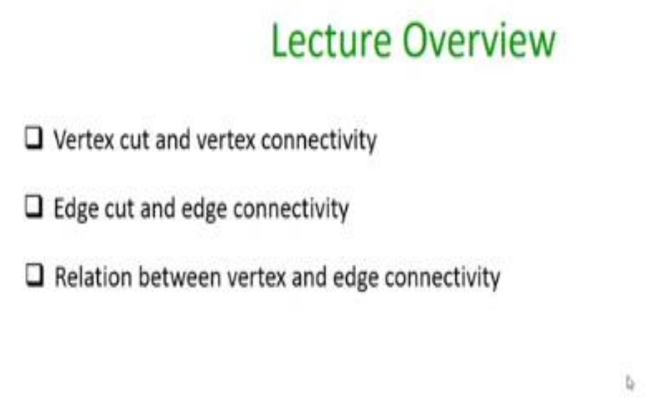
Discrete Mathematics

Prof. Ashish Choudhury
Department of Mathematics and Statistics
International Institute of Information Technology, Bangalore

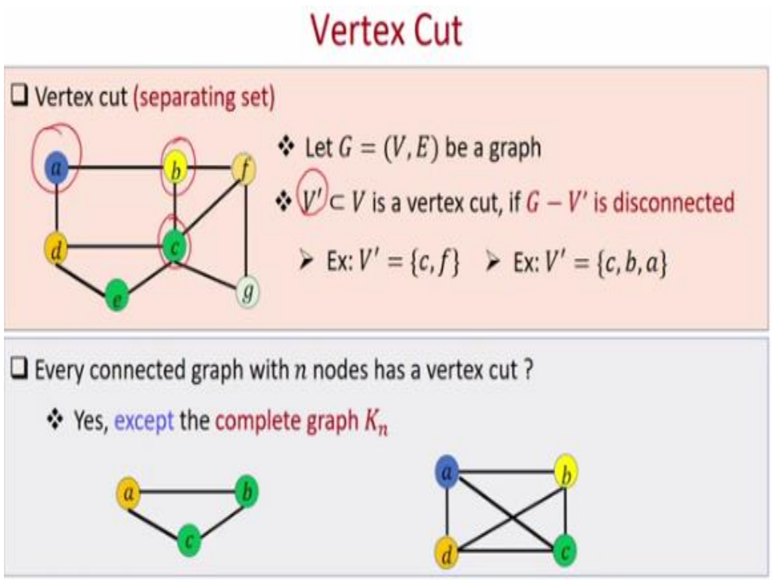
Lecture -48
Vertex and Edge Connectivity

(Refer Slide Time: 00:26)



Hello, welcome to this lecture. In this lecture we will continue about discussion on our vertex connectivity, cut vertices, cut edges and we will introduce vertex cut and vertex connectivity, edge cut and edge connectivity. And we will formally prove the relationship between the vertex connectivity and edge connectivity.

(Refer Slide Time: 00:43)



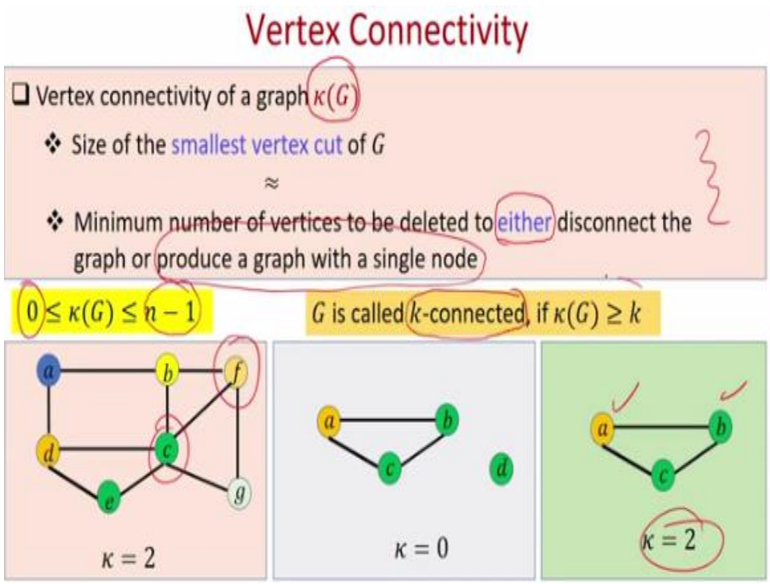
So, let us start with the definition of a vertex cut. It is also called as a separating set. So, imagine you are given a graph $G = (V,\ E)$. Then a proper subset $V' \subseteq V$ of the set of vertices is called the vertex cut if removing the vertices in V' disconnect your graph. So, remember if your graph has an articulation point then that articulation point itself can constitute a V' of potential V' whose deletion will disconnect the graph.

But it might be possible that your graph may not have an articulation point in which case you may need to delete more than one vertex in the graph to disconnect it. So, the basic idea here is we are now try to generalize the definition of articulation point in terms of a subset of vertices V'. So, if I take this graph and if I remove the nodes c and node f then this node g will get disconnected from the network, because these edges will also go away.

Whereas I can remove the vertices c, b and a that will ensure that d and a separate out from current diagram. So, here as of now the there is no criteria on the cardinality of V', |V'|. I am just checking with a V' constitutes a vertex cut or not whether deleting the vertices in V' disconnects the graph or not. So, now the question is, can I say that every connected graph which has nodes as a cut vertex and answer is yes, except when the graph is a complete graph.

So, if you take a complete graph of a n nodes even if you remove up to n - 1 nodes your graphs still remains connected, the reduced graph. So, remember my $V' \subseteq V$. You cannot say that you can delete the entire graph itself because if you delete the entire set of vertices the entire graph vanishes. So, in a complete graph at max, you can delete up to n - 1 nodes with the hope to get a disconnected graph, but that is not possible.

(Refer Slide Time: 03:18)



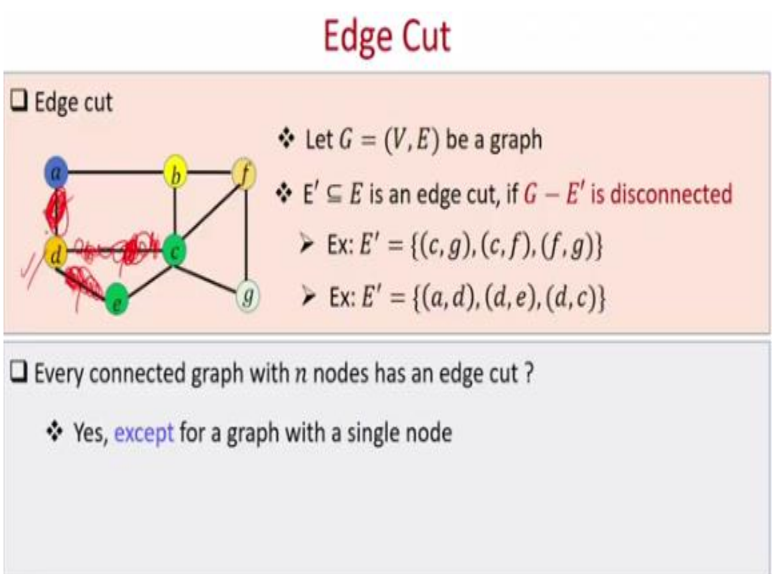
So, now let us next define vertex connectivity of a graph. And the vertex connectivity of a graph is also denoted by this parameter kappa, $\kappa(G)$. So, the vertex connectivity of a graph is the size of the smallest vertex cut. That means the size of the smallest V' whose deletion will disconnect your graph or equivalently it is the minimum number of vertices to be deleted to disconnect your graph.

So, consider this graph and for this graph your $\kappa(G)$ is equal to 2. When your $\kappa(G)$ will be 1 if your graph has an articulation point. But if there is no articulation point and you may need to delete more than one vertex to disconnect your graph and in this graph we do not have any articulation point. So, we need to delete at least two nodes to disconnect the graph namely the nodes c and f.

Whereas if I take this graph G, then my $\kappa(G)$ here will be 0, because my graph is already disconnected, and I do not need to delete any additional node to further disconnect it. Whereas if I take this graph, then the definition that I have given for vertex connectivity does not make sense here. Because even if I delete up to n - 1 nodes here namely 2 nodes if I delete say a and b, I will be left with a connected graph. So, that is why to take care of this special case of complete graph, I slightly change my definition of vertex connectivity and my definition of vertex connectivity is the following. It is now the minimum number of vertices which needs to be deleted to either disconnect the graph or produce a graph with a single node. This later condition is to take care of the complete graph. So, with respect to this new definition the $\kappa(G)$ or the vertex connectivity for this triangle graph will be 2.

So, it turns out and it is easy to verify that the vertex connectivity of a graph will always be in the range 0 to n - 1, where n is the number of nodes in your graph. Because if your graph is already disconnected then you do not need to delete any vertex. Your vertex connectivity will be 0 whereas if your graph is a complete graph, then you need to delete up to n - 1 nodes to produce a graph with a single node. Now, my graph will be called as k-connected, if the vertex connectivity of the graph is at least k. That means the size of this smallest vertex cut is k.

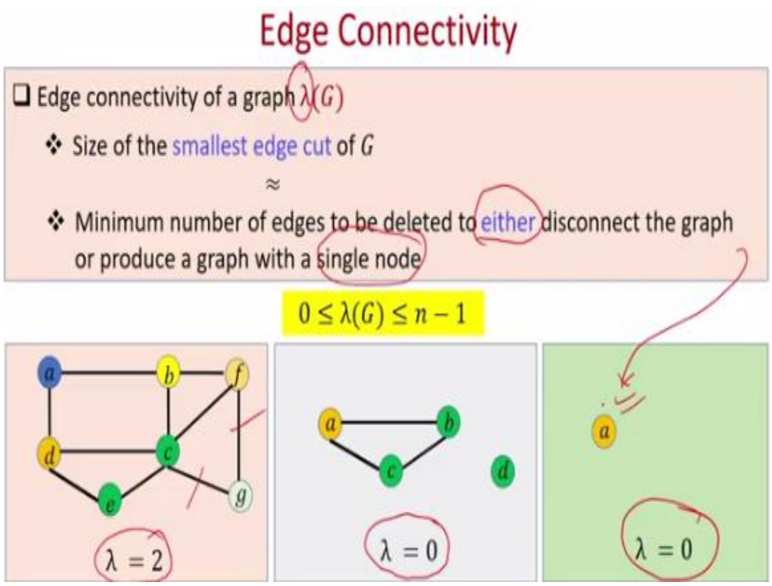
(Refer Slide Time: 06:26)



Now we can keep similar theory with respect to a collection of edges whose removal will disconnect the graph. So, we define what we call as an edge cut. So, imagine you are given a graph and a collection of edges E' will be called an edge cut, if deleting the edges in E' from the graph G disconnects your graph. So, there might be several E's possible. Now when I give the definition of edge cut, I am not focusing on the minimum sized E'.

I will be just given an E' and I have to check whether deleting the edges E' deletes or disconnects my graph or not. So, for instance if I take this graph, if I remove the edge between c and g, c and f and f and g then I get a disconnected graph because the node g now gets disconnected from the rest of the network. Similarly, if I remove the node edges between a and d, d and e and d and c, then I think this should not be if I remove the edges, it constitutes an edge cut. Because now the node d gets disconnected from the entire graph. So, again similar to the question that we asked for the vertex cut, let us answer this question whether every connected graph which has nodes as an edge cut or not. Again, the answer is yes, except for the case when your graph is already a graph with just a single node and no edges. Because if you have a graph with a single node and no edges then you do not have any edge to delete at the first place.

(Refer Slide Time: 08:42)



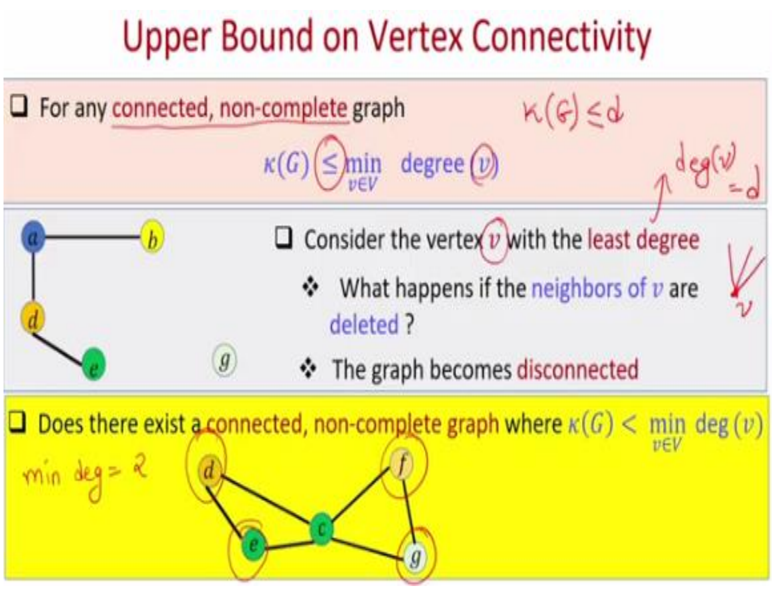
Otherwise, you always have a set of edges in a connected graph which you can delete to disconnect it. So, we now define what we call as the edge connectivity of a graph and this is denoted by λ. So, what is the edge connectivity of a graph? It is the size of the smallest edge cut or equivalently the minimum number of edges to be deleted which disconnects graph. So, if I take this graph then the edge connectivity is 2 because I need to delete two edges to disconnect the graph.

If your graph was a bridge or cut edge then λ will be 1. But if your graph does not have a bridge then you need to delete more than one edges to delete the graph to disconnect the graph. If you take this disconnected graph, then edge connectivity will be 0, because it is already disconnected, and I do not need to delete any edge to disconnect it. But now if I take this graph, which has no edges and just a single node then this definition does not make sense here.

Because I cannot delete any edge in this graph to let make it disconnected graph. So, to take care of the special case, I modify my definition and my modified definition of edge connectivity is the following. I define edge connectivity to be the minimum number of edges which needs to be deleted to either disconnect the graph or produce a graph with single node. This latter condition is added for taking care of this special case.

Because if I apply this definition to the special case then I get λ equal to 0. Because my graph is already a graph with a single node and I need to delete 0 number of edges to produce a graph with a signal node. Again, it is easy to verify that your edge connectivity will be in the range 0 to n - 1, 0 for the case when your graph is already disconnected and or for this particular case when your graph is just consisting of a single node. And n - 1 for the case when your graph is the complete graph. If your graph is a complete graph, then you just delete n - 1 edges incident with any node then that node gets disconnected from the rest of the network.

(Refer Slide Time: 11:12)



So, now we want to fix some upper bounds on the vertex connectivity and edge connectivity. So, let us first prove an upper bound on the vertex connectivity. So, my claim here is that for any connected, non-complete graph, I stress connected not complete graph the vertex connectivity is always less than equal to the minimum degree that is possible in your graph. That means you take the vertex v which has the least degree in the graph say the degree is d, then my claim is that $\kappa(G)$ is always less than equal to d.

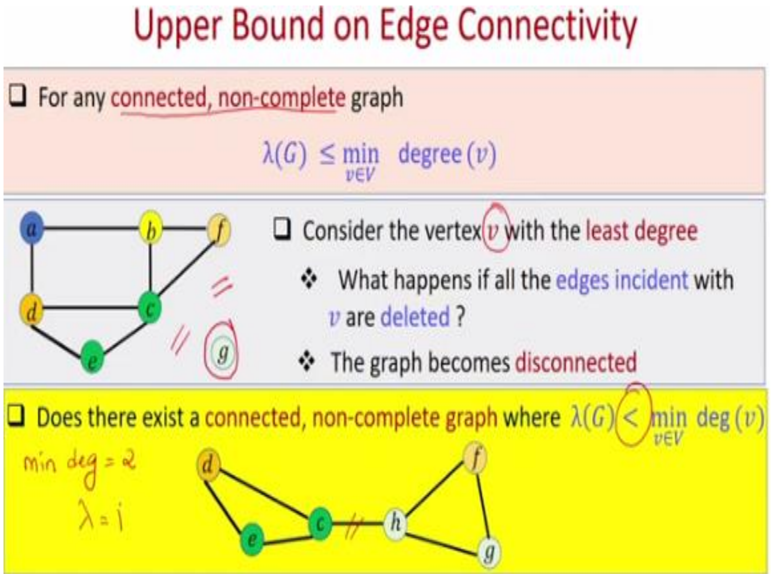
So, a simple proof for this fact is the following. So, consider this graph or consider any arbitrary connected non-complete graph and focus on the vertex v, which has the least degree, namely the degree of the vertex v is d. That means the node v has d number of neighbours. Now what happens in this graph if all the neighbors of the vertex v are deleted?

So, for instance in this graph the vertex c has the, Sorry the vertex g has the least degree. When the vertex e also has the same degree as vertex g so what I am saying is if you remove all the neighbors of the vertex which has the least degree then that vertex v gets disconnected from the rest of the graph. So, that shows that you do not need to delete more than d number of nodes in the graph to produce a disconnected graph.

The maximum number of nodes that you need to delete is d. But then now you might be wondering that as per the argument that I have given why it is less than equal to, why not exactly equal to. You might argue that I definitely need to delete exactly d number of nodes to disconnect the graph or to disconnect the node v from the rest of the graph. Well that is not necessarily true if you consider this graph then here the minimum degree is equal to 2.

Because you have the node g which has degree 2, you have a node f which has degree 2, you have node d which has degree 2, you have node e which has degree 2 and so on. So, you might argue that definitely I need to delete at least 2 nodes to disconnect the graph. Of course, if you can delete all the neighbors of d or all the neighbors of e or all the neighbours of f or all the neighbours of g you get a disconnected graph. But you do not need to do that much.

(Refer Slide Time: 14:16)



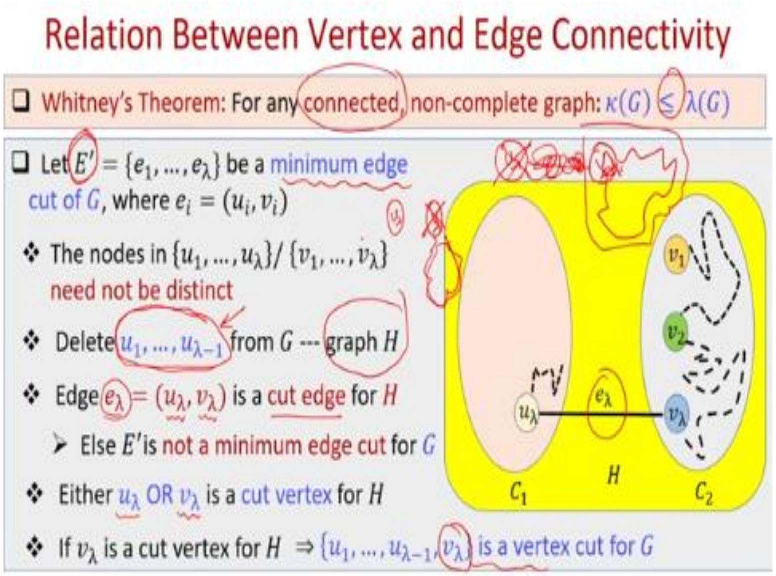
Because if you just delete the node c your graph gets disconnected. So, the vertex connectivity comes by here is 1 which is strictly less than the minimum degree in the graph. Now let us put, let us derive some upper bound on the edge connectivity and again I take the case of a connected, non-complete graph and upper bound here remains the same. My claim is that the edge connectivity of a graph is always upper bounded by the minimum degree that is possible in your graph.

So, the proof is again very simple, let v be the vertex with least degree, of course, you can have multiple vertices with the same least degree focus on one of the vertices v. And then argue that what happens if all the edges incident with the vertex v are deleted. So, for instance in this graph, g is the vertex v which has the least degree. So, what I am arguing here is that if you delete all the edges which are incident with this vertex g.

The vertex g gets disconnected from the rest of the graph. So, that shows you do not need to delete more than least degree number of edges in your graph to produce a disconnected graph. But then again you get the same question that as per this argument one may get the feeling that λ should be exactly equal to the minimum degree in the graph, why less than equal to? Again, consider this graph here the minimum degree is 2.

So, the vertex f has degree 2 the vertex g also has degree 2. So, that means to remove these two edges the node f gets disconnected and in the same way if you remove these two edges the node g gets disconnected and so on. So, you might say that I definitely need to delete two edges to produce a disconnected graph from this connected graph. The answer is no. Because if you remove this edge which is a bridge in this overall graph your graph gets disconnected. So, the λ here is 1 not 2.

(Refer Slide Time: 16:47)



So, now we want to establish a relationship between the vertex connectivity and edge connectivity. And there is a very nice theorem statement which says that you take any connected, non-complete graph then the vertex connectivity is always less than equal to the edge connectivity. So, let us prove this. The proof is very simple, so imagine you are given a connected, non-complete graph. Why I am taking non-complete? Because for a complete graph this inequality is always true, both $\kappa(G)$ as well as $\lambda(G)$ are n - 1.

So, that is why I am taking the case of a non-complete graph and connected graph. Because again if I take the disconnected graph, both $\kappa(G)$ and $\lambda(G)$ are zero and inequality holds. So, I take a connected, non-complete graph G, which is an arbitrary graph and imagine that E' constitutes an edge cut for G, the minimum edge cut for G which has $\lambda(G)$ number of edges. That means if I remove the edges $e_1, e_2, \ldots, e_\lambda$ from the graph, my graph gets disconnected.

And say the end points of the ith edge e_i is (u_i, v_i). So, these are the end points of the edge e_i. So you can imagine that if I remove the edges $e_1, e_2, \ldots, e_\lambda$ from my graph G, I get 2 connected components, component C_1 and component C_2. So, you can imagine the structure of your graph is something as follows. You have the vertices images structured in such a way that you can interpret $e_1, e_2, \ldots, e_\lambda$ as kind of bridges between component C_1 and C_2.

So, that is one of the endpoints e_1 is in component C_1 and other endpoint namely v_1 is in C_2 and so on. By the way here, it is not necessary that all the nodes u_1 to u_λ or v_1 to v_λ are distinct. It might be possible that you have just u_1 in component C_1 and all the edges $e_1, e_2, \ldots, e_\lambda$ has u_1 as one of its endpoints. That is also possible but similarly you might have a case where the edges are such that you only have v_1 as the only node in C_2 and all this edges $e_1, e_2, \ldots, e_\lambda$ as v_1 is one of its end point.

But it is not necessary that we want to u_λ and we want to v_λ are all distinct. But for simplicity and for pictorial understanding I am representing them as λ number of u vertices and λ number of v vertices. And as per my property as per my definition E' or this collection of edges e_1 to e_λ constitutes an edge cut. That means even if you do not delete a single edge from this collection of edges the graph G still remains connected.

Only when you delete all this λ edges the graph gets disconnected into components C_1 and C_2. That is the property of this collection of edges E'. Now from these edges I have to show the existence of at least, I have to show the existence of some λ number of vertices whose deletion will definitely disconnect my graph. Because if I can show you the existence of some λ number of vertices whose deletion will disconnect my graph then that shows that my vertex connectivity is also upper bounded by λ, and hence it shows that the vertex connectivity is upper bounded by edge connectivity.

So, let us do that, so first delete the $\lambda - 1$ u vertices namely to delete the end point of e_1 from the component C_1, you take the edge e_2 and delete the edge u_2 from the component C_1. And similarly, you take the delta minus 1 edge and delete the u vertex from the component C_1. So, you have deleted λ-1 vertices from your graph G.

And remember as per the definition of deletion of a vertex and I delete a vertex all the edges incident with that vertex also gets deleted from the graph. So, my new graph I call it as the graph H and my new graph H will look something as follows. The edge u_λ is still there in my graph G. And because of that the end point u_λ is still there in my graph G, now what can I say about my graph H? My graph is still connected graph.

It is not yet disconnected, because I still have one bridge or one edge going from u_λ to the component C_2 and that will ensure that everything in C_1 is reachable to C_2 via the edge e_λ. Now here is the crucial claim, my claim is that the edge e_λ constitutes a cut edge or a bridge for the reduced graph H. And the poof is very straight forward.

If this edge e_λ does not constitute a cut edge for the reduced graph H, that means that even if I delete this edge e_λ, still somehow everything in C_1 is reachable to C_2. That is the implication but if that is the case then I get a contradiction that the collection of edges E' which I assumed to be the minimum number of edges whose deletion will disconnect the graph G is not a valid assumption. So, I remember I assume that as soon as I remove the edges e_1 to e_λ, my graph get splitted into 2 parts, C_1 and C_2 such that nothing in C_1 is reachable to C_2 and vice versa.

So, I have already removed e_1 to $e_{\lambda-1}$. And now I mark queuing that if I remove e_λ my graph H gets splitted into C_1 and C_2, such that C_1 is completely disconnected from C_2 and vice versa. If that is not the case that means I need to further delete more edges even after removing e_1 to e_λ to disconnect my original graph G, which goes against the assumption that E' was the collection of minimum edges whose deletion will disconnect my graph.

So, I get this implication, remember I have already removed λ-1 vertices. And now I have to show that if I add one more vertex to this collection of λ-1 vertices, which I have already deleted from G, I get a vertex cut for my graph G. Now, you might be tempted to say that if I include u_λ to this collection of vertices u_1 to $u_{\lambda-1}$, it will always constitute a vertex cut for G, or you might be tempting to say that if I include v_λ the collection of vertices in u_1 to $u_{\lambda-1}$, that will constitute a vertex cut for G.

You can not necessarily do that you have to argue here based on cases. So, since I have argued that the edge e_λ constitutes a bridge or a cut edge for the graph H, I can definitely say that one of its end point is a cut vertex for the reduced graph. Because then only the edge e_λ can constitute a bridge. And this is a property which is there with respect to any bridge or any cut edge of a graph.

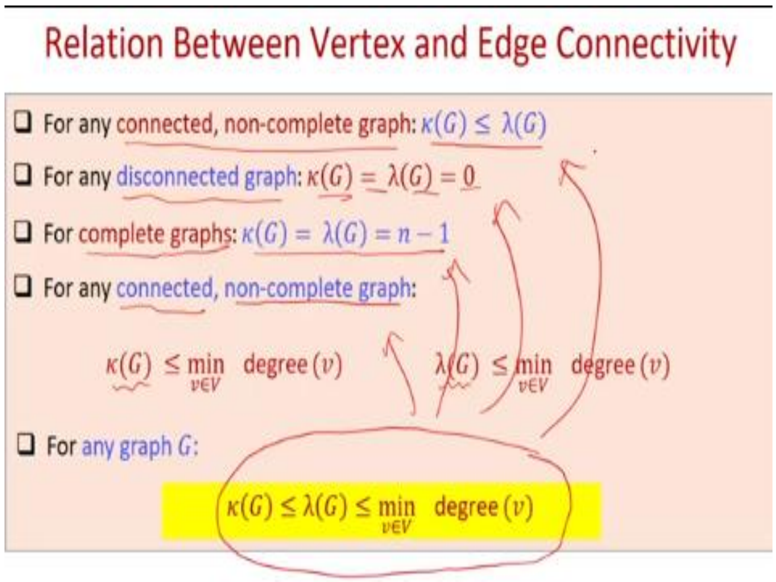
I can always say that if I have a cut edge in a graph, one of its end point definitely a cut vertex. Because if none of the end points of the edge is a cut vertex, then in the first place that edge was not a cut edge, a very simple fact. Now, I have two possible cases. If u_λ is a cut vertex for H, then I can say that the collection of u_1 to $u_{\lambda-1}$, and the vertex u_λ constitutes the vertex cut for G.

Whereas case 2 is the following, if it is the if v_λ is the cut vertex for the graph H then I can say that the collection of u_1 to $u_{\lambda-1}$, along with v_λ will constitute a vertex cut for G. So, you cannot always say that its u_1 to u_λ which is always a cut vertex for the graph G. It depends, once you have removed the first $\lambda - 1$, edges and one of the end points of those $\lambda - 1$ edges in one of the components you will be left with of cut edge and your graph may be something as follows.

So, your graph may be something like this. So, you might have got u_λ and then you have v_λ and then the rest of the graph is still, that means your reduced graph H is something like this. In that case, you cannot say that if I just delete u_λ along with u_1 to $u_{\lambda-1}$, I get disconnected graph. No, by removing u_λ you get the whole graph C_2 as you reduced graph, which is connected.

So, in that case, you have not obtained a cut vertex. It is only when you remove v_λ that this portion of C_2 gets disconnected from u_λ. So, depends upon which end point of the cut edge e_λ is the cut vertex for the reduced graph, and that cut vertex along with the λ-1 vertices, which you have already removed will give you a vertex cut for the original graph G. So, that is the subtle point, the two cases. Otherwise the rest of the proof is straight forward.

(Refer Slide Time: 28:23)



So, now let us unify the relationship between the vertex connectivity and edge connectivity taking care of various cases. We just proved that for any connected, non-complete graph the vertex connectivity is always less than equal to edge connectivity. We know that for disconnected graphs, $\kappa(G) = \lambda(G) = 0$. And for complete graphs, we know that $\kappa(G) = \lambda(G) = n-1$.

And you also prove that for connected non-complete graphs, individually the vertex connectivity is upper bounded by the minimum degree, $\kappa(G) \leq \min_{v \in V} \text{ degree } (v)$ and the edge connectivity is also upper bounded by minimum, $\lambda(G) \leq \min_{v \in V} \text{ degree } (v)$. So, unifying all these things we can say the following. Irrespective of whether my graph is connected, disconnected, complete, non-complete the vertex connectivity is always less than equal to edge connectivity and edge connectivity is always less than equal to the minimum degree in the graph. $\kappa(G) \leq \lambda(G) \leq \min_{v \in V} \text{ degree } (v)$.This relationship takes care of all the cases here, complete, non-complete, connected disconnected and so on.

(Refer Slide Time: 29:45)

References for Today's Lecture

So, that brings me to the end of this lecture. Just to summarize, in this lecture we discussed about the vertex connectivity, edge connectivity and we prove the general relationship that exists between the vertex connectivity and edge connectivity. Thank you!