Now, to disconnect x from y , what you can do is remove this set S ; then in the resulting graph so, nothing will be there. So, the resulting graph after removing S will look like this. x , y and then z , say, z dash. So, x is one of the components, z is a different component. You have no path starting from x and reaching z in the resulting component. So, this is such a vertex - xy vertex cut.

(Refer Slide Time: 11:51)



I can consider another example here. So, let us see. Consider this graph, where in this graph, if I want to disconnect say x from z ; this is an x ; x and z , take it as x , y and so then definitely I have to disconnect this vertex. So, I mark it with green; this vertex is to be disconnected and then this is to be disconnected, this is to be disconnected. So, these 3 vertices have to be disconnected, if I want to disconnect x from y . Therefore, this is a xy vertex cut here and its cardinality is three.

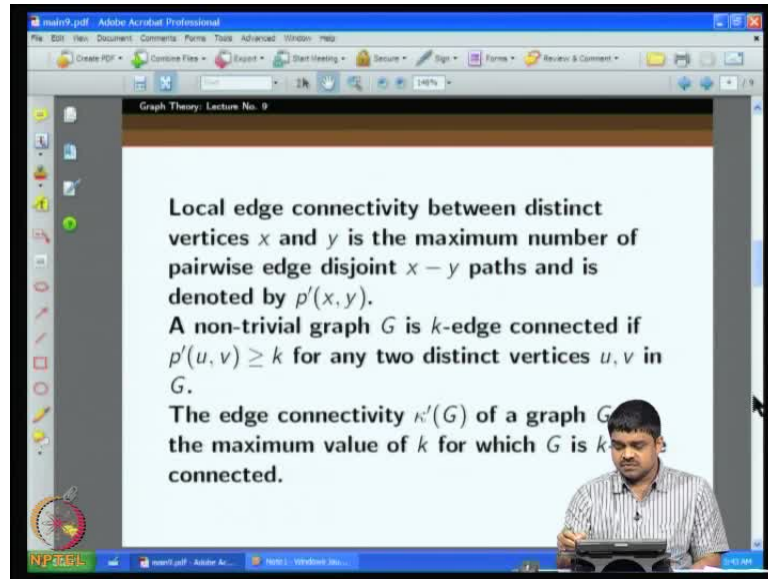
So, this xy vertex cut, the cardinality of the minimum xy vertex cut and over all possible x , y will give you the minimum number of vertices that we have to remove so that the graph gets disconnected. So, for instance, if the graph gets disconnected, in the remaining graph, there should be at least one pair such that x is on one side and y is on the other side and naturally, those 2 vertices are non-adjacent in the original graph.

Now, to find out the minimum number of vertices that we have to remove from the graph so that the graph after removal of this vertex is a disconnected graph. What we can do is over all pairs x comma y which are non-adjacent, we can find the cardinality of the minimum xy cut and this minimum will be the minimum number of vertices required to disconnect the graph.

So, this number is another way of defining connectivity because finally, we are talking about removing the minimum number of vertices so that the graph gets disconnected.

But these two notions look a little different at the first look, but it so happens that they are the same. We will later study a theorem called Menger's theorem, which states that both these notions are equivalent.

(Refer Slide Time: 14:36)



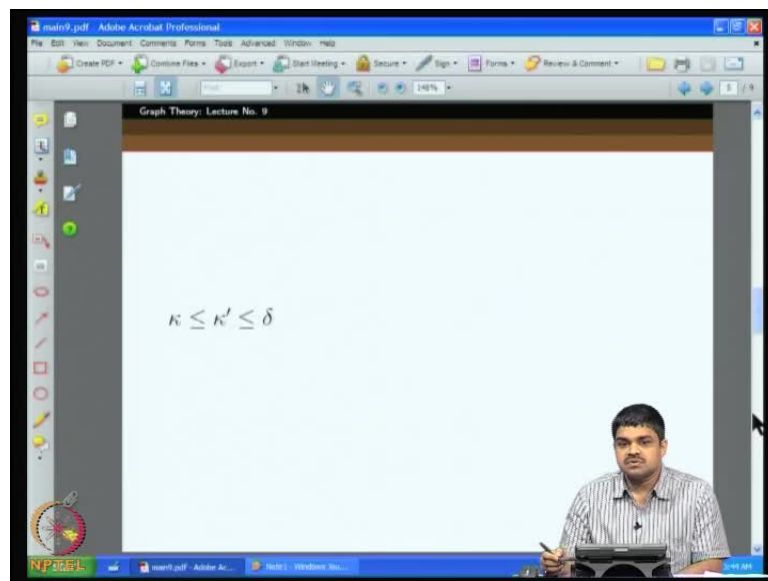
Now, just before moving further, I will mention that there is something called edge connectivity also. **Here, we were** Naturally, it is very parallel. So, we can define edge connectivity in two different ways, like we did for vertex connectivity. One is by considering any 2 vertices x, y - x and y and consider the number of pair wise edge disjoint x, y paths.

Now, there is nothing like internally edge disjoint here because you know, you are given 2 vertices, the edge set they should be totally disjoint; in fact, the edge set of the two different paths $x - y$ paths should be disjoint. That is, like that we have to get the maximum number of pairwise edge disjoint $x-y$ paths and this number is the local edge connectivity and this is denoted like P dash x, y . Remember P of x, y denoted the local x, y vertex connectivity, but p dash x, y will correspondingly denote the edge connectivity between x and y .

We will say that a non-trivial graph G is k edge connected, if for every pair u, v , the local connectivity is greater than or equal to k , which means that you have at least k edge disjoint paths between any two pair of vertices. **The edge connectivity k dash of G** So, the κ of G was used for vertex connectivity; now, edge connectivity is denoted by κ dash of G and κ dash of G of a graph G is the maximum value of k for which, G is k edge connected.

In other words, in the graph you can consider all pairs x, y , we can consider the minimum local edge connectivity and that is over all the pairs and that is going to be this value just like in the edge vertex version. Similarly, we can have a different notion like how many edges are to be removed from the graph so that the graph gets disconnected. This is another notion to define the edge connectivity and then, we will show that both these notions are equivalent. So, we will come back to these notions later again.

(Refer Slide Time: 17:30)



Now, we can quickly see one interesting observation. The vertex connectivity is always less than equal to edge connectivity and that is going to be less than equal to minimum degree because you see the vertex, **if you remove** if κ' happens to be the minimum number of edges that is to be removed so that the graph gets disconnected, you can remove one end point from each edge, although those edges will anyway get removed.

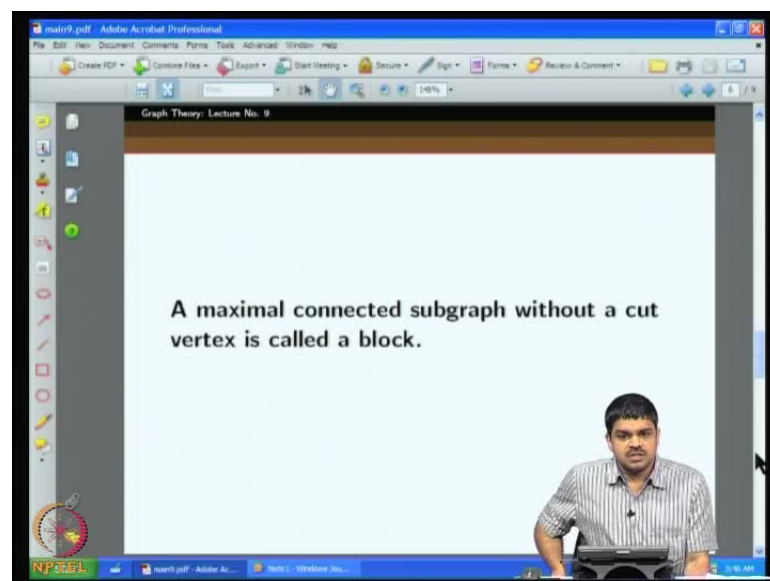
Therefore, the vertex connectivity is going to be at most the edge connectivity. What you do is you consider a collection of edges by removing which the graph get disconnected. Now, you can remove one end point of each edge; that means one vertex for one edge. **So, naturally the graph gets disconnected with this thing, this, if you remove these vertices because all the edges in the edge cut, that is what it is called; like vertex cut, we have edge cut; so, this will disconnect the graph.**

So, naturally that that gave us a vertex cut also and because the κ being the minimum cardinality of a vertex cut, we can infer that κ is less than equal to κ'

dash and that is less than equal to minimum degree. Why because if minimum degree is the smallest degree of a graph, the number of neighbours of a vertex, smallest number of neighbours of a vertex over all vertices. It is because we are talking about simple graph. So, they are also equal to the number of edges that is to be removed, that is connecting the vertex to the rest of the graph.

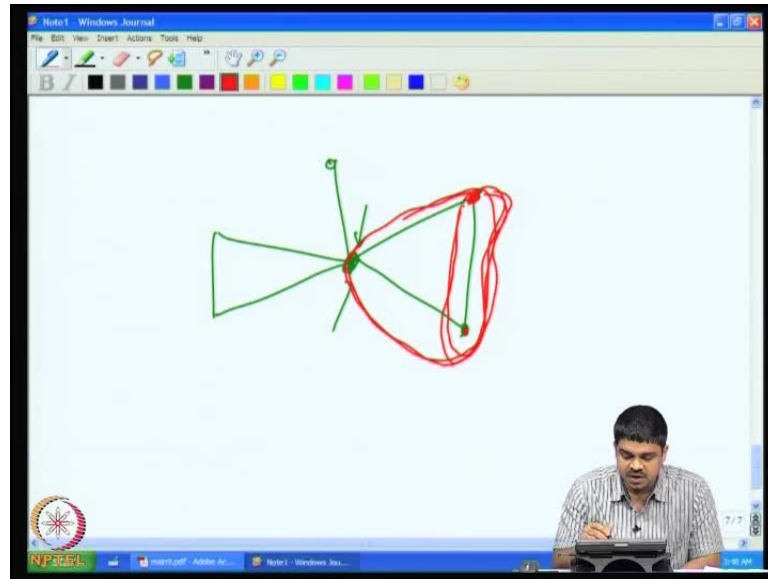
So, if you remove these edges, definitely that vertex gets separated from the remaining part of the graph. So, naturally that is an edge cut and therefore, the actual minimum edge cut is going to be even less than equal to that. It can even be less than that. So, κ dash is less than equal to δ . So, this is one quick observation that we can make about the **edge connectivity**, vertex connectivity, edge connectivity and minimum degree of a graph.

(Refer Slide Time: 19:54)



Now, before going further and looking at statements like Menger's theorem, where we say that the **two equivalent** two notions of connectivity that we discussed both for the vertex version and the edge version; two different types of connectivity discussed: one was based on the number of path between vertices, the other was based on the number of vertices to remove from the graph so that the graph gets disconnected. So, later we want to prove one theorem called Menger's theorem, which states that both this notions are equivalent, but before that we will just look at the 2 connected graphs and how will they look like; this is our aim.

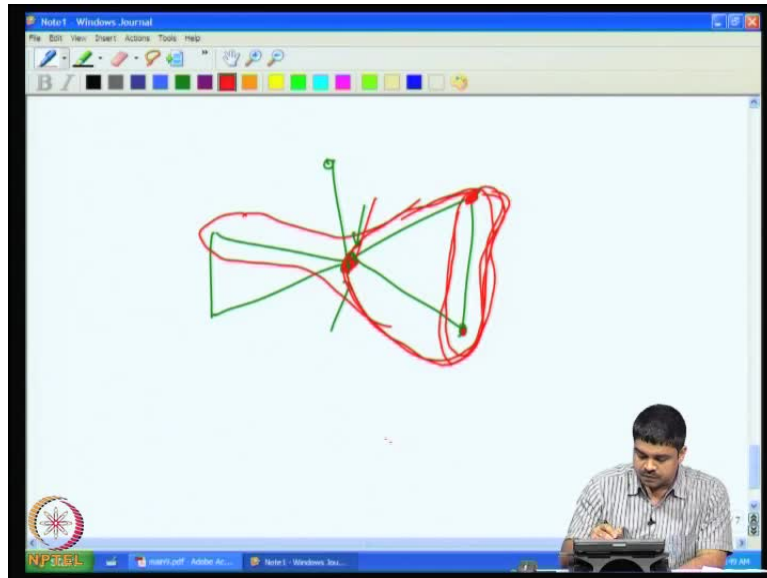
(Refer Slide Time: 21:13)



Here is one definition. So, you know that in a graph, a maximal connected sub graph is going to be a component. **So, like that you can take** In a connected graph, you can look for a maximal connected sub graph without a cut vertex; it is called a block. What is a cut vertex? So, in our examples here, you remember, we have this is a cut vertex. This is one vertex by removing which the graph gets disconnected.

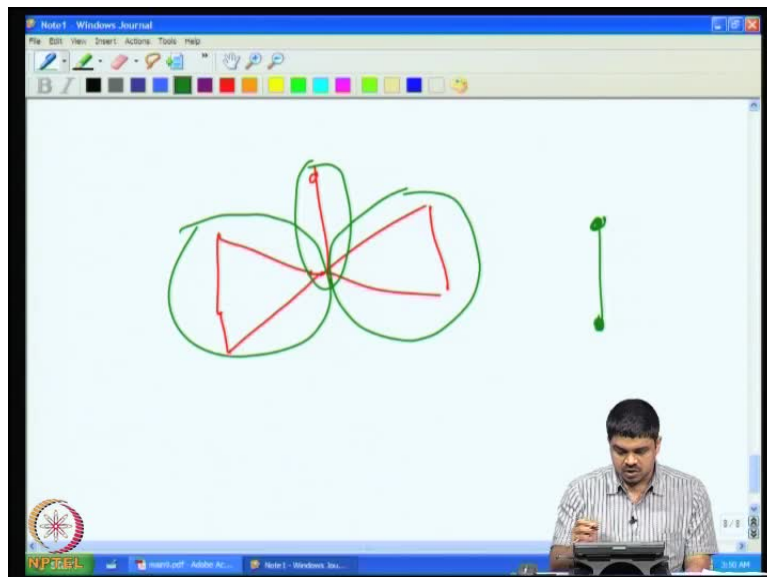
So, suppose, you are looking for a maximal and here, suppose, I add this also here. You look at this one; this is without a cut vertex because by removing this thing, the graph does not get disconnected or removing this thing, the graph does not get disconnected; for instance, this alone if I consider.

(Refer Slide Time: 22:15)



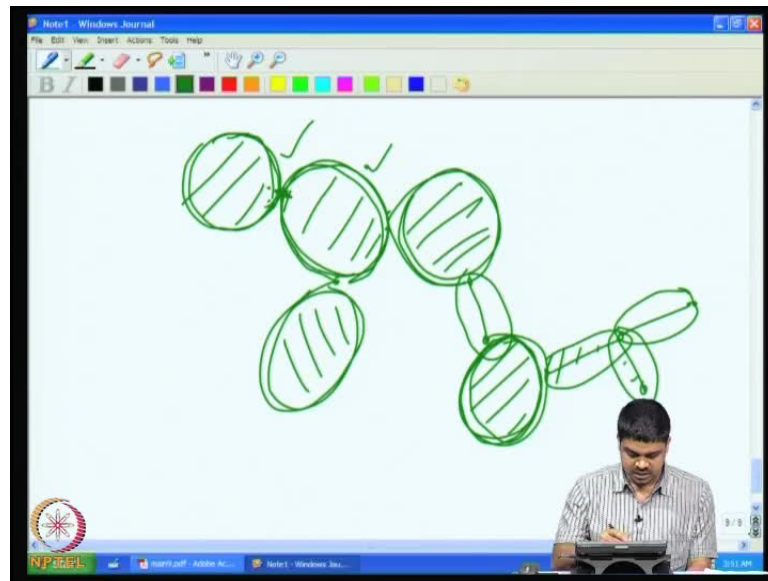
Now, I can add something more. I can take this thing; this is a maximal connected sub graph without a cut vertex because there is no cut vertex in this triangle, but now if I take a little more, for instance, if I take this much, then naturally, this will be a cut vertex. We cannot add anymore vertices there because this cut vertex comes.

(Refer Slide Time: 22:34)



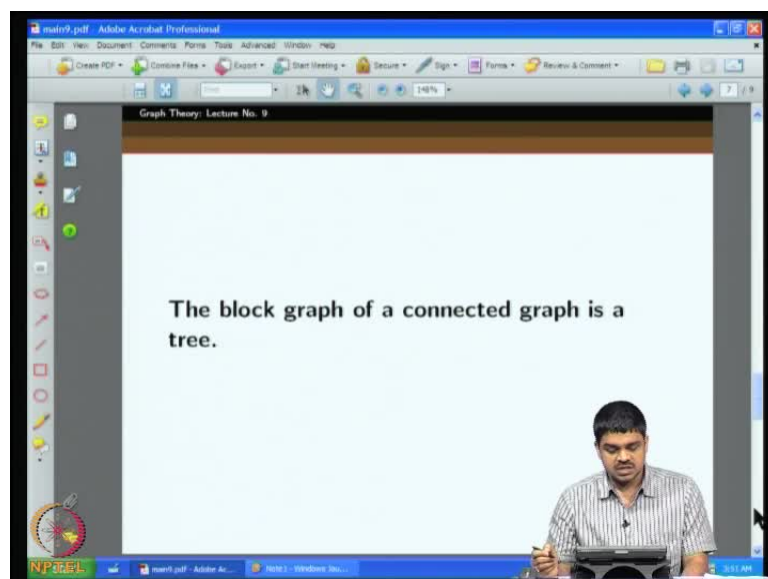
So, here in this graph, the maximal connected sub graphs without a cut vertex happens to be, in this graph, I mark it with this one and then this one and this one; this is also one, for instance, this is a edge, if you remove one vertex, this is still connected. Therefore, they are not cut vertices.

(Refer Slide Time: 23:20)

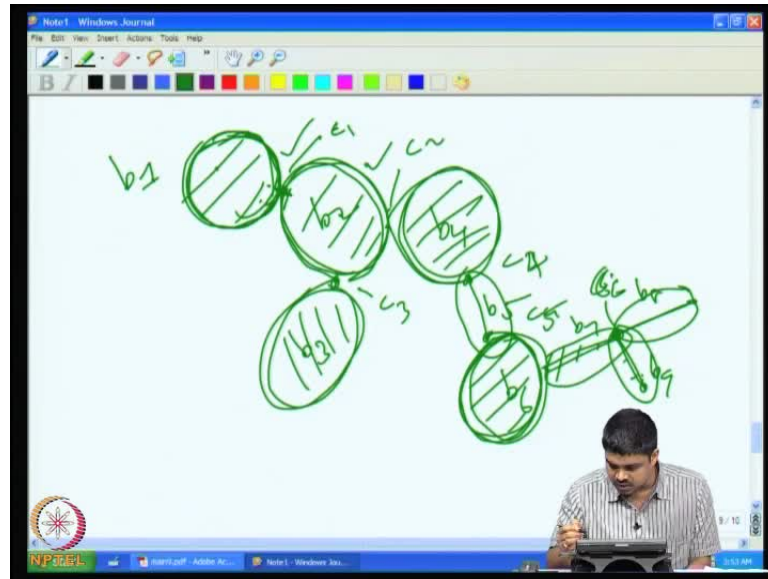


So, this is the way we can decompose the given connected graph into several maximal connected sub graphs without cut vertex. **So, it may look like little** Typically, the structure of a connected graph can be like this; some can be like this; sometimes you can get some edges, you can get some edges. It can be something like this. So, these are all maximal things, these edges I marked because they are important. So, here look in this thing; this is a 2 connected sub graph; this connectivity of this thing is two and this is also 2 connected because within that there are no cut vertices and within that there is no cut vertex.

(Refer Slide Time: 24:40)



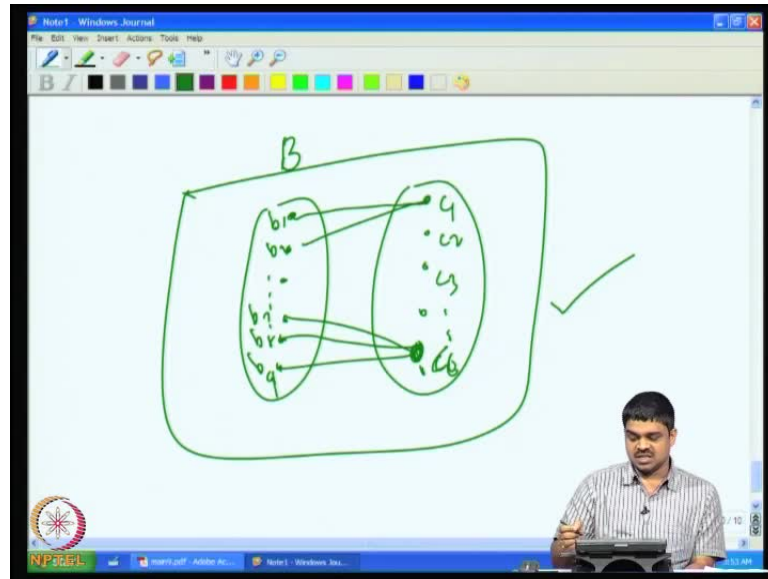
(Refer Slide Time: 23:20)



So, these things are one of the blocks and see this will be the picture we will get. For instance, if you look at the graph abstract, if you are looking from a distance, this is the look. Now, you can see some kind of tree structure here, among the blocks. While this you consider some super node and then the edges here is represented by the cut vertex on which these two blocks are pasted.

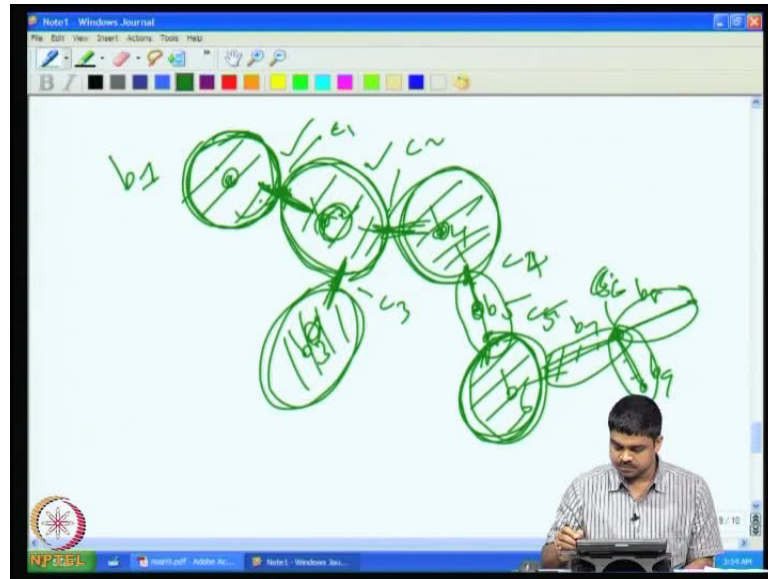
This idea is expressed in the next statement, which says that the block graph of a connected graph is a tree. What do I mean by a block graph? **In a block graph** Block graph is a bipartite graph. What do I do? Each of the blocks will be taken as a vertex. This is say b_1 the block 1, this is block 2, this is block 3, this is block 4, this is block 5, this is block 6, this is block 7, this is block 8, this is block 9 and each of the blocks will become vertices and yes, these are the blocks.

(Refer Slide Time: 26:27)



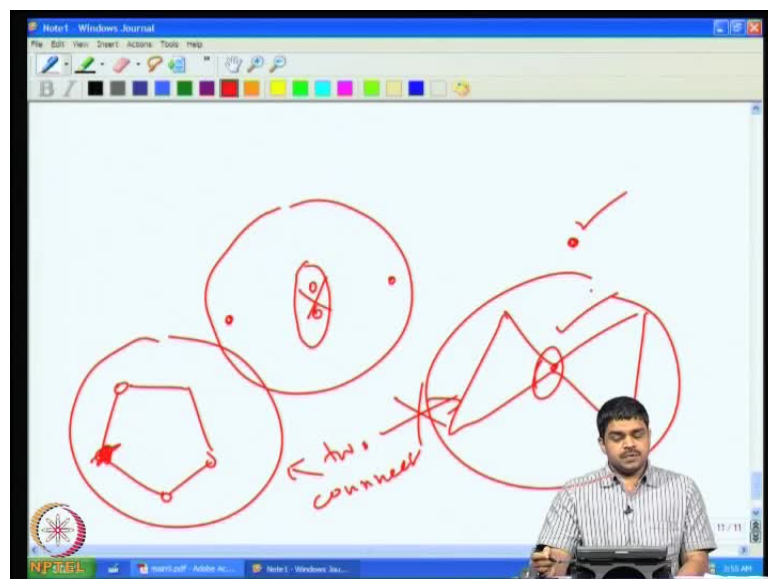
Then I can make a bipartite graph like this by putting b_1, b_2 up to b_9 here, the blocks and then on this side, I can collect the cut vertices of the graph and put. Which are the cut vertices of this graph here? c_1 , this is c_2 , this is c_3 , here this one and here this is c_4 , this is c_5 and this is c_6 ; like that, the cut vertices can be taken. Then, this is c_1, c_2, c_3, c_4, c_6 and this others and then, now, if b_1 contains c_1 , then you put this edge. Here, you know c_1 is contained in b_1 and b_2 ; therefore, both b_1 and b_2 will be connected to this. Similarly, let us look at this one; this is c_6 , c_6 is contained in b_7, b_8 and b_9 . So, b_7, b_8 and b_9 - all of them will be connected to c_6 ; this will be in the c_6 . This graph is called a block graph, you know, the blocks and cut vertices; blocks on one side of the bipartite graph, cut vertex on the other graph. **and then the cut vertex** If a cut vertex c_i is inside a block b_j , then b_i, b_j is connected. So, we say that this is a tree.

(Refer Slide Time: 27:02)



It is rather easy to prove, see you can more or less see it from this picture. For instance, if I had this cut vertices become the edges and this becomes the nodes, if I can see the tree structure here. **So tree structure here somehow** So, intuitively it is very obvious, but we will not prove it, but I will leave it to the students to prove it. (Refer Slide Time: 27:29)
Now this is the structure of connected graphs.

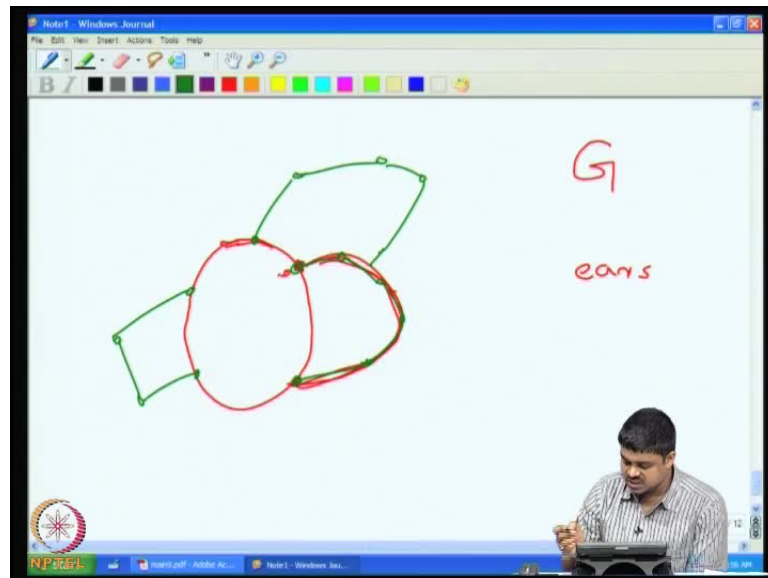
(Refer Slide Time: 27:41)



Now, what about 2 connected graphs? For instance, there are no cut vertices in it and also it is 2 connected; 2 connected in the sense that **between a** to disconnect the graph,

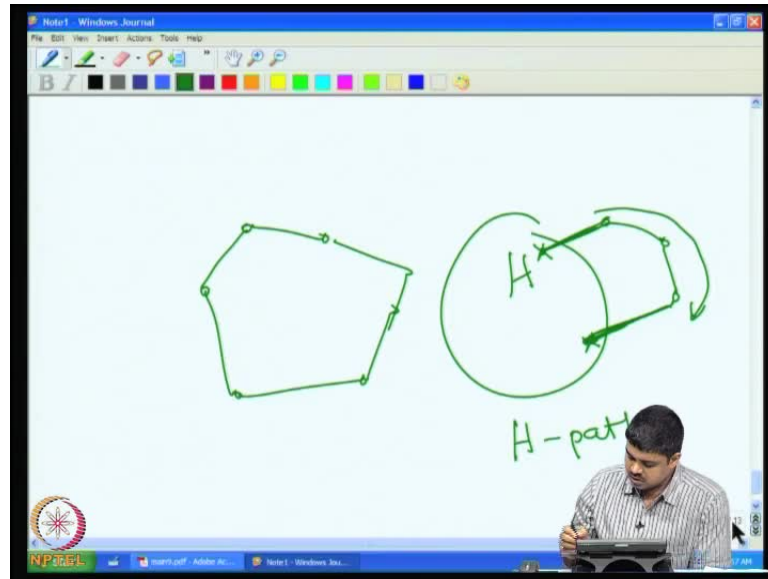
we need to remove at least 2 vertices from the graph. By removing one vertex, you cannot disconnect. For instance, this graph is not 2 connected because this vertex, if you remove then the graph gets disconnected. While this graph is 2 connected because you remove any one vertex from this thing, it will not get disconnected; a path remains here. Therefore, it is 2 connected graph. This is 2 connected, but this one is not 2 connected.

(Refer Slide Time: 28:46)



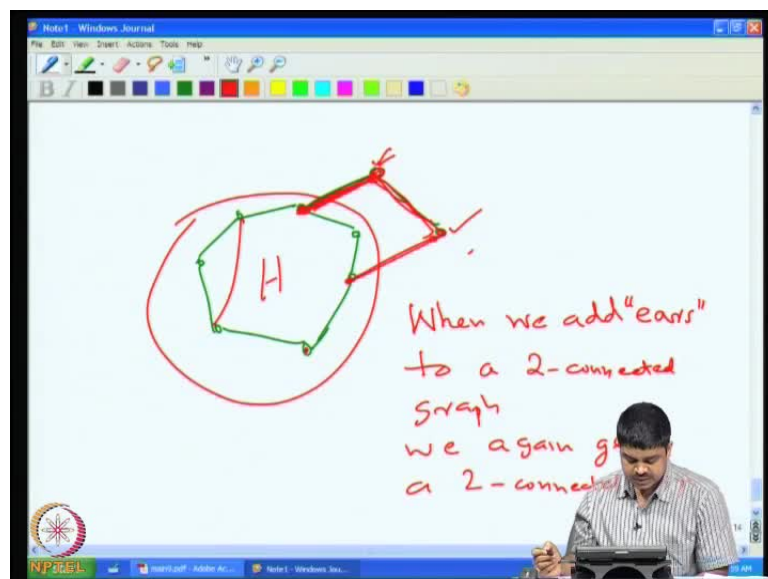
So, see this graphs have an interesting way to construct There is an interesting way to construct a 2 connected graph. What do you do? We can essentially construct by Give me any 2 connected graphs, what we can do is we can start from a cycle in it and then keep adding something called ears. What do you call ears? You start something from this already existing graph and then initially, it will be a cycle. So, this is an ear; that means, it can be just an edge starting at a vertex of this thing and reaching another vertex or it can be a something like this a path such that only the last 2 vertices are on this thing and the rest is just a simple path. Now this is a new graph and on top of this thing, you can add an ear like this say, may be like this or it can be like this. So, this new graph can be added.

(Refer Slide Time: 29:58)



Can we see that if I start with a cycle and keep adding ears, sometimes, the ears to an already existing graph H . This is an already existing graph and an ear is also called an H path, in the sense that it is a path outside H with just the end vertices placed in H . This is one **suability**, the end vertices placed in H . So, this is called an H path. So, we will keep saying ears rather than saying H path; H path is also a word, which is commonly used.

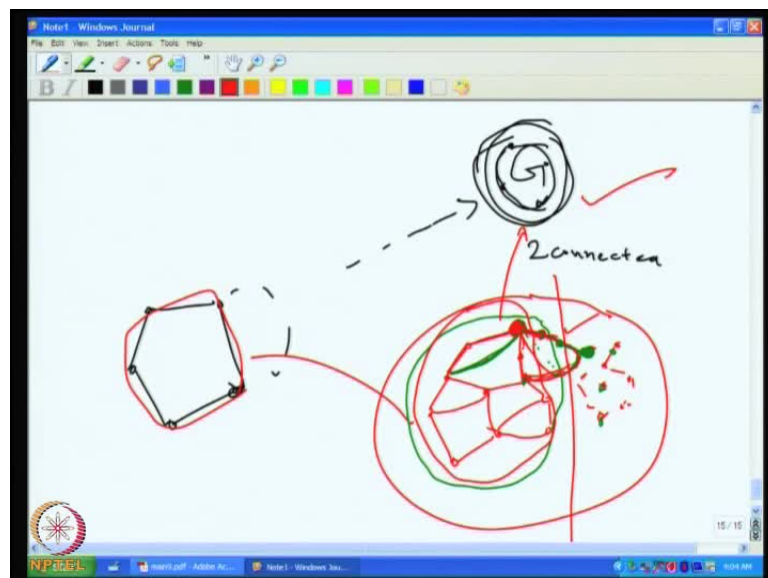
(Refer Slide Time: 30:41)



Now, suppose, I start with a cycle and then keep adding ears, as I mentioned earlier then is it possible that all the time I will end with a 2 connected graph or is it possible that I

may lose my two connectivity in some point. So, it is very easy to verify that it will never happen because if I am adding an ear of this sort, then it is **became** just adding an extra edge; it is not going to trouble you, but if you are adding an ear of this sort then also, **you know that within these any 2 vertices,** this cannot be disconnected into two components by removing less than 2 vertices; **that only one vertices vertex** removal of one vertex will never disconnect this graph, but then suppose it is getting disconnected by removing one vertex, it should be that some of these vertex, this or this, that should go out, **but then which vertex should disconnect it because see this is a path. Any vertex if it is going into one connected component.** So, you have two different ways to enter this already existing graph H . Therefore, it is not possible to remove just one vertex and disconnect any of them from here. Therefore, by adding an ear, you are not going to lose two connectivity.

(Refer Slide Time: 33:01)



So by When we add ears to a 2 connected graph, we again get a 2 connected graph. This is very obvious, but what can probably happen is that **what can probably happen is see so** there can be some graph G . So, this is what we are bothered about. Suppose, some arbitrary graph G – 2 connected graph, but see by starting from a cycle and adding ears to it like that adding ears one after the other, what is the guarantee that we can make this thing this way. So, we will prove this thing.

So, first we will identify a cycle in it because it should have a cycle because it is 2 connected. For instance, any 2 vertices if you take, there will be a two internally vertex disjoint path and there should be a cycle in it; so, we start with a cycle.

Now, to the cycle we will add ears. For instance, see this graph G . We will look at, maybe this cycle comes here like this. Now, **from the ear means** I cannot arbitrarily add an ear, should the ear should be existing in this thing. So, in this graph, if I try to locate an ear and then add it to that and **then already existing graph, I will add** already constructed graph, I will add another ear from as long as this path is available in G , I will add it as an ear. **and then** Sometimes the ears can be like this, simple edges, I will keep adding. As long as I can add an ear, I will add it and at some point I will have to stop.

So, there are two reasons to stop either all the vertices and edges in G are already added here. So, that means I have already constructed G starting with my original cycle and then by adding ears. So, that is a good situation. That means I have shown that G can be constructed by this procedure. Starting with a simple cycle in G and by adding ears one after the other, we can construct G .

Another way is that you may just stop and then something may remain outside, some vertices and possibly some edges among them, whatever, some graph and some vertices and then is it possible that something like this remains. So, this is not, but no, nothing like that because if such a thing is not yet added, you can always add because it is an ear by itself, you can add. We are saying that if you can add a new ear, you will add it.

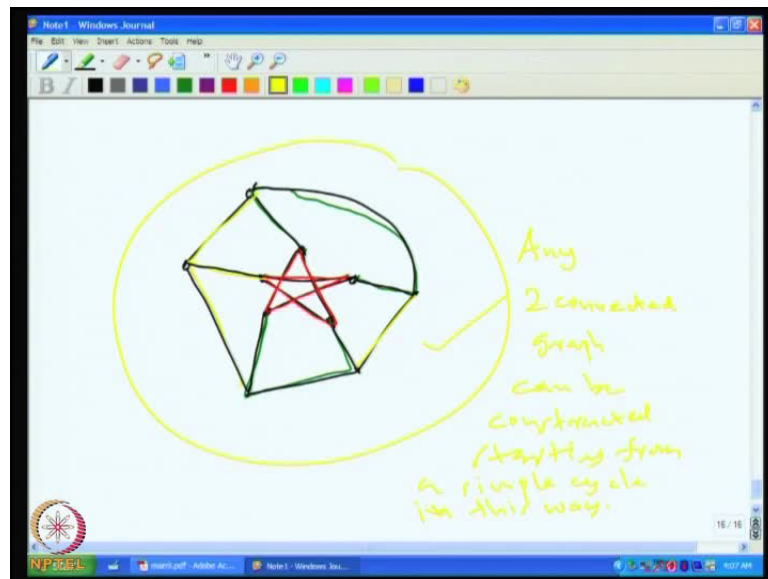
So, that means what we have already constructed is an induced sub graph of G , if not G . So, that means there are some vertices outside. Let us say, here is a vertex outside G and here is a vertex outside G and this G is definitely because of the 2 connected, there should be a path from here to here. There should be a path from here and then of case, it is not that all the paths which start from here goes this thing. There should be one more path starting from here and because otherwise if I remove this vertex, it will get disconnected. So, there should be two paths starting from here to here. So, this vertex to this thing and there can be some other vertices here of case, but then there should be 2 different paths because **otherwise what will happen is.**

More rigorously, what I can say is I will consider a vertex outside because it is connected there should be something which is directly adjacent and then from this thing, if I have

only this connection definitely by removing this vertex, I will disconnect it. So, there should be one more path; that one more path will come that takes some vertices outside and look at the first vertex it touches, the inner vertex it touches; it touches the vertex of this already constructed graph.

So, if it is this itself then again it is a cut vertex. That means if I remove this vertex, all these things will get removed. It should come and touch another place. That is going to be an ear. We can take it as an ear and add. So, our assumption that the graph, where we stopped, induced sub graph with which we stopped was one, where we could not make any progress; that is a wrong assumption. So, that is what it says, in other words.

(Refer Slide Time: 38:14)

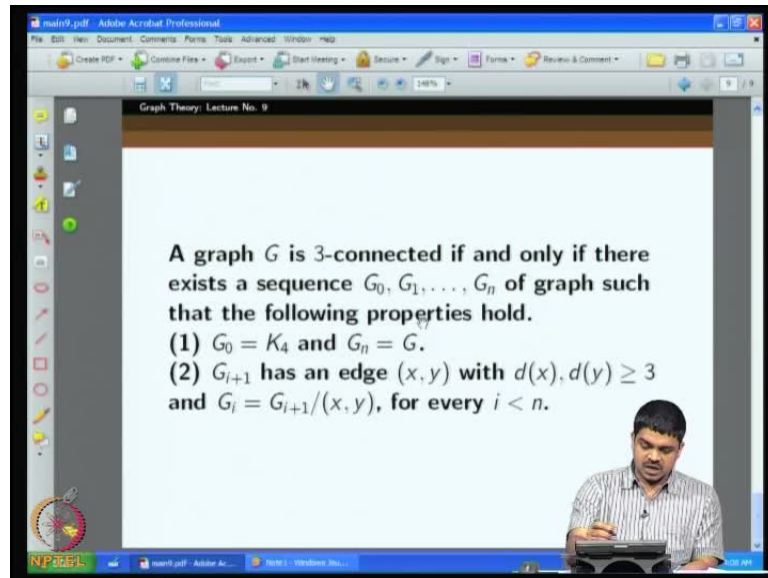


If you want to understand the ear decomposition of a Peterson graph, this is a Peterson graph like this. So, look at this graph, simple graph. It is a Peterson graph, if you want to look at ear decomposition. Now, I can start with a cycle. This cycle, this is a simple cycle and then you can add this ear, you can add this ear and then you can add this ear, then you can add say, the new ear, you can add this new ear and then the remaining ears are this edge, this edge. So, you could do this Peterson graph; construct it; this is the decomposition.

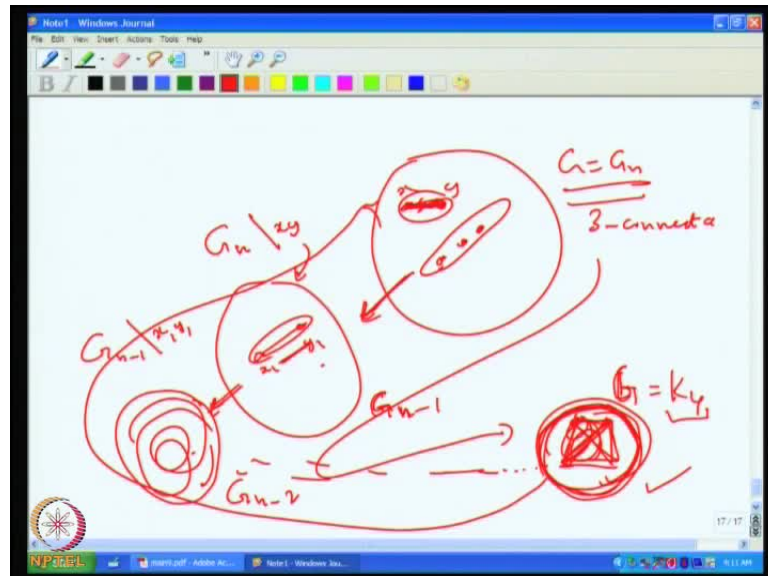
Our proof told that any 2 connected graph can be constructed starting from a simple cycle in this way; in this way means by adding ears one after the other. We will not get stuck. If you get stuck, there will be one vertex outside and there will be one vertex

which is adjacent to the already constructed graph outside and from that vertex there should be one more path coming back to the already constructed graph because of the two connectivity and this tree vertex the other end tree vertex cannot be the same vertex again because of the two connectivity and that will provide an ear; that is the idea.

(Refer Slide Time: 40:51)



(Refer Slide Time: 41:32)



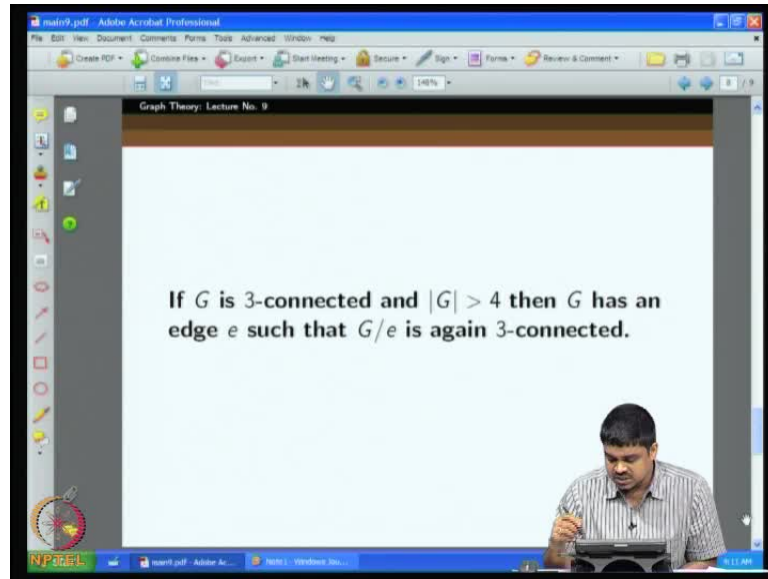
Now, what about 3 connectivity? So, 3 connected graphs also do we have some such theorem that starting from a simple graph, we can construct any 3 connected graphs. So, this theorem called Tutte's theorem; it says a graph is 3 connected if only if there exists a

sequence G_0, G_1, \dots, G_n of graphs such that the following properties hold. G_0 is equal to K_4 and G_n equal to G and G_{i+1} has an edge x, y with both the degrees of x and y greater than equal to 3 and from G_{i+1} , we can get G_i by contracting the edge x, y . So, such a sequence exists, but in other words, suppose, we are given the graph G . So, we say that it is 3 connected; that means to disconnect this graph, we need to remove at least 3 vertices - at least 3 vertices have to be removed, otherwise, it will not be disconnect at all.

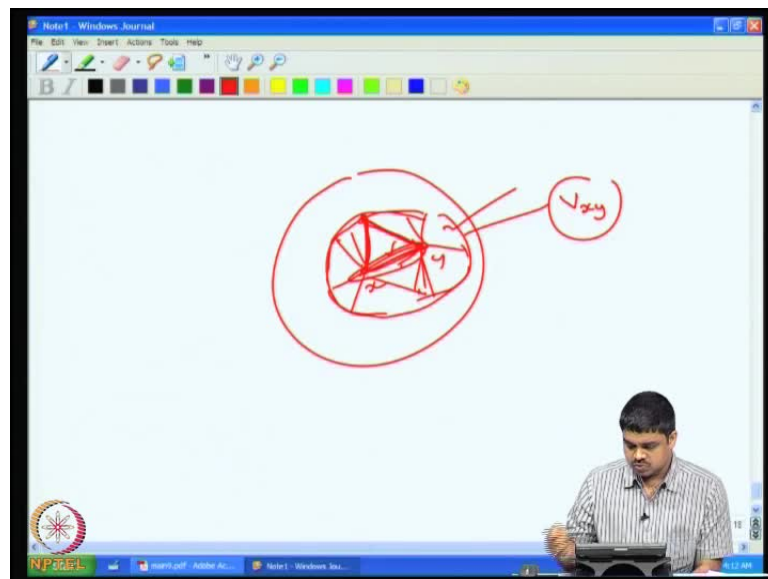
Now, in such a graph what we can do is we can find some edge such that it can be contracted and then you will get a smaller graph, which is again 3 connected. This is G_n , this is G_{n-1} . By contracting the vertex, number of vertices removed by 1, the point is we will be able to pick up a vertex in this edge, in this thing x, y such that by contracting x, y , we will end up a smaller graph G_{n-1} such that this is also 3 connected and here again, you will be able to find another vertex say x_1, y_1 such that when you contract this thing, we will end up with a graph smaller graph G_{n-2} , which is obtained by $G_{n-1} \setminus \{x_1, y_1\}$.

So, this graph will also be 3 connected. You can keep on doing it and finally, because every time you are reducing the vertices by 1, you end up with the beginning graph G , which is essentially K_4 equal to K_4 because why? To begin with, **we have** if there is 3 vertices only, then definitely, it is not 3 connected, it is only 2 connected. So, maximum because the triangle is only 2 connected. So, it should have at least 4 vertices and **you know it has to be** when it reaches this stage that it is only 4 vertices and because it is 3 connected, it has to be K_4 because smaller graphs are not 3 connected. So, also, if you remove one edge from K_4 , it is not 3 connected.

(Refer Slide Time: 44:28)



(Refer Slide Time: 44:58)



So, in K_4 we have to stop this thing. So, this is possible is what we are saying. The theorem says that any graph G which is 3 connected can be reduced to K_4 and this way by contracting an edge here, you will get a smaller graph contracting an edge here we will get a smaller graph like that you can reach K_4 .

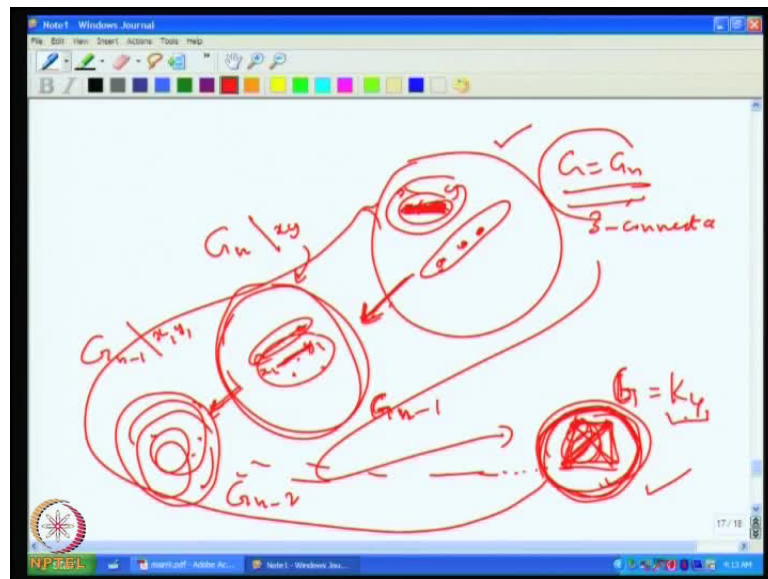
So, in other words, starting from K_4 , any 3 connected graph can be reached in a sequence like this. This is essentially what it says. Now, the key lemma we need to prove this thing is given here and it says, if G is a 3 connected graph and the number of vertices

and it has is more than 4, then G has an edge e such that G , the graph obtained by contracting a $G \bar{e}$ is again G .

See after the graph is obtained from G by contracting the edge e is again 3 connected. Remember, what is this contracting? So, this contracting essentially means if x, y is an edge and these are the neighbours of x and these are the neighbours of y , maybe some sharing is also there.

So, you make this one vertex and all the neighbours of x and y together will become the neighbours of this thing. If there is a common neighbour then multiple edges will come; that we will discard. So, essentially all the neighbours of x and y together will become the neighbour of the new vertex $V x y$. This is the contracted vertex, the vertex obtained after contraction.

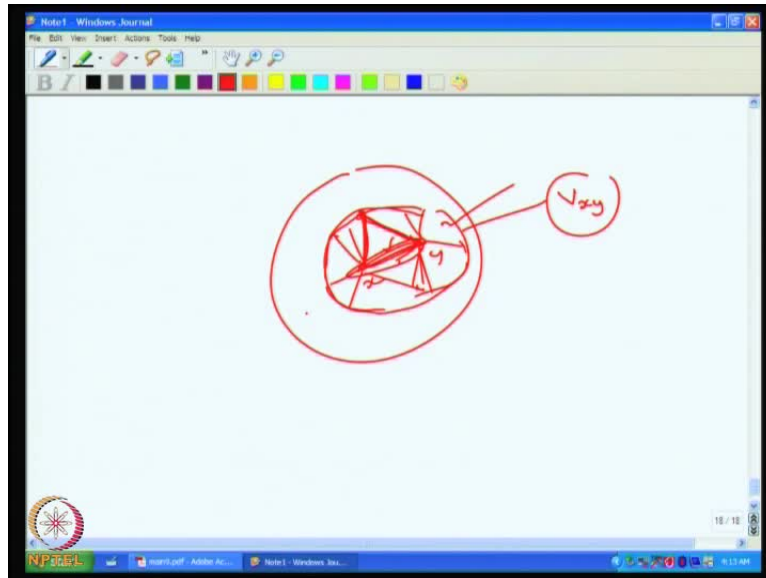
(Refer Slide Time: 45:40)



Now, we are going to say that As you can easily understand, to prove the previous statement at least one side, that means, given a 3 connected graph, you can reach K_4 by contracting edge. There is one edge here and then another edge here, another edge here. It is just to show that in any 3 connected graph, there is an edge that can be contracted so that the 3 connectivity is retained because this is a 3 connected graph, you will get such an edge if such a statement is true and then this is again 3 connected graph because we contracted an edge which will allow us to retain the 3 connectivity and this is again a 3 connectivity graph, there is another edge in it which when contracted will allow us to

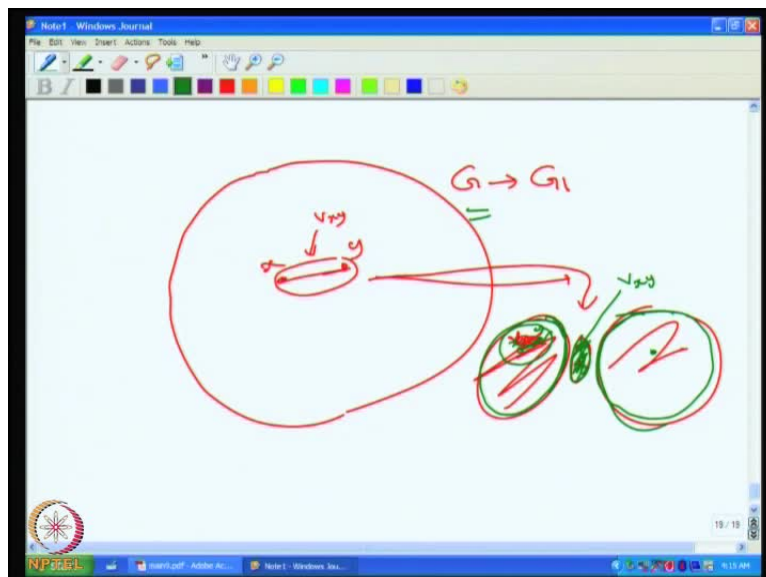
retain the 3 connectivity. So, we can contract it and what we get after that is again a 3 connected graph. So, this reduction should happen, reduction of number of vertices without losing the 3 connectedness.

(Refer Slide Time: 46:37)



So finally, when we reach K_4 , 4 vertices, it has to be a K_4 . That is all. Therefore, we only have to prove this that is, in a 3 connected graph, in any 3 connected graph there is at least one vertex which can be contracted without losing the property of 3 connectivity. How am I going to prove this thing?

(Refer Slide Time: 46:55)

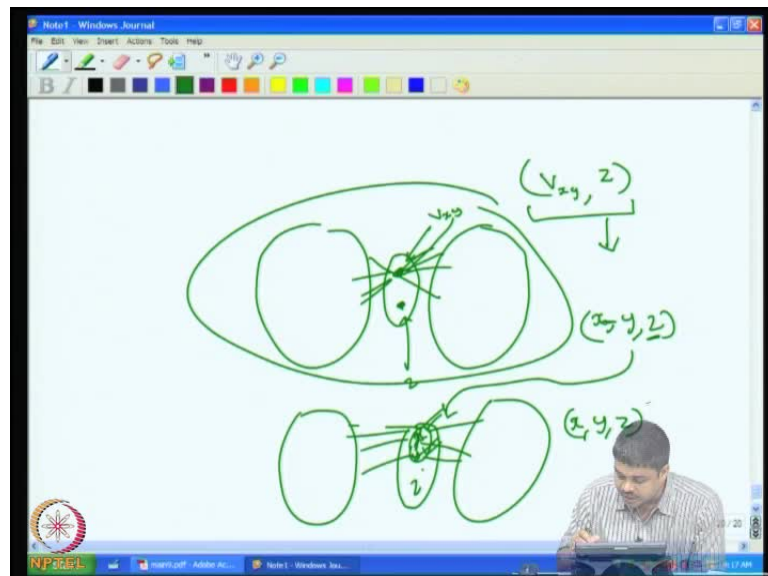


So, here this is the thing. You take this 3 connected graph G and then you can consider this edge x . Suppose, there is an edge x, y here. This edge x, y is if I suppose there is no edge x, y with the desired property. That means, any edge x, y if I take, if I contract it then what happens is, so **with this new** this I replace by $V_{x, y}$, the resulting graph G_1 that I get from G is such that its 3 connectivity is lost. That means, it gets a two separator - that means a vertex cut of cardinality two.

So, what does it mean? Suppose that means any edge, if I have any edge x, y , I will get some separator somewhere, some vertex separator with 2 vertices, just 2 vertices in it. This should be on one side; this should be on the other, after contracting this. **become**

But is it possible that this contracted vertex here but is not in the separator. It is here. Of course not because in that case, if I had done the reverse operation and then put back the edge x, y , then what will happen? This 2 separator, 2 length separator would have separated the original graph G itself. In the G also, this would work as a separator then you understand it because x and y after contracting it. If it so happens that there is this 2 separator then while before contracting because that is anyway here on one side. this separator would have separated that from the remaining part in anyway.

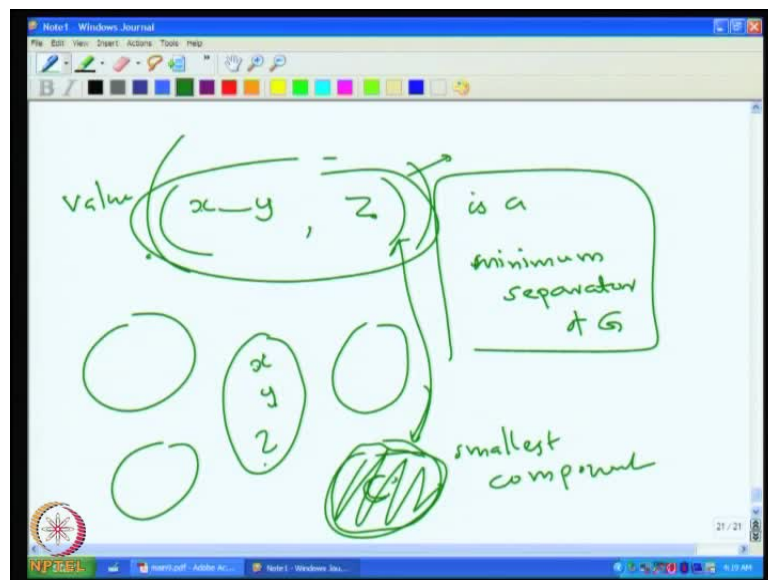
(Refer Slide Time: 49:07)



Therefore, neither can we have both x and y here. That means $V_{x, y}$ here or $V_{x, y}$ here then where can $V_{x, y}$ be? So, it can only be here $v_{x, y}$ can be only inside the separator so we can for any edge x, y when we contract the graph obtained will be such that it will

have a separator and that is a 2 separator and one of these vertices in the separator will be $V \times y$ and there will be another one let us say this vertex is called Z . So $V \times y, z$ is our separator. So, if I put this x and y back here. So, in the original graph we see x and z . These 3 vertices together is a separator for the original graph because if this separates, when this and this together can remove only those edges which this removes here because they are all adjacent to either this or this. So, if I remove x and y both, then the separation will happen even in the original graph. So, if $V \times y, z$ is a separator for this graph, after contraction, then x, y, z is a separator, 3 separator for the graph before the original graph G , the first graph itself.

(Refer Slide Time: 50:34)



Now, I will say that given any edge x, y I have z such that x, y, z is a separator of the graph. So, give me an edge x, y then I can find a z for it such that these three together is a minimum separator. It is not just separator, minimum separator. Why because it is a 3 connected graph minimum vertex cut; minimum separator means minimum vertex cut, minimum separator of G .

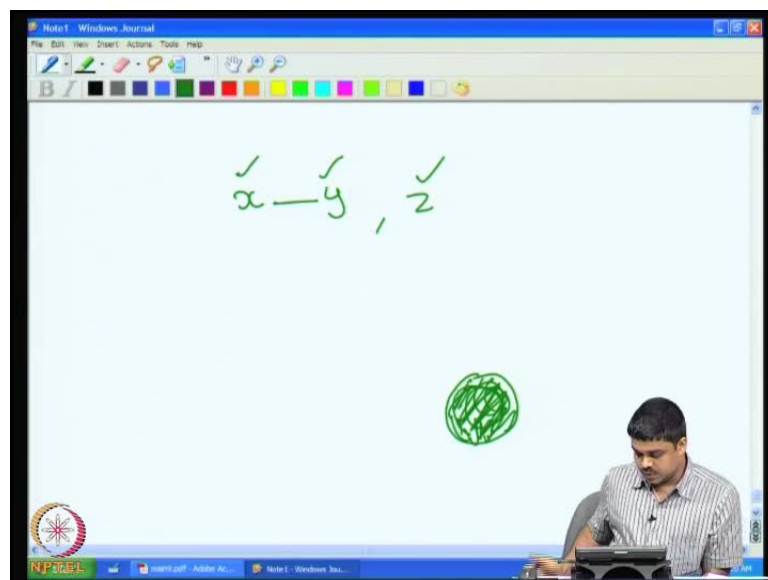
Now, this is for an edge, z is the partner of it in some sense and let say for each three tuple of this sort, let me assign a value. How do you compute the value? Now, I consider this separator x, y, z in the graph G and there will be several components.

So, for instance, there will be this component C also which happens to be the smallest component in G minus x, y, z . If I remove x, y, z the graph gets disconnected and among

the components which results I will look for the smallest component. The cardinality, the number of vertices in the smallest component will be the value of this one. I will say this is the value of this set, this x, y, z ; so, like that for each edge and its partner z , not that there is a unique z that I can get. So, you can consider the one, which will give you get you the minimum value if you want so.

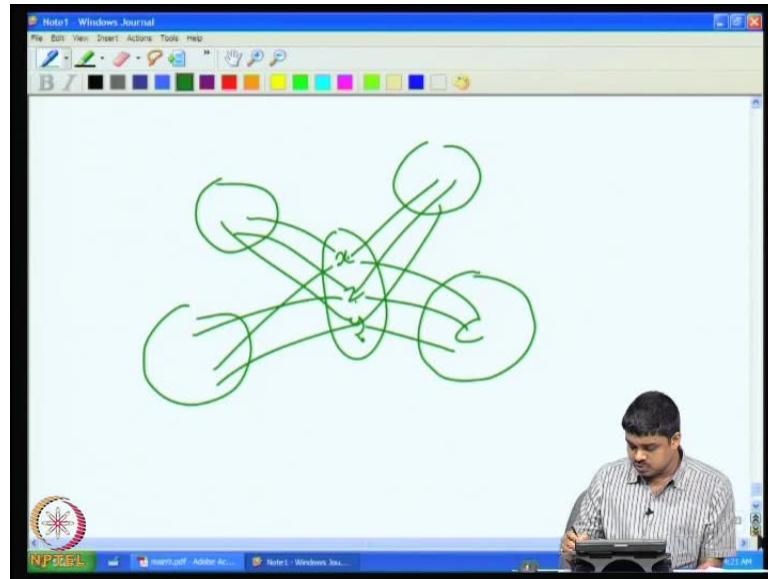
So, that way for every edge x, y , a partner is available and with respect to this x, y and its partner z together, we can evaluate its value by removing these 3 vertices in the graph and looking at the components and finding out the minimum cardinality component. The cardinality of the minimum component will be the value of that.

(Refer Slide Time: 53:00)

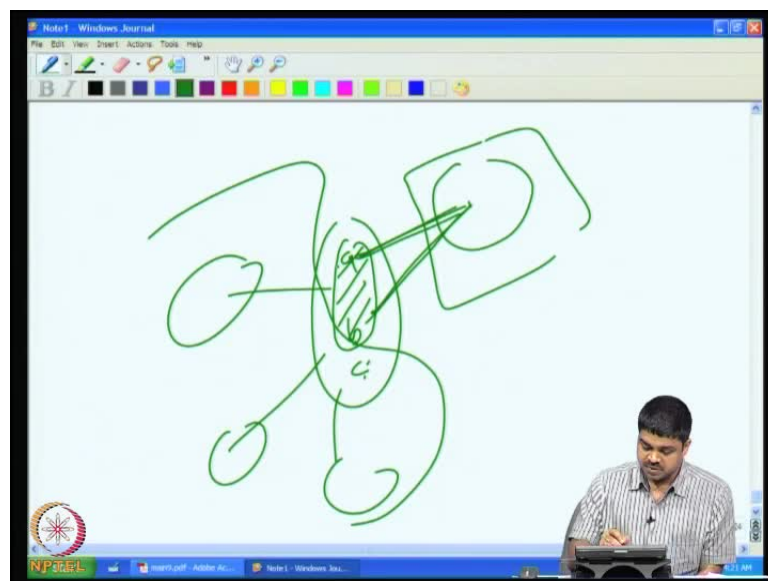


Now, among all possible edges x, y in the graph, I will look for the one which will achieve them minimum value; not that is unique. It may be possible that there are more than one edge, but I will pick up one such edge. From now on, when I say x, y , that is an edge and together with its partner. So, the value of it is going to be the minimum; that means, when I remove x, y and z from the graph, where x is a component C and then the cardinality of that component is such that if you remove **any other 2 edges** any other edge x dash y dash and its partner set z dash, any component which is produced will be greater than or equal to this **in cardinality**, in the size, in number of vertices.

(Refer Slide Time: 53:41)



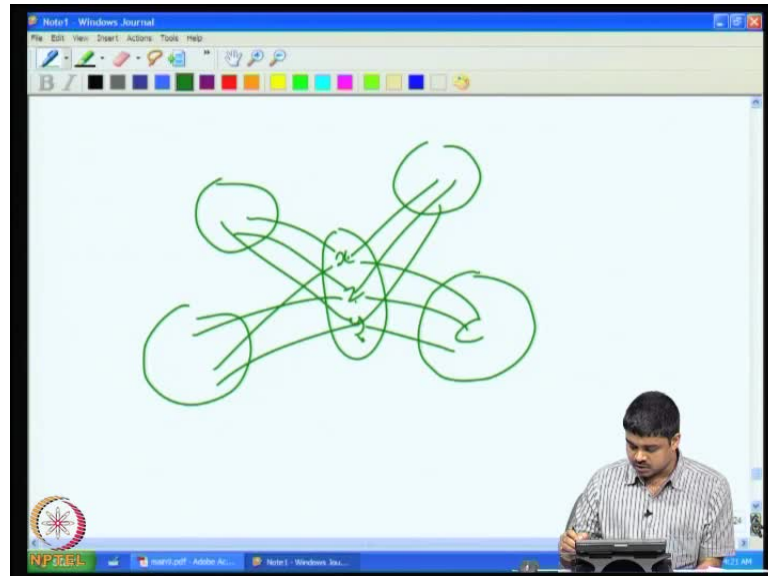
(Refer Slide Time: 54:16)



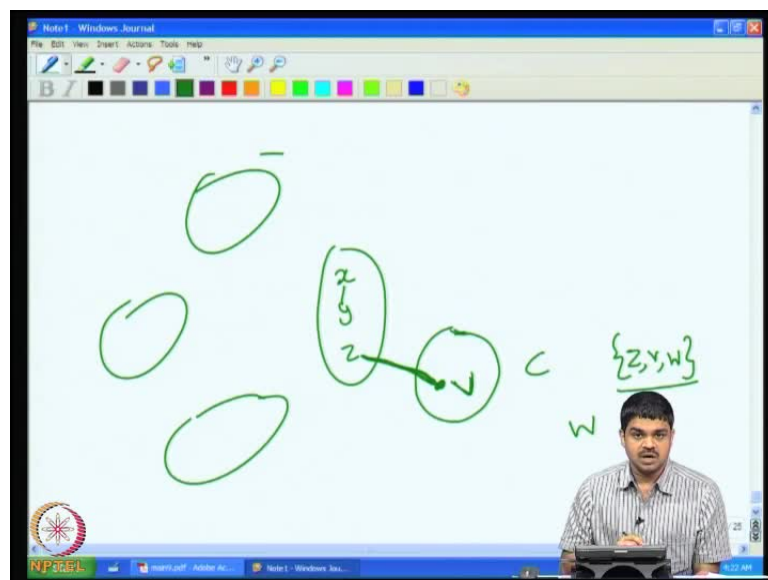
So, such an x, y Once you get such an x, y . So, I will show a contradiction. Here, we have x, y, z and you can look at the minimum component C of it. I should say x, y, z . Now, one property of a minimum separator is very important. That is, if it is minimum separator, this x, y, z is a minimum separator, it should have an edge. This x should have an edge to each of the component and z should have an edge to each of the component, y should have an edge to each of the component, why is it so? Suppose not. For instance, if there is this a, b, c is a minimum separator and there is a component here to which only a and b are connected, c is not connected, there are other components also and that is why

it is a minimum separator. Then naturally, if I remove a and b alone, I can cut away this portion from the rest. So, there will be a 2 separator and the assumption that the graph is 3 connected will be wrong.

(Refer Slide Time: 54:56)



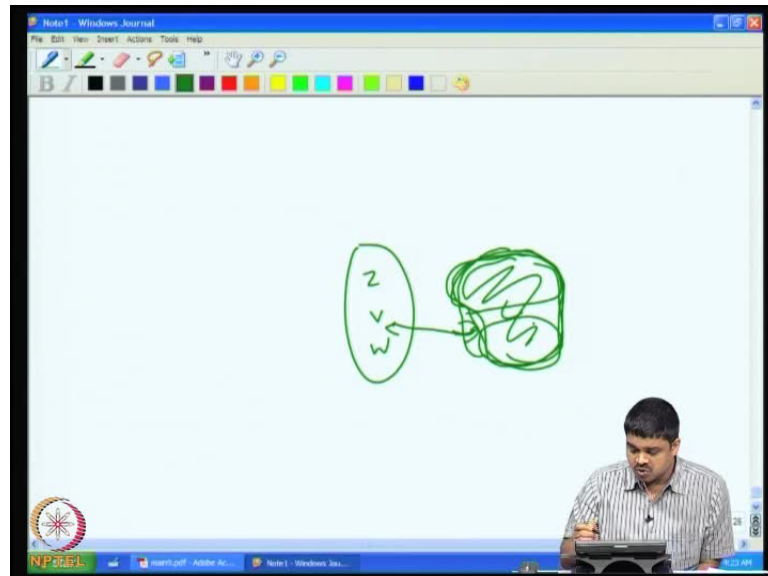
(Refer Slide Time: 55:12)



So, therefore, in any minimum separator, we will have the property that every vertex in the minimum separator is connected to every component by an edge like I showed here. Now, this property allows us to infer that, so going back here, allows us to infer that when I consider x , y , z and my smallest component C ; there are other components; this is

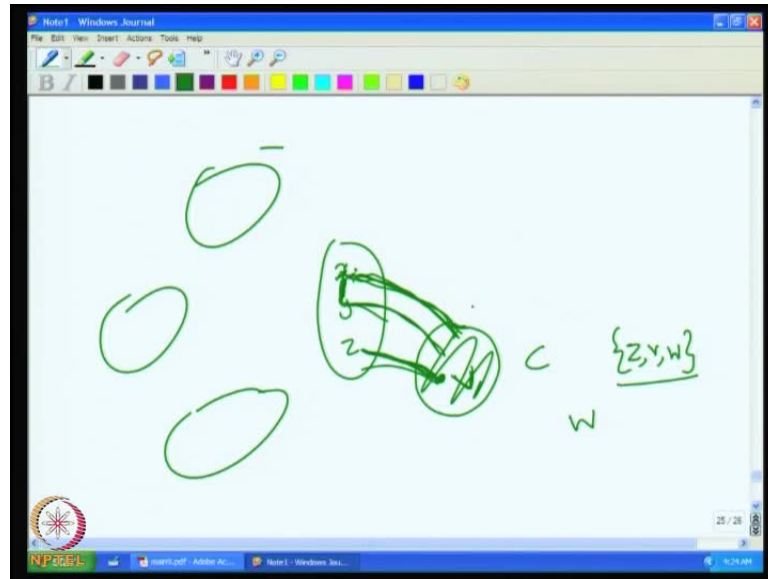
the smallest component and this happens to be the smallest among all possible x, y, z , which I select.

(Refer Slide Time: 55:53)



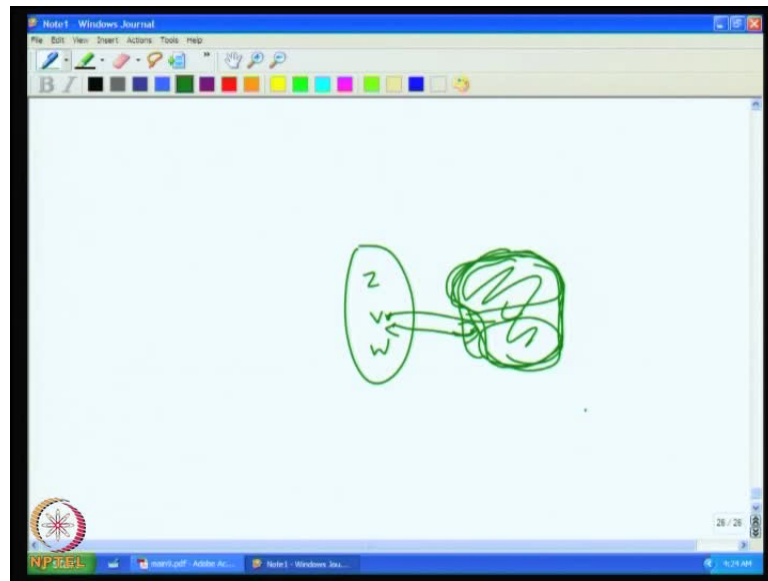
So, there will be one vertex V here, to which it is adjacent because there should be edge going here and that end of this edge be called V . Now z, v is an edge and there is another vertex w somewhere, I do not know where, somewhere such that z, v, w , these three set is a separator of the given graph. Now, you can consider this z, v, w ; this is the separator of the given graph.

(Refer Slide Time: 56:58)



Now, I want to see what will happen to my component C. Of course, the component C cannot break, become smaller at all. If it becomes smaller then it is a contradiction because C means it may not be as such, the vertices of C. You have already pulled v away from this. So, it is already becoming smaller. It is getting smaller. So, if it gets disconnected here because I pulled away v then we got even smaller component, but that is not allowed because you removed v, something more should be added to that and then you should get a component at least as big as the original C, but then the C is connected. So, if you go back to the earlier picture, the C is connected only to x and y, **other than this C component the component C is connected to only to x and y** other than say it is not possible to connect only to x, not connect to y because once you get x, y will automatically come because this is an edge. So, that means it should capture x and y and later once you will capture x and y, it can capture more things.

(Refer Slide Time: 57:33)



So, **it is possible that x and y should** it is important that x and y should be part of the component which contain C. You removed v, but then x and y should balance it, but the difficulty is if that happened, where will all the neighbours of v go? So, remember v was a vertex in C **and then we can now**. We will discuss it in the next class. So, we will complete the proof in the next class. We will need some time. Thank you.