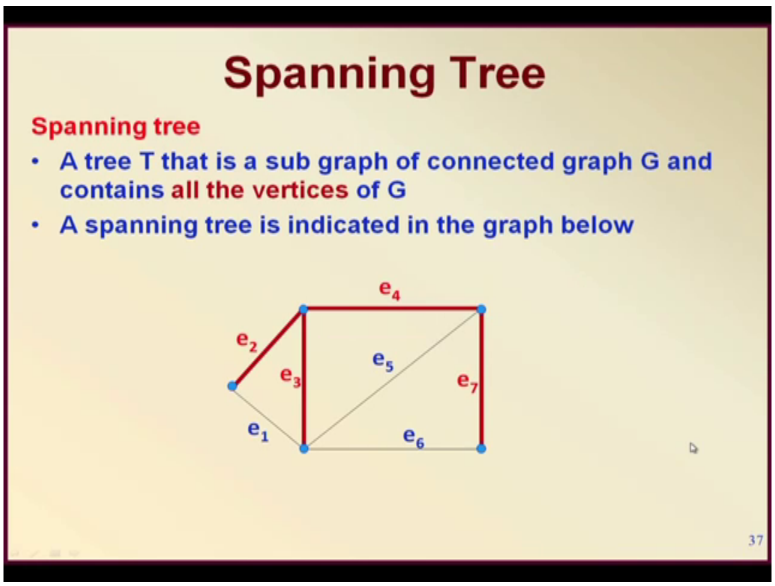
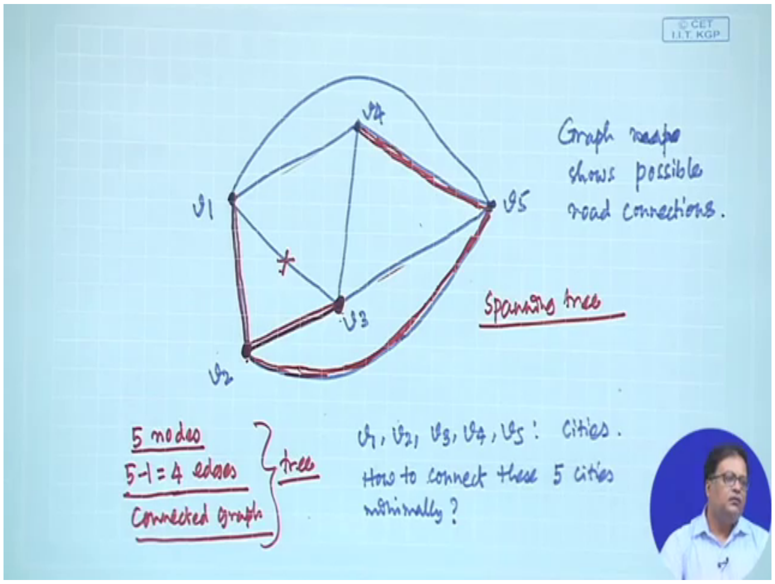
Course on Decision Modeling
By Professor Biswajit Mahanty
Department of Industrial and Systems Engineering
Indian Institute of Technology, Kharagpur
Lecture 34
Module 7
Minimal Spanning Tree

Right, so today we go ahead with our discussion on spanning trees and there after we shall see the how we can find out minimal spanning trees in a given problem, right? And we can actually solve some problems also, right? So let us go ahead and know this particular discussion on the minimal spanning trees.

(Refer Slide Time: 0:45)

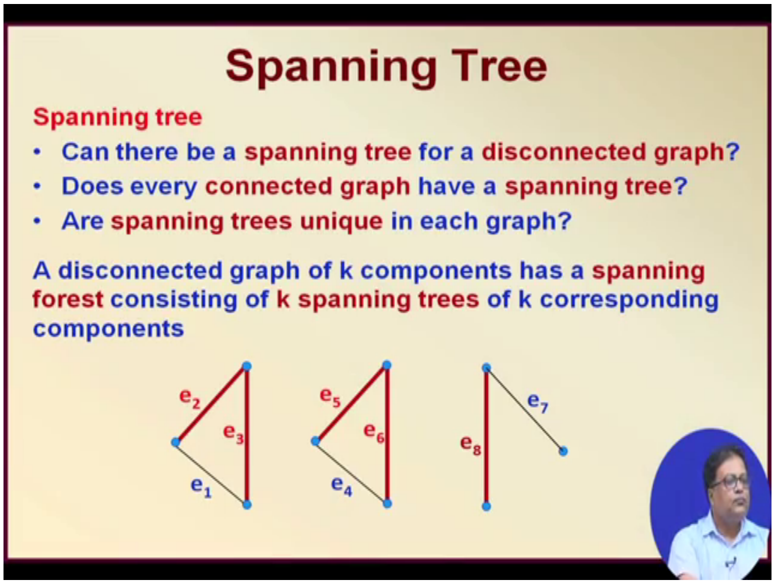




So the minimal spanning tress as you can see the minimal spanning first of all the concept of spanning tree, so what I discussed if you look at this particular diagram which I was showing on previous day that there are 5 vertices and we had to connect minimally to those 5 vertices these are possible road connections.

The question is that how to make a minimal connection in this particular road network on which road network to build. So obviously all the 5 cities should be connected, so if I draw a tree out of these 5 cities then that particular tree is called a spanning tree, right? So we have seen a concept of spanning tree in our previous class now we continue from there.

(Refer Slide Time: 1:39)



You see like we have a spanning tree for a connected graph suppose we start with a disconnected graph, then what happens? You see connected graph has a spanning tree.

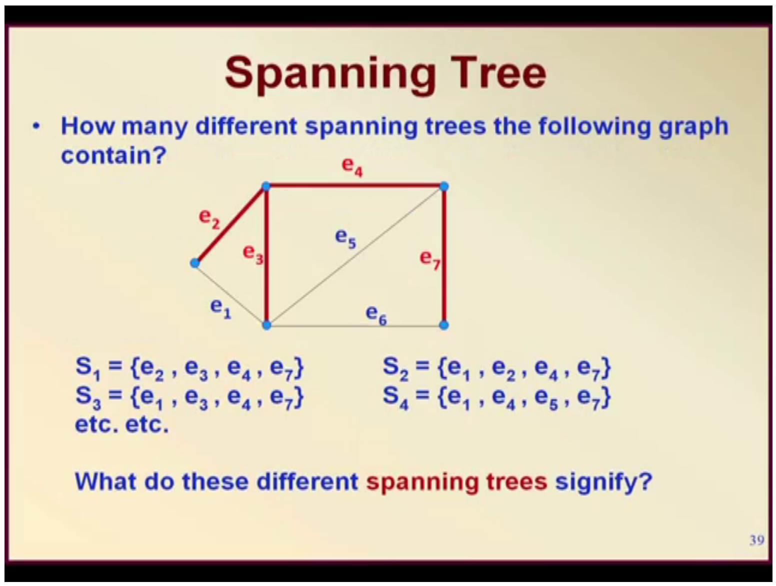
So what a disconnected graph will have? Here is an example of a disconnected graph you see all 3 are together makes a graph G, so they are not connected see from this vertex can you reach here? Answer is no, from this vertex can you reach here? Answer is no. So there are components, how many components let us say k components so here 3 components so k equal to 3. Now if I take this as a connected graph we can draw a spanning tree, similarly if I take this as a connected graph we can also draw another spanning tree and this one is another connected graph that has a spanning tree.

So for the entire graph we do not have a spanning tree we have a what is known as a spanning forest, alright? So now can you answer this questions can there be a spanning tree for a disconnected graph? Answer is no, we cannot have, right? For a disconnected graph has components and that kind of graph will not have a spanning tree but it will have a spanning forest and each component will have a spanning tree that is what happens in a disconnected graph.

The second question we ask here does every connected graph have a spanning tree? If a graph is connected can there be a spanning tree? You see in the worst case the graph could be a tree itself. So if the graph is a tree, then the entire graph is a spanning tree of itself (())(3:45) because that graph itself is a minimal connected so a tree can also have a spanning tree but that is nothing but the graph itself so the graph itself will be the spanning tree.

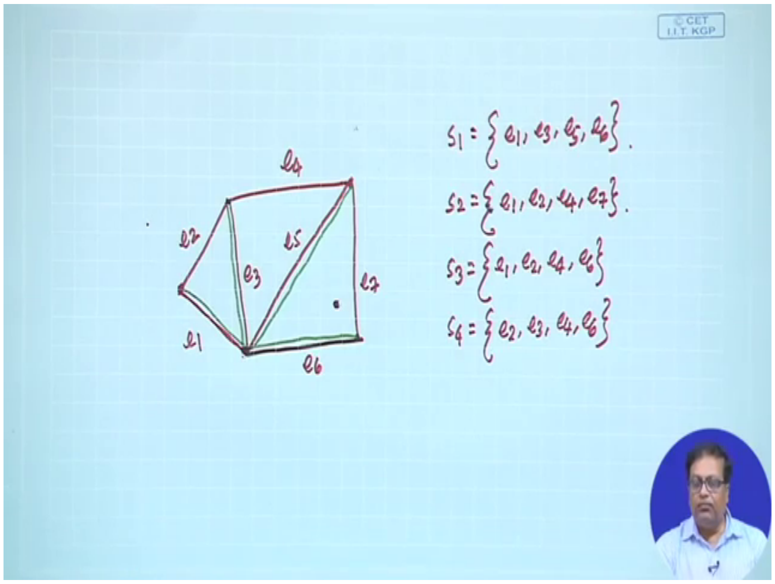
So then our spanning trees unique in each graph? That is also very interesting question that in a graph will there be a only one spanning tree or there will be multiple spanning trees?

(Refer Slide Time: 4:14)



Let us understand this question, so how many different spanning trees the following graph can contain, right? So look at this graph and try to write down all the possible spanning tree that you can think of. One spanning tree is shown that is e2, e3, e4, e7 let us draw it and let us see how many different spanning trees one can find out.

(Refer Slide Time: 4:40)

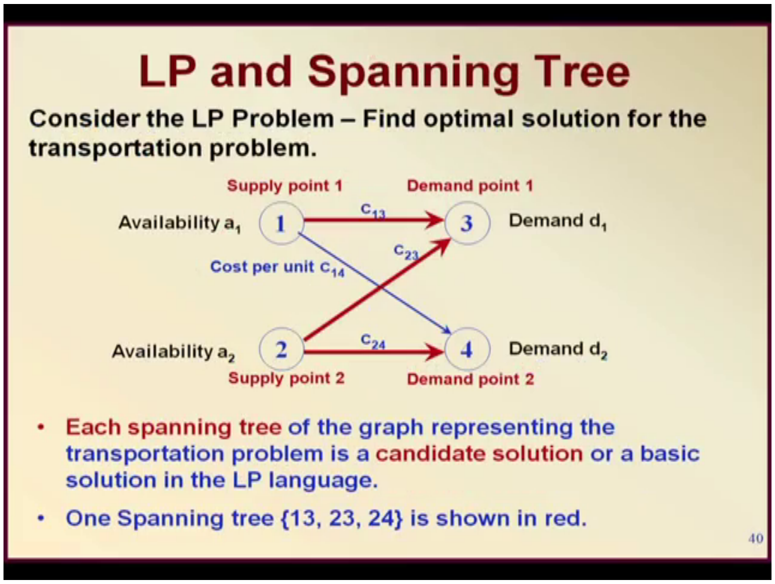


So supposing we have this, then we have this, then we have this, then we have this and then have this, then we have this and then we have this. So this is our graph this graph is e1, e2, e3, e4, e5, e6 and e7 so this is our graph. Now question is how many different spanning trees can you think of, right? So one spanning tree you can think of is say e1, e3, e5 and e6, right? This is also a spanning tree so there are 1, 2, 3, 4, 5 vertices so there will be 4 edges that should be a spanning tree.

So you see one spanning tree s1 could be 4 edges it is a set of edges so e1, e3, e5, e6. Now, can you think of another spanning tree s2 as e1, e2 if you take e1, e2 then you cannot take e3 because that will make a circuit, right? e1, e2, e4, e7, can you take another spanning tree as e1, e2 then e4 and then e6 e1, e2, e4 and e6, right? So you know you can think of so many of them why not e2, e3, e4 and e6, right? So you can have like this several spanning trees but what do they signify.

You see what is spanning tree? A spanning tree connects so if this is a 4 you know 1, 2, 3, 4, 5 cities and these are the possible road connections the spanning signifies the minimal connection. So these are the choices (())(7:29) or these are the potential solutions if the problem is that how do I minimally connect 5 cities so that you know optimally we find what is the possible connection of these 5 cities that will optimally connect these 5 cities minimally. Then our candidate solutions are the spanning trees and one of the spanning tree will be our answer, is or not? That is what they signify.

(Refer Slide Time: 8:07)



So you see this question is directly related to the problem that we call sometimes the linear programming problem and this linear programming problems let us take a very simple problem say transportation problem. What is a transportation problem? In a transportation problem there are some supply nodes like 1 and 2 you can see there are 2 supply nodes, there are some demand nodes or demand points so 3 and 4 are demand points and at each supply point there is some availability a1 and a2 and each demand there is d1 and d2 they are the demands.

So possible paths you see these are like possible paths 1, 2, 3, 4 these are the 4 possible paths through which supply can be made and let us say the cost per unit is c13, c14, c23 and c24. So question is how much should I supply through 1, 3 through 1, 4 through 2, 3 or through 2, 4 if you have solve linear programming problems usually what we do you know we try to find a basic fusible solution and invariably the basic fusible solutions that we find we do not cover all the possible options that we have.

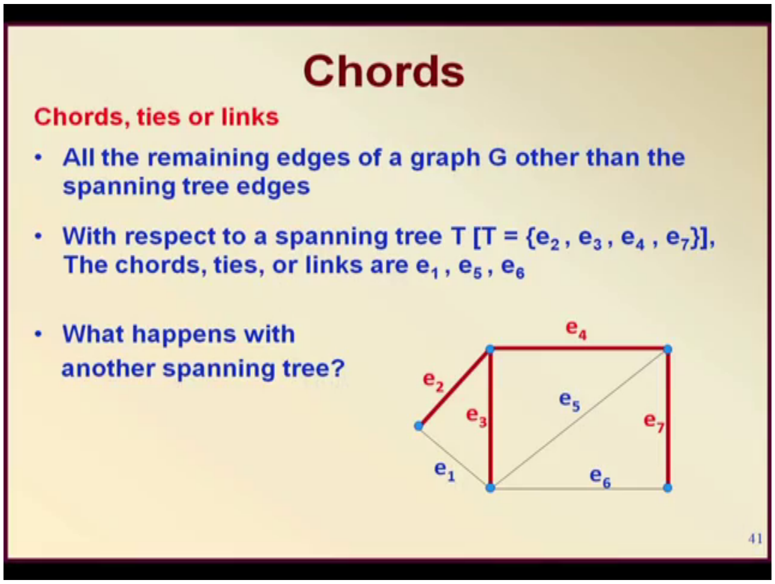
It could be found that out of all the choices only a few choices or some of the choices will be required the remaining choices would not be required, how many such choices? You will be very interest very very much you know this you know amazed to find that the linear programming process essentially puts the excess variable to zeros, right? So by putting excess variables to zeros if you solve a set of if there are m variables and there are n equations where m is greater than n and you put the remaining variables to zeros and additionally we have an overwhelming condition that is total availability equal to total demand, then how many will you able to solve?

You will be able to solve only n minus 1 and if you have n minus solutions n minus 1 solutions out of n what this n minus 1 solutions would actually mean? They mean it is a spanning tree, so basically the solution is nothing but a set of spanning trees, right? Which spanning tree combinations out of these say there are 4 you take all 4 you get a circuit you do not get a tree. So to get a spanning tree you have to choose any 3, which 3? Which is the combination which gives the least cost, right?

That is the spanning tree we have to actually choose. So each spanning tree of the graph representing the transportation problem is a candidate solution or a basic solution in the LP language, right? One spanning tree 1-3, 2-3, 2-4 is shown in red. So basically a linear programming problem is nothing but finding the difference spanning trees evaluating them and see which one is giving us the best possible solution.

Although we are not going to solve linear programming problems here, but it must be remembered that any network problems we can actually solve a linear programming problem by what is known as the network simplex method but today we shall discuss something else, right?

(Refer Slide Time: 12:11)

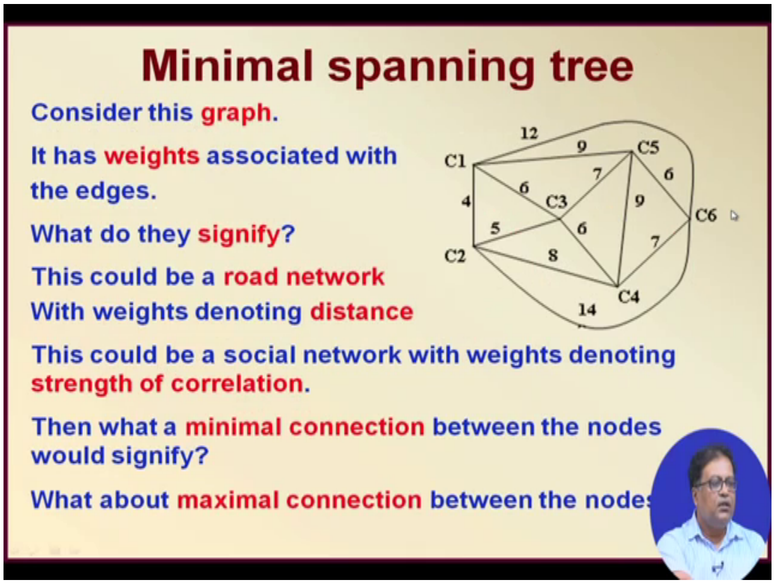


What is that something else? Before that let us go ahead little bit about the chords so all the remaining edges of a graph G other than the spanning tree edges they are called chords, ties or links, right?

So here if e2, e3, e4, e7 is a spanning tree, then e1, e5 and e6 are the 3 chords and look here these chords give exactly one circuit, exactly one circuit with the spanning tree edges, right? So you take e1 then we get a circuit here, we get e5 we get another circuit here, we take e6 we get the another circuit here, right? So this is what and what happens to another spanning tree if you have another spanning tree then we have a different set of circuits and all these circuits are called the fundamental circuits.

So e1 if you take with e2, e3 it gives one circuit that is called a fundamental circuit 1. With e5 it gives another fundamental circuit, with e6 it gives a third circuit which is called also a fundamental circuit. So we get several fundamental circuits in fact the number of such fundamental circuits will be nothing but the number of chords we will discuss more about these later I will skip this for the time being and we directly come to what is known as how to compute a minimal spanning tree, look at this particular problem.

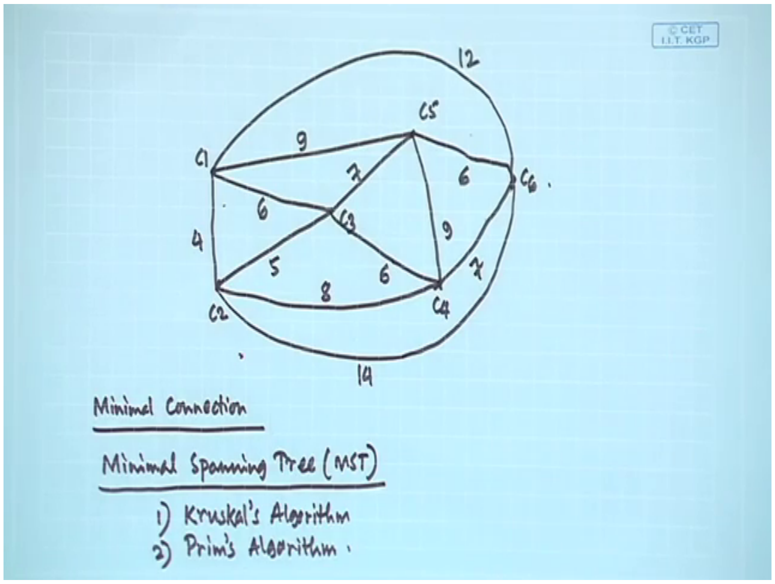
(Refer Slide Time: 13:52)



So consider this graph it has weights associated with edges what do they signify? Supposing there are 6 cities and these 6 cities are connected in a certain manner and these are the possible road connections, right? And there is a figure given the weight given what is that weight? That weight is nothing but the distance or the length of the possible road connection, right? Now if we have to minimally connect them, then how do we go about? That is a essential question, right?

So what is our essential question? The essential question is how do we go about minimally connect these 6 cities c1, c2, c3, c4, c5 and c6 so that we get a minimal connection between these particular edges. So if you want to solve these problem it may take a little time but you know that is an important time that we really put let us draw this graph here, right?

(Refer Slide Time: 15:18)

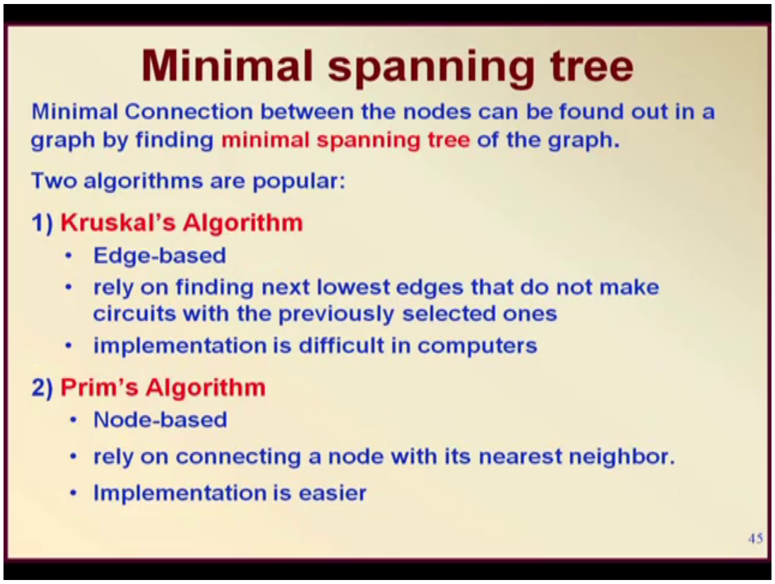


So this is c1, this is c2, then this is c3, this is c5, this is c4, this is c6 so these are the 6 this thing, now these are the connections this is 9, this is 4, then c2 to c4, c2 to c3 that is 5, this is 6, this is 7, this one is 6, this one is 8 and this large one it is 14, then c5 to c6 it is 6, c5 to c4 9, c4 to c6 it is 7 and finally c1 to c6 it is 12, right? So we have drawn this particular graph hope everything is covered, right? So once we have drawn this particular graph, now the question is how do I go about to find minimal connection.

So basically this one is nothing but to find out a minimal spanning tree or in other words an MST. So how to I find out a minimal spanning tree? There are 2 methods, what are the 2 methods? One is called Kruskal's algorithm and the other one is called Prim's algorithm. So there are 2 methods one is called Kruskal's algorithm and the other is called Prim's algorithm by which we can find out the minimal spanning tree.

Now why do we need minimal spanning tree? Because these are the possible road connections between these 6 cities and we have to find out which are the connections that we shall go for so that you know the minimally these 6 cities can be connected. This problem is finding nothing but the spanning tree which is having the minimum total weight and that is called a minimal spanning tree.

(Refer Slide Time: 18:36)

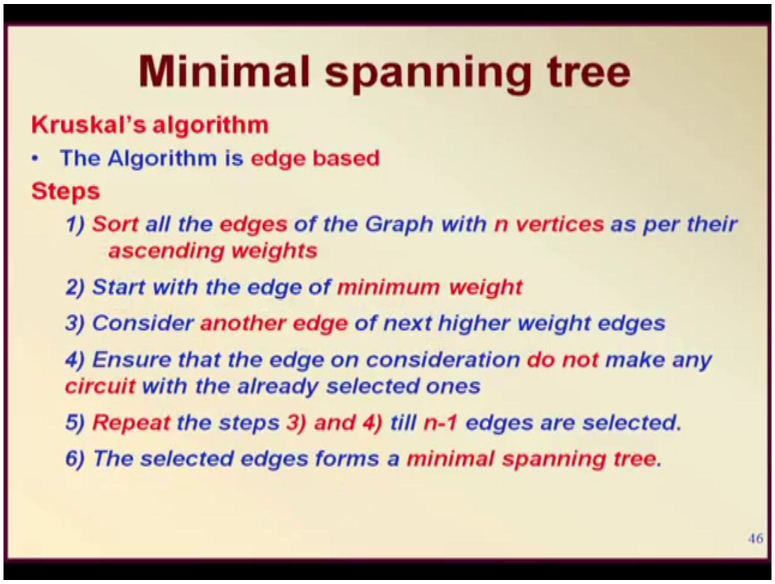


So you see this is a slide that I was talking of just now that minimal connection between the nodes can be found out in a graph by finding minimal spanning tree of the graph and there are 2 algorithms Kruskal's and Prim's.

The Kruskal's algorithm is an edge based algorithm it realize on finding next lowest edge edges that do not make circuits with previously selected ones and implementation is difficult is computers. Prim's algorithm on the other hand is a node based method and it realize on connecting a node with its nearest neighbor and implementation is easier. So there are 2 methods one is the Kruskal's which is edge based but idea is to finding that no circuit is formed and the other one is node based there the advantage is you do not have to really find whether it is making a circuit or not.

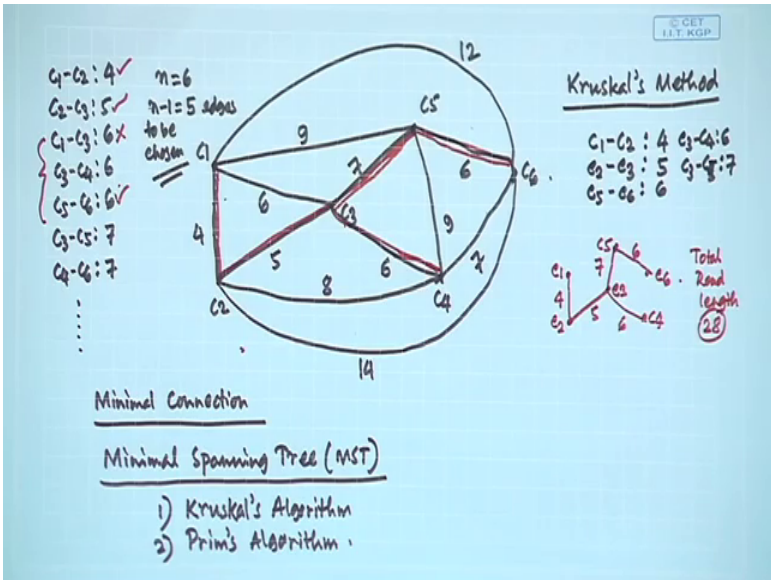
Please remember when you do computer implementation if you have to find whether it makes a circuit or not that is rather difficult process, right? So it is more involved in a computer while we are doing manually we can just look at and we see that it is forming but if you want to implement this on a computer you must have algorithms that can actually read it through a matrix, right? So those matrix things I am not talking right now but we shall give some hints later on.

(Refer Slide Time: 20:19)



So what exactly we do here is in the Kruskal's algorithm you can see the method we are sorting all the edges of the graphs with n vertices as per there ascending weights and then start with the edge of minimum weight. Consider another edge of next higher weight edges and ensure that the edge on consideration do not make any circuit with the already selected ones that is very important. you see it is to be not with the last selected edge that is the mistake people make, the idea here is you have already selected a few edges see that the next stage do not make circuit with the already selected ones and continue till how many edges n minus 1 edges are selected, right? When n minus 1 edges are selected you get a minimal spanning tree. So now let us look at how Kruskal's method can be applied for this particular problem, right?

(Refer Slide Time: 21:22)



So this is the problem that we are solving that and we are trying to use what is known as the Kruskal's method. So if you sort all the edges as per the ascending weight which one will come first? You see c1, c2 because look at this entire graph the minimum possible edge weight is 4 only, right?

So let us sort a few c1-c2: 4, then c2-c3: 5, then c1-c3: 6, then c3-c4: 6, then c5-c6: 6, then c3-c5: 7, then c4-c6: 7 etcetera etcetera, right? So we have chosen some 7 hopefully it is okay because how many we have to choose because here n equal to 6, so n minus 1 equal to 5 so 5 edges to be chosen, right? We have to choose only 5 edges because total number of vertices are 6 and out of 6 you know we can have only 5 to form what is known as the minimal spanning tree.

So what Kruskal's method is telling that first of all select a particular edge say let us say we chose first c1-c2: 4 already chosen so let us draw it here on a particular curve that is so c1-c2: 4 already chosen we can also mention it here that is this is already chosen, right? Next one we look at the list so next one is c2-c3 that is 5 should we choose, does it make any circuit with the already selected one? Answer is no, it does not make any circuit with the already chosen one so we can choose c2-c3: 5 we can choose and we can put it here also c2-c3: 5 so two are chosen only 3 more are to be chosen.

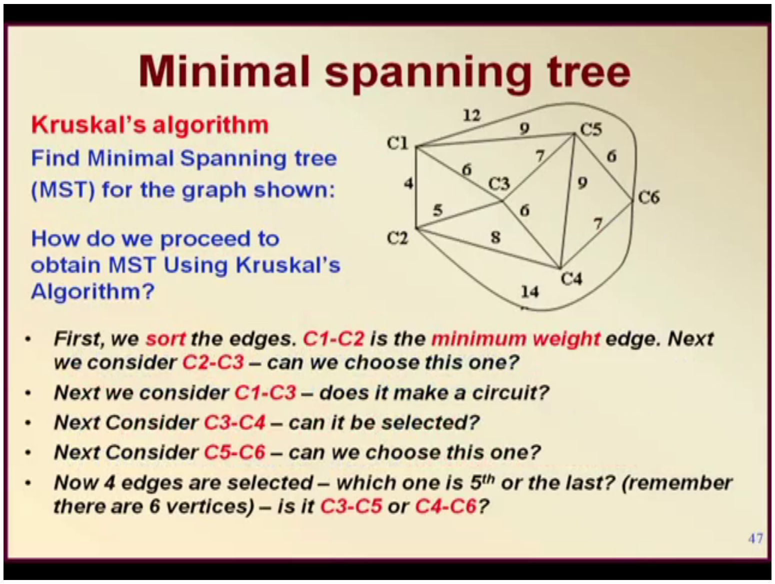
So we have got up to this. Now question is that which should be next? Look at the list now there are 3 choices before us we can choose c1-c3, we can choose c3-c4, we can choose c5-c6, but can we choose c1-c3? Look here if I choose c1-c3 it makes a circuit but Kruskal's algorithm says that a selection should not be a particular edge which is going to make a circuit with the already selected ones. So we have chosen these two so that means this third one cannot be chosen, right? So c1-c3 cannot be chosen so it is out.

So question is we can now choose either c3-c4 or c5-c6 you see I would like to choose c5-c6 because you know you may be thinking that we have to go from one corner and keep on going, so it is not required we can choose c5-c6 also there is no problem, right? So let us choose now c5-c6 so c5-c6 is 6 so we choose it so let us draw in this diagram so this is c5 and c6, right? So c5-c6: 6 is also chosen so we have chosen 3 and there is no circuit.

Now what is the fourth one? The fourth choice should be that again we have chosen this, we have chosen this, we have chosen this, we cannot choose this. So next again it will c3-c4 we can consider now because it is the next one coming so look c3-c4, does it make any circuit with the already chosen one? Answer is no, it does not make any circuit with the already chosen one so we can consider c3-c4: 6. So let us choose c3-c4 also that is 6, so c3-c4 is chosen, so four are chosen now we come to this, can we choose c3-c5 or c4-c6 both are there in fact now since there is a choice you know they do not make circuit so we can choose either we can choose either.

So suppose we choose c3-c5: 7 then we can have this one c3-c5: 7 so if we choose like this then we have this particular graph and look here we have got what is known as a spanning tree, can you see that? We have a spanning tree because we have already got 5 edges and all the 6 cities are connected minimally and what is the total distance? 4 plus 5 9, 15, 22 and 28 so total road length 28, so this is what is known as the Kruskal's method, right?

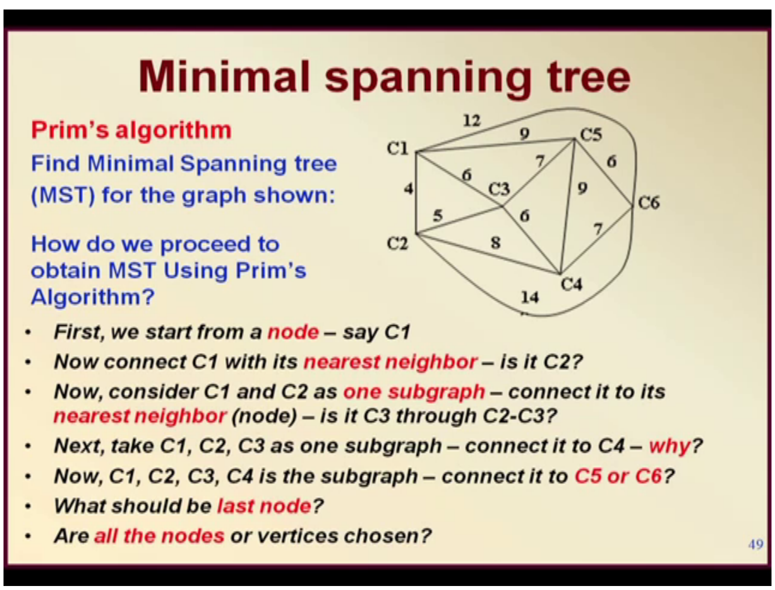
(Refer Slide Time: 28:56)

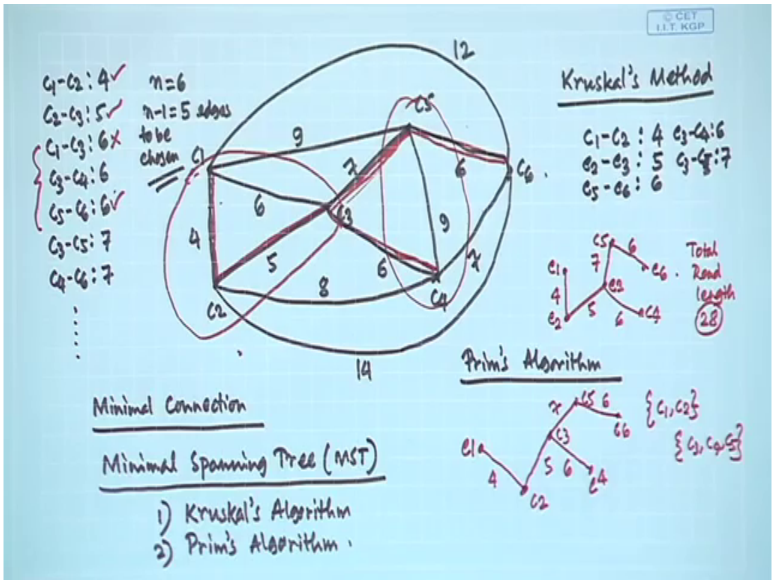


So in Kruskal's method 28 and if you really look at here, then you can see that depending on whether you choose c3-c5 or c4-c6 we get the two alternate solutions, right?

So that is a Kruskal's method for solving minimal spanning tree problems.

(Refer Slide Time: 29:16)





Now look at the second method which is more popular for computer based systems that is called a Prim's algorithm. In Prim's algorithm it is a node based method so what we do we first start with a node let us say c1, so now in c1 we have to see what it is nearest neighbor, so let us directly look here let us do here in this case (())(29:40) so this is the Prim's algorithm.

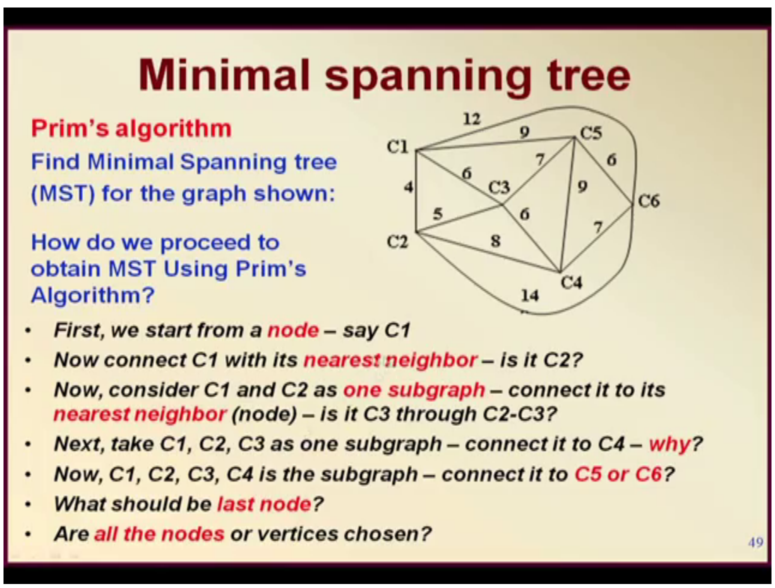
So like Kruskal's algorithm let us look at what is known as the Prim's algorithm also. So what happens in Prim's algorithm let us start with a particular node c1, right? So here the process is slightly different that is we start with a particular node c1 suppose we can start with any other node also but let us say we start with a particular node c1. Now the question is that which one should be its nearest neighbor what is the nearest city to c1? You see choice is c2 4, c3 6, c5 9, c6 12 direct edge connections there 4 edge connection, which one is the lowest? The lowest is c2.

So that means we have to connect c1 to c2 because that is the lowest edge connection nearest neighbor. So now consider c1 and c2 as 1 sub graph and what is the nearest neighbor to these 2 together? You see what has happened we have now you know differentiated or divided the entire space into 2, one is c1-c2 another is c3-c4-c5 so you see you have to now therefore look at all possible connections between c1-c3, c1-c4, c1-c5 c2-c3, c2-c4, c2-c5 and look at these 2 connections and see which one is the lowest. So if you see c1-c2 from c2 the lowest is 5 because from c2 you can go to c1 is already taken c3, c4 and c6 5 is lowest.

From c1 6, 9 and 12 so ultimately lowest is 5. So at this point we connect c3 and we get 5, now take the entire thing c1, c2, c3 all together and look at what its nearest neighbor. So one side is c1, c2, c3 these entire portion what is the nearest neighbor out of these c4, c5 at this point, right? So there is which one so you see from c1 12, 9, from c2 8, 14 and from c3 7 and 6. So we get c4 which is 6 so there is a fourth connection now if you to connect c5 there are two choices, right? So we have connected c1, c2 and c3 to c4 then we have this either this 7 or see this 7 because now c1, c2, c3, c4 this is what we have now we have to see that the lowest connection that we have is one of these two sevens, right?

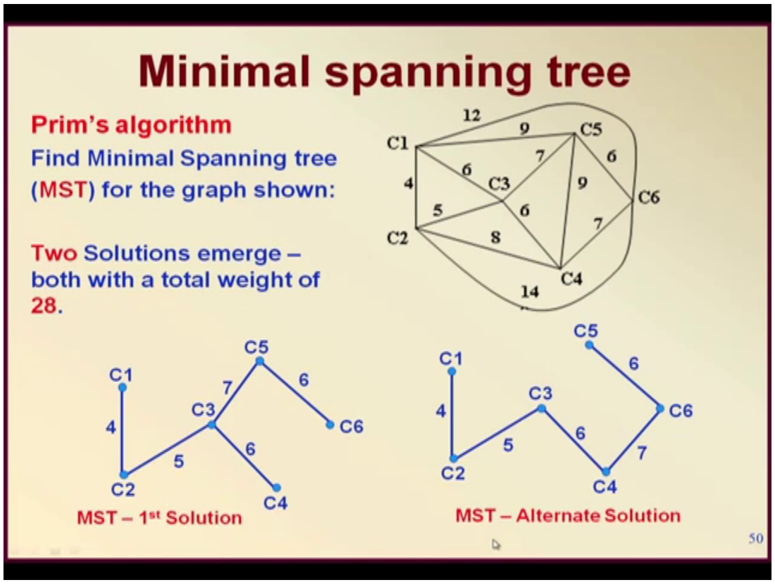
So maybe anyone you can take so maybe we take this one and we go to c5 that is 7. So if I know take these entire c1, c2, c3, c4, c5 the nearest connection to this is c6 and then we get 6. So you see look we have got the same graph then earlier the only difference is that approach is different the way we calculate is also very different.

(Refer Slide Time: 33:43)



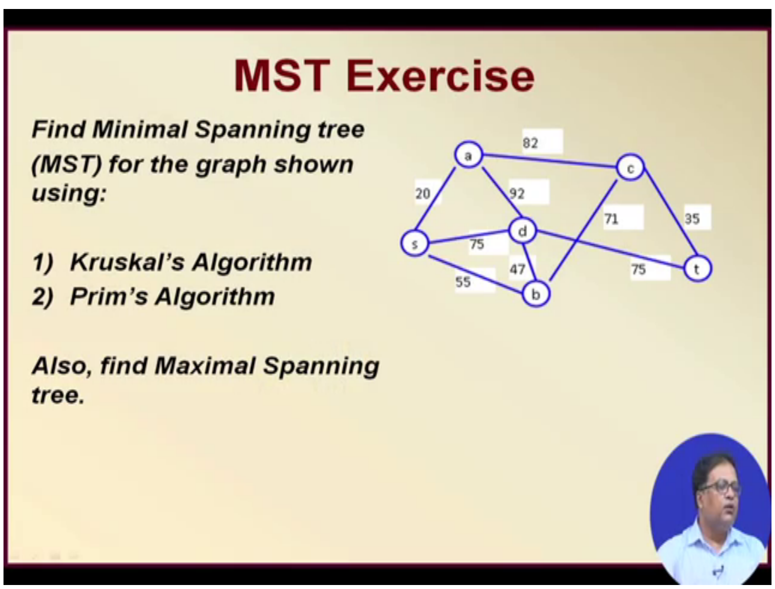
So look at this slide so start at node say c1 connect its nearest neighbor c2 consider c1, c2 as one sub graph connected to its nearest neighbor, is it c3? Yes then take c1, c2, c3 as a sub graph connected to c4, why? Because for this c4 is the nearest neighbor then take c1, c2, c3, c4 then should we take c5 or c6? Answer is both because both are same distance, right? Where should be the last node either? Either c5 or c6 depending on that, so all vertices chosen? Yes, stop.

(Refer Slide Time: 34:20)



So these are the same answers we get by the Prim's algorithm also, is it alright? So this is how we actually can calculate and you can try out different other problems.

(Refer Slide Time: 34:34)



For example you can try this problem you know where we have 1, 2, 3, 4, 6 nodes and there are so many connections so can you find what is known as the minimal spanning tree for this and additionally can you also find maximal spanning tree, right? So anyway I leave it here maybe we this question you take it up later, so thank you very much.