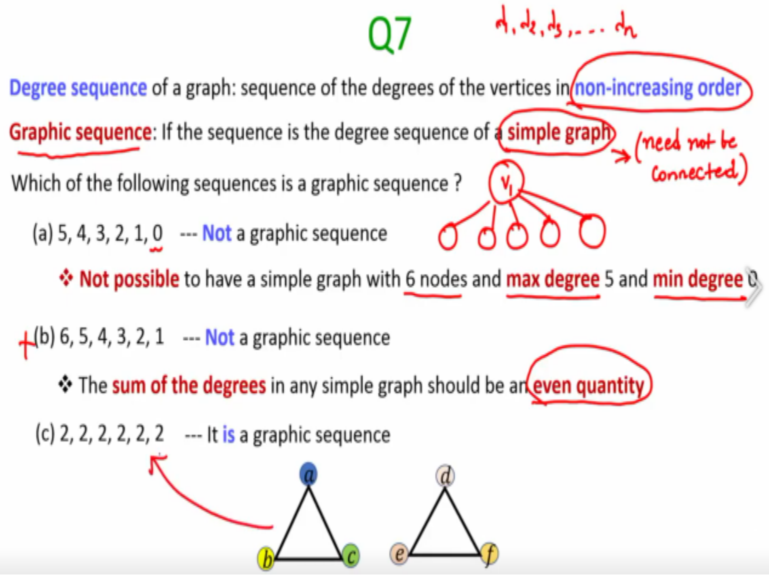
Discrete Mathematics
Prof. Ashish Choudhury
International Institute of Information Technology - Bangalore

Lecture - 54
Tutorial 9: Part II

(Refer Slide Time: 00:26)



Hello, everyone, welcome to the second part of tutorial 9. So, let us start with question number 7. So, here we first define what we call as the degree sequence of a graph and the degree sequence of a graph is basically the sequence of degrees of the vertices in non increasing order. So, you list down the highest degree vertex or the degree of the highest vertex first followed by the next highest degree, followed by the next highest degree and so on.

So, if you have n vertices, basically you are listing down the degrees of the n vertices in a non increasing order. And we say a sequence of n values as a graphic sequence, if you can construct a simple graph whose degree sequence is the given sequence, if you cannot draw any simple graph whose degree sequence is a given sequence, then the given sequence will not be called as a graphic sequence.

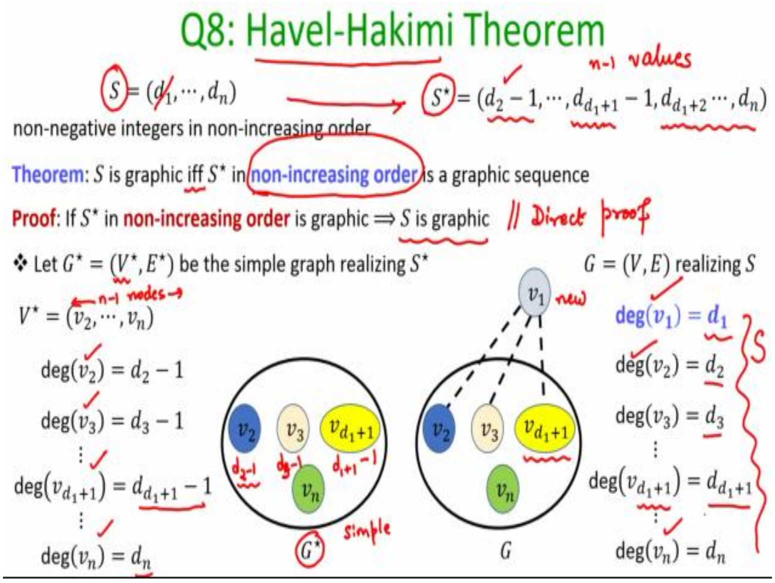
So I stress here that we need a graph only to be simple it need not be connected, it is fine if the graph is not connected. So the first few parts of question 7 basically asks you to prove or disprove which of the given sequences is a graphic sequence. So let us take the first sequence 5, 4, 3, 2, 1, 0. Of course, 1 obvious condition in a graphic sequence should be that values should be non negative, you cannot have a vertex with a negative degree so that is a trivial condition.

So in this case, we have to verify whether we can draw a simple graph with 6 nodes where the highest degree is 5 and the smallest degree is 0. And it is easy to see that this sequence is not a graphic sequence. Because you cannot have a simple graph with 6 nodes where the maximum degree is 5 and the minimum degree is 0. Because if say v_1 is the vertex which has the maximum degree, so if its degree is 5, then it should be a neighbour of each of the remaining 5 nodes.

That means each of the remaining 5 nodes will have a degree which is non 0, but you also need a vertex with a degree 0 among those 6 nodes, which is not simultaneously possible. So, now let us take the second sequence (6,5,4,3,2,1) and try to argue whether the sequence is a graphic sequence or not. And again, this sequence is not a graphic sequence, but there are several ways by which you can refute that this sequence is not a graphic sequence.

One simple way is that if you take the sum of the values that are given in this sequence is not an even quantity, but we know that for any graph, it may not be a simple graph for any graph the sum of the degrees of all the vertices is twice the number of edges which is an even quantity. So, 1 obvious condition that should be satisfied by any graphic sequence is that if you sum the values given in the sequence, it should be an even quantity, which is not the case for the sequence given here. Let us consider the third sequence and the sequence is a graphic sequence and this is a simple graph which realises or which has this degree sequence (2,2,2,2,2,2).

(Refer Slide Time: 04:00)



So, now in question 8, we want to characterise that, we want to find out a characterization for graphic sequences. So, if you are given a sequence with n values, how can you verify whether that sequence is a graphic sequence or not we cannot keep on drawing all possible simple graphs and then either prove or refute that a given sequence is not a graphic sequence, we need an algorithmic characterization, a necessary and sufficient condition and that is given by what we call as Havel-Hakimi theorem.

So here we are given the following you are given sequence S of n non negative integers in non increasing order and you have a reduced sequence S*. It is reduced in the sense it has n - 1 values whereas the sequence S has n values. So how exactly we construct a sequence S*. So, the way we construct S* from S is the following, we first to remove the value d_1, and then from the next d_1 degrees or from the next d_1 values in the sequence S we subtract 1 from each of those d_1 values in the sequence S.

So d_2 gets decremented by 1, d_3 gets decremented by 1, and d_{1+1} th term gets decremented by 1. Whereas, from d_{d1+2}th term to the d_n th term, the degrees remain the same as they were in the sequences. So, that is the way we obtained the sequence S*. And what Havel-Hakimi theorem says is the following it says that your sequence S is a graphic sequence if and only if the reduced sequence S* when arranged in a non increasing order is also a graphic sequence.

So, for the moment imagine that this theorem is true, how exactly we can use this theorem to verify whether a given sequence of S values is a graphic sequence or not? Well, we have to reduce the sequence S and build a new sequence S* and then rearrange that terms in S*, so, that the new degrees are in a non increasing order. And now, we have to verify whether the reduced sequence S* is a graphic sequence or not. To do that, I can again apply the Havel-Hakimi theorem.

Now, this reduced to sequence S* can be further reduced to n - 2 degrees, where I can remove the first degree from S* and to compensate that I subtract 1 from the next few degrees. And then the next reduced sequence again is arranged in a non increasing order and then we can verify whether that sequence is a graphic sequence or not. And I can keep on repeating this process; keep on decreasing my sequence till I obtain a very short sequence which I can very easily verify whether it is a graphic sequence or not.

If it is a graphic sequence then I can come back all the way and declare that my big sequence my original sequence S is a graphic sequence. Whereas if the reduced sequence or the small sequence at which I stop and inspect and find out that it is not a graphic sequence, then I can declare that my original sequence S also not a graphic sequence. So, that is a way I can apply the Havel-Hakimi theorem to verify whether a given sequence is a graphic sequence or not. So now, let us prove this theorem and this is an if and only if statement.

So, we have to prove 2 implications: let us first prove the easier one. So, we want to prove that if S* when arranged in a non increasing order is graphic, then so is the sequence S, what does this mean: so I will give a direct proof for this implication. And when I say I will give a direct proof, I mean to say that I will assume that my premise is true and I will arrive that my conclusion is also true so, assume that my premise is true.

That means, since my sequence S* as a graphic sequence, I can construct a graph, a simple graph G* with n - 1 vertices and some edges whose degree sequence is the same as the sequence S*, what does that mean? So, I can imagine that my vertex set V* has n - 1 nodes. I call those nodes as v_2, v_3, v_n. And since it realises the sequence S* that means, I have a vertex of degree $d_2 – 1$. Let v_2 be that vertex.

I will have a vertex with the degree $d_3 – 1$. Let v_3 be that vertex and like that I will have a vertex of degree this much. Let $v_{d1 + 1}$ be the vertex with that much degree and like that I will have a vertex of degree d_n and let v_n be the vertex with that degree. That is the implication of assuming my premise to be true. Now, my graph G* is a simple graph remember, apart from that, I do not know anything about G* whether it is connected or not connected and so on.

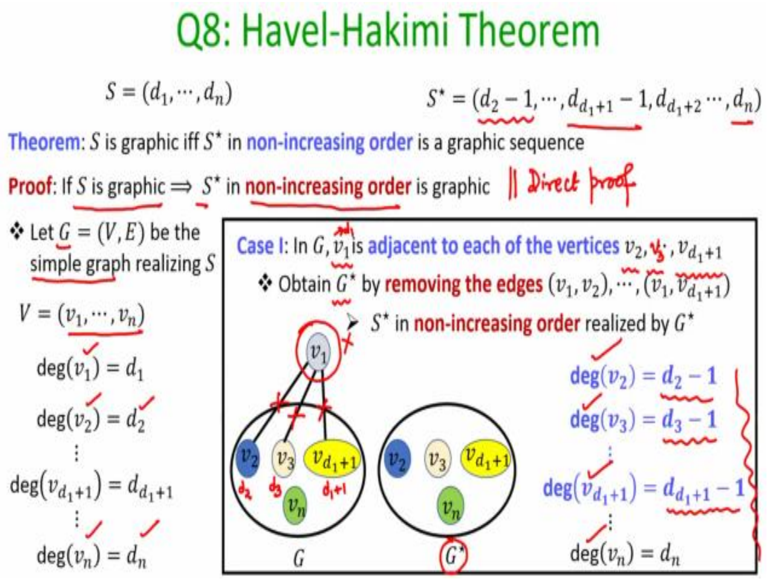
Now from G*, I have to build another graph G which has n nodes, which is simple and whose degree sequence is the same as the sequence S, that is what is the implication. So the construction of the graph G is very simple. I take a copy of G* as it is and since I have to give a graph which has n nodes, but since I have taken the graph G* I have currently n - 1 nodes. So, what I will do is I will now include a new node: call it v_1 and I have to give some edges to this vertex v_1.

So, what I do is, I add the edge between the vertex v_1 and the vertex v_2 which has earlier degree $d_2 – 1$. I add an edge between the vertex v_1 and the vertex v_3 which had earlier the degree d_3 - 1 and similarly, I add the edge between the vertex v_1 and vertex number d_1 + 1 which had earlier the degree d_1 + 1 and the remaining edges they remain as it is in the graph G. Now, what can I say about the new degree for the vertex v_2 it will be one more than what it was earlier.

So, earlier the degree was d_2 - 1, but now, since I have given a new edge to the node v_2 its degree will now become d_2; similarly, the new degree of the vertex v_3 will become one more than it was earlier so, it will become d_3 and like that degree of the d + 1th vertex will be one more than what it was earlier. So, it will become this much and the degrees of the remaining vertices will remain as it was earlier and what can I say about the degree of the vertex v_{1} : it will be d_1.

Because I have added d_1 edges incident with the vertex v_1 and now, you can see that this sequence is nothing but the sequence S that means in the sequence S you need to have 1 vertex of degree d_1. So, I have one such vertex namely v_1. You need to have a vertex of degree d_2. I have one such vertex namely v_2 you need to have a vertex of degree d_n I have one such vertex namely d_n. So, I have now a simple graph whose degree sequence is same as the sequence S. So, that shows that this implication is true.

(Refer Slide Time: 12:35)



Now, let us prove the implication in the reverse direction. So, I want to prove that if your sequence S is graphic, then the reduced sequence S* when arranged in a non increasing order is also graphic. And again I will give a direct proof; that means I will assume that my premise is true and I will arrive at my conclusion. So, since my premise is true that means, I have a simple graph call it G with n nodes.

And some edges whose degree sequence is same as the sequence S; that means, you have n vertices say v_1 to v_n and let v_1 be the vertex which has degree d_1, v_2 be the vertex which has degree d_2 and v_n has the vertex which has the degree d_n. Now, from this graph I have to arrive at another graph, which is simple with n - 1 nodes and which realises the sequence S* : the reduced sequence S*.

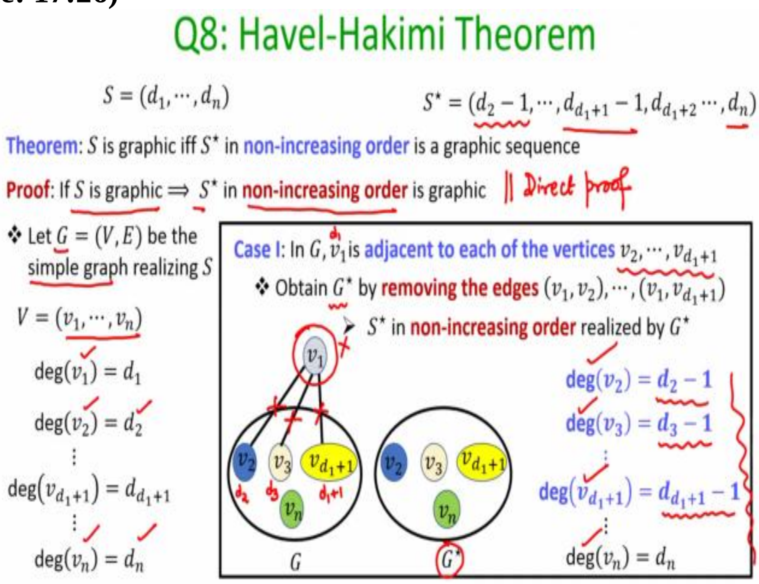
So, how do I do that, so, I will use now a proof by cases. So, once I assume my premise to be true, I will do a proof by cases because there will be 2 cases which will be happening depending upon what exactly is the structure of the graph G. So, your case 1 will be the following imagine your simple graph G is such that the vertex v_1 which has degree d_1 is adjacent to the vertex which has degree d_2 it is adjacent to the vertex which has degree d_3 it is adjacent to the vertex v_4 which has degree d_4 and like that, it is adjacent to the vertex which has degree $d_1 + 1$. Suppose that is the case. Case 2 will be when this is not the case. So, case 1 is when v_1 is adjacent to the vertex which has degree d_2, it is adjacent to the vertex v_3 which has degree d_3 and it is adjacent to the vertex which has degree $d_1 + 1$. Now, let us see what will happen if I delete this vertex v_1 and the edges which are incident with the vertex v_1 because if I delete the vertex v_1 of course, these edges will no longer be there.

So, I will obtain now, a new graph G*, which will be of course, simple because my original graph G was simple. So, I am not adding any edges I am deleting edges, so, by deleting edges, I will still obtain a simple graph. So, my graph G* will be a simple graph and it will have n - 1 nodes because I am reducing 1 vertex namely v_1. Now, what can I say about the new degrees of v_2, v_3 and vertex number d_1 + 1 well the degree of v_2 will be 1 less than what it was earlier, because the edge between v_2 and v_1 has vanished. The degree of v_3 will be 1 less than what it was earlier, because the edge between v_3 and v_1 has vanished and the degree of the d + 1th vertex will be 1 less than what it was earlier, because the edge between the d + 1th vertex and vertex number v_1 has vanished. The degrees of the remaining vertices will remain as it was in the graph G.

So, now, what can you say about this sequence, I can say that this sequence is nothing but the sequence S* in non increasing order, namely I can say that there is a graph, a simple graph namely G*, which realises the sequence S* because in S* in order that S* is a graphic sequence, you need a vertex of degree d_2 - 1 in G* and you have one such vertex namely v_2 you need 1 vertex of degree d_3 - 1 in G* and you have one such vertex namely v_3 and you need 1 vertex of degree this much.

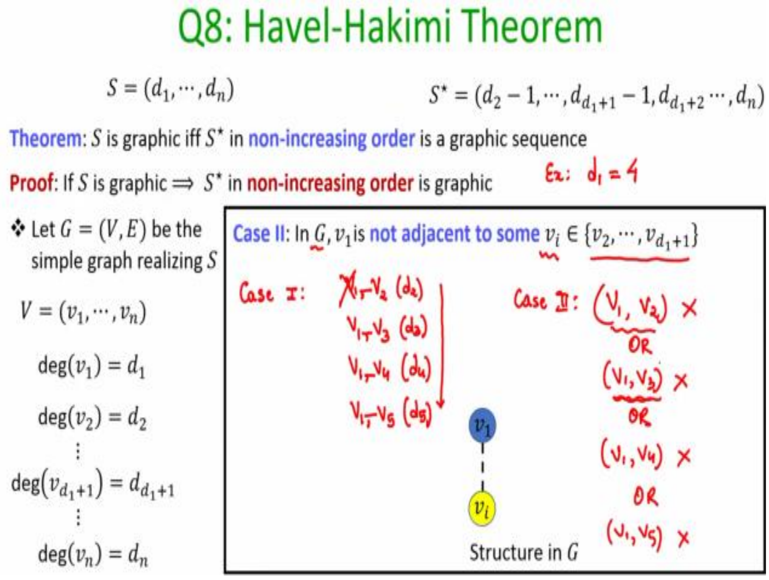
And you have a vertex in G* with that much degree you need a vertex of degree d_n in G* and you have a vertex whose degrees is d_n. So, that means, now I can say that G* can realise the sequence S* and hence my sequence S* is also graphic so, that is case 1.

(Refer Slide Time: 17:26)



Now, case 2 will be the following: case 2 occurs where in the graph G which realises your sequence S the structure is as follows: there is at least 1 vertex v_i in the set v_2 to the d + 1th vertex such that v_1 is not adjacent to that vertex. So, what do I mean to say here is the following in case 1 if you see the situation was that v_1 was adjacent, so, v_1 degree was d_1 and those d_1 edges were contributed from the next d_1 vertices namely the next d_1 vertices which has the degree d_2, d_3, d_4 and $d_1 + 1$ that was case 1.

(Refer Slide Time: 18:21)



In case 2 we are considering the case where this is not happening. That means, you have at least 1 vertex v_i outside this set v_2 to vertex number $d_1 + 1$ such that v_1 is not adjacent to v_i. So, what do I mean by that for instance, imagine that your d_1 is equal to say 4. In case 1, what was happening is the following you need v_1 to have 4 edges incident with v_1 that means 4 edges should be incident with v_1 that is why its degree was 4.

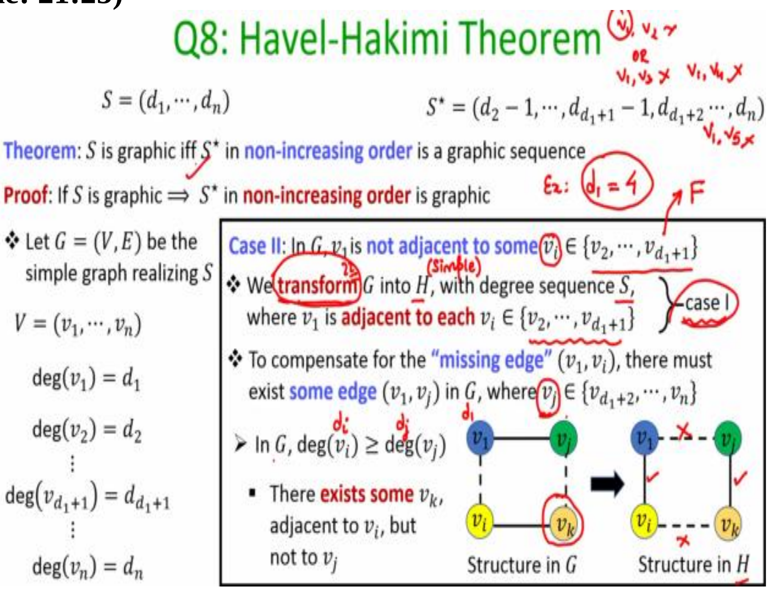
So, those 4 edges were between v_1 and v_2 where the degree of v_2 was d_2, it was between v_1 and v_3 where the degree of v_3 was d_3, it was between v_1 and v_4 where the degree of v_4 is d_4 and it was between v_4 and v_5 where the degree of v_5 is d_5 and of course the degrees are now in non increasing order that was happening in case 1, but in case 2 what is happening is your degree d_1 is still 4.

But either the edge between v_1 and v_2 is missing or the edge between v_1 and v_3 is missing or the edge between v_1 and v_4 is missing or the edge between v_1 and v_5 is missing where v_2, v_3, v_4 and v_5 are the vertices with degree d_2, d_3, d_4 and d_5 in the graph G respectively. So now I cannot run the same argument, which are used in case 1. In case 1, I simply deleted v_1 due to which all these edges which are there between v_1 and vertex 2, vertex 3, vertex 4, vertex 5, they vanished.

And the degrees of d_2, d_3, d_4, d_5 automatically got decremented by 1, I cannot run the same argument here. Because, say for instance, if the edge between v_1 and v_2 is missing, then by deleting v_1, I cannot say that the degree of v_2 gets decremented to d_2 - 1, because v_2 is not adjacent to v_1. Its degree will remain the same namely d_2 or say for instance, the edge between v_1 and v_3 is not there, then deleting v_1 will not change the degree of vertex v_3, it will still remain d_3 and so on.

So, I cannot run the same argument which I easily or conveniently used for case number 1, I have to do something more to handle the case number 2, and by the way, these are the only 2 cases either case 1 could occur or case 2 could occur, there cannot be any third case possible.

(Refer Slide Time: 21:25)



So, the proof strategy here will be the following. What I will do is I will do some transformation and we will see how exactly the transformation happens we will do some transformation on the graph G and convert it into another simple graph H with n vertices and with the same degree sequence S, that is important that means whatever what the characteristics of G were they remain the same.

So, G was simple that transformed graph H also will be simple, G had n nodes, that transformed graph H also will have n nodes, the number of edges in G will be the same as the number of edges in H and G was simple H will be simple. G realised the degree sequences and a transformed graph H also will realise the same degree sequences, but H will now have a characteristic which was not there in the graph G.

So, in the graph G there was some node v_i in this set. So, let me call this set as F so, there was some node v_i in the set F such that v_1 was not adjacent to that node v_i, but after transformation what we will do is we will ensure that v_1 is adjacent to each node in the set F, that means, the degree d_1 which was attributed to the vertex v_1 is coming because of the edges between the vertex v_1, v_2, vertex v_1, v_3, vertex v_1, v_4, vertex v_1 and d + 1th vertex.

That means, what I can say now is that my transformed graph H is exactly having the same structure as we had for the graph G in case 1 and now, I can apply the same argument that we used for case 1. So, now, I will say that I will forget about the graph G I will say that now I have a graph H, which is simple which has n nodes and which realises the degree sequence S and where the vertex with the highest degree d_1 is adjacent to the next immediate d_1 vertices.

So, I can remove the vertex v_1 and argue that because of the removal of the vertex v_1 the degree of the next to d_1 vertexes will get decremented by 1 and that will be an instantiation or realisation for the sequence S*. So, that is a proof idea. So, now, everything boils down to how exactly we do the transformation. So, the transformation is as follows so, remember the structure in the graph G is the following: there is at least 1 node v_i in the set F such that the edge between v_1 and v_i is missing.

And I also know that since the degree of the vertex v_1 is d_1 and edge between v_1 and v_i is missing. So, to compensate this missing edge namely to ensure that the vertex v_1 has the degree d_1 there must be some outside vertex and what do I mean by outside vertex namely that vertex S not in the set F, but in the remaining n - d_1 vertices. It is not among the first d_1 vertices. So, this vertex v_j is the outside vertex and there must be an edge between v_1 and that outside vertex v_j because we have to take care of the fact that the degree of v_1 is d_1.

So again, for instance, what I am saying here is if d_1 is 4. So in case 2, we know that either the edge between v_1 and v_2 is missing, or the edge v_1, v_3 is missing, or the edge v_1, v_4 is missing, or the edge v_1, v_5 is missing. But since I have to give degree 4 to the vertex v_1, that means v_1 is adjacent to either vertex 6 or vertex 7 or vertex 8 and so on. So, that is the vertex v_j that is outside vertex v_j in my current context.

And what I know is that in my graph G the vertex v_i, its degree d_i is as large as the degree of the vertex v_j because that is the structure of my graph G. So, that means there must be some neighbour of v_i call it v_k, which is not a neighbour of v_j. Because if every neighbour of v_i is also a neighbour of v_j and on top of that v_j is a neighbour of v_1. But v_i is not a neighbour of v_1, we arrive at the conclusion that the degree d_j is more than the degree d_i, which is not the case.

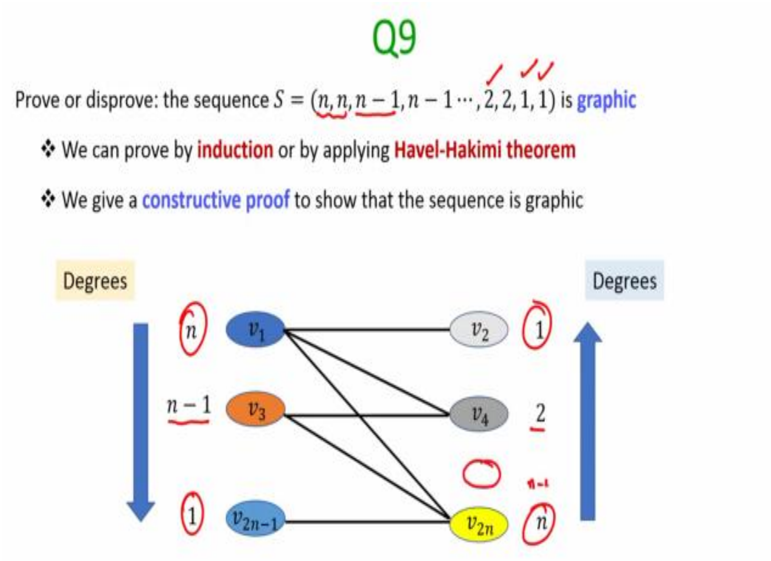
So that is a very simple proof of the fact that there must be some neighbour namely v_k, which is there must be some neighbour v_k of v_i, which is not a neighbour of v_j. So that is a structure present in your graph. Now, what the transformation does is the following. Since the edge between v_1 and v_i is missing in G, but after transformation, I want that edge to be present.

So, I add the edge but that will increment the degree v_i or degree of v_i, but I do not want to do that. So to compensate this new edge, which I have given to v_i, I take away the edge, which was earlier present between v_i and v_k. So, that ensures that the degree of v_i remains the same. And I have to take away the edge between v_1 and v_j because since I am giving a new edge to v_1, the degree of v_1 will get incremented, which I do not want to do.

So, to compensate that I take away the edge between v_1 and v_j, which was earlier there, but that will reduce the degree of v_j. again, which I do not want to do and to compensate that I add the edge between v_j and v_k and this whole process, I am not disturbing the property that my graph G or the transformed graph H is a simple graph. So, my transformed graph H still remains a simple graph.

But by doing this transformation, what I have done is the following: earlier this vertex v_i was not immediately a neighbour of v_1, but now in my transformed graph v_i is a neighbour of v_1. So, I can repeatedly apply this transformation for all the outside vertices v_i and after doing the required number of transformation, I will get my graph H which will have the same structure as in case 1 and then the proof becomes the same as it was in the case 1. So, that proves the implication in the other direction.

(Refer Slide Time: 28:29)

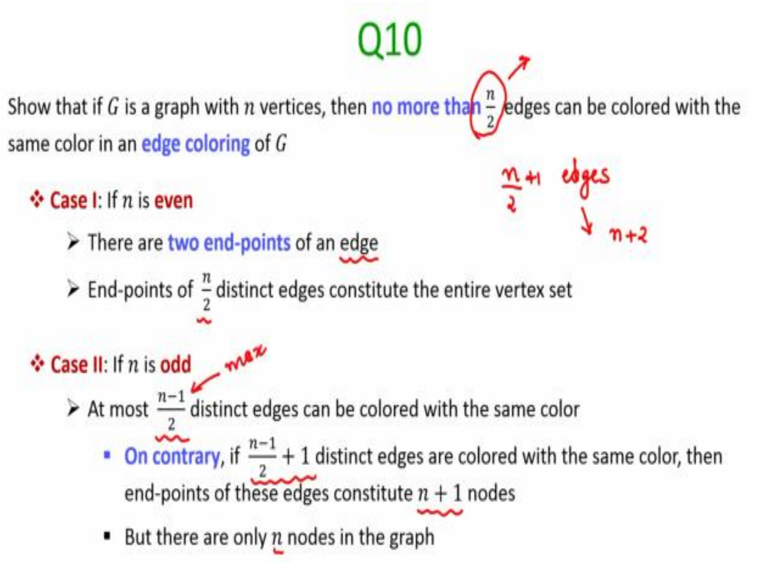


In question number 9, we want to either prove or disprove whether the sequence is a graphic sequence. So, there are 2 options: we can use either Havel-Hakimi theorem or we can use a proof by induction to prove that this sequence is a graphic sequence, but we will give a constructive proof to show that the sequence is graphic by showing a graph, a simple graph with 2n nodes whose degree sequence is same as S.

So here are the vertices: 2n vertices and what I do is the following. I take the vertex v_1 and add the edge with all vertices with even indexes. I take the vertex v_3 and I add an edge with all even index vertices except the vertex v_2 and I keep on doing this process and for the last vertex with odd index, I will give only 1 edge namely an edge with the last vertex with even index.

Now, what I can say about the degrees of the respective vertices here, so it is easy to see that these 2 vertices will have degree n so indeed, I need 2 vertices of degree n. I will have this vertex of degree n - 1 and this vertex of degree 2, so I got 1 vertex of degree n - 1 and 1 vertex of degree 2. And if I continue, I will find that I will get 2 vertices of degree 1 and then eventually I will obtain the second vertex of degree n - 1 and so on. So the vertex here will be of degree n - 1 and so on, that is a very simple construction to show that the sequence is a graphic sequence.

(Refer Slide Time: 30:07)

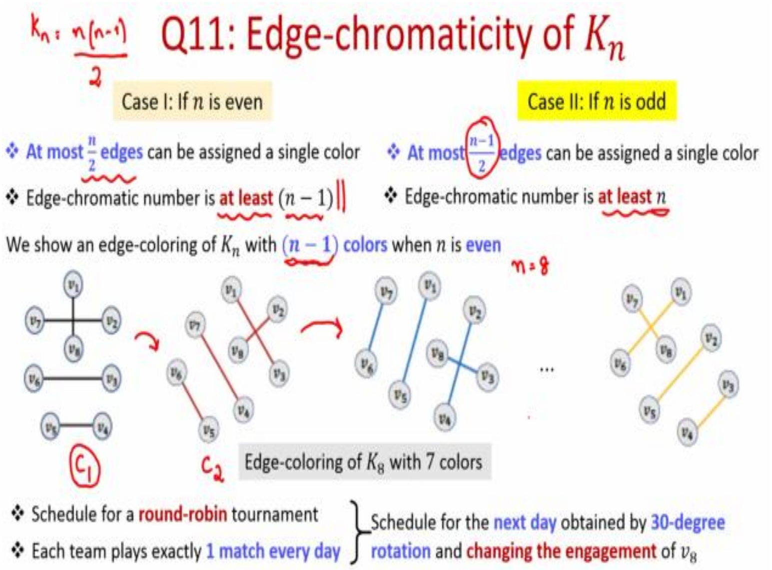


Now let us come to question number 10. Here, we want to prove that if you are given a graph with n vertices, and if you are doing edge colouring, then you cannot use a single colour to colour more than $\frac{n}{2}$ edges. And it is a very simple fact, depending upon whether your n is odd or even, we can prove this very easily. So let us take the case where n is even. So remember, each edge has 2 endpoints.

That means if I consider $\frac{n}{2}$ distinct edges of the graph, and if I focus on their endpoints, that will constitute the entire vertex set. So that means I cannot colour $\frac{n}{2}$ + 1 edges with the same colour, because if I do that, then their endpoints will give me n + 2 vertices, but my graph at the first place has only n vertices. So at most I can colour $\frac{n}{2}$ edges, distinct edges with the same colour, I cannot colour more than those many edges.

Whereas if n is odd, then this quantity $\frac{n}{2}$ is not well defined, it will not be an integer value. So $\frac{n}{2}$ in that context of an odd value of n will be $\frac{n-1}{2}$. And indeed, it is easy to see that I cannot use a single colour to colour more than $\frac{n-1}{2}$ number of distinct edges, because if I try to do that, say for instance, I tried to colour $\frac{n-1}{2}$ + 1 number of distinct edges, then their endpoints will give or constitute will n + 1 nodes, but my given graph has only n nodes. So that is a maximum number of distinct edges, which can be coloured with a single colour.

(Refer Slide Time: 32:04)



Now, based on these information, I will try to solve question 11 where I want to find the edge chromatic number of a complete graph with n nodes and my solution will be divided into 2 cases depending upon whether my n is odd or even. So, remember, from the previous question, I know that if your n is even then you can use 1 colour and colour at most $\frac{n}{2}$ edges, whether indeed you will be able to colour $\frac{n}{2}$ edges or not that depends upon the structure of your graph, but at max you can colour $\frac{n}{2}$ edges using a single colour.

Now in a complete graph, k_n I have $\frac{n(n-1)}{2}$ number of edges. So, with colour number 1, I can take care of at most $\frac{n}{2}$ edges. With colour number 2, I can take care of another set of $\frac{n}{2}$ edges. So, like that, how many colours I will require at least? So, I will require at least n - 1 number of colours, because to the first colour, I can take care of $\frac{n}{2}$ edges, the next colour I can take care of another bunch of $\frac{n}{2}$ edges and I have to take care of n - 1 such bunches of $\frac{n}{2}$ edges.

So, that is why the minimum number of colours that will be required will be n - 1. Whereas if I take the case when n is odd, then from my analysis of question 10, I know that through 1 colour I can take care of at most $\frac{n-1}{2}$ number of edges. And I have to take care of n bunches of $\frac{n-1}{2}$ number of edges. So, that means I will require at least n colours if my n is odd.

Now, what I will show is I will show that these bounds on the edge chromatic numbers, which were the lower bounds because they were the least number of colours which are required, they are actually tight in the sense I will give you a constructive colouring, a concrete colouring for colouring the edges of a complete graph with n nodes where n is even. And where the number of colours required is exactly n - 1.

And you cannot beat this bound because the lower bound says you will need at least n - 1 colours. So that is why I am giving you an optimal colouring. So let me demonstrate the colouring assuming the value of n = 8. So remember, edge colouring here corresponds to scheduling of a round robin tournament. So we have 8 teams, and we have to schedule matches among the teams.

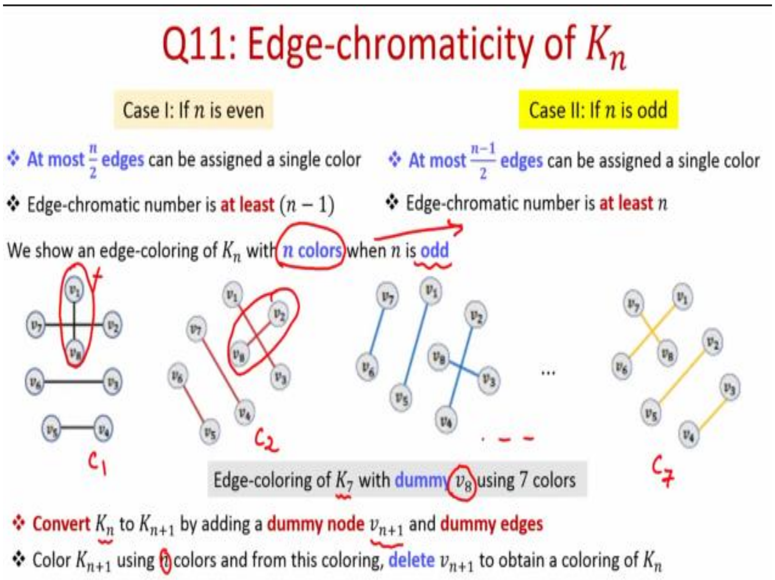
And the requirement is that each team has to play against every other team once but at the same time, we do not want to enforce a team to play more than a single match on any day. So the way I do the colouring here is as follows. So on the first day, I keep v_8 at centre and engage v_8 with v_1 and engage v_2 with 7, engage v_3 with 6, engage v_4 with 5. So, this is equivalent to saying that this colour c_1 is used to colour the edges between (v_4, v_5), (v_3, v_6), (v_2, v_7) and (v_1, v_8).

That means, I have coloured the maximum number of edges using colour number 1 and now, I have to use colour number 2 and using colour number 2, I will try to colour another set of 4 edges. So, which is equivalent to saying that I now want to find a schedule for the next day. So, the schedule for the next day is obtained by kind of rotating this diagram by 30 degrees and changing the assignment of or engagement of v_8.

So, earlier v_8 was engaged with v_1, but now v_8 will be engaged with the next team in the clockwise direction which is v_2. So, now, the assignment of the colours is the following. So, I use the colour number c_2 to colour these 4 edges, or equivalently I schedule these matches on day number 2, then again I shift it by 30 degree and change the engagement of v_8 . Now v_8 will be engaged with v_3 and so on.

So, now you can see that I have to do this rotation 7 number of times, and then I will be able to colour all the remaining edges of my complete graph with n nodes: 8 nodes.

(Refer Slide Time: 32:04)



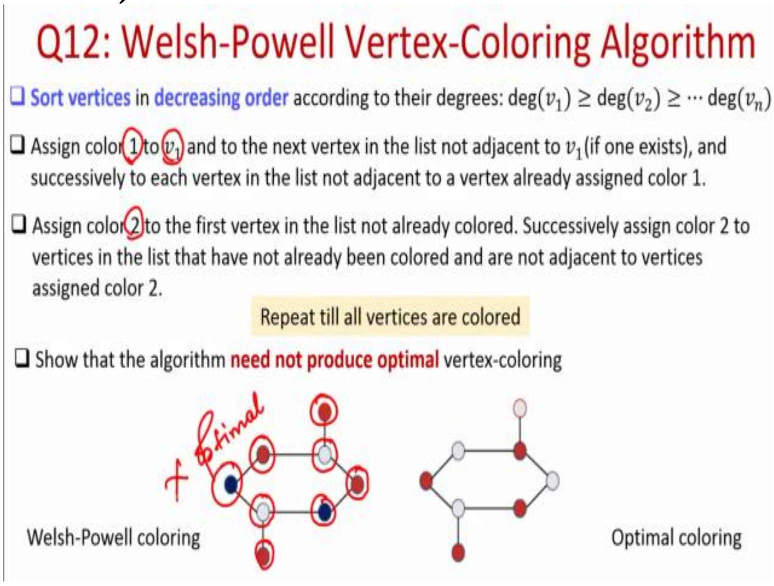
Now, let us see how exactly we can colour all the edges of a complete graph with n nodes where n is odd. And I will be using exactly n colours, which is optimal because the lower bound says that for n being odd, I need at least n colours. So, the idea here is I can convert the complete graph with n nodes to a complete graph with n + 1 node by adding a new vertex and the required dummy edges.

And since n was odd, n + 1 will be even. And I know a colouring mechanism to colour a complete graph with n + 1 nodes where n + 1 is even using n colours, namely the colouring that I had discussed just now. So, take that colouring and now you delete the dummy node and the corresponding edges. That will give you the colouring for the original complete graph with n nodes where n was odd.

So, for instance, what I am saying here is if you have only 7 teams, and you want to come up with a schedule, you imagine that you have included a dummy team, say the 8th team and now you want to come up with a round robin scheduled tournament for 8 teams with the same restrictions that you had earlier. So, this will be the schedule you will require 7 days. Now in the first day you can see that; on the first day you can see that v_8 is engaged with v_1.

So you can forget about that you can imagine that match is not going to be held; remaining other matches will be held as per the colouring assignment namely v_2 will play with v_7, v_3 will play with v_6 and v_4 will play with v_5 . On the second day v_8 is engaged with v_2. So, you can imagine that match will not be there actually and remaining 3 matches will be played and so on. So, this now gives you a colouring; edge colouring for complete graph with n nodes when n is odd.

(Refer Slide Time: 38:37)



Now, question 12 we are giving a greedy strategy for vertex colouring and we want to prove that this strategy need not give you the optimal vertex colouring. So, the colouring strategy is the following. We first sort the vertices according to their degrees and then use colour number 1 to colour the vertex which has the highest degree that you have arranged as per your degrees and then the next vertex in the list which is not adjacent to be v_1, if at all it exists and successively try to colour as many vertices as possible according to the colour number 1, keeping in mind that the next vertex which you are selecting is selected according to their degree. That means, you are following a greedy strategy and trying to occupy or colour as many vertices with that colour and now do the same process with the next colour and so on.

So now we have to give a counter example, namely a graph where Welch-Powell algorithm will end up utilising more colour than the optimal number of colours. So consider this graph and let us see how many colours we need. Actually we need 4 colours as per the Welsh-Powell algorithm because this vertex has the highest degrees so I will colour it and then I can assign the same colour to this vertex which has also the same degree.

And now I cannot use the same colour to colour any other vertex. Now, I will focus on the next set of vertices which has the highest degree. So let us use this vertex, this vertex and then the same colour I can assign to this vertex and this vertex. So that is the maximum number of vertices which I can colour with the second colour. Now among the remaining vertices I will pick the vertices which have according to their degrees.

So I can pick this vertex and the same colour can be assigned to this vertex. So we need total 3 colours; but optimal colouring is 2 this will require only 2 colours and 2 colours will be sufficient to colour all the vertices in this graph so that shows this is not optimal colouring, so with that I conclude this tutorial. Thank you.