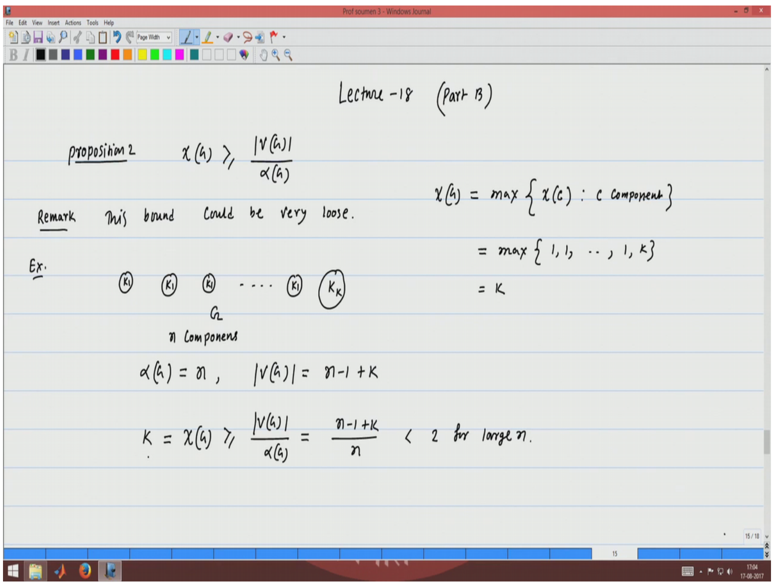
Graph Theory
Prof. Soumen Maity
Department of Mathematics
Indian Institute of Science Education and Research, Pune

Lecture – 18
Part 2
Chromatic Number and Max. Degree

Welcome to the second part of lecture 18 on Graph Theory. In the first part of this lecture we talked about 2 lower bounds for chromatic number of a graph. That chromatic number of a graph is always greater than equal to the tick size of the graph, and the other one is that the chromatic number of the graph is greater than or equal to the cardinality of b by maximum independent set size.

(Refer Slide Time: 01:00)



So, the second bound that we got is the proposition 2 probably that chi G is greater than equal to V G by alpha G. Now one remark sometime this bounce works fine, but they could be very loose also sometime. So, this bound could be very loose, and we give an example to sort of support this remark. Let me concentrate this graph. This graph has n components, n components. The first component is k 1; that means, just one vertex.

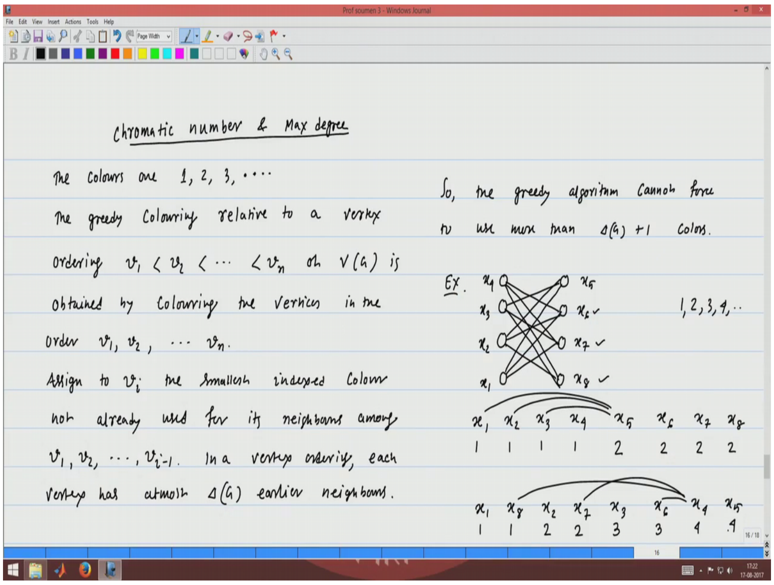
The second component is also complete graph with one vertex; that means, just a single vertex. And you have such components n minus 1 such components right. And then the nth component is complete graph with k vertices. So, this is the graph G it has n components is the graph G. Now what is the size of maximum independent set for what is alpha G here? Alpha G is clearly n because you can pick in the independent set this n minus 1 vertices from n minus 1 components, and one vertex from k k. So, that will give you independent size maximum independent set of size n. And what is the number of vertices here V G is equal to here you have k vertices, and in this n minus 1 components you have n minus 1 vertices, so n minus 1 plus k.

Now, what is the chromatic number of this graph G the chromatic number of this graph is like we have used that you can use that result that chromatic number of a graph G is maximum of chromatic number of different components. So, c is a component. So, this component required one color this component required, one color this component required one color, but this component required k colors. So, the chromatic number of this graph is maximum of 1, 1, 1 and k which is equal to k right.

So, as we can see that the chromatic number is k, and our this bound gives that chi G is greater than equal to V G by alpha G, which is n minus 1 plus k by n. And this will be less than 2 for large n right. That is easy to see, but actual chromatic number is k. So, the lower bound says that the chromatic number is something less than 2, but actual chromatic number is k. So, that is mean for this graph the bound given by this proposition 2 is very loose ok.

So, we talked about 2 lower bounds for chromatic number. Now we move to upper bound for chromatic number.

(Refer Slide Time: 06:54)



And we a give a greedy algorithm to color the vertices of a graph. So, chromatic number and it is relation with maximum degree. And also this talk about the upper bound for chromatic number. So, I will start with an algorithmic technique to color the vertices of a graph, the colors are 1 2 3 like this. You are not really going to use different colors we will use numbers instead of colours ok.

Now, what is this greedy colouring algorithm? The greedy colouring relative to a vertex ordering v 1 v 2. So, this colouring algorithm depends on how you order the vertices of the graph G. And the greedy colouring relative to a vertex ordering is obtained by colouring the vertices in the order v 1 v 2 v n. So, you choose an arbitrary ordering of the vertices. This algorithm does not say how you ordered the vertices.

So, you have to just decide on an arbitrary ordering of the vertices. And then you colour the vertices in this order you colour the vertex v 1 first. And then v 2 first with this specific rule you assign to v i in the ith step assign to v i the smallest indexed colour not already used for it is neighbour among. I will explain this one among v 1 v 2 v i minus 1. And in a vertex ordering, what about ordering you take it does not matter each vertex has at most delta G neighbour on the left or earlier neighbours.

So, the greedy algorithm cannot force to use more than delta G plus 1 colors. So, this is what the algorithm is, but I need to explain it. So, what you do is that a given a graph you just decide on an arbitrary ordering of the vertices, and then you colour the vertices in this order only. So that means, you colour v 1 first and then v 2 first. So, when you are assigning a color to v 2 for example, you use the smallest index colour not already used for it is neighbour on the left hand side; that means, you use the smallest index colour for v 2 which is not used for v 1 provided v 1 is neighbour of v 2. And the number of colours required to greedily colour the graph it depends on what ordering you have taken.

If you change the ordering of the vertices the number of colours required to colour the graph might differ. So, I will give an example that how the number of colours required by the greedy algorithm changes when you change the order of the graph order of the vertices. So, let me give this example. So, I considered a bipartite graph, say this one. You have 4 vertices in the left side and 4 vertices on the right side and I call them or level them x 1 x 2 x 3 x 4 x 5 x 6 x 7 x 8.

So, x 1 is adjacent to x 7 x 6 x 5 x 2 is a adjacent to x 8 x 6 and x 5. X 3 is adjacent to x 5 x 7 and x 8 and x 4 is adjust adjacent to x 6 x 7 and x 8. Now the algorithm does not know that whether the this is a bipartite graph or some other graph. So, it will decide on some arbitrary ordering if I choose this ordering for example, is x 1 x 2 x 3 x 4 x 5 x 6 x 7 and x 8.

Suppose this is the ordering I decide one and then I will start colouring with x 1 and I will colour this vertex x 1 1. So, I give colour 1 2 x 1. Now when I colour x 2 I will use the smallest index colour not already used for it is neighbour among the previous on it is it is neighbour on the left hand side. So, x 2's neighbour on the left hand side is no neighbour on the left hand side. So, I can use the smallest index colour that is available. So, I can use one again for x 3, it does not have. So, x 1 and x 2 are not neighbour of x 3. So, I cannot you I can use the same colour one x 4 x 1 x 2 x 3 are not neighbour of x 4.

So, I look at the neighbour of a vertex x I on it is left hand side only. I do not look at the right hand side at this moment. So, I can give colour one. Now for x 5 I look at it is neighbour on the left hand side. So, x 5 has neighbour x 3, x 2, x 1, x 3, x 2 and x 1. So but all of them are coloured one, so the colour among the colours 1 2 3 4. I can use the smallest index colour for x 5 that is 2 right. For x 6 again I look at it is neighbour on the left hand side and the neighbour of x 4 are sorry, neighbour of x 6 are x 4 x 2 and x 1 all of them are coloured one. So, I can use colour 2 for x 6.

Similarly, for x 7 you can check that all it is neighbour on the left hand side I will coloured one. So, I can use colour 2 for x 7 and similarly for x 8 I can use colour 2 again right. Now it is good that you got a the greedy algorithm coloured this bipartite graph with 2 colours, as I said that if you change the ordering of the vertices the required number of colours might change. So, if you take this ordering for example, say x 1 x 1 because you can start with any arbitrary ordering, and the number of colours required depends on what ordering you choose.

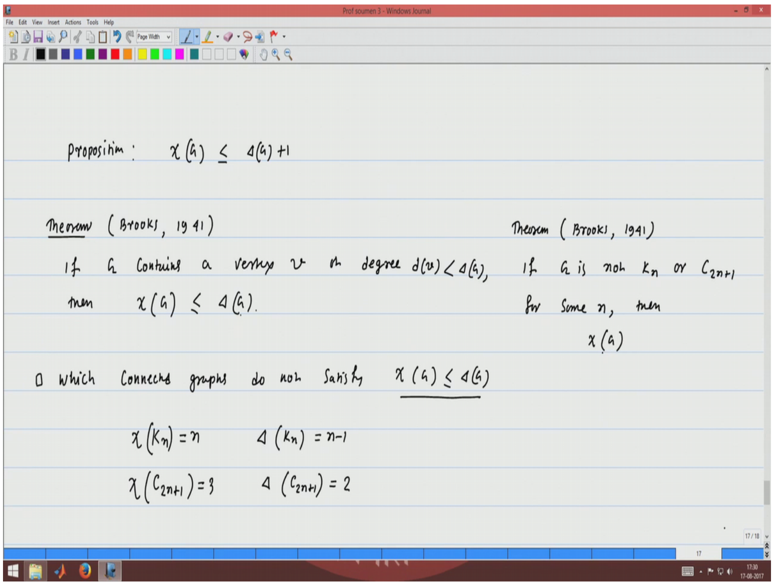
So, x 1, x 8, x 2, x 7, x 3, x 6, x 4, x 5. Now if you choose this ordering, because the ordering is an arbitrary ordering you do not know in which order you have to color the vertices. So, if you choose this ordering then first for x 1 you can use colour one fine for x 8 it has no neighbour on the left hand side. So, again you can use the colour one. Now for x 2 has neighbour on the left hand side 8. So, you cannot use colour one again for x 2 because x 2 which is adjacent to x 8. So, you have to use the smallest index colour which is has been not used for it is neighbour on the left hand side. So, you have to use colour 2 for x 2.

Now, look at x 7, x 7 has neighbour x 1 and x 1 is coloured with one. So, for x 7 you have to use the smallest index colour available that is 2. Now go to x 3, x 3 has 3 neighbours x 5 x 7 and x 8. So, the neighbour of x 3 is x 7 and x 8 on the left hand side. So now, you can see that the 2 neighbours of x 3 they are already coloured 1 and 2. So, x 3 cant. So, x 3 has to use the next smallest index colour available. So, x 3 will be coloured with colour 3. And similarly x 6 is on the left hand side it has neighbour x 2 and x 1 x 2 and x 1.

I can remove this 2 now and they are coloured 1 and 2. So, the smallest index colour not used for it is neighbour on the left hand side that is 1 2 are already used. So, you have to use colour 3 for x 6. Now look at x 4. X 4 has neighbour x 6 x 7 and x 8. X 8 is one neighbour for x 4 on the left hand side. X 7 is another neighbour for x 4 and x 6 is also a neighbour of x 4. And so, the neighbour have already used the colour 1 2 and 3. So, you have to use the next smallest available colour that is 4, smallest indexed colour that is 4. And similarly for x 5 you can check that you have to use colour 4.

So, for this ordering you can see that the number of colours used by the greedy algorithm is 4. So, this explains that how many colours you need to use by the greedy algorithm that depends on completely on the ordering of the vertices how do you order the vertices. And this algorithm does not talk about how to order the vertices. So, that the number of colours required will be less. So, this example explains that if you change the ordering of the vertices you required different number of colours right. So, we talk and clearly when you are colouring the ith vertex v i, it can have maximum delta G neighbours on the left hand side.

(Refer Slide Time: 24:12)



On it is left because the degree of this vertex is maximum delta G right. And it might happen that all the neighbours on it is all the delta G neighbours they have different colours.

So, for v i you have to assign delta G plus 1 the colour delta G plus 1. So, that is why this proposition is a obvious consequence of the greedy algorithm, that this gives upper bound that delta G is always less than or equal to delta G, sorry chromatic number of G is always less than equal to delta G delta G is the maximum degree of the graph G plus 1. So, this is also explained in the in the greedy algorithm, that in any vertex ordering each vertex v i has at most delta G neighbours on it is left. So, the greedy algorithm cannot force to use more than delta G plus 1 colours.

So, this clearly says that the chromatic number of the graph G is always less than equal to the maximum degree plus 1. So now, just for information we talk about some theorems results known in this area theorem, brooks theorem 1941. It says that if G contains a vertex v of degree d v strictly less than delta g; that means, it is not a delta G regular graph. That is at least one vertex which has degree less than the highest degree.

Then you can improve this upper bound slightly that delta G sorry the chromatic number of the graph G is less than equal to delta G. So, in this case when there is a vertex of degree strictly less than delta G, you can this theorem gives an ordering of the vertices which requires depth first search, which is a technque in algorithm you start with one vertex and systematically you visit all the vertices which are accessible from the vertex v. So, since we do not know depth first search algorithm we will not talk about the proof of this algorithm.

But the outline of this outline of the proof of this theorem that you start with vertex v which has degree strictly less than delta G and applied d f s. And so, the d f s will give you ordering of the vertices. And you colour the vertices of the graph in a reverse order of the d f s. So, this is just for information that you can improve your result slightly instead of delta G plus 1 it can be delta G only. Now one which connected graphs do not satisfy this condition. This is just a remark of observation that chi of G is less than equal to delta G.

You look at this complete graph with n vertices. We know that chi of k n is equal to one sorry, is equal to n. And delta of k n is equal to n minus 1. So, here this connected graph k n the complete graph with n vertices does not satisfy this condition. Another graph is cycle with odd number of vertices c 2 n plus 1. And we know that what cycle has chromatic number 3 and the maximum degree of cycle c 2 n plus 1 is equal to 2.

So, chi of c n plus 1 is delta plus 1 basically right. So, here is the other theorem again by brooks 1941, that if G is not k n it is not a complete graph or not a odd cycle c 2 n plus 1 for some n, except this 2 graphs for the other graphs chi of G. Of course, I am talking about connected simple graphs chi of G is less than equal to delta G. So, that is all. So, we have learnt what is a chromatic number of a graph. And we talked about it is lower bound using the clique number maximum clique size and maximum independent size, size of maximum independent set. And also we talked about upper bound based on the maximum degree of the graph G. That is all for today.

Thank you very much.