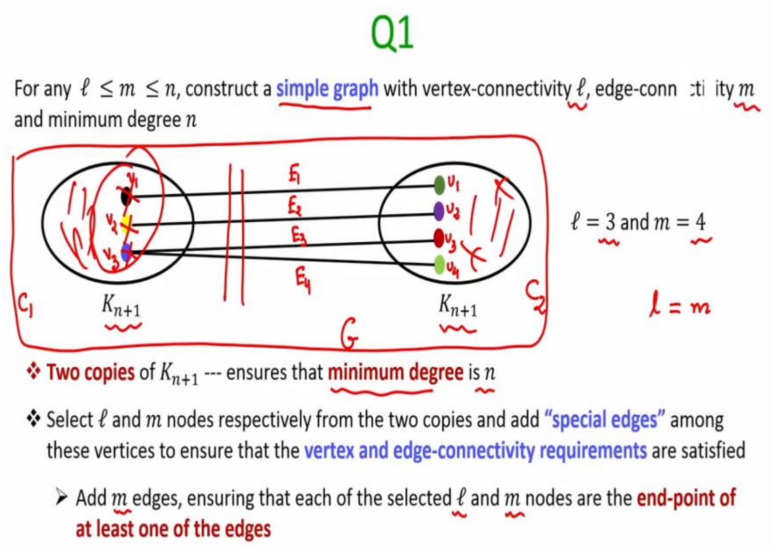
Discrete Mathematics
Prof. Ashish Choudhury
International Institute of Information Technology - Bangalore

Lecture - 53
Tutorial 9: Part I

(Refer Slide Time: 00:24)



Hello everyone, welcome to the first part of tutorial 9 so, let us start with question number 1. So, in this question you are given 3 positive numbers ℓ, m, n not be positive, non negative integers. So, ℓ, m, n such that ℓ is less than equal to m and m is less than equal to n. And what we want here is a simple graph where the vertex connectivity is ℓ, edge connectivity is m and minimum degree is n. So, remember the relationship between the vertex connectivity edge connectivity, and the minimum degree is that: vertex connectivity is less than equal to edge connectivity and edge connectivity is less than equal to the minimum degree in the graph. Basically in this question we are asking you to give the construction of one simple graph which satisfies the inequality with respect to the ℓ, m, n values that are given to you. So, here is how we can construct a graph. Since we need the minimum degree in the graph to be n, to ensure that my resultant graph; my final graph has the minimum degree n, I take 2 copies of a complete graph with n + 1 nodes. So, this is my copy number one and this is my copy number two C_1 and C_2. Both of them are complete graphs with n nodes so I am not drawing the edges within the complete graph. The whole graph I am denoting by this circle. Now I have taken care of the minimum degree in my graph.

Now I have to take care of my vertex connectivity and edge connectivity. So, how do I do that? I randomly pick l nodes and m nodes from the two copies. So, l nodes I pick from the first copy and m nodes I pick from the second copy. Remember the values of l and m are given to you and l and m are both less than equal to n. So, it is possible to pick l nodes from the first copy. And it is possible to pick m nodes from the second copy.

Feel free to pick any l nodes from the first copy, m nodes from the second copy. So, I am going to demonstrate assuming that l = 3 and m = 4. So I have picked 3 nodes arbitrarily from the first copy. And I have picked 4 nodes arbitrarily from the second copy. Now I have to take care; I have to ensure that my vertex connectivity should become l and edge connectivity should become m.

So, I already have edges in these copies of complete graph which I have not highlighted here but now I will add; I will give extra edges in my graph; those edges will be special edges and these special edges will ensure that my vertex connectivity of the overall graph is l and the edge connectivity of the overall graph is m. How do I do that? So, I add edges between the l nodes which I have picked in the first copy and m nodes which I have picked in the second copy in the way that it is ensured that: I add basically m edges between the l nodes and m nodes that I have picked in the 2 copies respectively. And m edges are added in such a way that those edges ensure that each of the l nodes and m nodes which I have picked in the 2 copies they occur as the end points of those edges which I am adding here. So, for demonstration purpose m = 4. So, I am adding 4 edges apart from the edges which are already there in my copy 1 and copy 2.

So, I am adding edge number 1, edge number, 2 edge number 3, edge number 4. And these edges are added in such a way that if I call this vertices as v_1, v_2, v_3 and u_1, u_2, u_3 and u_4. Then if I take any vertex among v_1, v_2, v_3 it is occurring as one of the endpoints out of these 4 edges. And in the same way if I take any of the vertices u_1, u_2, u_3, u_4 it is occurring as one of the endpoints of these 4 edges. That is the way I am adding the edges.

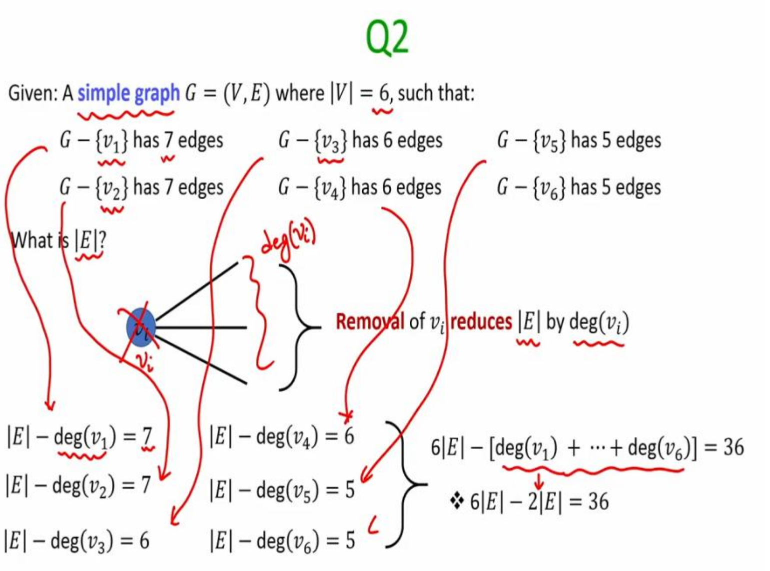
So, if l = m or if l would have been equal to m what I would have done is I would have picked l edges in first copy m edges in the second copy and just add distinct edges. That means between the one node here and another node here I would have added 1 edge between the second node in the first copy and the second node in the second copy I would have added 1 edge and between the third node of both the copies I would have added 1 edge.

But since my m could be more than ℓ it might happen that some of the vertices out of the ℓ vertices which I have picked in the first copy are the endpoints of multiple edges or endpoints of the multiple special edges. Now you can see that the way I have given these special edges it is ensured that my vertex connectivity is ℓ why vertex connectivity is ℓ? Because if I delete the l vertices which I have picked or which are the endpoints of the special edges from the first copy of the K_{n+1} graph sub graph then my entire graph get disconnected.

So, remember my entire graph is this whole graph which has 2 copies of the complete graph of n + 1 nodes and these m special edges. So, if in this whole graph G, I removed the vertices v_1, v_2, v_3, v_ℓ then that will ensure that all these edges also vanishes. And that will ensure that the first copy of K_{n+1} gets separated from the second copy of K_{n+1}. So that takes care of vertex connectivity being ℓ.

And it is easy to see that the edge connectivity is m because the m edges which I have added across the 2 copies of K_{n+1} that constitute the edge cut because if I remove all these m edges again the 2 copies of K_{n+1} separates out. So that ensures that edge connectivity is m and as I argued that since I have taken 2 copies of complete graph with n + 1 nodes which are a sub graph of the entire graph the minimum degree in the graph is at least n so that is the construction.

(Refer Slide Time: 07:18)



So, now let us go to question number 2. In question number 2 you are given an unknown simple graph G, the graph G is not known to you known to you in the sense that you are just given that it has 6 vertices but exact cardinality of edge set is not given. But it is given to you that your graph G is such that if you delete the vertex v_1 from the graph then you are left with 7 edges. If you delete the vertex v_2 from the graph you are left with 7 edges you delete vertex v_3 from the graph you are left with 6 edges and so on.

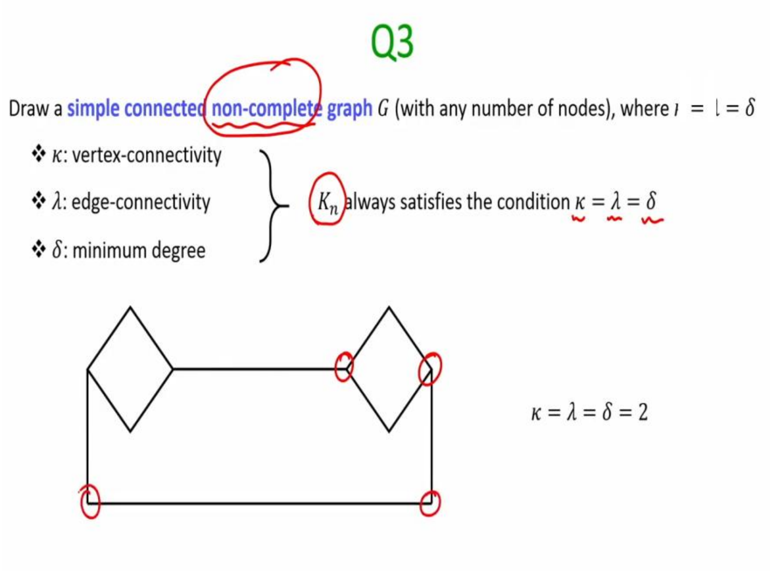
So, if that is the case the question asks you to find out the cardinality of the edge set of the original graph. Again do not try to do a brute force and try all possible graphs in your mind and then hit upon the answer because that will take enormous amount of time. Instead we will try to apply some rules of logic and properties of graph here. So, the property that we would like to explore here is that if I take the graph G and if I have a vertex v_i here.

If you have the vertex v_i here and if you remove the vertex v_i from the graph the cardinality of the edge set gets decremented by the degree of v_i because the deletion of the vertex v_i will delete how many edges from the graph? All the edges which are incident with the vertex v_i namely degree of v_i number of edges from my edge set will be removed that means the cardinality gets reduced by degree of v_i. That is a simple fact that we are going to follow here.

So that means what I can say is that my graph G is such that the edge set cardinality minus the degree of v_1 is 7 because it is given that after deleting vertex v_1 in the graph you are left with 7 edges. That means in the leftover graph which I obtain after deleting v_1 if I would have added the edges which were incident with vertex v_1 and how many such edges would have been there? : degree of v_1 number of edges that would have given me the cardinality of the edge set. In the same way from the 2nd fact I get this equation. And from the 3rd fact I get this equation from the 4th fact I get this equation from the 5th fact I get this equation and the 6th fact I get this equation. Now if I sum all these 6 equations I get that 6 times the cardinality of E minus the summation of the degree of 6 vertices in the graph is 36.

And now I can apply the handshaking theorem which says that if you take the summation of the degrees of all the vertices in your original graph it is same as twice the number of edges. So, now I have 1 equation just involving the unknown which is my cardinality of the edge set. So, I get my edge sets cardinality to be 9.

(Refer Slide Time: 10:46)



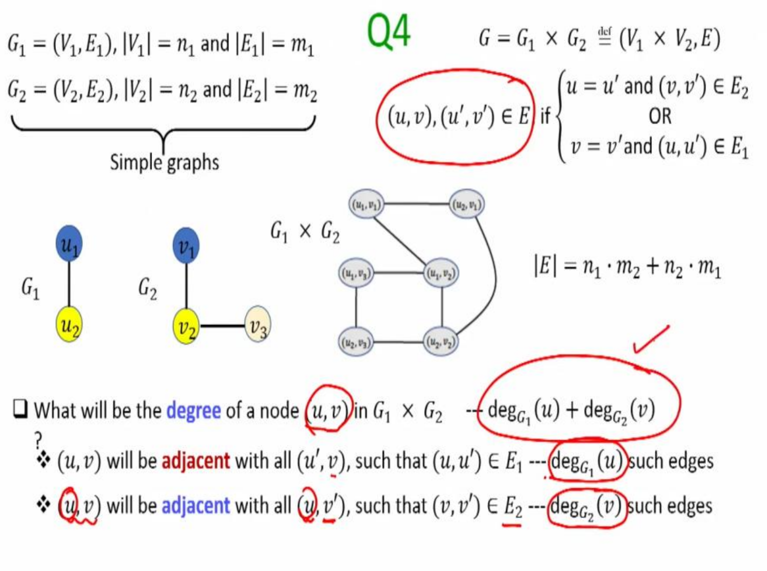
In question 3 I want to draw a simple connected non complete graph with any number of nodes where the vertex connectivity, edge connectivity and minimum degree are all same. Again I think you can draw the graph from the answer from the graph which we constructed in question number 1 but again let us do this question. So, why I am focusing on a non complete graph here?

Because if I do not put that restriction then you can always give me the example of a complete graph because in a complete graph with n nodes the vertex connectivity is n because as per the definition of vertex connectivity the vertex connectivity of a complete graph will be n - 1 because I cannot disconnect a complete graph the only thing I can do is after deleting n - 1 nodes I am left with a graph with a single node.

And in the same way the edge connectivity is defined for a complete graph to be n - 1. And of course the minimum degree of a complete graph is n - 1. ? So that is why complete graph is always an example if I do not put this restriction of non complete graph. So, here is an example of a non complete graph where vertex connectivity edge connectivity and minimum degree are all 2 why 2?

So, you can see that my edge connectivity is 2 because if I remove this edge and this edge then this portion of the graph gets disconnected from this portion of the graph. So, I need to remove 2 edges if you just remove 1 edge then the graph does not get disconnected and due to this I can say that if I delete this vertex and if I delete this vertex then again my graph gets disconnected. So, my vertex connectivity is also 2 and the minimum degree is 2 because if you take this vertex and this vertex then their degrees are 2 which is the minimum degree.

(Refer Slide Time: 13:04)



Let us go to question number 4. Here you are given 2 simple graphs G_1 and G_2 their vertex sets are V_1, E_1, V_2, E_2 respectively there are n_1 number of vertices in the first graph and m_1 number of edges in the first graph. Whereas there are n_2 and m_2 number of vertices and edges respectively in the second graph. Now I am defining a new operation on the graph which I call as the Cartesian product of the graphs.

And it is possible to define the Cartesian product because remember the vertex set and edge set they are sets, so I can always define Cartesian product of the sets. So, the way I define the Cartesian product of the 2 graphs is the following the vertex set will be now ordered pairs basically the vertex set here is the Cartesian product of the vertex set of the first graph and the vertex set of the second graph because I am defining the Cartesian product of G_1 and G_2.

If I would have defined the Cartesian product of G_2 and G_1 then the ordered pairs would be the first vertex from the second graph and the second vertex from the first graph that means the vertex set would have been v_2 x v_1 but since I am defining the Cartesian product of graph 1 and 2 the vertex set is the Cartesian product of the first vertex set and the second vertex and the edge set is E.

Now how the edges are defined here? So remember now my vertex set in the graph is ordered pairs. So, I will say that 2 ordered pairs which represent 2 vertices they will be connected by an edge if the following holds: either the first component of the 2 vertices should be same and the second component of the 2 vertices should be an edge in the second graph under that condition I could add an edge between these 2 ordered pairs or another condition in which I can have an edge between these 2 ordered pairs is the following: the second component of the 2 vertices are same and the vertices which appear as the first component they have an edge among them in the first graph. So, if any of these 2 conditions hold I will add an edge between these 2 ordered pairs in my Cartesian product of the graph.

So, it might look slightly tricky so let me demonstrate with an example. So, imagine my graph G_1 is this which has 2 nodes and 1 edge so my $n_1 = 2$ and $m_1 = 1$ and I have a second graph here which has 3 vertices and which has 2 edges. Now let us construct the Cartesian product of the graph. So, the Cartesian product of the graph will be this I have not drawn it in a very beautiful way.

But you can now see that the vertex set will be u_1 paired with v_1 that is one vertex, u_1 paired with v_2 that is another vertex, u_2 paired with v_1, u_2 paired with v_2, u_1 paired with v_2, u_1 paired with v_3 and so on. So, you will have 6 vertices here, so u_1 paired with v_1 is here, u_1 paired with v_2 is here, and there is an edge between them. Why so? Because the first component u_1, u_1 or same here and there is an edge between v_1 and v_2.

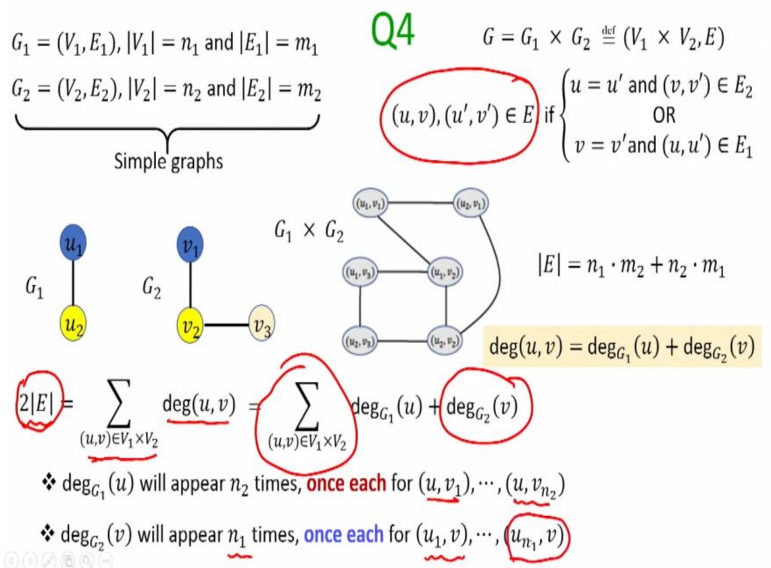
So that is why this ordered pair and this ordered pair you have an added an edge. In the same way you can see that the second component here v_1, v_1 are same and there is an edge between u_1 and u_2 in the first graph. So that is why this edge is added that is how we have built the graph G_1 x G_2 here. So, now we want to prove that the cardinality of the edge set for the Cartesian product of the graph is this value : namely $|E| = n_1 \cdot m_2 + n_2 \cdot m_1$.

And before going into the proof you can at least verify that this is actually the case for the example graph that we have here. So, we have total 1, 2, 3, 4, 5, 6, 7 edges and your number of vertices in the first graph is 2, number of vertices in the second graph is 3, the number of edges in the first graph is 1, the number of edges in the second graph is 2 and you can check that this is indeed the case.

But now we want to prove that cardinality of edge set is $|E| = n_1 \cdot m_2 + n_2 \cdot m_1$ for a general graph which is the Cartesian product of 2 graphs. So, for that what we are going to do is the following we will first consider an arbitrary vertex in the Cartesian product of the graph and try to argue what exactly will be the degree of that vertex. So, let us consider an arbitrary vertex (u, v). My claim is that the degree of the vertex (u, v) in the Cartesian product of the graph will be the summation of the degrees of the vertex u in the first graph, and the degree of the vertex v in the second graph for that we observe here the following if I take this ordered pair (u, v), to how many vertices it will be adjacent with? So, it will be adjacent with 2 categories of vertices: category 1 of vertices where the second component is v and the first component is u' such that (u, u') constitutes an edge in the first graph that comes from the definition of the edge set of the Cartesian product of the graph.

So, you have (u, v) here it will be adjacent to all (u_1, v), (u_2, v), (u_n, v) if u is adjacent or if it is neighbor of u_1 in the first graph, if it is a neighbor of u_2 in the second graph and so on. That is a category 1 type of neighbors for this vertex (u, v) and category 2 neighbors of this vertex (u, v) will be all vertices of the form (u, v') where the u component is same here. And the second component v' is actually neighbor of the component v in the second graph. So, these are the 2 categories of vertices which will be adjacent to the node (u, v) in the Cartesian product of the graph the first category will have these many number of nodes: $\deg_{G1}(u)$, second category will have these many number of nodes: $\deg_{G2}(v)$ and that shows that this will be the degree of any arbitrary vertex (u, v) in the Cartesian product.

(Refer Slide Time: 22:03)

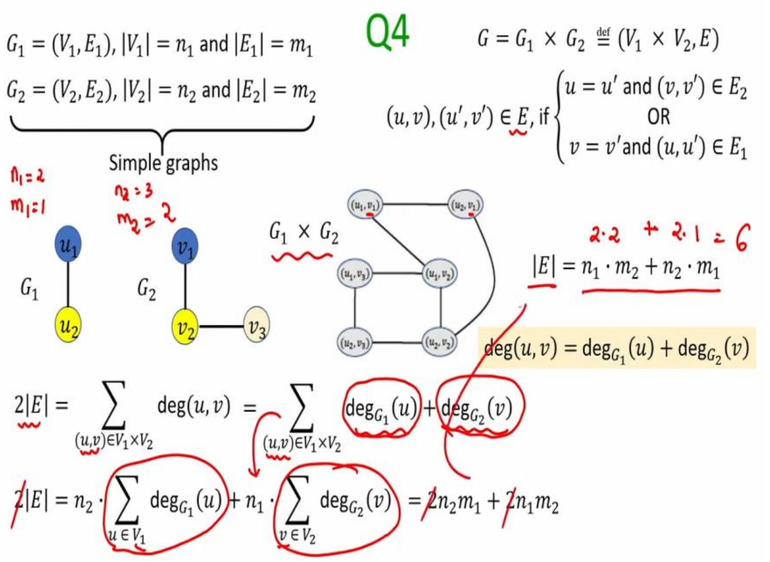


So, I have written down this result here $\deg(u, v) = \deg_{G1}(u) + \deg_{G2}(v)$ which will be useful now we will be applying the handshaking theorem on the Cartesian product and the handshaking theorem says that if I sum over the degrees of all the vertices in the Cartesian product that will give me the same value as twice the number of edges in the Cartesian product of the graph. Now I can substitute the value of degree of (u, v) as per this formula.

And now it is easy to see that this term degree of u in the vertex G_1 that will appear n_2 number of times in this entire summation. Why n_2 number of times? Because once it will occur when this (u, v) would have taken the value (u, v_1) again it will be encountered when this (u, v) would have taken the value (u, v_2) again this term will be encountered when I will be considering (u,v_{n2}) and so on.

So that is why the contribution of this term in the overall summation will be n_2 number of times. In the same way I can say that if I take this second term here in the overall summation here this term degree of v in the second graph will be appearing n_1 number of times; once it will come when I am summing over (u_1, v) again it will be coming next when I am summing over (u_2, v) and then finally it will be again coming when I will be summing over the vertex (u_{n1} , v).

(Refer Slide Time: 24:01)

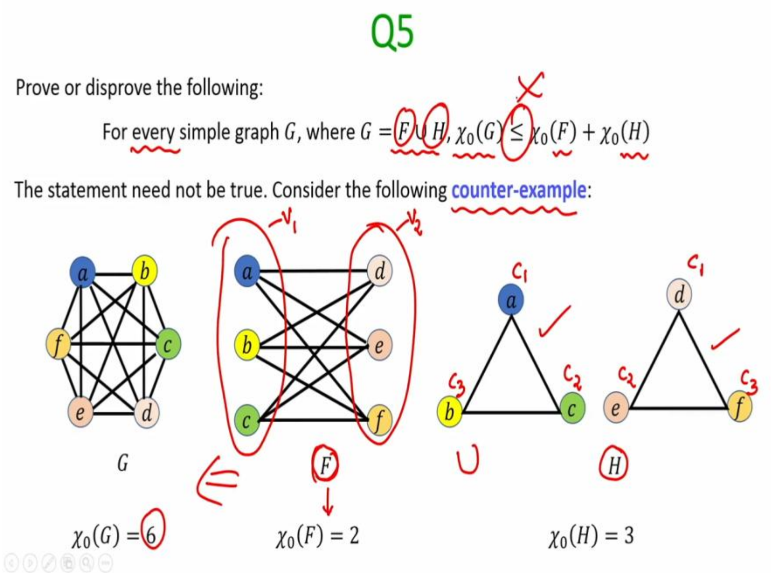


So, based on this observation I can say the following that this overall summation can be splitted down into these 2 individual summations. Since as I said that the contribution of this term will be n_2 times; it will be appearing n_2 number of times I can take n_2 outside in the same way the contribution of the second term will be n_1 number of times. So, I can take n_1 outside and then individually the summations will be now over single vertices namely over all the vertices in the first graph and all the vertices in the second graph.

And now I know that I can apply the handshaking lemma on the individual graphs G_1 and G_2 as well. So, if I take the summation of the degrees of all the vertices in the first graph I will get 2 times m_1 and in the same way if I take the summation of degrees of all the vertices in the second graph I will get 2 times m_2. And now if I cancel out 2 and 2 on both the sides I will get the cardinality of the edge set which I claimed earlier.

The summation of the degrees of all the vertices v in the second graph is twice the number of edges in the second graph which is 2 times m_2 and 2 and 2 cancels out and hence I get my desired result.

(Refer Slide Time: 25:31)



In question 5 we either want to prove or disprove the following. So, you are given a simple graph G where G is the union of 2 graphs namely the graph F and graph H; your graph F and H could be any arbitrary graph you have taken the union of those 2 arbitrary graphs and you have obtained the graph G. Now we want to prove or disprove that vertex chromatic number of the bigger graph G is always upper bounded by the summation of the vertex chromatic numbers of the 2 sub graphs irrespective of what are the 2 sub graphs.

So, intuitively it might look that the theorem is true because we are taking two small sub graphs F and H and combining them to get a bigger graph. And whatever is the number of colours that I need for colouring the 2 small things I will not require more than the combined number of colours to colour the bigger graph. That is an intuition you might get and you might end up saying that this inequality is true.

But we will prove that this inequality need not be true by giving a counter example namely we will give an example of a G. And an example of an F and H such that is inequality is not true even though G is equal to the union of F and H. So, remember this is a universally quantified statement because the claim is with respect to every simple graph. But the way I can disprove a universally quantified statement to be true is by just giving 1 instance a counter example for which violates that statement.

So, consider this complete graph with 6 nodes. And imagine that my G is the union of these 2 graphs. So, my F is a complete bipartite graph and H is now disconnected graph so you might be wondering how exactly I have constructed this instance of F and H. What I have done basically is I have taken this complete bipartite graph and I have put 3 vertices in one collection and the remaining 3 vertices in another collection.

So, as per the property of bipartite graph I cannot have edges within the collection a, b, c. And I cannot have the edges within the collection d, e, f. And since it is complete I need to give the edge between every vertex in the first collection and the second collection. But if I just take F then I am missing the edges within the first partition namely the edges involving a, b, c.

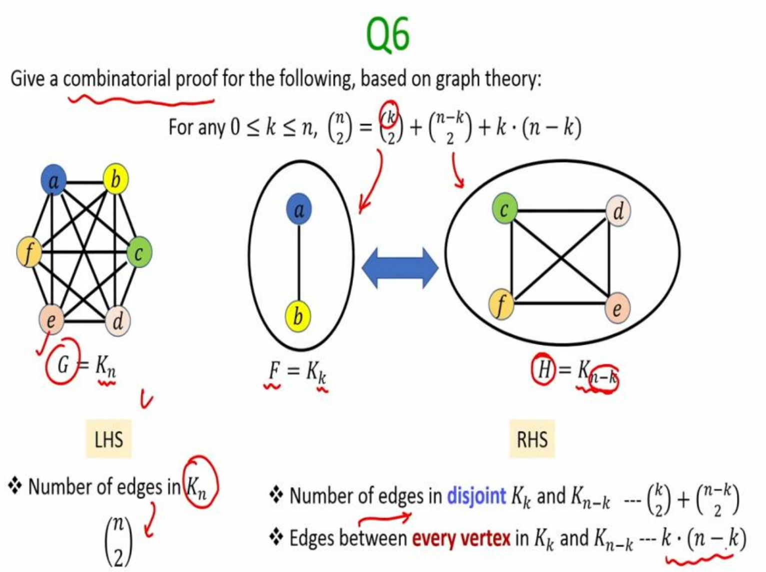
So that is why I am taking this triangle. And similarly I am missing the edges d, e, f. That is why I am taking the second triangle. And now I can say that if I take the union of F and H I will get this complete graph with 6 nodes because in the complete graph of 6 nodes a will have an edge between with b, c, d, e and f. So, a got all the edges involving d, e and f through the bipartite graph and the missing edges it is getting to the triangle graph and so on.

Now what is the vertex chromatic number of this complete graph of 6 nodes? It is 6 I need 6 colours. But what is the vertex chromatic number of this complete bipartite graph F it is 2 because any bipartite graph can be coloured with 2 colours so do not get the impression that I am colouring a, b, c with different colours in the graph F. So, I can colour all the vertices in the partition v_1 with colour number 1 and all the vertices in the partition number 2 with colour 2.

So, I just need 2 colours for colouring all the vertices of the graph F. And for colouring the graph H I need 3 colours. I can give C_1, C_2, C_3 I can give C_1, C_2, C_3. How many total colours

I need? Now for F and H 2 and 3 which is summing up to 5. So, now you can see that this inequality is not true for this instance of G, F and H. So that means the statement is not necessarily true.

(Refer Slide Time: 30:05)



Now based on this example I want to give a combinatorial proof for the following equality but the proof should be based on graph theory. So, remember what is a combinatorial proof? We give a counting argument and argue that expression in the left hand side and the expression in the right hand side give the count for the same number of things. And we do not do any kind of simplification or expansion and show that LHS and RHS are same.

What I am asking you here is that argument now should be based on some concepts in the graph theory. So, this is the equality which we want to prove. So, again what I will do is the following I can imagine that my left hand side expression is nothing but the number of edges in a complete graph with n nodes. So, if I take a complete graph with n nodes then the number of edges is ${}^{n}C_2$ (n choose 2). So that will be the interpretation of my left hand side.

Now I have to show that indeed the right hand side expression also counts the number of edges in a complete graph. How do we argue that so the right hand side expression brings a quantity k. So, you can imagine ${}^{k}C_2$ as the number of edges in a complete graph with k nodes. And the expression ${}^{(n - k)}C_2$ you can interpret as the number of edges in a complete graph with n - k nodes but I have to relate this k with it n somehow.

So, the way I can interpret the right hand side expression is that you have taken a complete graph and you have divided into 2 sub graphs, sub graph 1 which has only k nodes out of those and nodes and sub graph 2 which has the remaining n - k nodes I stress the k nodes which you are taking in F are disjoint from the n - k nodes which you are taking in the remaining complete graph.

Now whatever edges are there in the sub graph F they are definitely also present in your graph G. And similarly whatever edges are present in the sub graph H they are also present in the sub graph G. But there are still some edges which are missing; the some edges which are still there in the graph G but they are not yet counted because you have now till now counted only the edges which are there in F only the edges which are in H.

So, what you can do is if you imagine that there is an edge between every vertex in the sub graph F and every vertex in the sub graph H that will take care of the missing edges. And now if you include those edges as well in the edges which you have already counted that will give you the total number of edges in your complete graph with n nodes. But now how many edges I can have between every vertex in the set F and every vertex in the set H I will have k into n - k number of nodes.

So that is why the summation of these 3 quantities can also be viewed as the total number of nodes in a complete graph with n nodes and that shows that my RHS expression is same as the LHS expression. So, with that I conclude the first part of tutorial 9. Thank you.