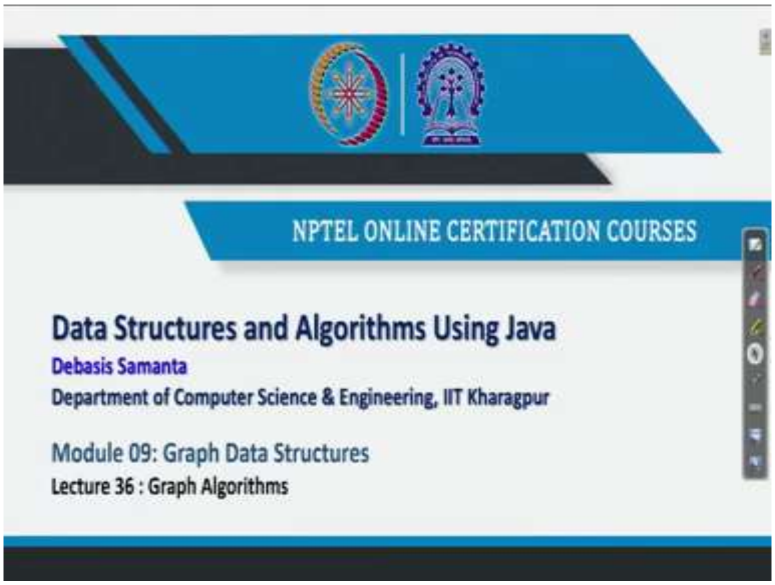
Data Structure and Algorithms using Java
Professor Debasis Samanta
Department of Computer Science and Engineering
Indian institute of Technology, Kharagpur
Lecture 36
Graphs Algorithms

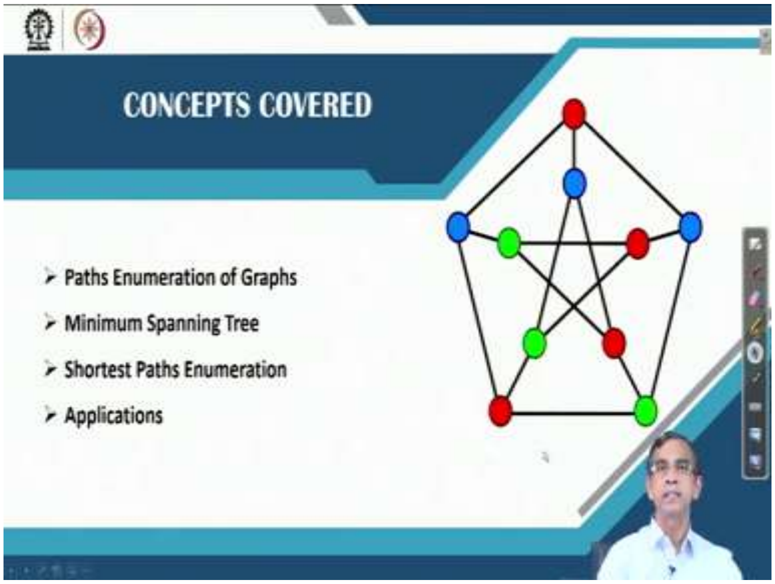
(Refer Slide Time: 00:34)



We are discussing about graph structure, this is our 9^{th} Module in our total discussion out of 15 modules. And we have discussed we have learned about basic concept of graph and how graph can be stored and few operations related to the graph.

And I was referring that graph is an important concept in the discipline in computer science and engineering discipline, and it has been an exercise like anything, and many algorithms, many interesting problems related to the graph have been solved. We will try to cover few very interesting concept related to the solving graph related problems, actually.

(Refer Slide Time: 01:09)



So, todays topic is basically how we can trace the paths of a graphs. Now, what are the different paths that is possible, I will discuss. And then one important concept of path problem in graph theory, it is called the spanning tree. Now that will be discussed. And then there are many algorithms to find the shortest path.

As you can know from one node to another node you can find many ways to go there from source to a distance or node. But out of the many paths we have to find one path which is called the shortest path. So, this is the one example and finally I will conclude these lectures by giving or hinting some applications of the structure.

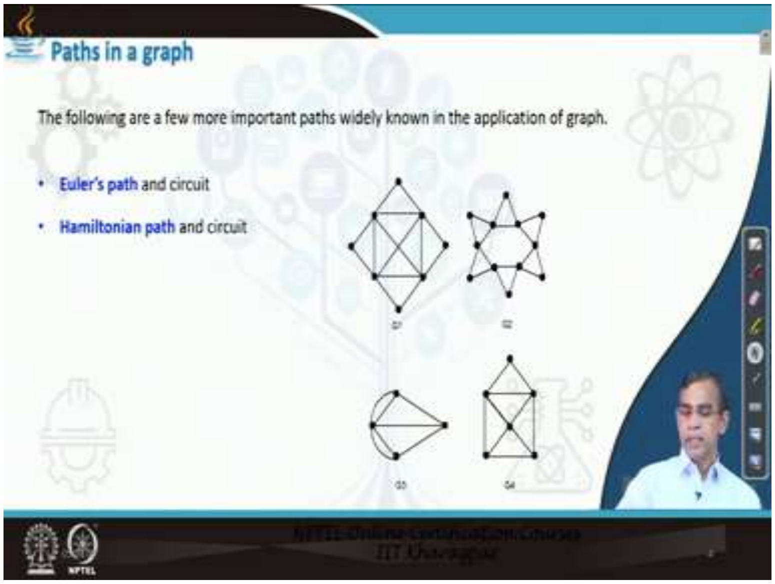
(Refer Slide Time: 01:56)

Now, let us talk about Path Enumerations of Graphs what exactly the path enumerations is there. I can tell you a graph is given and you have already studied about the traversals, two traversals depth first and breadth first. Basic objective all those traversal to visit all nodes but there are certain constant can be given. So, for example, you have to visit all nodes, but two nodes should not be visited more than once.

You can exactly visit each node only once. So, if this constant is given, then how we can traverse the inter-graph definitely DFS, BFS cannot help you to solve this problem. On the other hand opposite problem maybe so, we have to visit it all edges. There is a possibility that in order to do it, you can visit a node more than once. Now here basically, the idea is that you should visit all edges exactly once, you cannot you are not allowed to visit two edges more than, now one edges more than twice or whatever or more than once.

So these are the different constant that can be given and then you have to solve this problem. You can think that is a puzzling problem like but very new interesting algorithms are there by which you can solve this problem.

(Refer Slide Time: 03:13)



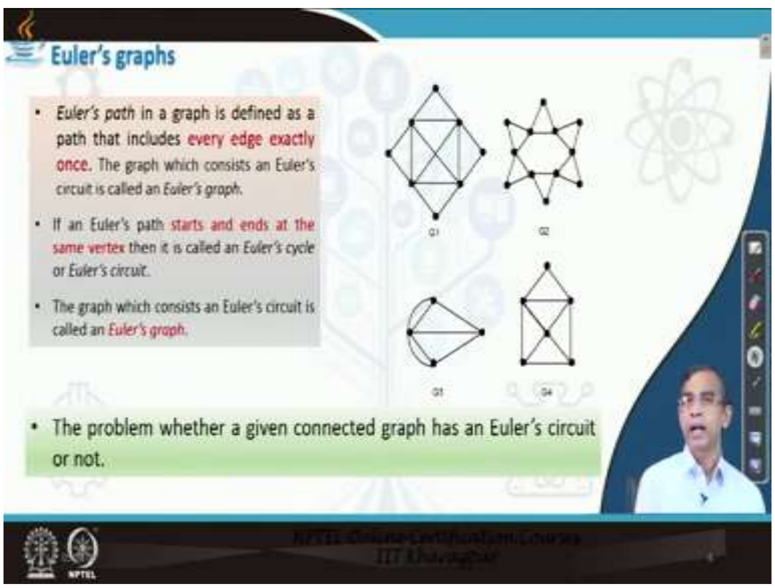
So, today basically we will discuss about how the different, two different paths are there. And the first one path is called the Euler's path. And the circuit means if you start from the source to come back to this and visiting a path that is Euler's path then it is called the Euler's circuits. Like Eulers path there is another path is called Hamiltonian path.

So and then, even starting from a node to return to the node is called another circuit is there or loop is there is called Hamiltonians circuits. Now Euler path is basically the concept is that we have to visit all edges exactly once.

In case of Hamiltonian path is basically, we have to visit all nodes exactly once. Now, here are few graphs, it is given there whether you will be able to accomplish or each both the paths Euler path or Hamilton path like. So, these questions let us see how we can solve it. So, there are a graph will be given to you input graph and then a question will be asked that, okay whether this graph contains an Euler path?

An Euler path means you can visit all edges exactly once or Hamilton path that means there is a path which basically traverse all the nodes exactly once. So, that needs to be here. Now we will see exactly how this kind of problem can be solved.

(Refer Slide Time: 04:45)



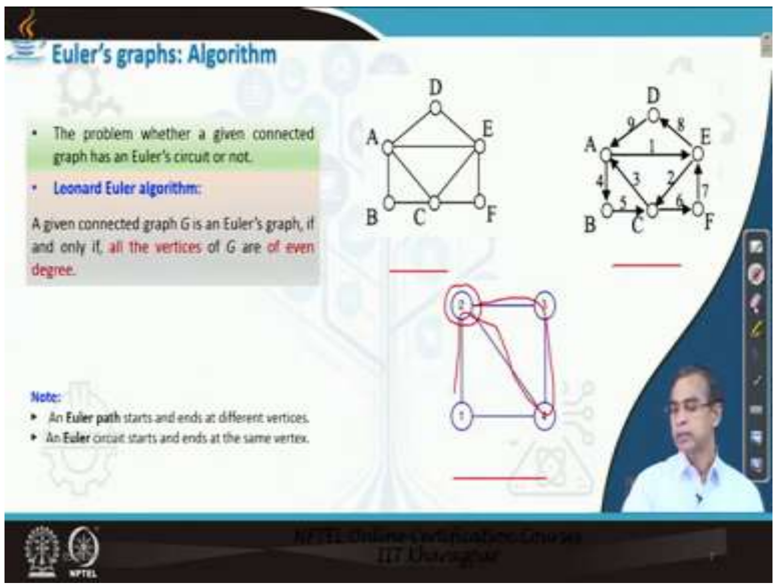
Now let us first start with the Euler's graph. Now, there is a Euler is basically this is the I told you that father of graph theory, who is known as Euler, who basically invented this problem and then solved this problem also. Now here a graph will be defined as a Euler graph if it has a path which includes that every edge exactly once.

The graph which consists of, which contains an Euler path is called Euler graph or Euler circuit also that means circuit is starting from the same node, if you want to return to the starting from a node, if you can come back to the same node, then it is called the Euler circuit.

But if it is starting from one node, but you have to visit all edges not necessarily return to the same node, then it is just simply not circuit it is called the path. Now so there is a basically difference between path and circuit. Circuit is a closed loop, a path is not a closed loop. Now, the graph which contains an Euler circuit is called Euler graph. So, that is the definition it is there.

Now, as I told you the problem is that out of these 4 graphs which are the Euler graphs okay weather G1, G2, G3, G4 or whatever it is there. Now, this interesting problem has been solved by Euler's.

(Refer Slide Time: 06:22)



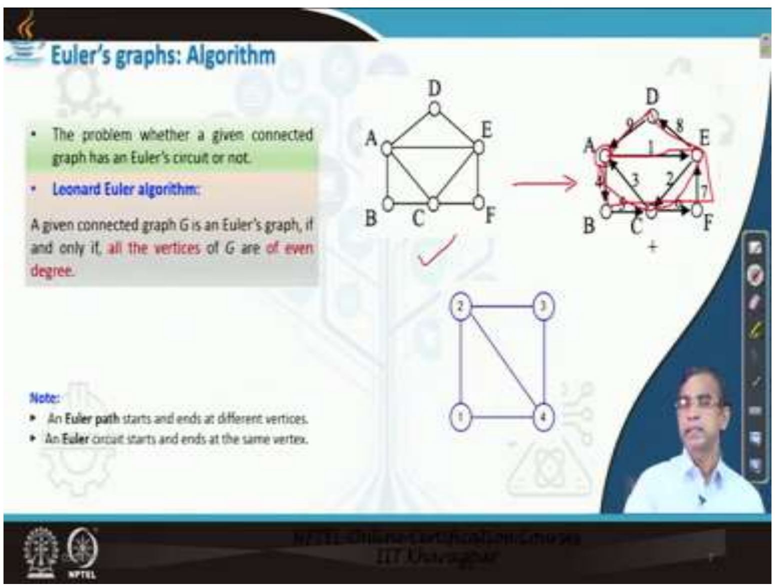
And this problem is like this a given connected graph actually there is a graph called connected and disconnected. There are so many nodes and there is no connection from one node to any other node then it is called disconnected or there is a connection from one node you can visit to any other node then you can say connected.

But here disconnected graph means from a given node, from some node together we can say that you will not be able to visit all nodes or some nodes then it is called the disconnected. But this solution definitely valid or this problem also valid for the connected graph that means given a graph you will be able to visit to any nodes. So, in this example, as you can see, all are basically the connected graph.

So, this is also connected, connected and these connected, connected means a starting from any node, you will be able to visit to other nodes like this one. Now, a given connected graph we will be termed as Euler graph if this condition is satisfied. What is the condition? Condition is that all vertices of G are with even degree. Even degree means if there is any node having that degree odd then it is not an Euler graph.

Now here for example, in this graph. So, this is the one node having the degree 3 that means it cannot be Euler graph. That means starting from these we will not be able to visit all the edges exactly one. If you start here, then here, then here, then here and sorry, so I am. Okay fine, so there is a mistake now yes.

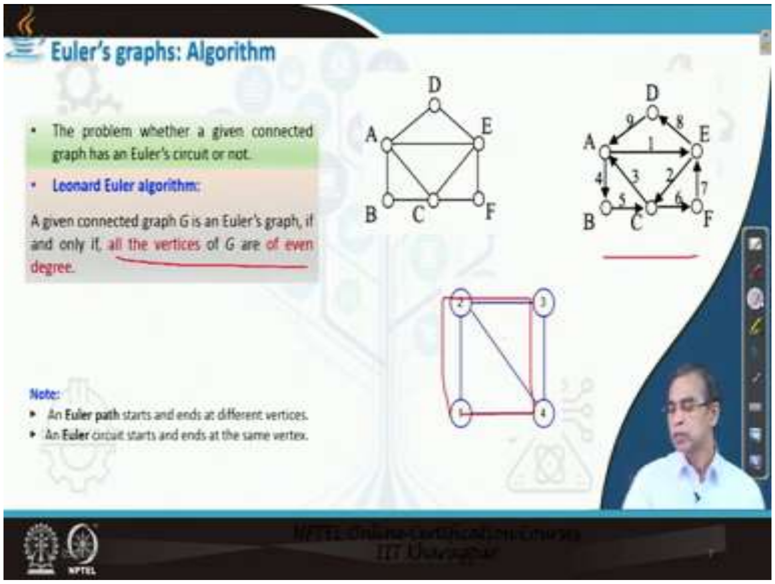
(Refer Slide Time: 08:07)

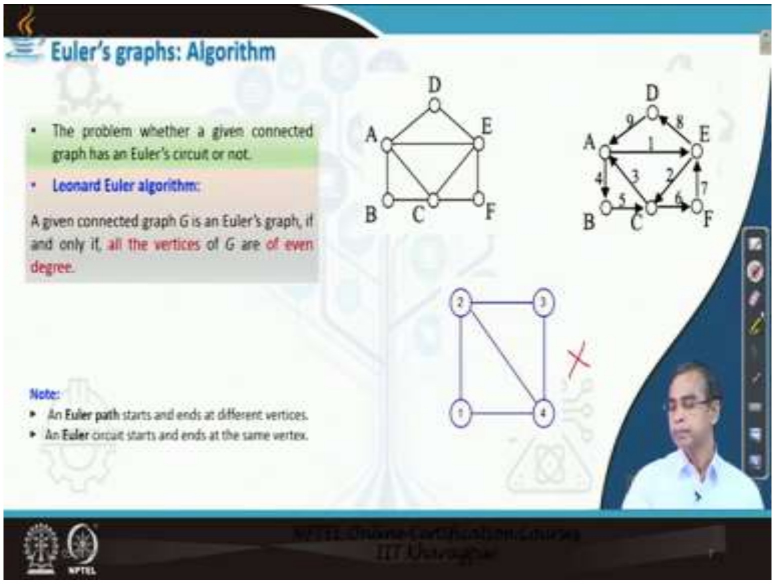


Here if you see so all the vertices of G are of even degree. Now here in this example as you can see this node having the degree 2. So, even this is also even, this is also even, this is also even, this is also even. So, all the vertices are of degree even. This means that this is basically is an, this is basically an Euler graph. Now let us come to this one and this basically here, how we can visit all the paths, all the edges, all the edges rather all the edges exactly once. So, we can start from A to. We can go to E then this one, then we can go there then here 5, 6, 7, 8, 9.

So starting from, so here we can come back to one. So, this basically is a, we can say Euler circuit also it is an Euler path starting from A to return to A is an Euler circuit and here we can say that we could traverse all the nodes exactly once and following this path.

(Refer Slide Time: 09:27)



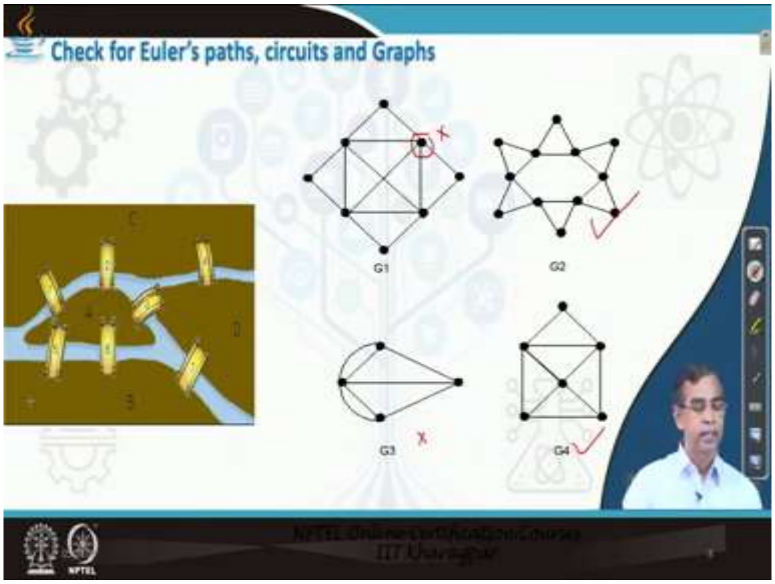


So, this means that in this examples, this basically satisfied, that all the vertices of they are even degree, this means this basically is an Euler graph. Now, let us see whether this is an Euler graph or not. We can start from here. We can go there and there, then there, then there. Now we will not be able to visit this path, according. Because if we visit then either any one edges to be traverse from here but you can start from any other nodes also, again, you can check it whether you will be able to I mean, go and come back there or not.

So in that sense, this actually this graph is not a, not an Euler graph, okay. So this is a very simple solution rather, but it solution was not so easy initially to find this I mean to solve this problem. It takes long time to solve this problem and ultimately who introduced this problem himself solved this problem. And he is a Leonard Euler and this algorithm solving this problem, is called Euler algorithms.

And then how the different path can be traced, this algorithm is beyond the discussion in this course. You can find a book that I will give a reference and you can find algorithm, how the different path can be traced there.

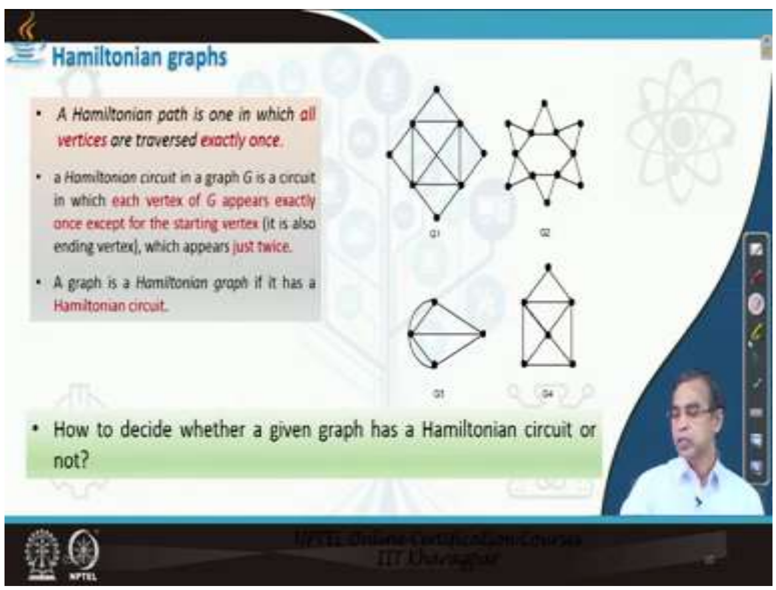
(Refer Slide Time: 10:47)



Now I have given some input graph you can check easily based on this formula that weather all these 4 graphs are Euler graphs or not. Now, what in your… So this example G3, as you see, these are basically (even) odd degree. So, it cannot be Euler, right. And here exactly even degree all are even degree these are even degree. So, this is an Euler graph and here also this is an Euler graph. Now, what about this one, right? So, this is not an Euler graph because this degree is odd. So this way, given a graph we can quickly check weather a graph is an Euler graph or not.

Now, again this is a (())(11:35) from problem. Now you can understand that, okay, visiting all bridges, exactly once weather you can reach to all lands or not you can think about. And if you can do it, you can do. So, this is the idea about the concept of right Euler graph.

(Refer Slide Time: 11:55)

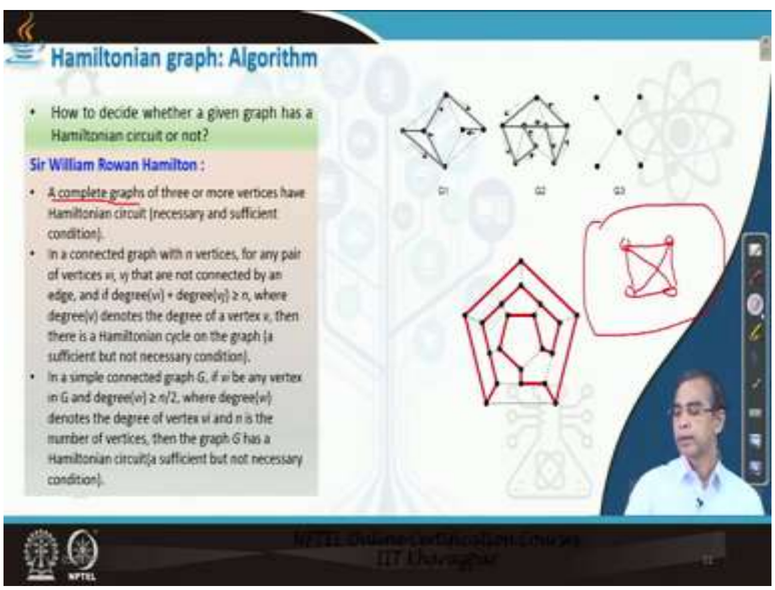


Now, let us come to the another path enumeration, it is called the Hamiltonian Graph. Now, likewise Euler, Hamiltonian graph is a constant problem, path constant problem. Here problem is that given a graph, you have to visit all the node exactly once, so and then the path which basically can, you can think about visiting all node exactly one is called the Hamiltonian path. So, a Hamiltonian path is one in which all vertices are traverse exactly once.

Now, a Hamiltonian circuit it is basically same if we if the starting vertices and then n vertices are the same then that path is basically is a closed loop and it is called the Hamiltonian Circuit. So, Hamiltonian circuit in a graph is a circuit we need each vertices of G appears exactly once except for the staring vertices. That means starting vertices is basically to be visited twice, whereas all other vertices will be visited exactly once.

Now, a graph which contains a Hamiltonian graph, Hamiltonian circuit is called a Hamiltonian Graph. Now, again I have given few graphs say G1, G2, G3, G4. Now, can you check weather a graph is an Hamiltonian graph or not. That means you will be able to find a path such that you can visit all nodes exactly once. And a circuit, if you can find a path or circuit which basically traverse all nodes exactly once except the starting point, starting nodes. Now, so this problem is a puzzle we can say. Now let us see how this problem can be solved.

(Refer Slide Time: 13:55)

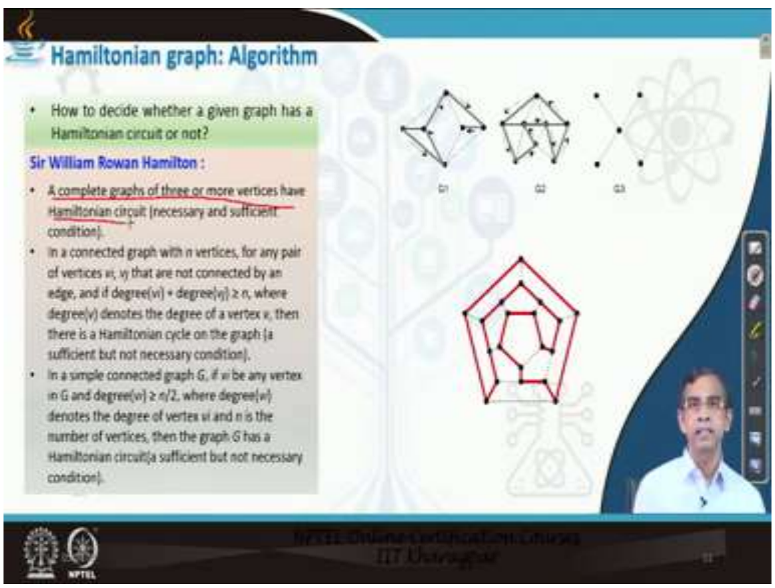


And again, there is this problem is proposed by William Rowan Hamilton, is a (())(13:59) mathematician, and he himself again solved the problem again. So is basically the problem is how to decide whether a given graph has a Hamiltonian circuit or not. Now, there are few compositions, he solve that there are some conditions that needs to be satisfied if a graph satisfy this condition, then we can say that graph is an Hamilton graph or not.

Now so here the first condition it is basically a complete graph. Now, a graph is called complete graph if from a node we will be able to visit to all other nodes in the graph. Now here, obviously no one graph is complete graph, but okay and here I can give you an example of a complete graph.

Say suppose this is the one node and this is the one. So, there are 4 vertices are there from these vertices we can go there, we can go there. We can go there. So, this is basically there is a that means from this node we can visit to all other nodes. Similarly, from there we can go here, we can go there and we can go there. From there also we can go there and so this is an example of a complete graph.

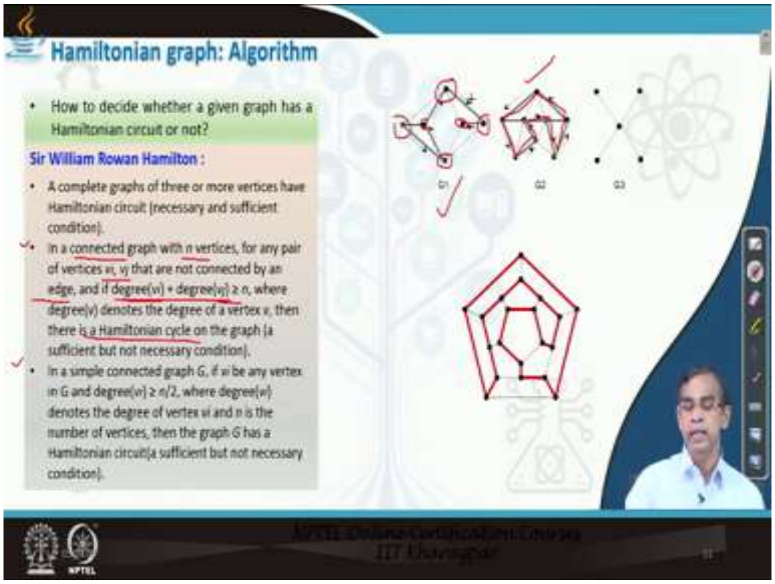
(Refer Slide Time: 15:24)

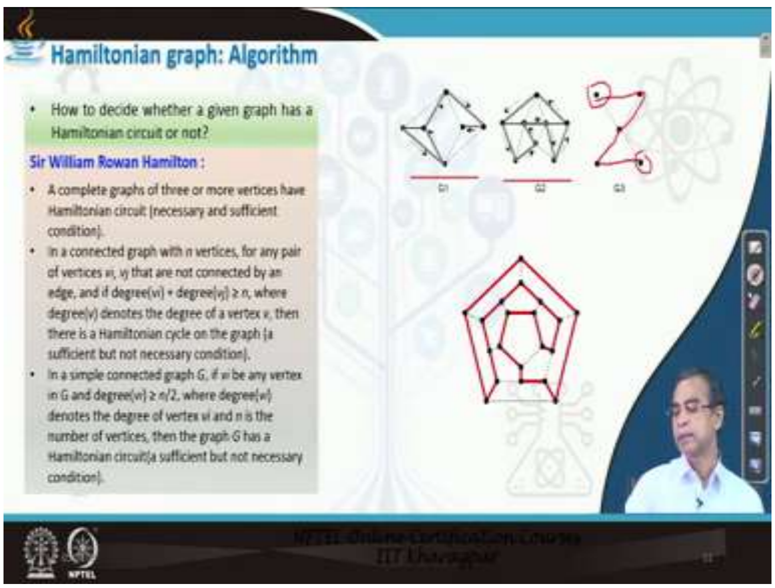


Now, we can verify that if there is a graph with n number of nodes, then total number of edges that is possible in a complete graph is n into n minus 1 by 2 that you can verify and you can prove also. But here you can see no one graph that is included here in this example are complete graph in that sense. Anyway, let us come to the problem solving. A complete graph of 3 or more vertices have Hamiltonians circuit it is always that mean if given a complete graph, you can tell that without any check, without any other conditions, that it is basically that Hamiltonian Graph or not.

A complete graph contains always, right it is always true and this is a necessary and sufficient condition. If the graph is complete, this implies that it has always the Hamiltonian circuit. Now, this is another way of checking weather graph is a Hamilton graph or not. This is very interesting one problem and then proof and everything there is a formally proof is there.

(Refer Slide Time: 16:33)





So, proof is beyond the scope of this discussion. Here see that there is not necessarily to be… So this is valid for complete graph, but not valid for other graph. But these are the second these points and this point is for any other graph actually. In a connected graph obviously, graphs has to be connected, otherwise there is no possibility of visiting all nodes in the graph. So, there is a connected graph and let the number of nodes in the graph is n. So, n vertices are there.

Now you choose any two pair, let that two pair is vi and vj that are not connected by an edge. So for example here so this is on vi and this is the another vj. They are not connected to each other by an edge. So, now if the degree of vi so for example vi, the degree is 3 and degree of vj, this is the degree of vj it is also 3. So, degree of vi plus degree of vj, if it is greater than equals to n.

Now number of nodes in this case for example, number of nodes is 6, okay. So, 6 and now degree of vi is 3 and degree of vi is 3 if it is greater than equals to n then we can say that there is a Hamiltonian circle on the graph. Now, we have considered for this path, but we have to check for all pair of vertices that is the problem. So, if this pair of vertices and this is connected we should not take. Now we can take this one not connected.

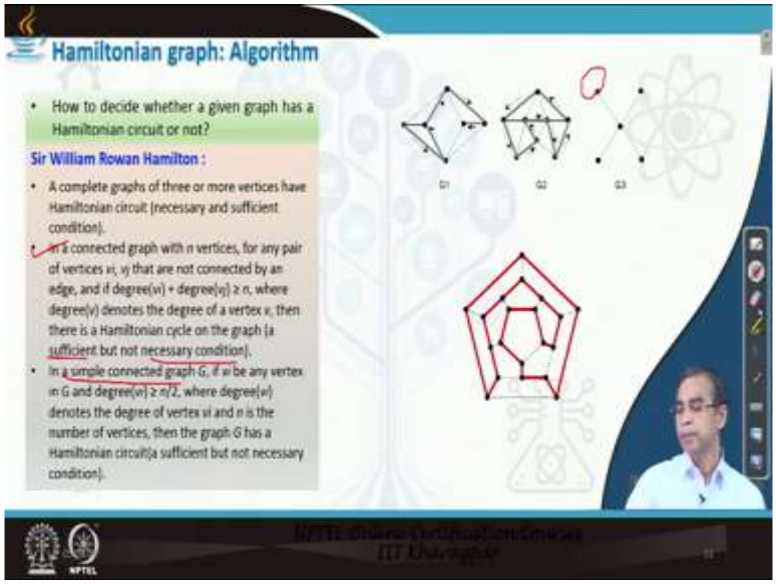
Now you can consider, you can apply this formula here is a 4 and 4, 8. So, it is greater than n, n is 6 in this case. So, it is also valid. Now here again, this and again this, not, this is a pair of vertices which are not connected here. So, this is also satisfied. This means that this G1 graph is basically is a Hamiltonian graph. Now, what are the different this basically shows you the path which you can follow to visit all nodes exactly once.

Starting from any one nodes, in this case, you can start from here, right. So, here you can go, then go this, this, this, this and then come here, come here. So, this is the Hamiltonian circuit that is possible for this graph actually. And if we see the Hamilton path starting from these two go there and this is the Hamiltonian path.

So, in this graph, so this graph, this is the graph shows that Hamilton path and this is the graph shows that Hamilton circuits that we are coming to return in the same path it is like this one. Now, here again check weather this graph G3 satisfy this formula or not. For example, in this case total number of nodes is 5. And so this is the one and this is the one, two disconnected pair of vertices you can say.

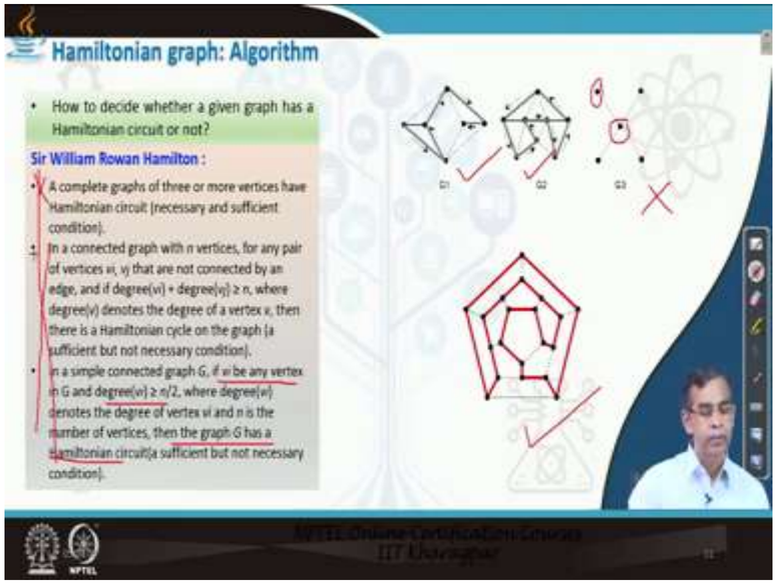
Now some of the degrees of edges here 2 plus 2, 4 and which is not greater that equals to 5. That means you will not be able to visit all node is exactly once traversing this path. You will not be able to find it. For example we can start from here. We can go there. We can go there. And then right that path is there, okay. But we have to find a path you will not be able to because if you want to find a path, then you have to go I mean, traverse the node twice, whatever it is there. So, this is the problem that you can think about.

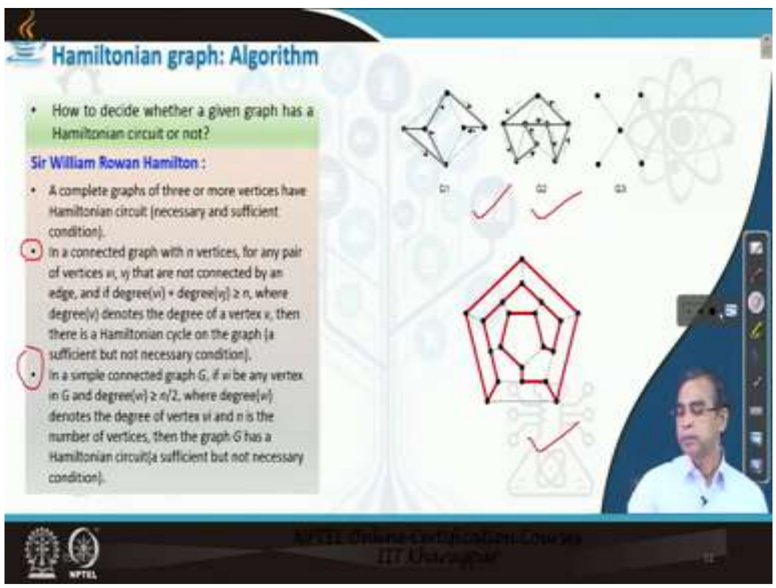
(Refer Slide Time: 20:15)



Now so this is the condition that we have discussed so for this on one condition but this is again not sufficient. So, that is why this condition, this is a sufficient but not necessary condition in some cases it will, but in some cases it cannot be possible also. Now there is another composition by which we can test also this in a simple connected graph. Now a graph is called simple connected if there is no loop. So, loop means from this node, if you can start this one. So, if there is no self-loop, then it is called the simple connected graph.

(Refer Slide Time: 20:58)





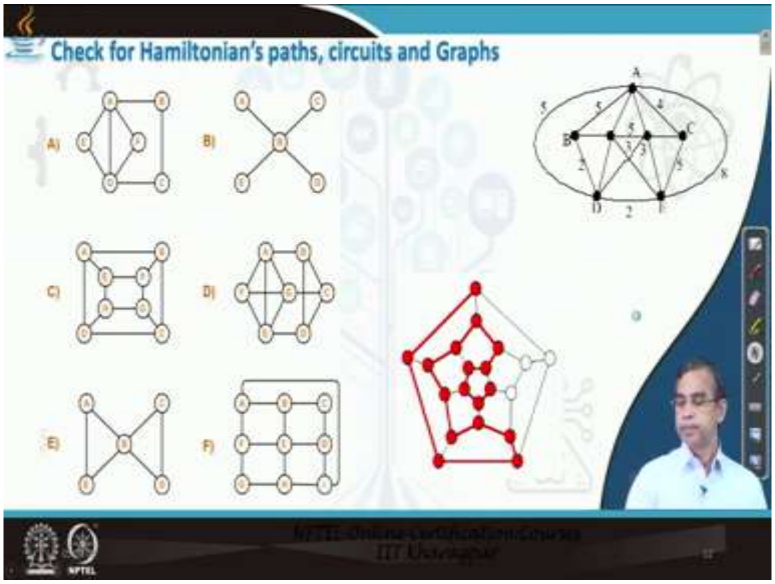
So if in a simple connected graph if vi be any vertex in G and degree of vi is greater than or equals to n by 2 then the graph G has a Hamilton circuit, and it is also sufficient but not necessary condition. It is true, but not oppositely true also. There may be some cases where it is not possible, but if it is satisfied then it is necessary.

Now, combining the two things also we can have the condition also that is there. Now for example, here n is in this graph n is 5 and degree of vi should be greater than or equals to n by 2. So, greater than 5 by 2 means, it is 3 like. So here degree of this node is not degree greater than 3 but it is true. So, this way this graph is again also not a Hamiltonian graph. But this graph also we can check it. It is hollow edge and this is also hollow edge.

And this is also another example which having the Hamiltonian path, which basically satisfied all these three conditions actually. Anyway, so this is not a complete graph, so we cannot tell this one. But we can, we can apply these two conditions, we can apply these two conditions. So, these two conditions we can apply to all these graph and this graph also, you can check that they satisfy this one and therefore you can say that all these graphs are Hamiltonian graph and like this one.

So, this way you will be able to detect weather a graph is an Hamiltonian graph or not. Likewise the here. And another thing is that in case in Hamiltonian path or circuit not necessarily that unique path is there. There may be more than more than one path is also possible. That means see you can have more than one path as a Hamiltonian path or more than one circuit as a Hamiltonian circuit that is quite possible, particularly for a complete graph there is more than one path is possible, actually.

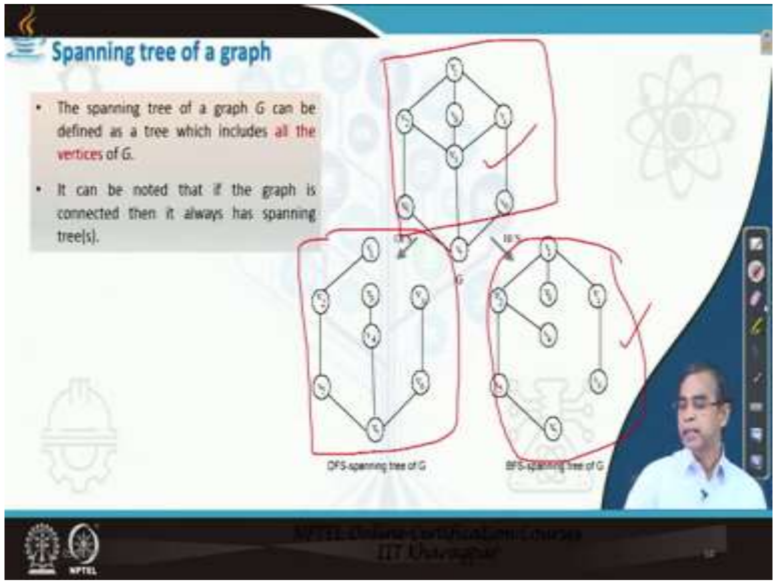
(Refer Slide Time: 23:07)



Now, here I have given few examples to apply your concepts. So, learning so that you can check it and so you can think about whether these are all graphs, either Hamiltonian graph or which one is Euler graph and whether a graph is both Euler graph or Hamiltonian graph or like this one. So, that is left as an exercise for you. And this is a very interesting problem that you can think about to test the Euler path as well as the Hamiltonian path if it is there.

So, first of all, you have to decide whether the graph contains Euler circuit or not or Hamiltonian circuit or not. Then next thing is that if circuit if the graph is an Euler graph, then what is a Euler circuit or graph is an Hamiltonian graph then which is the Hamiltonian circuits. So, finding a path or circuit, either is a Euler or Hamiltonian algorithm is there. But I will not be able to discuss those algorithms here.

(Refer Slide Time: 24:12)



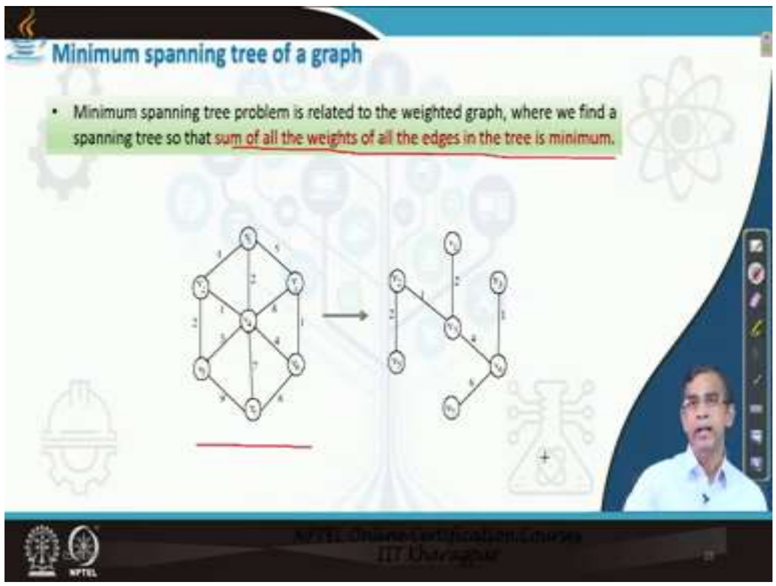
Now I will discuss another important, path enumeration problem, and it is called the spanning tree problem. The spanning tree is basically is a minimally connected among the nodes. It is actually, as the name implies that this is basically tree. Now, for an example here suppose this is the input graph. And if we follow certain DFS, then it basically this is basically the graph, if we apply the DFS traversals it will give the order, it will give a path in which the different nodes will be visited.

So here v1 to v2, v5, v7, then v4, v8, then v7, v6. So, this basically as you see it looks like a tree. Then this is called the Hamiltonian. This is called a spanning tree. Now here this is also another spanning tree. If we apply the BFS traversal to this graph and this kind of tree will be available. So, this is also another tree we can say. So, this tree is basically the idea is that in this tree, all nodes are connected.

But it is connected in a acyclic way this graph actually here in this graph, all nodes are connected, but they are connected in cyclic way. Cyclic way means from one node we can come back to the same node. But in the acyclic means form a node we will be able to reach to a node but cannot come back. So, these two examples that we have given here these are the two special tree and they are called spanning tree.

Now, the concept of spanning tree come that way is that if we can have a connection from one node to any other node, if all nodes are connected in a graph, then it is called the spanning tree. Now, the question is that how the spanning tree can be obtained? Definitely DFS traversals and BFS traversals can be applied and then we can find its spanning tree. And you can note that spanning tree problem exist for the connected graph. If it is the disconnected, then there is no question of spanning tree, finding the spanning tree is there.

(Refer Slide Time: 26:38)

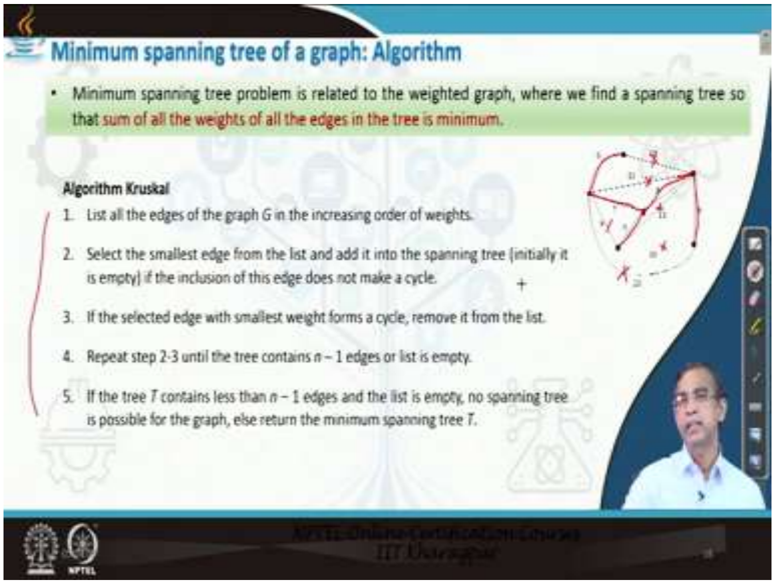


Now there is another specialized problem related to the spanning tree. If the edges are with weight and then there are basically more than one, for example, in the previous slides we can have two spanning tree DFS, BFS.

In fact, there are many more spanning trees possible. If we do not follow any particular order DFS, BFS sometime DFS, some nodes. Then again BFS another nodes and so mixing way then you can find many spanning trees are there. But out of all those spanning trees, there are some spanning trees where the sum of edges of all, sum of all the edges, sum of all weights of all edges are with minimum value then we can say that it is a spanning tree with minimum weights.

So, if a spanning tree with minimum sum of weights then it is called the minimum spanning tree. So, that is why the minimum spanning tree is defined as a sum of all the weights of all the edges in the tree is minimum then we can. Now here for an example, you can check again. So, this is the graph again, weighted graph we can say. But it is undirected weighted graph and this is the one example of spanning tree that means it connects all nodes are acyclic connected, but this connects some of weights is basically minimum. Now, so problem is that if a graph is given, then how you can find the spanning tree with minimum weight. So, this problem has been solved by many algorithms.

(Refer Slide Time: 28:33)



And two popular algorithms in this context is very popular it is called the Kruskal algorithm. And there is another algorithm is also called the Prims algorithm. Now, I will discuss only Kruskal algorithm, Prims algorithms so you can find in any book. And that algorithm is very simple. And it is also called like the greedy algorithm because it solve the problem in greedy manner. So, greedy manner is like this. So, first you have to sort all the edges according to their weights in ascending order.

So here, for example, in this input, all the edges are there, if we sort all the edges so these are the sorting of edges will obtain. In the previous examples if we sort it. So v2 to v3 and then v2 to v5, v3, v6. So, all the edges will be sorted there. Now so, basically the first step is that you sort all edges in ascending order, according to their weights. Then second thing is that you choose that edge, which has the lowest weight, and then add this weight into the spanning tree initially empty.

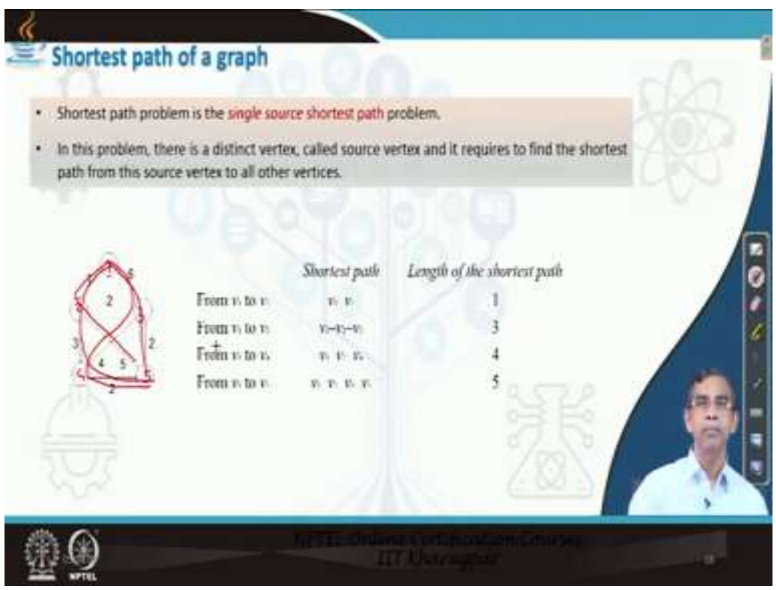
So, for example, in this example say suppose these are the different weights are there if we choose the smallest weight. So, this is basically there so we can connect the path here. So, this is here. Then next smallest weight is basically this one we can connect it here. Then next smallest weight is there. Now, we can add weight if we can see that there is no cycle because a tree should not have a cycle, then after that they are again, we can… because it will not add any cycles. Then here also we can, because it is there, okay.

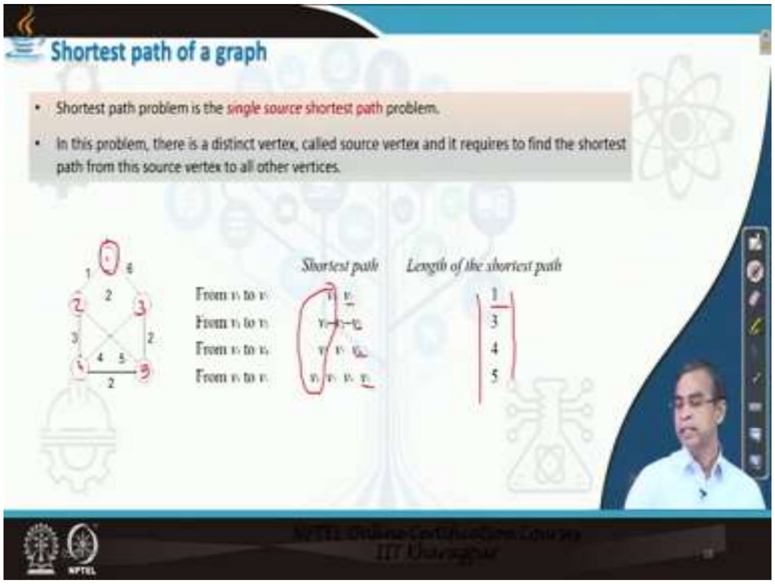
But 9 we cannot, next is 9. We cannot do because 9 if we add it add cycle so it cannot be added. So, it cannot. Then we can take weather this 10 can be here. It cannot because it so it cannot then this also 10 cannot be. Then remaining, weights is this one either this one or this one, but we cannot take this one because it adds weights, but we can take this one, it also we cannot because it is also is. Now this way and this 21 also we cannot because it adds weights.

So, now we can see this is the tree that we obtain which connects all the nodes. And this is a minimum. Some of the weights are minimum, if you consider any other paths that any other tree that you can construct then compare to this one. However, this algorithm gives you a number of minimum spanning tree error. So, this is the one example of spanning tree that we have enumerated.

There are many mores, if more than two edges are a minimum weights, then there is a ambiguity that which edge you should include. Anyone you can include, but anyone if you include, then it leads to a different spanning tree like this one. So, an another thing is that you can notice that in case of… So, this is basically the idea about you can take any graph and then exercise these algorithms which have been mentioned here, these are the say 5 procedures it is mentioned there. And you can check that how the graph can be, how the minimum spanning tree for a given graph can be obtained.

(Refer Slide Time: 32:25)





Now next, I want to discuss about shortest path of graph. And shortest path of graph is very interesting problem. And it has many applications in many situations. The idea about the shortest path is that starting say any two nodes given and then you have to find a path from that two nodes.

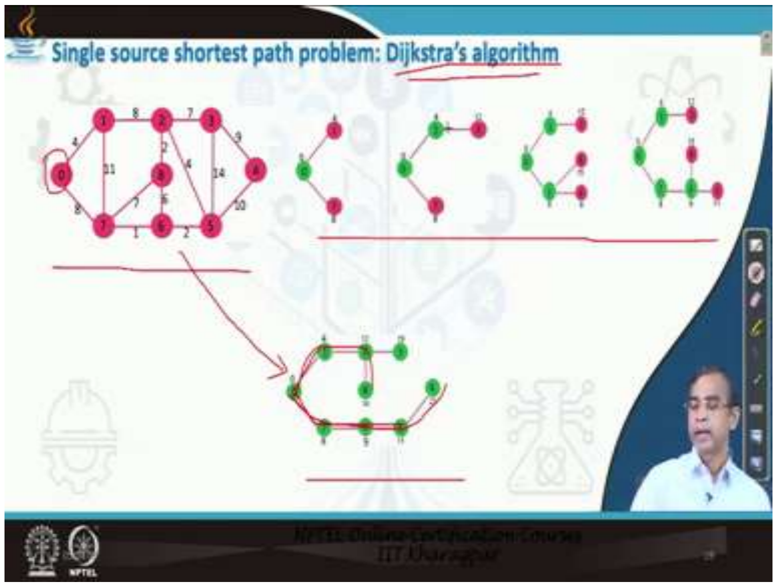
Obviously shortest path problem applicable if the graph is connected. And second is that if the graph weighted. That means all edges are with some positive weights. Now, the shortest path means you can visit. For example, you can visit from this node 1, 2, 3, 4 and 5. So, these are the 5 nodes, you can visit 1 to 5 you can visit many path. So, this path or you can visit this path or you can visit this path or whatever, you can visit this path. So, there are many paths are there.

Now out of many paths which paths having the minimum, what is called the some of the weights. Then it is called the shortest path problem between this one. Now, there are many algorithms related to shortest path between any two pair of vertices or shortest path from any source node to any other destination node or shortest path between any two pair of vertices.

There are many versions of this algorithm or problem are there. So, we will discuss only one problem here. So, given a source vertices how we can find the shortest path from these vertices to any other vertices in the graph. For example, in this graph v1 to v2 this is the shortest path of length one you can check it that this is the shortest path. So, it is 1, 2, 3, 4 and 5.

So, these are the few path (source) these are the single source v1 to any other v2, v3 and v4 and v5 and these are the path lengths we can say that it is possible from v1 to v2 or v1 to v3 or v1 to v4 and v1 to v5. So, this is the one problem the shortest path from a single source namely, v1 here v1 to any other nodes and that path is obtained here. Now, here is a problem is that how this path can we solve this problem? Now this problem can be solved very nicely.

(Refer Slide Time: 35:11)



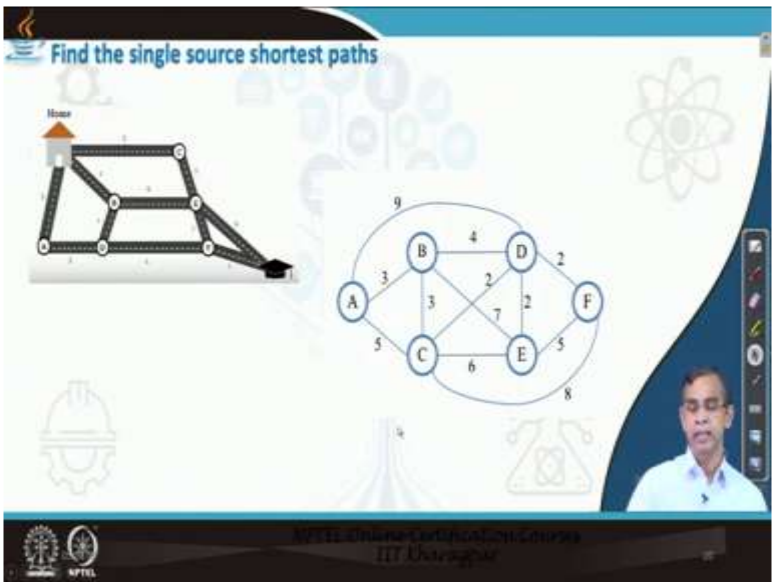
There is an algorithms this algorithm is very famous in the graph theory algorithm. It is called a Dijkstra algorithm. Now, this is the one, Dijkstra algorithm again likes minimum spanning tree problem using Kruskal also a greedy algorithm. So, this is the input graph and suppose this is the source vertices. And we add two enumerate from the source vertices to any other path.

Now, this algorithm, again, you can follow any standard books I have given few steps that how this path can be enumerated. And this is the final solution from this algorithm. It can give graph and this will give you from this path, if you want to go to 1, then this is the shortest path. If you want to go to, say, 8, then the shortest path will be this one. If you want to go to 5, then shortest path is this one. If you want to go to this one, then shortest path is this one.

So, these are the different paths, all are shortest path from a starting node, this is the starting node to any other nodes in the graph. Again, this algorithm is applicable. If the graph is connected and all weights are positive, if the some weights are with negative value, you will not be apply to this problem because this will give you wrong results. So, that is also one limitation of this and the negative weights also does not have any meaning, actually.

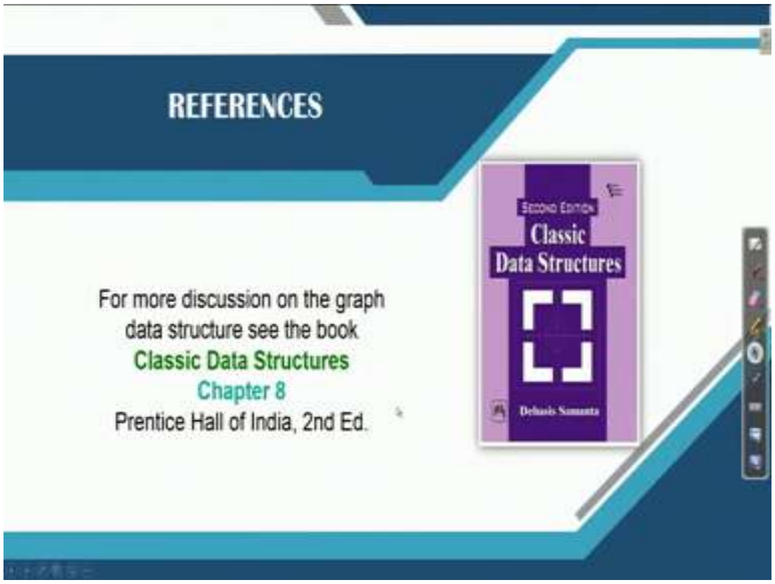
So, in that sense, this algorithm is very standard algorithm for solving this kind of problem. And so this algorithm, you can follow any books to understand about how this problem is solved. And as we are not discussing the details about the algorithm and I advise you to solve the problem and learn algorithm consulting any books.

(Refer Slide Time: 37:06)



So this is the algorithm again, you can apply your learning and then you can solve this problem so that you can understand.

(Refer Slide Time: 37:13)



So, regarding the different other type of shortest path problem and then other many graph related problem thorough discussions you can find in this book, Chapter 8 includes the discussion of graph theory, where you can find many algorithm illustrations and then many other things, which is basically in details about, and particularly for those are interested for computer science study and preparation for the gate examinations and others. So, this chapter you should learn thoroughly this concept you should learn thoroughly okay. Thank you. Thank you very much.