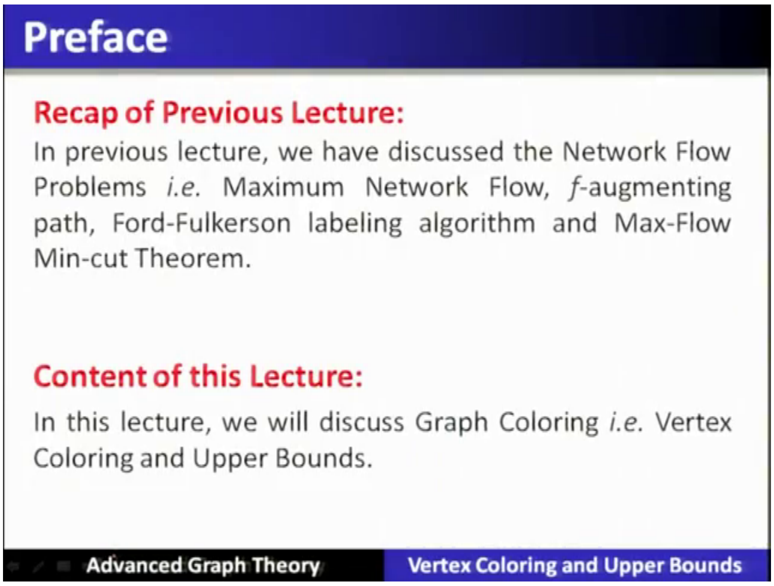
Advanced Graph Theory
Prof. Rajiv Mishra
Department of Computer Science and Engineering
Indian Institute of Technology, Patna

Lecture - 16
Vertex Coloring and Upper Bounds

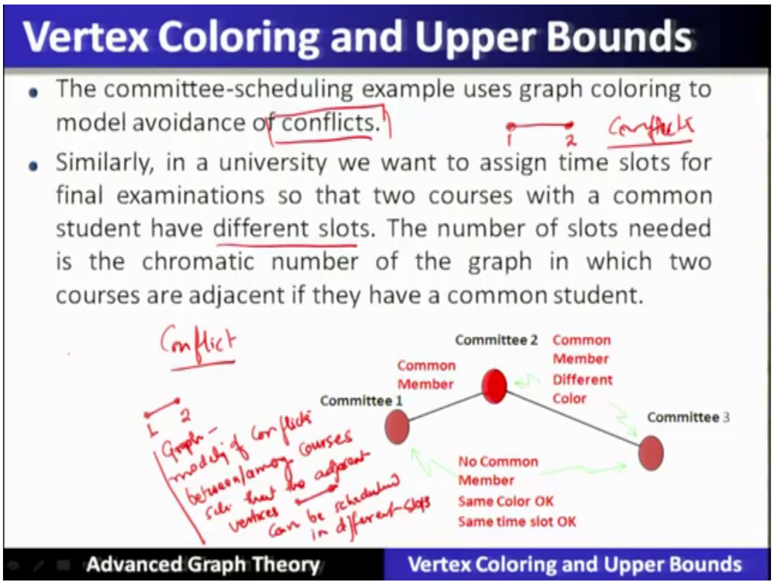
Vertex Coloring and Upper Bounds.

(Refer Slide Time: 00:17)



Preface recap of previous lecture, we have discussed the Network Flows maximal; Maximum Network Flow, f-augmenting path, Ford-Fulkerson labelling algorithm Max-Flow Min-cut Theorem. Content of this lecture: this lecture we will discuss graph coloring that is; the vertex coloring and it is upper bound.

(Refer Slide Time: 00:44)



Vertex coloring problem rises in many applications.

For example wherein the two committees if they have a common member, then they can basically join with an edge. So, the graph which may form out of these particular conflicts which will model the conflicts can be applied graph coloring. So, that we can schedule with a minimum number of slots a different committees meeting.

So, this particular model if it is modelled form of a graph and applied the graph coloring algorithm. Graph coloring problem then it will be basically nothing, but it is a modelling and solving the problem of the conflicts another such problems is about finding out the timeslot for the examination.

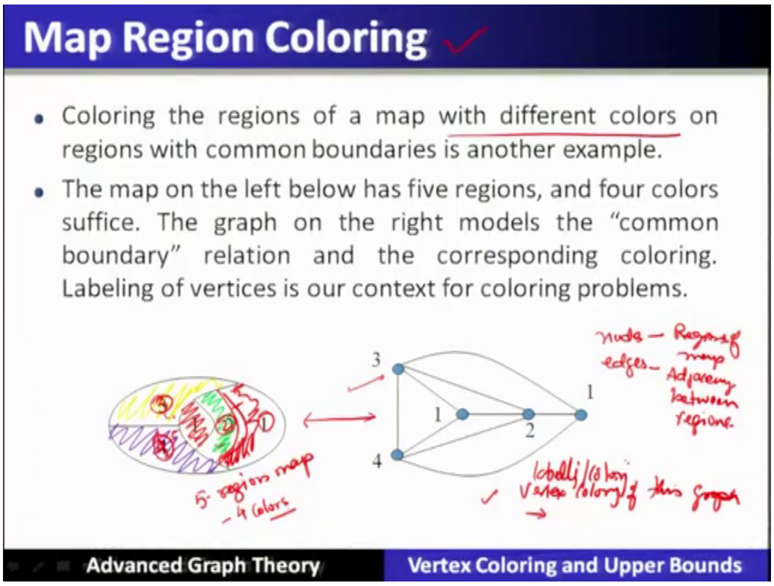
Now, here in these particular scenario two courses having the common students cannot be scheduled in the same time slot. So, the two courses having a common students can basically be represented again with the help of the graph. So, the graph is nothing, but the modelling of the conflicts between courses such that the adjacent vertices like this can be scheduled in different slots hence this way of representing the problem of you know t examination is nothing, but modelling the conflicts and then the graph coloring or the chromatic number of the graph is the minimum number of slots required to conduct the entire examination.

So, again this is another such problems so; that means, whenever there is a problem of a conflict it can be modelled in form of a graph and the chromatic number of that particular graph; will become the minimum number of slots or minimum number of timeslot required to solve that particular problem.

So, hence how this particular; coloring is to be done in these graphs which are modelling different problems for example, committee scheduling with a minimum number of slots similarly examination to be conducted with a minimum number of slots. So, that kind of particular problem can easily be modelled in a form of a graph and that will basically captures the conflict relation and then basically the chromatic number; that means, what is basically? How the coloring is to be done in that particular graph.

And that number of colours minimum number of colours required, that is; called chromatic number of the graph will tell, what is basically the best possible way that particular problem can be solved.

(Refer Slide Time: 04:48)



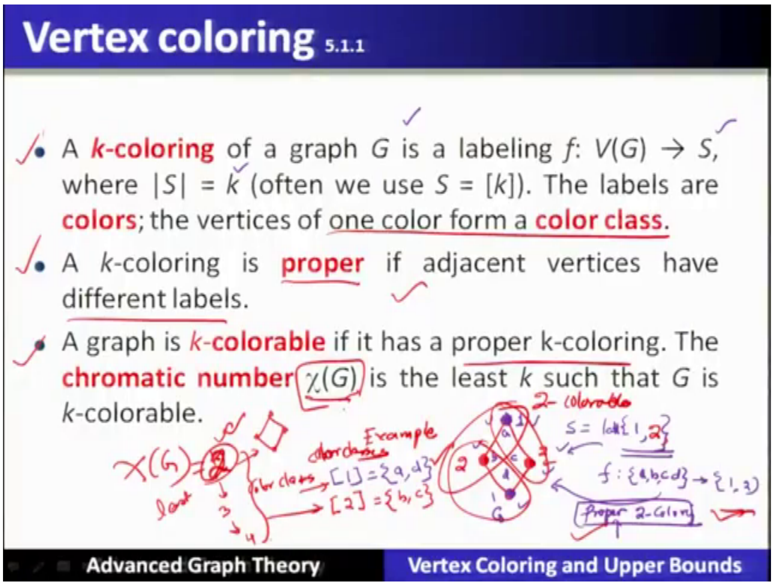
Another such problem of coloring arises in a map coloring. So, coloring the regions of a map with different colours on the regions with common boundaries another such example called map coloring problem. So, on the left here we can see that it has five regions: 1, 2, 3, 4, 5, but the constant the different the adjacent region should not receive the same colour at the same region should receive the different colours.

So, the adjacent regions are receiving different colours that we have already shown you. So, that no two regions which are basically having the common boundary are receiving the same colour they are receiving different colours.

So, how many colours are required one two then this is already done three and four different colours are required to basically map this five region map. This particular map can also be represented in a form of a graph, where these regions are represented as a notes regions of a map and edges are the adjacent relation between the regions.

So, having model this particular a graph, now we apply how many minimum number of colours required in this particular vertex coloring of this graph that will be the solution for map coloring region. So, this particular labelling of these vertices or we also say it is a coloring of the vertices will be the context for coloring problems.

(Refer Slide Time: 07:19)



Vertex coloring k-coloring of a graph G is a labelling function f; which maps vertices of G to S where S is basically set of k different values these are called labels are these labels are the colours the vertices of one colour will form the colour class. So, a k coloring is a proper if adjacent vertices have different labels a graph is k-colourable if it has a proper k coloring the chromatic number of chi of G is the least k such that G is three colourable.

So, take this particular example. So, let us take a graph of having four vertices now we do the labelling. So, let us call it as labels. So, let us say how many labels are required for this particular graph G; this label is proper if the adjacent vertices are receiving different colours. For example, if we colour and say that this particular colour is having label 1, then these two adjacent vertices cannot receive the same colour.

So, let us give it another colour and let us call it as two, this vertex is concerned this vertex can also be coloured with the same colour to and yet it is a proper coloring. Now as far as this vertex is concerned this vertex cannot be coloured with colour number 2. So, it can be coloured with colour number 1. So, with only two labels we can basically colour these particular a graph so; that means, labelling is nothing, but the set of vertices let us say a, b, c, d they are being mapped to a set S and that S is 1 oblique two and we have shown this particular mapping here in this particular graph.

So, we have found out a proper here two proper two labelling this particular labelling, whether it represents these numbers are if this numbers are immaterial are not having any significance then we can call them as a colour. So, this is called proper you see every two adjacent vertices are receiving different colours. Hence, this particular coloring is called proper two coloring is required, because with one colour you cannot colour this particular graph. So, minimum two colors are required.

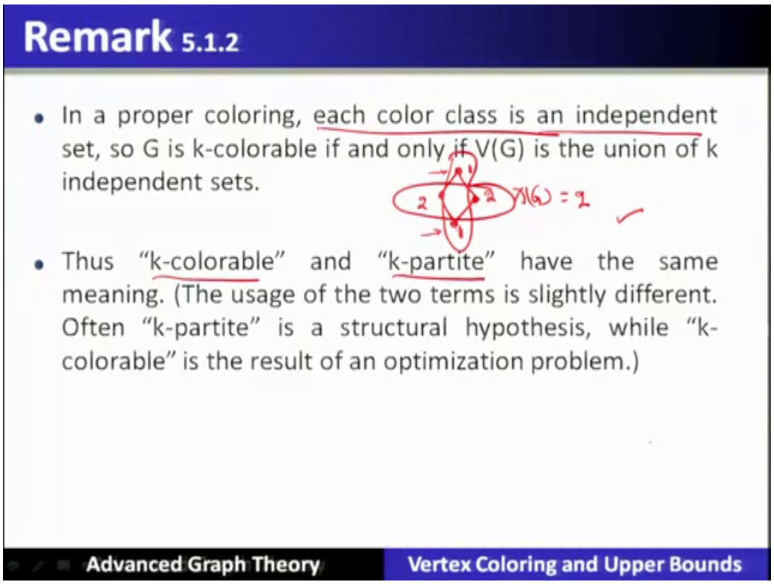
So, we have obtained a proper two coloring of this particular graph and let us revisit the definition again. So, a k coloring of a graph G is the labelling which is nothing, but a function of vertices of G, which will map to some set of labels let us call k different labels. So, these labels are called colours why? Because, they are values are immaterial from with set they belong that also is not having any significant. So, let us call it as colours the vertices of one colour for example, the vertices having colour one are represented as a and d. Similarly, the vertices of colour two are represented b comma c and they are called colour classes.

So, the vertices of one colour is called the colour class. So, in this example we have shown you two colour classes 1 and 2. Now a k coloring is a proper adjacent vertices have different labels here you can see vertex a and b, they are adjacent, but they have colours or labels 1 and 2, they are not having same hence any two vertices any pair of vertices which are adjacent you pick they may receiving different colours or a different labels hence this coloring is proper.

So, graph is k-colourable you would has a proper k coloring. So, we have shown a proper two coloring of a graph, hence this particular graph is 2 colourable. Now the chromatic number chi of G is the least cases that G is k colourable. So, here we have taken the value two, hence chi of G is 2. So, the chromatic number of this particular graph which we have shown you which we have taken in this particular example is 2, because this is the least value you can also colour with three number of colours you can also colour with four different numbers of colours, but this is the least of these all hence this particular least value k such that this graph is k-colourable this value becomes two here in this particular example.

So, a graph is k-colourable if it has proper l coloring the chromatic number chi of G is the least k such that G is k colourable.

(Refer Slide Time: 13:53)

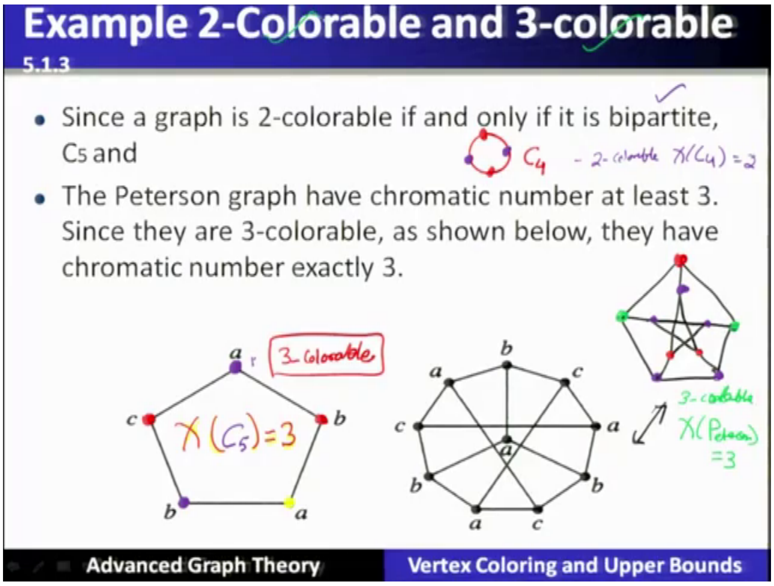


In the proper coloring each colour class is an independent set. So, a graph is k-colourable if and only if vertices of a graph G is the union of k independent sets take this particular graph same graph which we have taken up. So, it required two different say it is a two colourable graph. So, chromatic number of this particular graph we have shown is two so; that means, this is colour one these and this end will receive the second colour this is again receives colour.

So, we can partition or we can collect the colour class. So, all the vertices of having same set of colours is an independent set; why? Because they do not have any edge if they have an edge or they are adjacent that then they cannot receive the same colour hence. So, each colour class represents an independent set. So, here this particular graph has two independent sets; why? Because it is chromatic number is 2.

So, a k-colourable and k partite sets have the same meaning the usage of these two terms is slightly different often k partite is a structural hypothesis by k-colourable is the result of an optimisation problem that we will see.

(Refer Slide Time: 15:21)



Further let us take the example of a two colourable graphs. So, a graph is two colourable if it is bipartite let us see this particular bipartite graph. So, this is C 4. So, this is 2-colourable means you we should be able to colour this particular graph in two different colours for example, this is the adjacent. So, it cannot be read. So, we have turned it to other colour this is also adjacent to read it is turned to the other colour this can be read. So, this two colours are required. So, this graph is 2-colourable are the chromatic number of C 4 is equal to 2, hence this is the bipartite graph.

So, two colourable graphs are bipartite graph now this is a C 5 graph. Let us start the coloring and see how many minimum numbers of colours can be coloured with two different colours. Let us colour it with this colour these two adjacent notes cannot receive this particular colour one which we have a sign this particular note is not adjacent. So, it can also be coloured with; hence this cannot be coloured let us take another colour to colour the remaining vertices.

Since this is colour one. So, we have chosen another colour. So, this is the colour number two this also can receive the same colour; why? Because they are not adjacent either to the second colour not to the first colour as far as this is concerned either it can be coloured with the colour number one or it can be coloured with colour number two. So, you we have to select from the palate another colour.

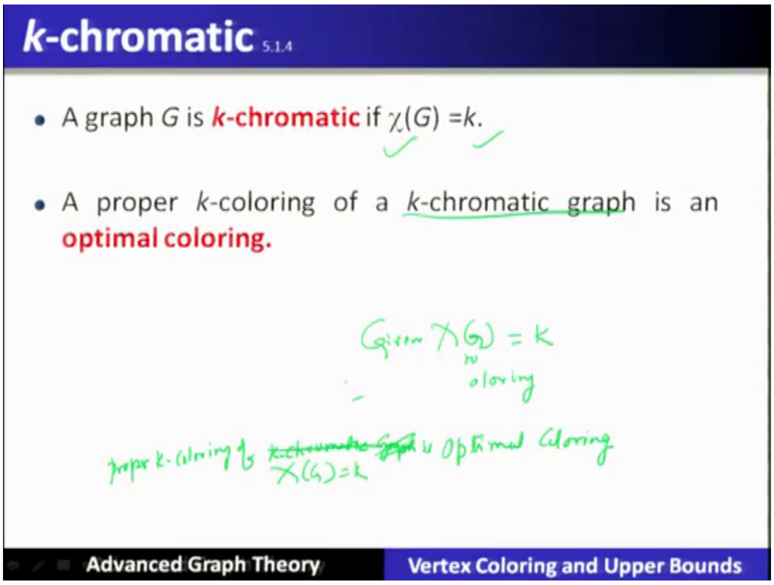
So, c three required how many colour three different colours. So, this particular graph is 3 colourable; why? Because we cannot colour this graph in with two colours, hence minimum number of colours required to colour this particular block is three hence it is called three colourable. Similarly, let us draw a Peterson graph and then you will see this is also the isomorphic to the Peterson graph so, another way to represent it.

But, let us see this particular and start the coloring, how many colours we need to colour this Peterson graph? Let us take this particular vertex and colour with colour number 1, this is adjacent it cannot be receiving the same colours. Let us colour these two vertices again with the same colour, then let us choose other colour. So, that the remaining vertices now this is adjacent to this red. So, it should receive different colours. So, different colour can be used to colour this particular graph.

Similarly, we can colour these two vertices also, why? Because they are not adjacent they may basically get the same colour as far as these notes are concerned they cannot be coloured with the second colour. Similarly these colours these two notes they may be coloured by, because they are not adjacent to these colours. Now we require another third colour to colour the remaining part of the vertices this can be coloured with the third colour.

So, three colours so, this graph is 3-colourable are the chromatic number of Peterson graph is equal to 3 that we have seen. So, these graphs are called three colourable while bipartite graphs are called 2-colourable graphs.

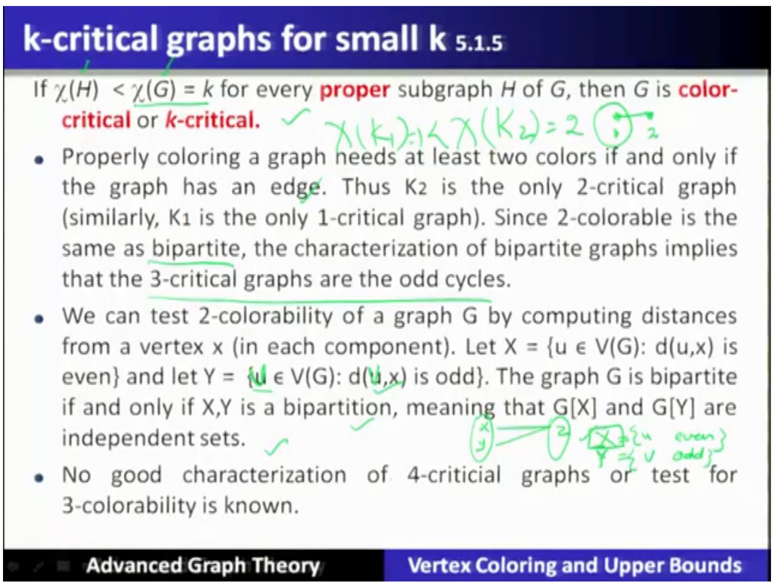
(Refer Slide Time: 20:06)



So, with this let us see the definition of a chromatic number of a graph a graph G is k chromatic if the chi value of G is equal to k, that we have already seen a proper k coloring of a k chromatic graph is an optimal coloring. Again I am explaining it. So, if a graph is k chromatic graph; that means, the chromatic number of a particular graph let us say k if it is given if you obtain proper k coloring of a of a graph with a chromatic number k, then it is called an optimal coloring.

So, optimal coloring is nothing, but finding out a proper k coloring of of a chromatic graph is an optimal k coloring instead of k optimal you can say that for a graph having the chi value is equal to k this particular coloring is the optimal coloring.

(Refer Slide Time: 21:08)



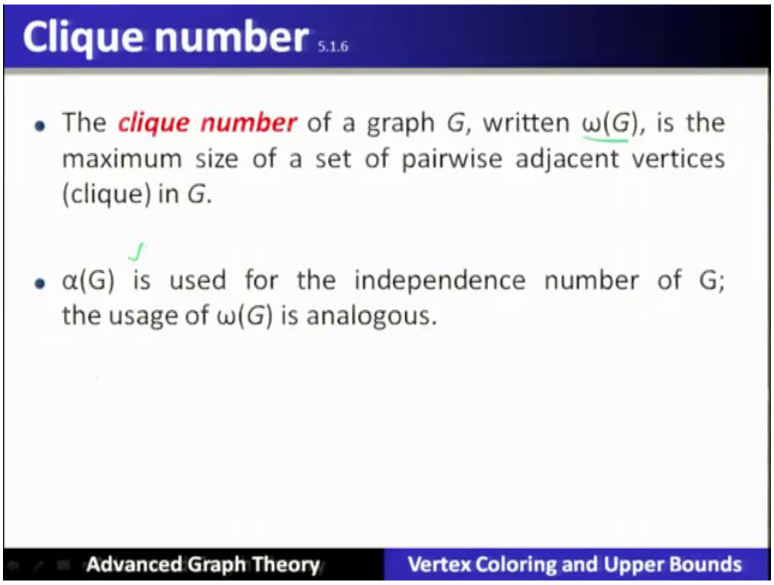
Let us see the k-critical graphs for a small value of k. Now if we take a subset of a graph G let us call it as h and the chromatic value of every subset of a graph G that is h is always less and then for every proper sub graph h of G then G is called colour critical or a k-critical graph. So, again I am repeating; let us take a graph G whose chromatic number of a graph is k and every subset h of that particular graph is having the chromatic value less than k or less than that chromatic value of this particular graph G, then G is colour critical or it is called k-critical. Let us understand this particular concept.

So, let us take a graph K 2 k two is two vertices how many colours are required; obviously, this can be coloured with 1 this is 2. So, the chromatic number of K 2 is 2, if you take a subset subset will be K 1. So, the chromatic number of K 1 is 1 and that is less than K 2. So, the proper coloring needs at least two colours of a graph even only the graph has an edge, thus K 2 is the only two critical graph here. Similarly K 1 is only one critical graph; since 2-colourable is the same as the bipartite set or it is a same as bipartite. So, it will characterize the bipartite means it will employ that three critical graphs or having the odd cycles we can test two colourability of a graph by computing the distances from the vertex X.

So, capital X and capital Y there are two big sets which we have obtained, where all the set of vertices u which are basically having the distance from X is the even parity is basically forming an X set. Similarly the Y set is all such pair of all such vertices let us call it as again V whose distance from X is always odd in parity. So, we will obtain two different sets or we obtain a bipartition of a particular graph, hence this particular graph is a bipartite graph and each X is nothing, but they are the independence sets.

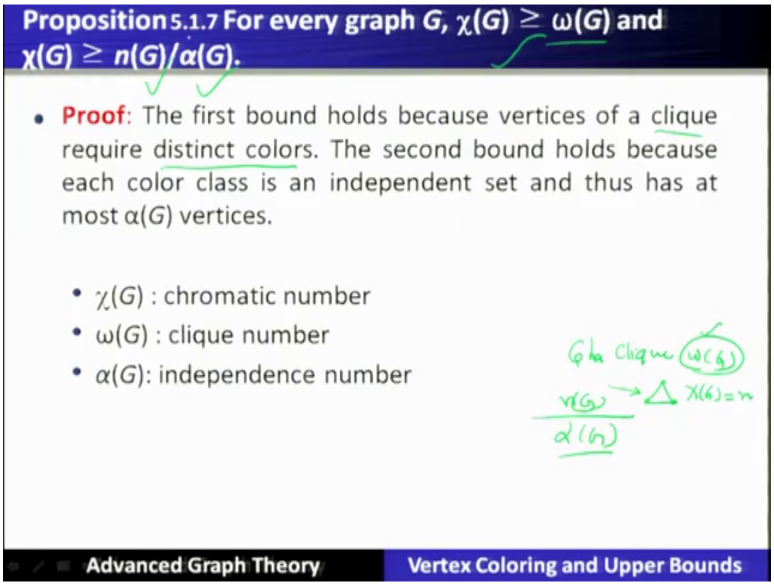
Now, there is no good characterisation of four critical graphs or a test for three colourability is known to us let us take another definition the clique number.

(Refer Slide Time: 24:43)



So, the clique number of a graph is little omega G, of a graph is the maximum size of pairwise adjacent vertices in a graph. Earlier we have already given the alpha G is the independence number of a graph G; that is the number of independence sets or a maximum number of independent sets available in the particular graph.

(Refer Slide Time: 25:09)

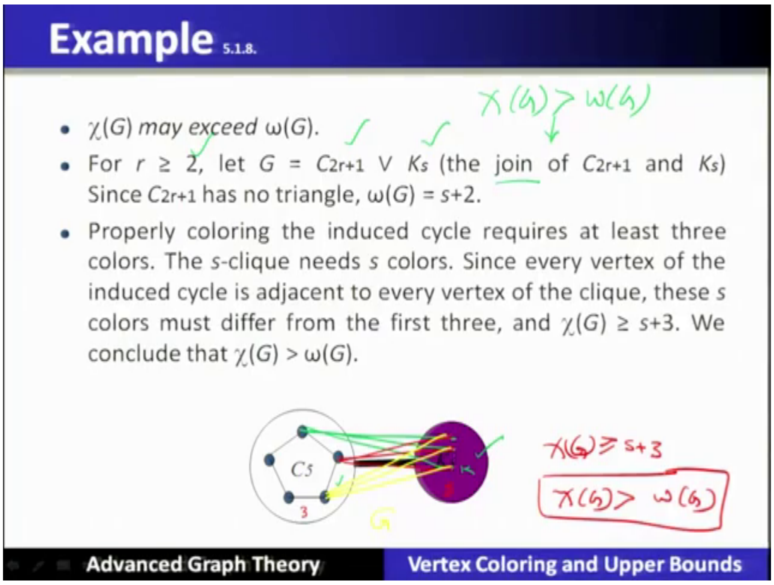


So, with this particular introduction of those concepts; let us take a proposition for every graph G chi of G or a chromatic number of graphs G is at least omega G so; that means, if a graph has a clique if a graph has a clique of size omega G. Now, clique whenever clique is there you can obtain or you know that the chromatic number of a clique is n or basically, the size of the clique you cannot obtain a proper coloring with less than that number of vertices of that particular click.

Hence, the coloring which you are obtaining should contain at least the size of that particular clique. Hence this particular bound is proved the second bound says that each color class; that means, after obtaining the coloring we collect the independent sets in the form of the colour class and alpha G, when indicate the number of independent sets.

So, at least that many number of colours are required because each independent set will be given a separate colour. So, alpha G different colours are required hence total number of vertices are n G, if we divide then at least that many number of colours will be required hence the second bound is also proved.

(Refer Slide Time: 27:06)



Now let us see that this particular bound how good this bound is in the terms of tightness. So, chi of G may exceed omega G in this particular construction of a graph. Let us take for the value of r is greater than 2 greater than or equal to 2. So, if you take r is equal to 2, then it will form a cycle odd cycle C 2 r plus 1 becomes an odd cycle C 5 and Ks is any complete graph which is represented here.

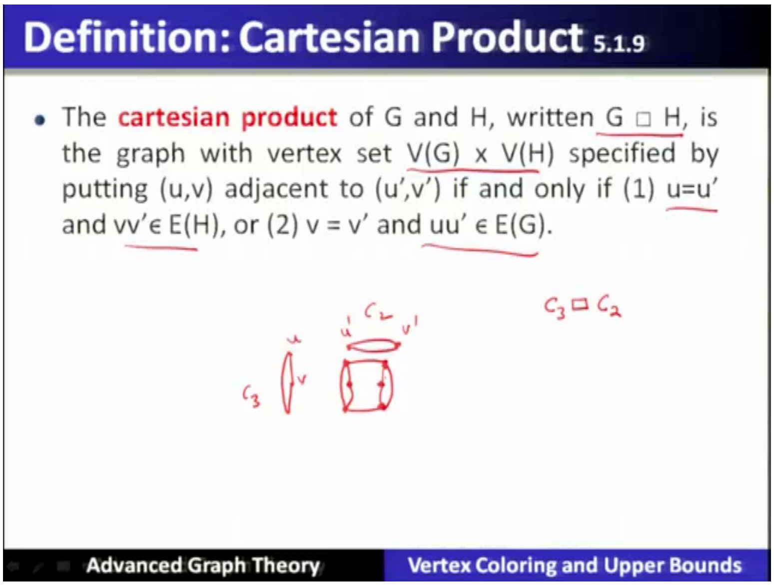
Now, union operation we have defined earlier of two graphs is nothing, but although vertices of these two graphs are adjacent to each other. So, if that is the condition then every vertex will be adjacent to all the vertices of Ks similarly this vertex also will be adjacent to all the other vertices of Ks this also will be adjacent.

So, let us see how many colours are required definitely this particular graph G is going to exceed this particular value, why? Because, every vertex of C 2 r plus 1 is adjacent to Ks so, more than omega of G that is; this is the complete graph that many number of colours are required, let us see how many different colours are required?

Now, here you can see that this is an odd cycle it require how many colours three colours and this requires S colours so; obviously, the chromatic number of this particular graph will be S plus three is at least this particular value will be there, hence we conclude that in this particular scenario is strictly greater than the size of the clique. So, told you that we are looking up this particular bound how good this particular bound is? So, for some of the graphs here the bound is quite loose, why? Because it is strictly greater than omega G.

Now, let us see we will see further that a particular graph where.

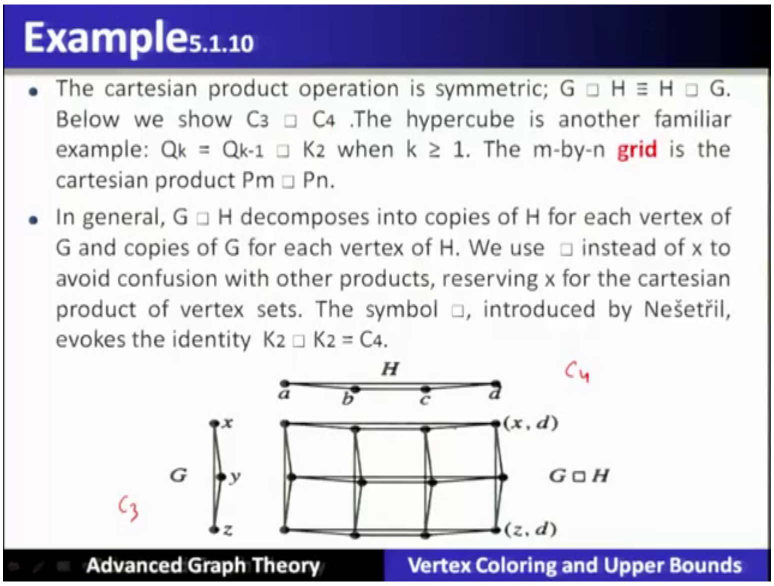
(Refer Slide Time: 30:09)



This particular bound is tight or equal hence the coloring will be looked upon, now let us take the definition of a Cartesian product of two graphs G and H, which is represented by this particular symbol Cartesian product of two graph is a graph with a vertex at V G cross or a Cartesian product vertex set of H it is specified by putting uv adjacent to u prime, v prime, if and only if u is equal to u prime and v prime vv prime will have an edge in H.

Similarly, v is equal to v prime and uu prime has an edge in H.

(Refer Slide Time: 31:03)

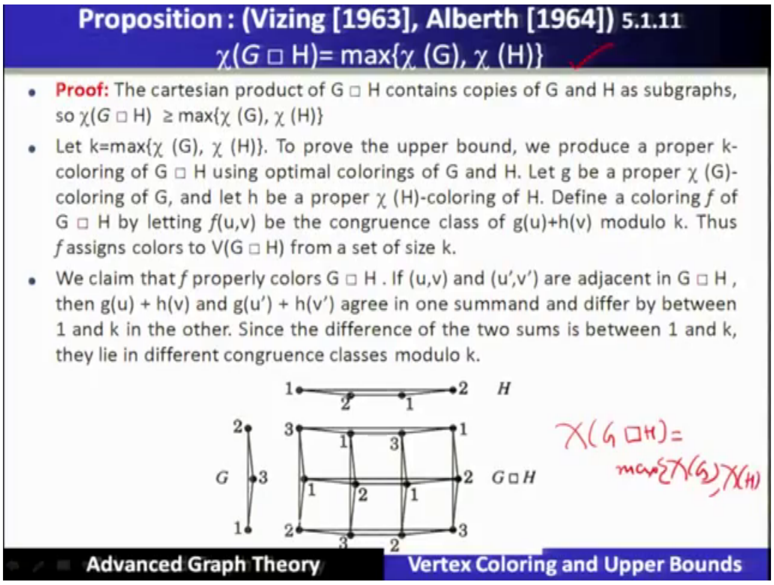


So, these are the example of such construction; but let us see another example. So, this is C 3 recycle and let us takes a C 2 cycle and let us draws a cross product of C 3 Cartesian product C 3 and C 2. So, this particular vertex if you take a Cartesian product, what will happen? Let us see that these two vertices will join and they will basically form this one and then again another pair of vertices and this is also joined.

So, as for as these definitions are concerned, u and u prime this is let us say u and this is u prime. So, u and u prime is equal and v and v prime let us say that this is v prime and this is let us say v. So, v and v prime has an edge. So, that will represent something here no something like u and u v and v prime.

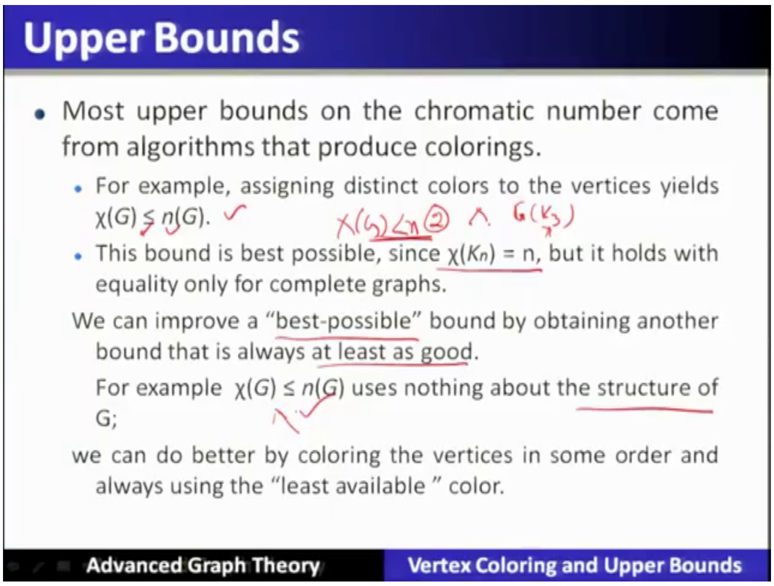
So, let us see in this particular example x, y, z this is you know that C 3 cycle 1, 2, 3, 4 this is C 4. So, let us take this particular example, and we will see in this particular example the Cartesian product of these particular graphs. So, you see that this particular cycle is repeated at every pair of vertices and these particular cycles also repeated on the other side and that becomes a Cartesian product.

(Refer Slide Time: 33:27)



So, having defined this particular Cartesian product, then let us see the proposition given by using that the chromatic number of the Cartesian product of two graph is nothing, but the maximum of the chromatic number of the graph G and the chromatic number of graph H.

(Refer Slide Time: 33:50)

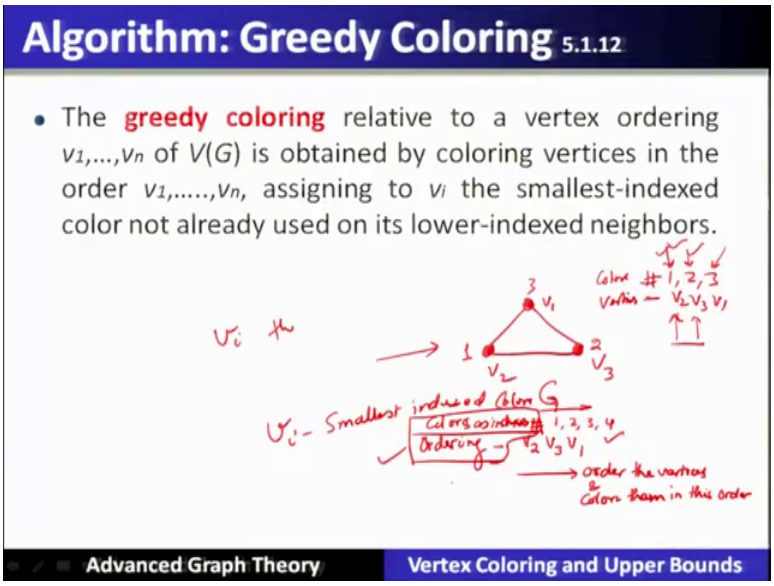


Now, let us find out the upper bounds in terms of the colourability. So, most upper bounds on the chromatic number come from the algorithm that produced the colourings. For example, assigning distinct colours to the vertices will yield that the chromatic number of G is at most n G, in the sense that if you are given a graph let us say that this particular graph here is K 3, then you require how many different colours that is nothing, but the number of the vertices that becomes equal, but if the graph is not complete, then what will happen then this is not equal to n G, but it is less than n G.

For example if there is no edge. So, you can colour using two colours only that is not with the three colours so; that means, then chi of G is basically not equal to n G, but it is less than n G. So, this particular makes a possibility to find out the bound. So, this bound is the best possible, when the graph is the complete graph that is equal to n, but we can improve this particular best possible bound with another bound that is always at least as good, why? Because if we say that the upper bound of a chromatic number is n G; that means, we have not seen inside the structure of that particular graph and we have assumed the structure of complete graph and we have established a bound.

So, if the graph is not a complete graph then a better bound is to be ascertained. So, we can do better by coloring the vertices in some order and always using the least available colour.

(Refer Slide Time: 36:04)



Greedy coloring algorithm the greedy coloring relative to the vertex ordering that is V 1, V 2 and so, on up to V n a particular ordering of the vertices of a graph G is obtained by coloring, the vertices in this particular given order assigning to a vertex V i the smallest-index colour not already used on it is lower-index neighbours.

So, we are going to give you the first algorithm for coloring and this is called greedy coloring algorithm. Then we will check the bound which we obtain using this particular greedy coloring, how good the bound is? We will know that once we compare it with the chromatic number of a graph G, then we will see how we can improve this particular algorithm and more insight in the graph coloring.

So, again I am explaining this particular algorithm a greedy coloring algorithm is related to the vertex ordering. Again I am taking let us take the simple graph V 1, V 2, V 3. So, a particular ordering let us say we say that V 2, then V 3 and V 1 and we want to colour in this order. First we have to order the vertices and then we will be coloring them in the same order.

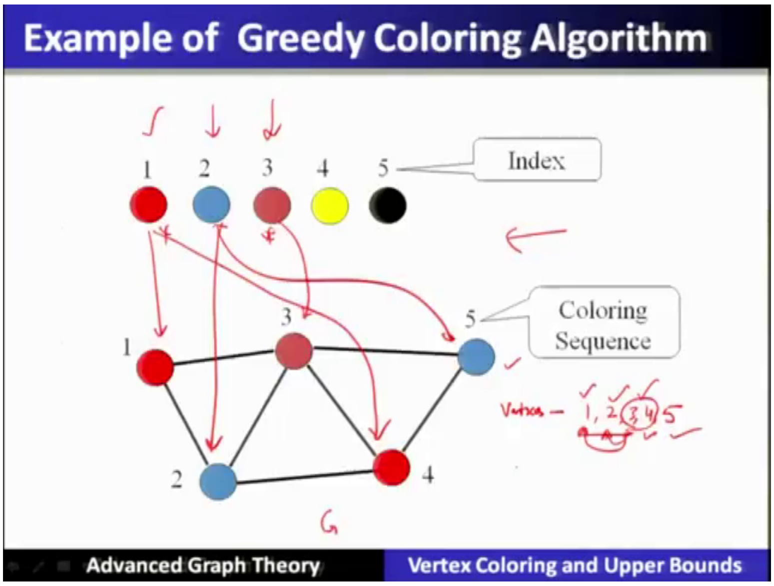
So, let us we have obtained a particular order now according to greedy coloring algorithm we will assign to V i we will assign to V i by smallest indexed colour. So, we will obtain the colours as the indexes for example, colour number 1, 2, 3, 4 and so, on. So, one may represent red blue green and so, on. So, colours are also indexed and vertex is also in particular order.

Now, let us see that greedy coloring algorithm says that a greedy coloring relative to the vertex ordering is obtained by coloring the vertices in that same order assigning the vertex V i by smallest indexed colour not already used on it is lower index neighbours. So, smallest index colours; that means, we will pick the colours from a particular index that is a lower index colours which are not applied earlier to it is neighbour will be applied to that particular vertex let us take this particular example.

So, let us we have the coloring colours 1, 2 and 3, they are indexed. Now vertices related to the given order let us say these orders are V 2, V 3 and V 1. So, first we have to colour V 2 we will colour V 2 by picking the these indexed colour which is not used in it is lower indexed neighbours one is never used. So, let us colour with this particular vertex with the colour number 1.

Now, comes to V 3; V 3 is a neighbour of V 2 who has already used lowest indexed or smallest index colour 1. So, the next is smallest index colour which is not used by the neighbour is colour number 2. So, V three will be coloured with colour number 2 as far as V 1 is concerned V 1 is neighbour to V 2 and V 3 both both are it is neighbours. So, the V 2 and V 3 have already used colour number 1 and 2; that means, the smallest index colour which is not used by the neighbour is colour number 3. So, this can be coloured with 3.

(Refer Slide Time: 41:10)



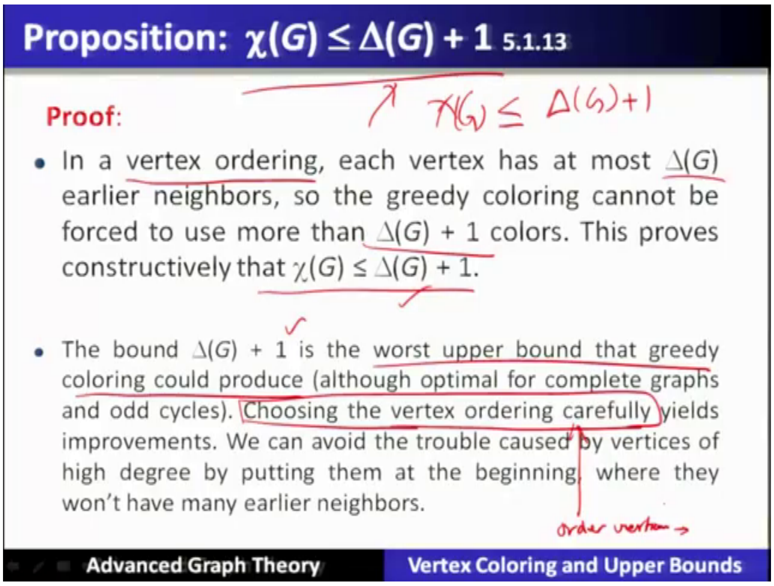
So, this particular coloring is called a greedy coloring. So, this particular graph is also explaining the greedy coloring algorithm here. We are given 5 node graph, let us call it as G and we are given the vertices in this particular relative order 1, 2, 3, 4 and 5. So, the vertices are given in this order and we have two colours relative to this particular order in this algorithm.

Similarly, we are also given the indexes of the colours 1, 2, 3, 4 and so, on starting the algorithm, what it does vertex one will be coloured first with the smallest indexed colour not used by the neighbour of one earlier. So, the colour number one will be assigned second vertex is vertex number 2, vertex number 2 is the neighbour of vertex number 1. So, the smallest index colour which is not used by the neighbour of 2 is 3 is basically, the second index colour. So, which is assigned to that particular vertex?

Similarly, the vortex number 3 in that order. Now vertex number 3 is neighbour to the earlier used colours by their neighbours that is 1; and 2 are it is neighbours 1 and 2 colours already used. So, smallest colour index which is not used by the neighbour of 3 is 3 itself. So, 3 will be coloured to this particular vertex.

Now, coming vertex number 4. Now vertex number 4 the neighbours of 4 who have used the colours earlier are 3 and 2. So, three cannot be used and two cannot be used. So, the smallest index colour which is available is 1. So, one will be assigned to this particular vertex finally, we have to come to 5. So, vertex number 5 it is neighbour are 3 and 4, let us see what colours 3 and 4 are assigned. So, 3 and 4 are assigned this colour and this colour. So, the smallest index colour which is not used by the neighbours of 5 is 2. So, 2 will be assigned here. So, this becomes a greedy coloring algorithm.

(Refer Slide Time: 43:54)

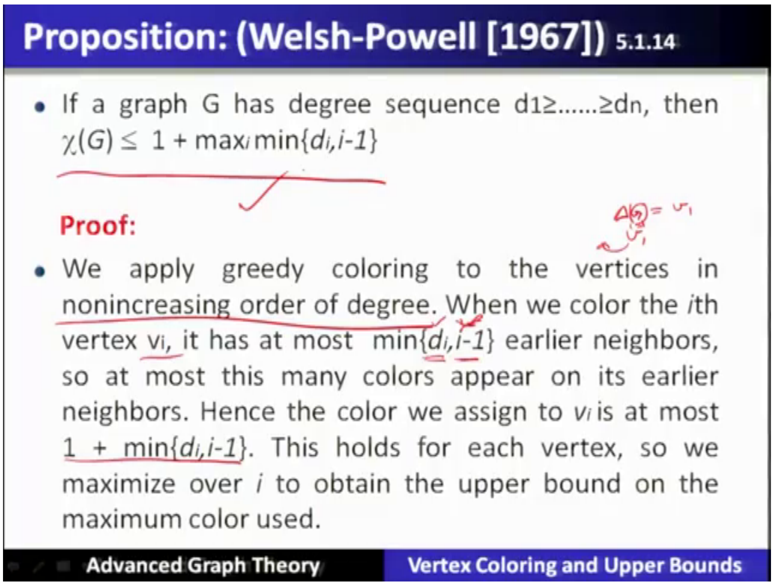


Now, let us see the proposition which will estimate the basically performance in terms of the bound of this greedy coloring algorithm. This greedy coloring algorithm will give the bound which says that the chromatic number of a particular graph is less than or equal to the maximum degree of a graph plus 1. Let us see the proof says that for a relative vertex ordering each vertex has at most maximum degree earlier neighbours. So, the greedy coloring cannot be forced to use more than maximum degree plus 1 number of colours, hence this proves constructively that the chromatic number of a graph is bounded by or is at most maximum degree of a graph plus 1.

Now, this maximum degree plus one bound is the worst upper bound that that greedy coloring algorithm could produce although this becomes optimal for a complete graph and odd cycles. Now choosing this particular greedy coloring algorithm depends upon choosing the vertex ordering.

So, if the vertex ordering is chosen very carefully it will improve this particular bound. Further, now one heuristic which will be applied here is that if we order the vertices according to the degree sequence. So, that the highest degree vertices are placed ahead or in the beginning in the sequence and the next higher degree will be placed and so, on because if the highest degree vertex is placed there will be no other neighbour, before it hence you can easily start the colour and; obviously, the number of colours required will be quite less or it will improve this particular bound of the greedy coloring algorithm.

(Refer Slide Time: 46:29)



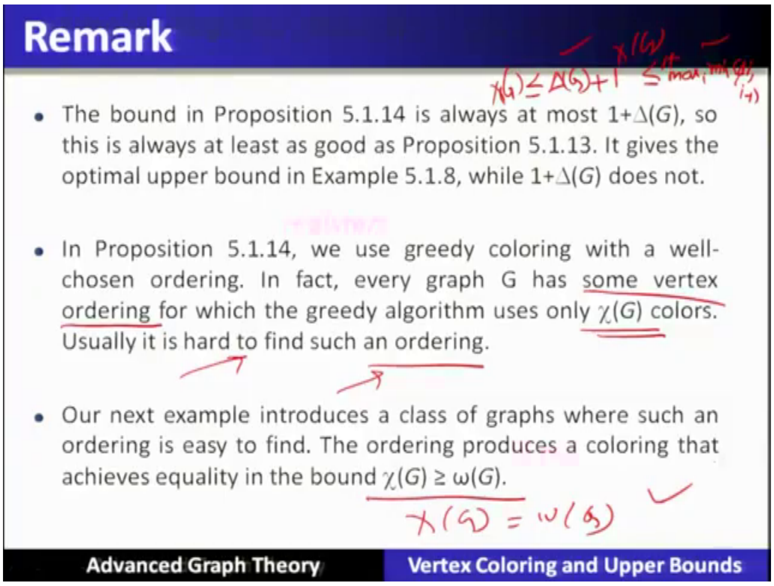
Let us see this particular heuristic which is given by Welsh Powell and hence the proposition which will give you a further modification in the bound. So, if a graph G has a degree sequence d 1 is greater than or equal to d 2 is greater than and so, on up to d n; that means, nonincreasing order of the degree sequence of a graph if it is given; that means, vertices are ordered in this particular degree sequence.

Then, when we apply the greedy coloring on this particular relative vertex, ordering of this degree sequence. So, that vertex V 1 is having the highest degree is placed first, then second and so, on. Then this particular kind of coloring or using this heuristic which is called Welsh Powell heuristic; let us estimate that it gives 1 plus max of i minimum of d i and i minus 1, and this particular bound is better than maximum degree plus 1 some of the situations.

Now, let us prove this. So, we apply greedy coloring to the vertices in the non increasing order of the degrees. Now, when we color ith vertex i it has at most either the d i or if the i has degree is placed, before it is i minus 1 is lesser than the d i, then the minimum of either the degree or the i minus 1. So, i minus 1 says that the highest degree is let us say that V 1 node is V 1. So, V 1 is placed. So, before V 1 there is no other neighbour hence d i will be not used. So, i minus 1; that means, I means the index i is 1. So, the 0 index colours will be use. So, minimum of that see these two smallest index and the degree values are taken together in Welsh Powell heuristics.

So, again I am repeating that vertex i has at most minimum of d i and i minus 1 earlier neighbour. So, at most these many number of colours appear on other neighbours, hence the colour we assigned to v i is at most one plus minimum of d i and i minus 1 this holds for each vertex. So, if we maximise over i to obtain the upper bound on the maximum number of colours used will be this particular bound that we have already proved now.

(Refer Slide Time: 49:39)



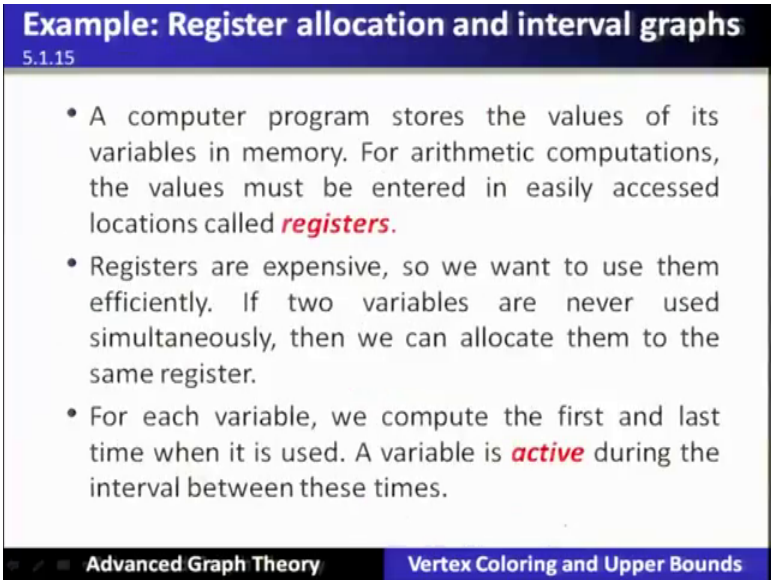
Let us see the remarks on these two discussions, that is; first we have obtained a greedy coloring whose bound was big delta G plus 1 and then we have seen the Welsh Powell, where it is 1 plus max of i min of d i and i minus 1.

So, the bound is always at most 1 plus maximum degree of a graph. So, this is always at least as good as the proposition which we have seen so; that means, this particular bound is at least as good as the earlier bound which we have seen it gives optimal upper bound for a particular example, that we have seen the union of odd cycle and the complete graph while 1 plus maximum degree does not give it.

So, in proposition 5.1.14 that is Welsh Powell we use the greedy coloring with welsh chosen ordering. In fact, every graph has some vertex ordering for which the greedy coloring uses only the optimal number of colours, but it is very hard to find out such an ordering. Hence the graph coloring problem to find out an optimal coloring through an algorithm is a bit hard it is not a easy algorithm.

So, our next example introduces the class of graph where such an ordering is easy to find which ordering means that ordering which gives an optimal coloring, through the greedy coloring. So, the ordering produces the coloring that achieves the equality in the bound that is chi of G is equal to little omega of G so, that we will see.

(Refer Slide Time: 51:59)

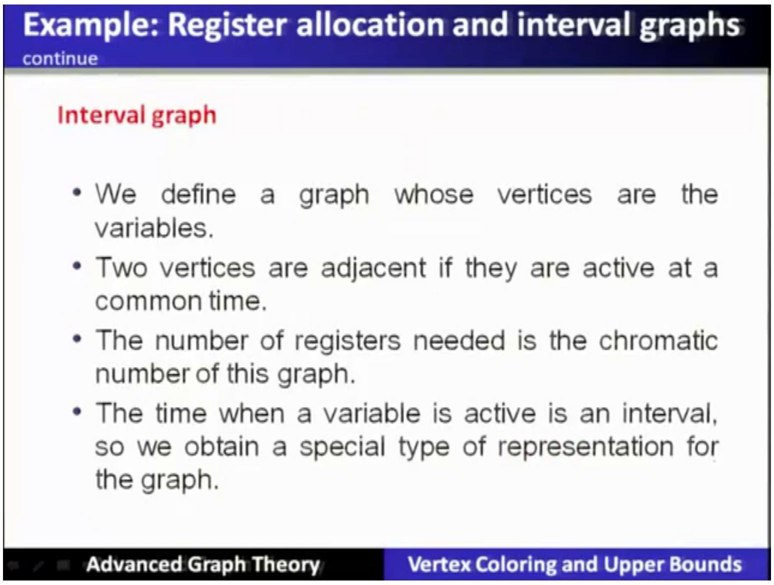


So, this kind of graph or this kind of ordering which we are talking about here we can obtain using a particular kind of graph which are called a interval graph and this interval graphs are basically if you see the problem of a register allocation in the compiler or in the pitch compiler does. Normally, it uses the register allocation through the understanding; how it is going to be used in the programs? And that is represented in the form of a interval graphs and that interval graph will give the vertex ordering in that ordering is basically the optimal gives an optimal coloring through the greedy coloring.

Let us see all that things in this example. So, a computer program restores the values of it is variables in the memory for arithmetic computations. The values must be entered in easily axis locations which are called registers; Registers are expensive. So, we want to use them efficiently two variables are now are used simultaneously then we allocate them to the same register.

For each variable we compute the first and last time, when it is used a variable is active during the interval between these.

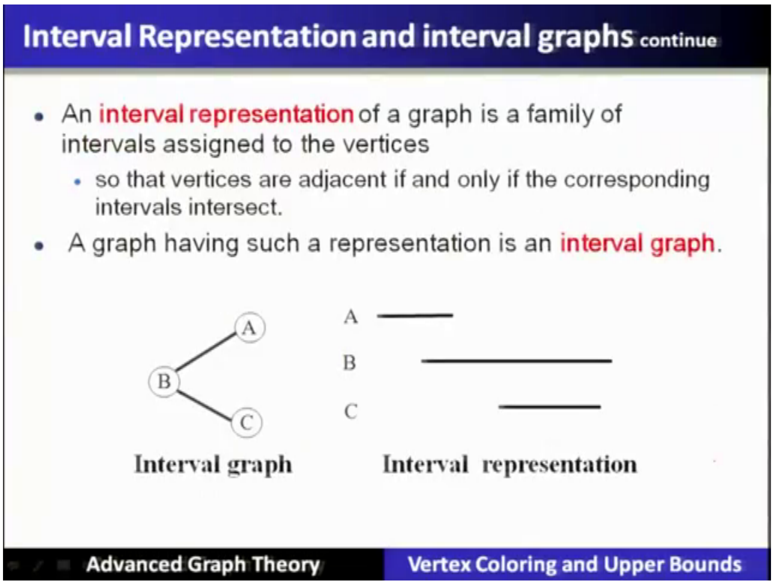
(Refer Slide Time: 53:20)



Times this is represented in the form of a graph which is called interval graph. So, we define a graph each vertices are the variables two vertices are adjacent, if they are active at the common time the number of registers needed is the chromatic number of this graph.

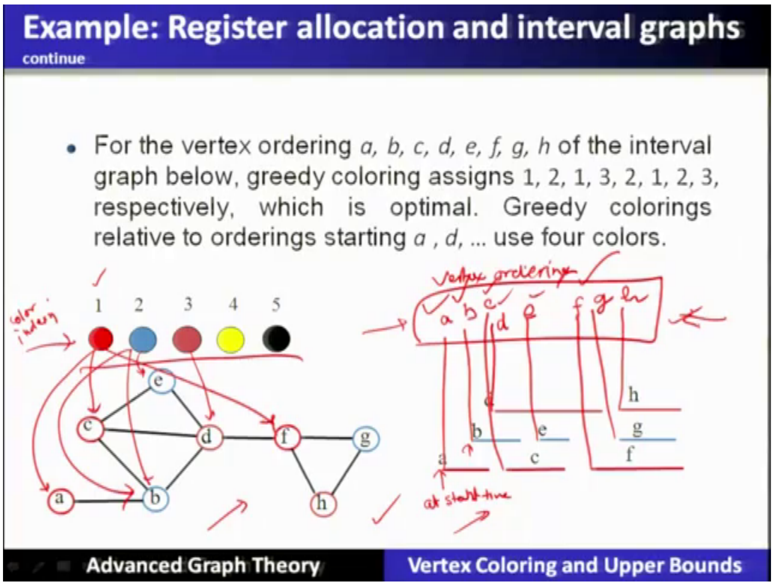
The time when the variable is active is an interval. So, we obtain a special type of representation for.

(Refer Slide Time: 53:41)



The graph which is called a interval graph.

(Refer Slide Time: 53:44)



So, let us take the example that given this particular graph here is called interval graph and the intervals are given in this particular manner.

So, the vertex ordering corresponding to this particular interval graph becomes a, then b, then c, then d, then e, then f, then G, then h. So, this particular vertex ordering we have obtained from the intervals when a particular variable is at start time. So, for every variable the start time according to the start time we order the vertices and we got this particular vertex ordering having got this vertex ordering.

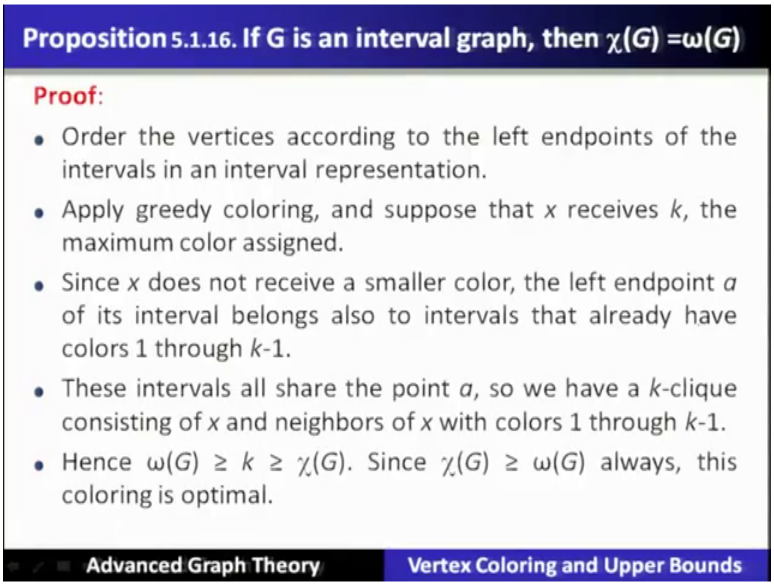
Now, we will see the particular ordering of or indexing of the colours this is the indexing of the colours colour indexing. Now we apply the greedy colouring algorithm. So, here vertex a is not yet coloured. So, the first smallest index colour which is not earlier used by it is neighbours a is 1. So, one will be coloured here.

Now, as far as b is concerned a and b they are neighbour. So, b cannot be coloured with colour number 1. So, colour number 2 will be given to b. So, b is also coloured c is concerned c is the neighbour of b. So, c cannot be given this particular colour, but c can be given the smallest index colour that is 1 as far as d is concerned d is the neighbour of c and b. So, c is basically having the colour number 1 b is having 2. So, the third colour is given to d.

Similarly, e is concerned e is a neighbour of c and d. So, c is having colour 1 it cannot be used colour 1. So, colour 2 is the smallest 1. So, colour 2 will be assigned to e as far as f is concerned f will be given red and all the colours will be assigned as we have shown in this particular figure.

Now, this particular coloring which we have obtained in this particular vertex ordering in the interval graph is optimal and that uses only four colours.

(Refer Slide Time: 57:04)

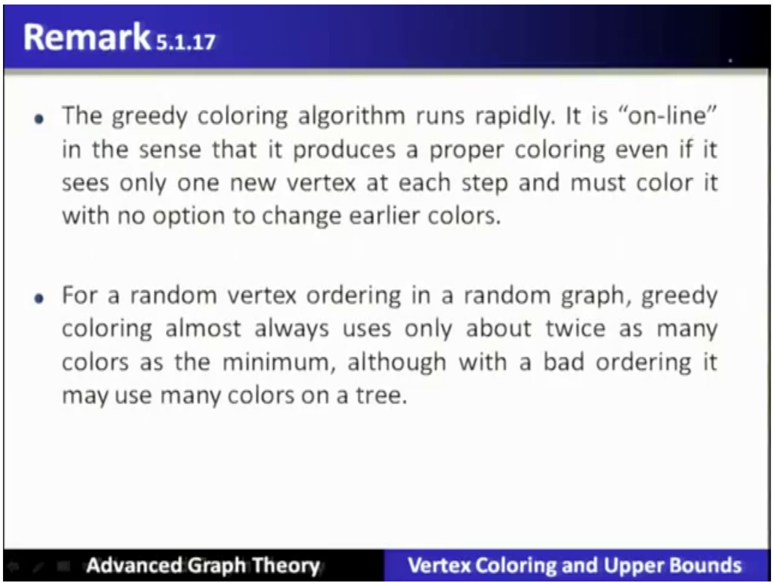


Now, we will see the proposition which says that G is an interval graph, then chromatic number of a graph is equal to omega G; that means, we can colour with only the clique value of that particular offence this becomes optimal, why? Because we have seen that it is at least omega G. Now, it will becomes equal to omega G, hence this particular coloring is optimal for interval graphs.

Let us see the proof or other vertices according to the left end points of the intervals in the interval representation that we have already seen earlier in the example. Apply the greedy coloring and suppose that x receives k, maximum colours assigned. Since x does not receive a smaller colour, left end point a of it is interval belongs to the interval that already have the colours one through k minus 1.

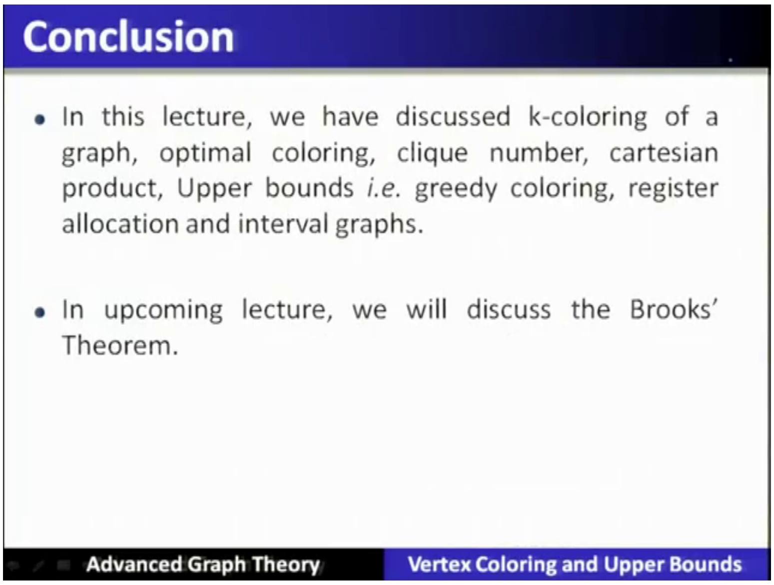
These interval all share the point a, and we have k-clique consisting of x and the neighbours of x with colours 1 through k minus 1. Hence omega G is at least k and that is; basically at least or that is less than chromatic number of that particular graph since chromatic number is always greater than omega G greater than or equal to omega G always. So, if you take both these equations then this particular coloring is optimal.

(Refer Slide Time: 58:40)



Hence it is proved. Now, remark the greedy coloring algorithm runs to rapidly. It is "on-line" in the sense that it produces a proper coloring even if it sees only one new vertex at each step and must colour it with no option to change earlier colours. For a random vertex ordering in a random graph, greedy coloring almost always uses only about twice as many colours as the minimum, although with the bad ordering it may use many colours on a tree.

(Refer Slide Time: 59:14)



Conclusion in this lecture, we have discussed k-coloring of a graph, optimal coloring, clique number, cartesian product, the upper bounds, that is. with reference to the greedy coloring. We have also seen an interval graph based coloring and example which is based on it that is about register allocation in upcoming lectures we will discuss the brooks theorem.

Thank you.