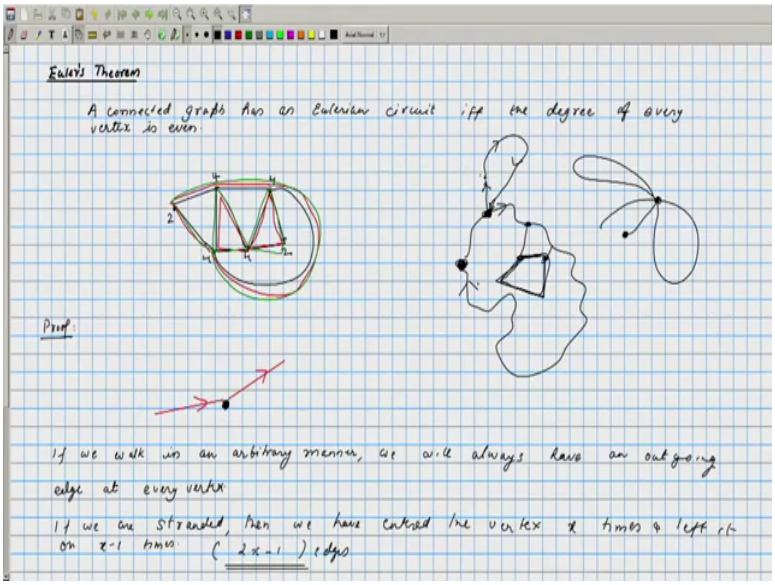
Discrete Mathematics
Professor. Sajith Gopalan
Professor. Benny George
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Lecture 12
Trees, cycles and graph coloring

In the previous lecture, we defined the notion, we introduce the notion of graphs, and then we defined when the two graphs are same we, talked about what is the isomorphism between graphs. And, then we saw a theorem called as Euler theorem.

(Refer Slide Time: 00:47)



With did not proved this so that is the first thing that we will do today. So Euler's theorem states that, a connected graph has an Eulerian circuit if and only if the degree of every vertex is even. So Eulerian circuit was a walk in the graph where no edges traverse twice every edges traversed exactly once. For example, if we consider the following graph, this graph will not have an Eulerian circuit as per the theorem because, if you look at these vertices they all have degree 3 but we can convert them into i mean if you modify the graphs this particular graph this will have an Eulerian circuit because the degree of vertex is even.

I am just writing down the degrees of the vertices this is not the vertex labels. So, this graph will surely have an Eulerian circuit and we will in fact construct one such and convert that construction into a proof. So, let us say we start from this particular vertex, so if we went about in this particular fashion there are lots of edges which are not traversed we can separately look at them.

This is a yet another one, so we drew this as we traversed this graph by means of three cycles, we could have done that more systematically we could have started from here and at this point, we can just come back here or again we can take this come back here then do this. So that gives us an Eulerian circuit.

So, the method by which we drew look like an arbitrary, which is arbitrarily trace the edges, we can essentially convert that method into a proof of our theorem, we will do that first thing. So, the key idea is since every vertex has degree even, if you enter any vertex you have to obviously leave the vertex. You will never be stranded at any particular vertex. So let us say we are randomly moving from one vertex to other, may not being particular clever about which is the outgoing edge.

For example, if we come to this particular vertex, one of the edges have been used, but, the fact that the vertex has an even degree means that there is at least one leaving edge. So for every vertex if we start at any arbitrary vertex and keep on moving that is tracing edges, the only requirement or the only restriction that we will impose on our walk is that we will not take the same edge twice. Now if you do this what happens, we can write the following statement.

If we walk in arbitrary fashion, we will always have an outgoing edge at every vertex, of course, there is particular case that we have to be careful about, suppose, we come back to a vertex and all the edges have been exhausted at that point. So, you start at some place and you move around and finally come back at this place and suppose these are only exactly two vertices then what we will do? So, that is one situation that we have to be careful.

It could be the case that once did this, there are other edges lying outside of this particular circuit. What we have ruled out so far is that if we reached some particular vertex after lot of travels, either it is one of the starting vertices we could reach at the starting vertex. If we are at in between vertex and you have no edges to move, that means you have come there some number of times and you have left that vertex one less time.

So formally we can say this as following, if we are let us say stranded suppose this a possibility that we are stranded. Then, we have entered the vertex the stranded vertex x times and left it only x minus 1 times. The starting vertex is different because starting vertex we never enter we just leave the starting vertex. So, but for every other vertex you are entering that from some other place and then you are leaving it. So you would have left it one less time than you have entered.

So, the total edges of that particular vertex which is been traversed is equal to that accounts only for 2 times x minus 1 edges, and that is an odd number of vertex and since we have assumed that every vertex has even number of even degree we will never be in such a situation. We could of course start at a vertex and then come back to same vertex, that is a possibility and at that point it could be that there are other parts of graph which are not traversed.

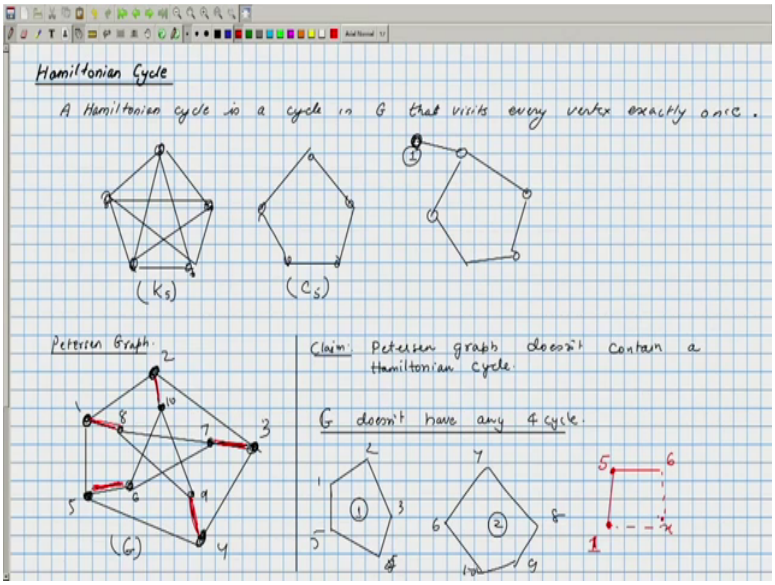
But here we can use a reduction now by virtue of this graph been connected, we can say that if there are other edges which are traversed, which are left behind then, there should be some connection to that to the left out edges and what we can do is first point from where there is an edge which is not accounted for, so look at this path and look at the first place first vertex which has an edge which is not used we could start their and then continue And we must essentially come back somewhere here.

We will never be stranded the only way we can be stranded is if we had come back here. So we can just traverse along the particular path, complete that cycle and then continue until original cycle. So this can be done inductively and therefore that will guarantee that if the degree of every vertex is even then we will automatically have an Eulerian circuit. The other part is simpler if we had some vertex of odd degree then there are no Eulerian circuits this can be seen because, if you look at the Eulerian circuit, the Eulerian circuit goes through every edge.

So, if you look at any particular vertex and count the edges it is, the Eulerian circuit has entered every vertex and left every vertex. The number of times it is entered is exactly equal to the number times it is left to that particular vertex, so since these are equal and together they are count for total degree of any vertex.

We will automatically get that the vertices must have an even degree every vertex must be of even degree. So that concludes the proof of Euler's theorem. So the next thing that we will learn today is another kind of walk or a path on graphs which are called as Hamiltonian cycles.

(Refer Slide Time: 09:55)



So, we will first define what are Hamiltonian cycle. A Hamiltonian cycle is a cycle in which every vertex of the graph is visited exactly once. It is a cycle in g so, we are talking about Hamiltonian cycle in a graph G. So, of course natural question to ask is where it does given a graph does it have a Hamiltonian cycle or not. So, we will see some examples, so if you take complete graph and vertices, this of course has many Hamiltonian cycles you can start at any vertex and go to any vertex in this particular graph.

So complete graph on n vertices this, is here the value of n is 5. So, k 5 has a Hamiltonian cycle if, you look at and say the cycle graph this is also a graph which contains a Hamiltonian cycle. Whereas, if you take this particular graph this does not have a Hamiltonian cycle because, the vertex if the number is as vertex 1 you start at vertex 1 you can never end up back at vertex 1 and, if you start at any other vertex the movement, you reach the vertex 1 you are stranded there.

And therefore this is graph, which does not have Hamiltonian cycle. So in these examples it was easy to see whether it contain Hamiltonian cycle or not. And like the Eulerian cycle problem, Hamiltonian cycle problem is not very easy to solve. So we will see a particular example. So, we will again look at the Peterson graph, so this is a graph on 10 vertices so this is 10 vertices and 15 edges and we want to know whether this graph contains a Hamiltonian cycle.

You can look at the graph for stair it for few minute and figure it out if it does contain a Hamiltonian cycle. If it does not contain a Hamiltonian cycle, what we have to show is that no matter how you walk in this graph you can never start at the vertex and reach back at the

same vertex after visiting every vertex exactly once. So in fact that is what we will show that this graph does not have a Hamiltonian cycle. Peterson graph does not contain a Hamiltonian cycle. Why is that the case? We will prove this claim via very crucial observation about Peterson graph.

So, let us call this Peterson graph as G, so first claim is G does not have any 4 cycle. This is a crucial fact that we are going to use. How do we see that this a graph without 4 cycle? Now if you look at this graph carefully, you will see that there are two cycles in this graph each of length 5. There are many 5 cycles but, you can see that this outer pentagon and the inner star which also can be drawn as a pentagon.

So, if I number this is 1 to the outer once is 1, 2, 3, 4, 5 and the inner once is a 6, 7, 8, 9, 10. So you can see that 6, 7, 8, 9, 10 also forms a cycle. So 1, 2, 3, 4, 5, is a pentagon or a 5 cycle and 6, 7, 8, 9, 10 also form a pentagon. Now, this cycles mean if you consider these cycles there are no other edges between them, for example between 1 and 3 there is no direct edge between 5 and 3 also there is no edge. So these cycles are two disjoint cycles and they do not have, if you just restrict the graph those vertices there are no smaller cycles in it.

So if I look at the graph consisting of just vertices just 1 to 5 they do not have 4 cycle and if I restrict to the graph to vertices 6 to 10 they also do not have a 4 cycle in it. And therefore we can say that, it if at all there was a 4 cycle it should have edges, which go from cycle 1 to cycle 2. There should be an edge any 4 cycle should have an edge which goes from the outer circle to the inner circle. And outer circle to inner circle the edges are we need to draw that in red, so these red edges at least one of this red edges must be present if, there is any 4 cycle.

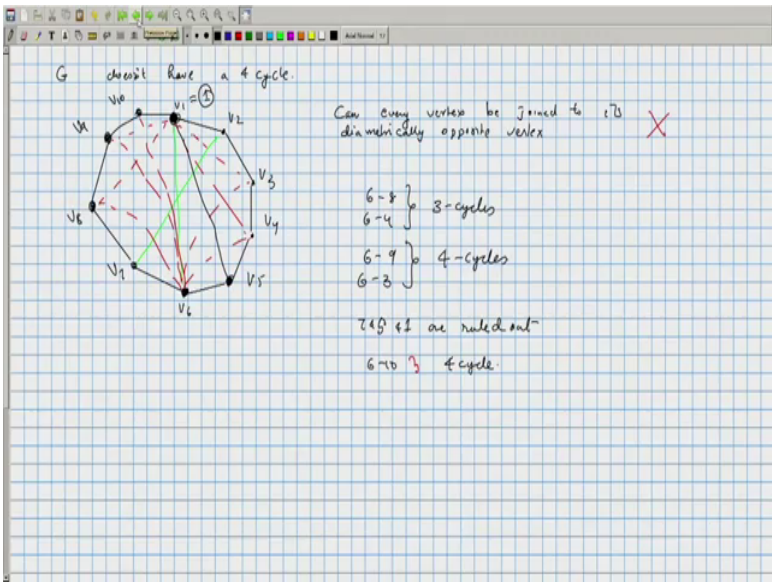
If there is a 4 cycle then that four cycle must contain one of these red edges. Without red edges, it is impossible to have a 4 cycle, what we will show is? Even with the red edges, there is no chance of having a 4 cycle. So let us say that one of these red edges is their all the red edges you can see that from the symmetry we can assume that all of them are one of same kind. So if there is a 4 cycle involved in one of red edges, then certainly there are 4 cycles involving in any other red edge as well.

That is from the symmetry of this diagram. So let us say 5, 6 is an edge which is there on some 4 cycle 5 to 6. Now 5 to 6 from 5 if there is 4 cycle there are only 2 possible edges 5,1 and 5,4 and from symmetry again we can say that we can just look at 5,1. If there was a 4 cycle 3 of its vertices are already in place 5, 6 and 1. If there is 4 cycle 6 and 1 should both be

connected to a common vertex. And you can see that there is now such vertex 6 and 1 and there is no other vertex such that, that vertex is connected to both 1 and 6.

And since there are no other, so that is you cannot have a vertex x such that 6 x is an edge and 1 x is an edge. So that basically means that Peterson graph does not have any 4 cycle. So how is this fact is going to help us show that Peterson graph does not have any Hamiltonian cycle?

(Refer Slide Time: 17:58)



So at that the moment we have shown that the graph G or Peterson graph does not have a 4 cycle. We need to show that g does not have any Hamiltonian cycle. So now suppose G did had a Hamiltonian cycle, we can say that all the vertices could have been traversed in a systematic manner. So let us say the ordering of the vertices is V1 to V2 to V3 so there is some ordering of which we can just place these 10 vertices. So I do not know whether V1 is the vertex 1 or V2 is the vertex 2 and so on but there is some order, I mean some way that you can traverse all of them.

So let us assume that starting vertex is V1. Now this accounts it means if you look at these 10 cycles that accounts for 10 of the edges in the graph. Now we can think about what are the other edges present. Just look at the Hamiltonian cycle if there was 1 and let us just draw it and we will get a cycle consisting of 10 edges. And there are 5 other remaining edges, which we have to add to this, get our original graph.

Where all can we add that edge V1 is an arbitrary vertex, if you had a Hamiltonian cycle, surely there is one starting at any particular vertex could have, because you look at the cycle start at look at any particular vertex in the cycle start at that point and traverse you will get

another Hamiltonian cycle. So if you have 1 Hamiltonian cycle you can have the Hamiltonian cycle start at any particular vertex. So we can assume that v1 is equal to vertex number 1. Now from V1 there are 3 edges 2 of them have already been accounted for the third edge must go somewhere. Where all can it go? So here we have shown that G does not have a 4 cycle, it is apparent that G also does not have any triangles, if you take any 3 of them they are not going to form a triangle. So G does have 4 cycle or 3 cycle. So V1 let say V3 or V9 that would make a triangle in the graph. The graph does not have triangle the same applies if you connect it to V4.

Because then you will have a 4 cycle and the same applies if you connect it to V8. So now the only 3 options are V1 could be connected to V7 V6 and V5. So V1 be can think of V6 as the diametrically opposite vertex. Note that all of these vertices 1 to 10 has 1 edge missing from there and together there are 5 edges missing. Now can it be the case that every vertex has joined to its diametrically opposite vertex along this particular path.

By diametrically opposite what I mean is V1 and V6 are diametrically opposite, V2 and V7 are diametrically opposite, V3 and V8 are diametrically opposite and so on. So can every vertex here be joined to its diametrically opposite vertex? If that was case V1 is joined to V6 and V2 is joined to V7. If every vertex is joined to its diametrically opposite vertex and this is what would have happened but that automatically creates a 4 cycle V1, V6, V7, V2. So this is not possible.

So there is at least 1 vertex which is not joined to its diametrically opposite vertex. So we may assume without loss of generality then that one vertex is V1. So V1 we may assume that V1 is joined to one of V5 and V7 and these are cemetery cases. So we will just say that V1 is connected to V5. So we can assume without loss of generality that V1 V5 is an edge in the graph. So our starting point was we assumed that there is a Hamiltonian cycle and we some have argued that Hamiltonian cycle has missing edges. And if you look at the starting vertex as 1 it is missing edge is towards goes to vertex V5.

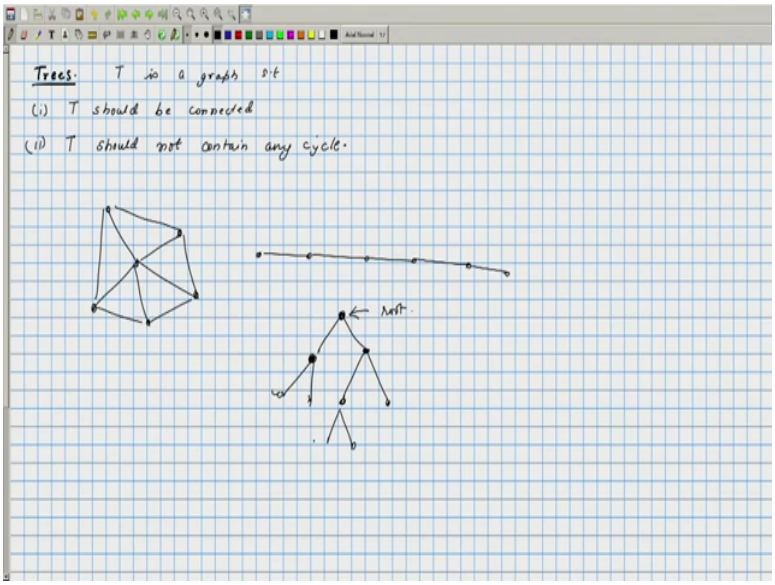
So this is part of the graph there are other edges also. So if you look at vertex 6, where all can 6 be connected to? We argued that V6 and V1 are not connected, so there are other possibilities but we will see that if you assume, I mean based on our claim that G does not have 3 cycle or 4 cycle, we cannot connect 6 to any other edges, clearly 6, 8 and 6, 4 are ruled out because of 3 cycle. V9 and V3 are ruled out because of 4 cycles, 6, 8 and 6, 4 causes 3 cycles, 6, 9 and 6, 3 causes 4 cycles. So they cannot because V6, V9, V8, V7 forms a

4 cycles so the only remaining vertices are V7, V5 any way are connected they cannot be considered. So 7 and 5 ruled out 7, 5 and 1 are ruled out. Only remaining vertex is 10 if you look at 6, 10 you again have a 4 cycle, 6, 10, 1, 5 forms a 4 cycle.

So this is also causes a 4 cycle and therefore we cannot have any edge out 6. So that contradicts are initial assumptions that there is a, I mean if we have this particular edge we look at Peterson graph surely there is an edge between 6 and 10, I mean some other vertex we argued that 6 cannot have any outgoing edge other than V7 and V5. So by this argument we could show that Peterson graph is a graph without any Hamiltonian cycle.

So this required careful examination of many of the properties of Peterson graph, we could not have a we did not have a general result like we had in the case of Eulerian cycle, Wherein we told that we just have to look at the degree of vertex if each vertex is the degree 2, then automatically every connected such graph was going to have an Eulerian cycle. So now we will move on to some different notions. So we learned about cycles, we learned about two different types of cycles Eulerian cycle and Hamiltonian cycle. The next special kind of graphs we will look at is what are called as trees.

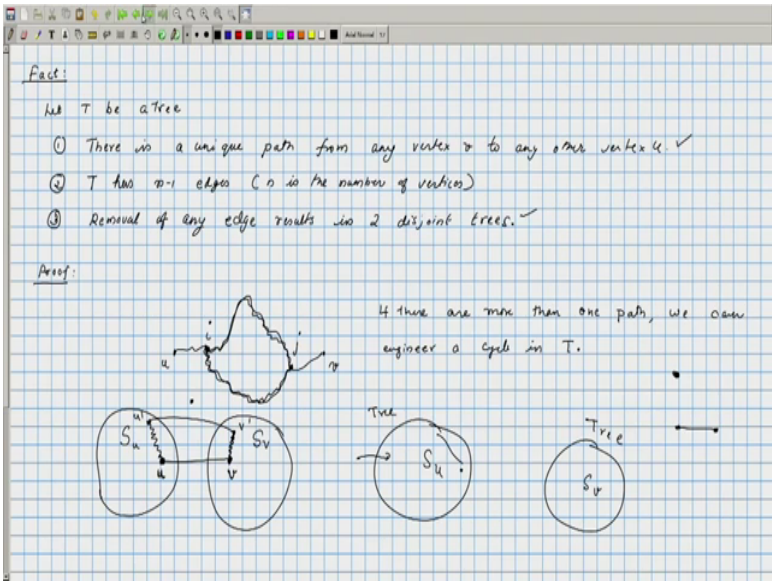
(Refer Slide Time: 26:32)

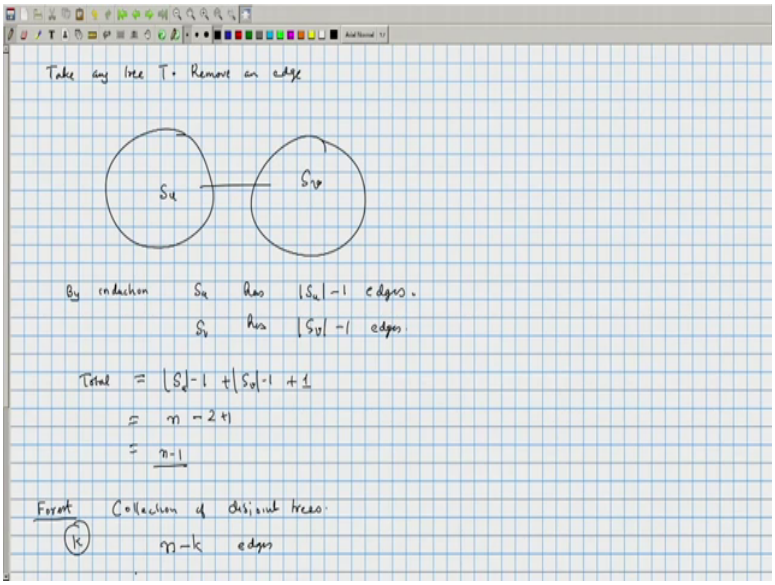


So trees are graphs, which has two crucial properties. So let us call it by T. So t also there is a graph first requirement is T should be connected. The second requirement is that T should not contain any cycle. So let us see some examples if you look at this graph on 6 vertices this is called as star graph. So this graph does not have any cycles and it is a connected graph. It is a line graph this is also an example of a tree.

If you look at wheel graph, this is not a tree because there are different cycles in this graph. You can have trees of different kind, so this an example of rooted binary tree, root because we usually but when we draw it like this, this signifies the root and binary because every intermediary note has two children, we will study about those things later but we can see that this is a graph in which there are no cycles and that it is a connected graph. So these kind of graph are called as trees and see some crucial simple properties of trees.

(Refer Slide Time: 28:32)





In a tree, let T be a tree first fact is there is a unique path from any vertex V to any other vertex U. So choose two vertices between those two vertices there is a unique path, whenever the graph is a tree. We will prove all these facts. The second facts is T has N minus 1 edges where N is number of vertices. Third fact is removal of any edge results in preciously two

disjoint trees. So if you take a tree and remove one edge from it you will get another graph which will have two connected components and both those connected components will itself be trees.

So we will quickly see a proof of these facts. So you look at any particular vertex U and any other vertex V, we want to say that there is precisely one path between them. So we may assume the contraries that is let us assume that there are more path between U and V. So let us say this is some path and then there is some other path. So if you look at these two paths there, is a first place where this paths divert. This paths start at the same vertex and, end at same vertex so clearly there is some place at which these paths divert.

So let us say this the point at which they divert and then of course they will together at some place. So if you call this first point as I and then there is surely first point after i where they come together because surely they come together at V so they must come together at some point and the first place where they together it is called as J. So in between I and J there are no common vertices, the paths do not have any other common vertices in between I and J. So these are all edges in the graph.

Now if you look at vertices I and J and if you start walking from I to J along the path 1 and then from J to I along the path 2 that is basically a cycle in the original graph. If there are two paths then surely there is cycle in the original graph. If there are more than one path, we can engineer a cycle in T because by looking at this two paths we can say that their will surely be a cycle in the original graph and the original graph by virtue of being a tree we know that it cannot contain any cycle.

So this cannot be the case that there are more than one path. So there is at most one path, how do we say that there is a at least 1 path? Well because this a connected graph between any two vertices you can go from 1 vertex to another vertex. So that proves statement 1 that there is a unique path. We will now prove statement 3 that removal of any edge results in two disjoint trees. How do we show this? We will let us look at one particular edge U, V so there is direct edge between U, V.

Suppose we removed this, now look all the vertices which are connected to U and all the vertices which are connected to V. Now after removal of this edge there cannot be any third kind of vertex which is neither, connected U nor connected to V because if there was one such vertex then it is not connected to either U or V in the original graph and therefore we argue that every vertex in the original graph either belongs to this set S U or S, V.

Now if you remove U V, S U and S V are not going to be connected, why? Because if they were connected then it would mean that from U to V there is more than 1 path because U to V there is a direct path and if S U and S V where connected by some other thing, clearly by virtue of any vertex been in S U there is path to U and here is a path to V. So you can go from U to this particular vertex U prime and from U prime we can go to V prime and V prime we can go back to V.

So that is an alternate path but by are first fact we know that there is at most there is exactly1 path. So we can argue that S U and S V are disconnected components. And by virtue of this being connected to I mean all these definition of all those things which are connected to U that is a connected path and of course there is no cycle in S U if there were cycle in S U then there is a cycle in original graph.

So, this is a cycle free connected path and therefore that is a tree. And this is also another tree so, we know the removal of any edge results in exactly 2 disjoint trees. And, now we can prove the third fact that any tree has at most N minus 1 adjust. So if you take any tree with 1 vertex clearly there are no adjust in it. So the base case would be that we are going to prove this statement by induction.

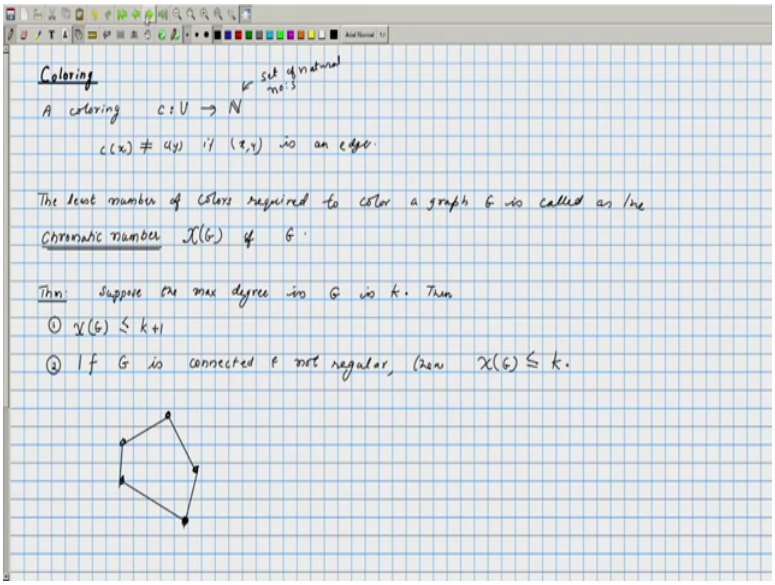
And the induction hypothesis would be true for the graph with one vertex. Now you take any other tree, suppose it is more than one vertex and surely it should have an edge remove that particular edge. So take any tree T and if the number of vertex is greater than 1 then surely there is at least one edge and remove any of those edge and then we would get two components, let us say S U and S V, S U is a tree and therefore the number of edges by induction.

So S U has size when we write size S U that means the number of vertices in that particular component minus 1 edges. And S V has size S V minus 1 edges. So, together the total number of edges that will be S U minus 1 plus S V minus 1, plus 1 for the edge U V that we had removed. So that would be size of S U plus size of S V that will be N minus 2 plus 1 which is equal to N minus 1. So any tree will have exactly N minus 1 edges we will just define something for later. So this is a notion of forest, a forest is just a collection of trees which are disjoint.

So if the collection had k disjoint trees and total number of vertices was N then we will have N minus K adjust in it. Which can be proven by the by using fact 2 repeatedly. We have just 1

connected component the number of edges was N minus 1. I will start of with the next topic, which is known as colouring.

(Refer Slide Time: 38:35)



So, we will define what is called colouring. So we have a graph and then we want to colour the vertices. What is a valid colouring is what we will first define then we will look at simple algorithm to colour a graph. So let us formally define what is a colouring? A colouring is basically a function from vertex this vertex set to natural numbers. This is a set of natural numbers and this one so any function from the vertex set to natural number is called as a colouring if it is satisfy certain requirements.

The requirement is C X should not be equal to C Y, if X, Y is an edge. All the way we are writing it as ordered pair X, Y since we are looking at undirected graphs they essentially mean the set X, Y. So each vertex is been assigned a number and for vertices which share an edge between them they should be assigned different numbers. So let us look at our Peterson graph if we give let us say red colour to this particular vertex then vertices 2, 8 and 5 cannot be given red colour they should be given a different colour.

These colour you could assigned them as numbers, so this should be a different colour. 8 can be the same the colour there is no problem because there is no edge between 2 and 8. So these vertices can be coloured in particular manner and then 10 can be given red colour itself 7 can again be given red colour, but 6 cannot be given red colour because 6, 7 is an edge it cannot be given yellow colour, so it should be given some other colour blue colour let us say.

Now if look at vertex 9 there also can be given blue colour because 9 is not, I mean the vertex 9 cannot be given either red or yellow. And if you now look at vertex 4 that can be given red colour and vertex 3 can be given blue colour. Vertex 3 cannot be given red or yellow because it is neighbours which is coloured with same colour. So here we can see that Peterson graph can be coloured using 3 colours. If you say red is 1 blue is 2 and yellow is 3 you get a function from vertex 8 set to natural numbers.

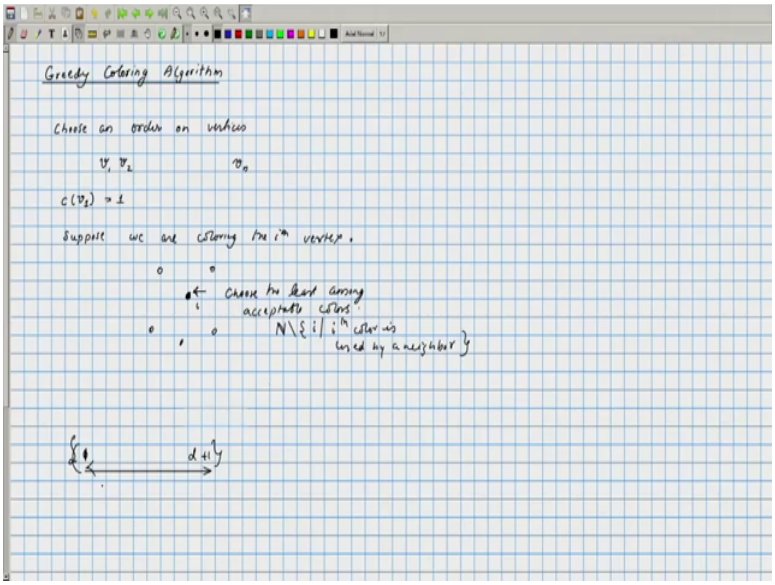
So that is the notion of a colouring. Clearly if you have N vertices you can surely colour it using N colours and if you take the complete graph on N vertices you would require N colours but what we are interested in is, if we given graph, how do we determine the minimum number of colours required to colour the graph, that we will define that particular property the least number of colours required to colour a graph G is called as the chromatic number of the graph.

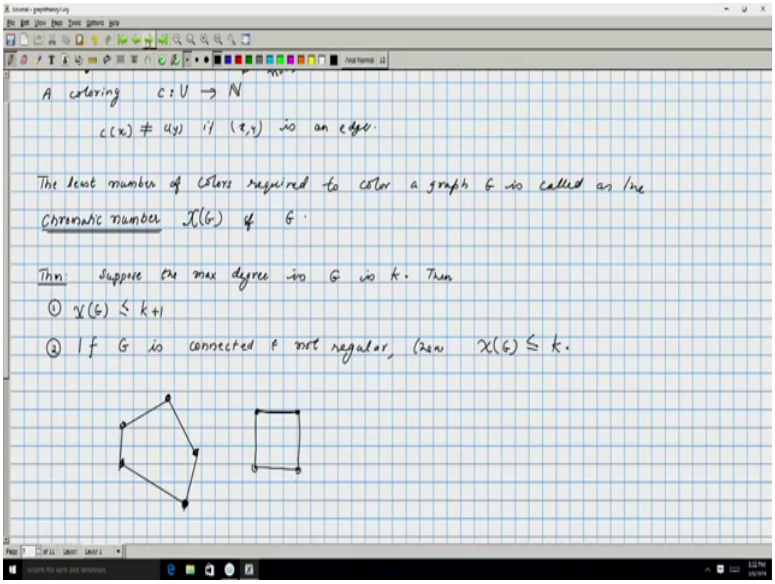
And this is denoted by kai G. So what we are interested in is given a graph, how do we determine its chromatic number? This is a difficult problem for arbitrary graphs, finding an algorithm which will determine this is not a easy task. We will look at a greedy algorithm for colouring and we will say the chromatic number is going to be certainly less than some quantity. So we will basically prove the following theorem.

Suppose the maximum degree in G is K that means if you look at vertex the maximum degree that it has among all the vertices, the vertex which is maximum degree, has degree K. Then kai G or the chromatic number is less than or equal to K plus 1. That is the first statement the second statement says if G is connected and not regular then, kai G is less than or equal to K. Not regular means so we will first define what are regular graphs. So a graph is called a regular if every vertex has the same degree.

If you look at our Peterson graph, look at every vertex its degree is exactly 3. So this is the example of a 3 regular graph. So if it is not regular then it will require less than or equal to k. If you look at this cycle graph, every vertex has degree exactly 2 and since the degree of vertex is exactly 2, kai G is less than or equal to 3 that is what first statement says but, this is an example of graph which is regular. If you had taken some other graph where the degree at most 2 but it is not a regular graph then you can argue that you don't require more than 2 colours. So the proof of this theorem will be via an algorithm, so we will show that the greedy colouring algorithm.

(Refer Slide Time: 45:45)





We will correctly colour the graph without using more colour. So what is the greedy colouring algorithm? The algorithm is very simple, so let us describe the algorithm systematically. So, choose an order on vertices, so let us say you are going to examine vertices in this particular order V1, V2, Vn and colour of V1 we will assume to be 1 and at any stage suppose you are colouring ith vertex look at the least number that can be assign to that without violating the colouring property.

So look at the vertex I and look at its neighbours, so they would have got some colour and you can choose for I the colour which is least amongst the acceptable colours. So acceptable colours would mean any number which is not a colour of its neighbour. So I such that ith colour is used by a neighbour. Consider the set of colours that neighbour has already used and

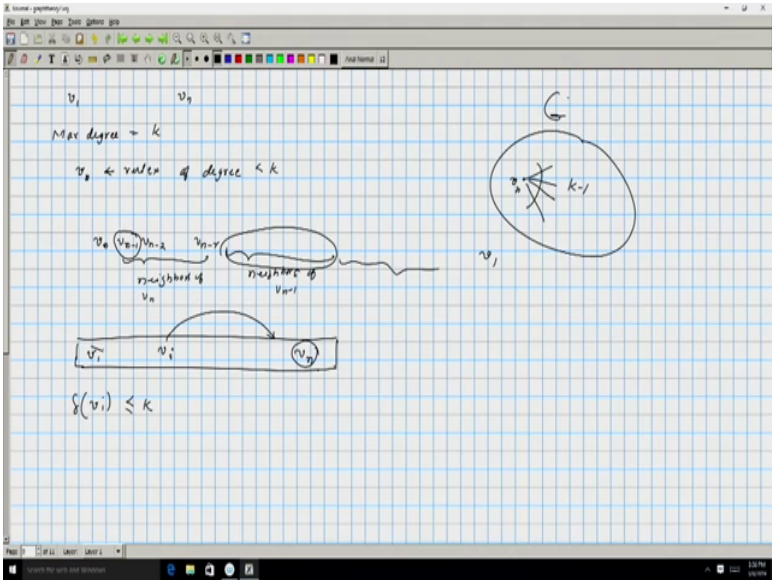
remove those colours from natural numbers and whatever is the least remaining colour that would be the least amongst the acceptable colours and that is the colour you choose for I.

This is the greedy colouring algorithm. Now if you look at the greedy colouring algorithm whenever you are examining a particular vertex since it at most the neighbours if you look at 1 to D plus 1, if look at the set of 1 to D plus 1 amongst these numbers at least 1 number would be missing and that number can surely be chosen as a colour for I. Now we are using colours from 1 to D plus 1 and using these colours you can colour the entire graph and therefore, the greedy colouring algorithm finds an acceptable colouring where you do not use any colour greater than D plus 1.

The second part of the theorem states that, if you have a connected graph which is not regular then you can colour it with less than or equal to K colours. So max degree is K and if the graph was not regular than K colours suffice. Clearly if the graph was regular then we cannot guarantee that it will be colourable in K using K colours. For example, if you take pentagon, which is a 2 regular graph, we can see that we require at least 3 colours.

But if you look at let if say the square this is a two regular graph and it can be coloured with 2 colours. The second condition just states that if you are guaranteed that is not regular, then you do not need more than the max degree number of colours. So let us prove that part again we are going to use our greedy colouring algorithm and what we will do is fix the order in which the vertices are to be coloured.

(Refer Slide Time: 49:56)



So we want to fix the order of vertices V1 to Vn, we will start describing this by first stating what is V N. So what we know is that max degree is equal K and this is not k regular graph this is a not regular graph, if is not regular graph then it means that there is at least 1 vertex whose degree is less than k. So we will choose that vertex as Vn. So Vn is the vertex of degree less than K. So there is vertex whose degree was strictly less than K and we will take that vertex as Vn. So if you take Vn it is going have lot of neighbours in the graph.

So suppose this is the graph G and V N is the particular vertex it is connected to some other vertices. There it can be at most K minus 1 of them, we first write all of them in any order, so V N minus 1, V N minus 2, up to V N minus R. So only requirement is R plus 1 will be less than K. And then we will take V N and look at neighbours of V N, so write down neighbours of V N minus 1 so these are neighbours of V N minus 1. And, then we can write down neighbours of V N minus 2 and so on.

This is the first block is neighbours of V N and second block is neighbours of V N minus 1 and so on. So of course when you are writing neighbours of V N minus 1, if you have included if there is some common neighbours with end you do not include them. So all the fresh neighbours of V N minus 1 neighbours which is not been included so far is included into this block and you do systematically since, it is connected graph this will list out all the neighbours and finally you will get V1.

So if you look at vertices in the order V1 to V N and apply greedy colouring in this particular order we can argue that we do not require more than k colours at any point. The reason being look at any vertex that you pick let us say V I is some particular vertex one of its neighbour is

somewhere ahead, there is at least one neighbour which is further ahead except for V N, V N does not have any neighbour ahead of it all its neighbour are appeared and would have appeared previously. So if you take V I and look at all its neighbours and since at least one of the neighbours is in front the number of colours used by the greedy colouring algorithm to colour the neighbours V I would be at most K minus 1.

Because V I degree of V I going to be less than or equal to K, but amongst these since at least 1 neighbour is in front of V I it comes after V I we know that the colours so far used will strictly less than K. So there is at least 1 colour left amongst colour 1 to K and that colour can be used for V I, of course this argument does not work for V N but V N any way guaranteed that its degree is less than K because that is how we choose V N. So even for V N we will have a colour left with when we are colouring via the greedy colouring method. So, that concludes the proof.