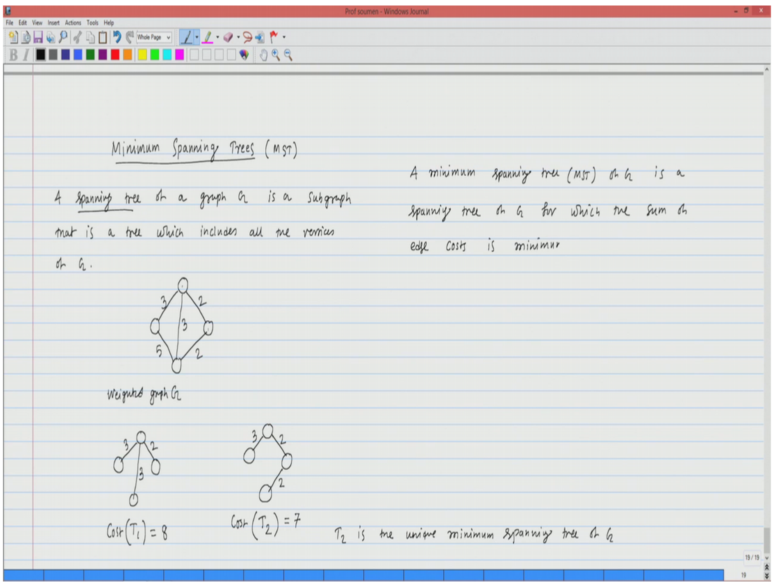
Graph Theory
Prof. Soumen Maity
Department of Mathematics
Indian Institute of Science Education and Research, Pune

Lecture - 05
Part 2
Minimum Spanning Tree

Welcome to the second part of lecture 5 on Graph Theory. So now we will learn; what is the minimum spanning tree of a given weighted graph? First we learn; what is weighted graph that every edge is given some weight that weight would be the distance between these 2 vertices, or it could be the cost of moving from one vertex to another vertex. And then we learnt; what is a spanning tree of a graph, and we will try to find there could be several spanning trees of a graph of a given weighted graph and we will find the spanning tree which is having the minimum cost.

So, let me just formally introduce the things; so minimum, minimum spanning trees.

(Refer Slide Time: 01:17)



So, this is with respect to weighted graph. First we learn what is spanning tree. A spanning tree of a graph G is a sub graph, sub graph that is that is a tree which includes all the vertices of G. So, a spanning tree of a graph is a sub graph that is a tree which includes all the vertices of G.

Let me just give an example to illustrate this thing, consider this graph. So, this is so now, this is not a simple this is simple graph, but it is a weighted graph. So, this edge weight means every edge is having some weight. So, this edge has weight 2; that means, you can interpret that weight in different manners like the weight might represent the distance between these 2 vertices. It could also represent that what is the cost of moving from here to here and many more.

So, this edge has weight 2, this has edge has weight 2. This edge has weight 3, this edge has weight 3 again, and this edge has weight 5. So, this is this is called weighted graph. Now this graph has several spanning trees. So, this is one spanning tree well. So, this is a spanning tree because it is a let me call it T 1. So, T 1 is a spanning tree of the graph G, because it is a sub graph of this graph G, it is a tree and it includes all the vertices of the graph. So, this graph has 4 vertices and this tree is also having 4 vertices. And there could be several spanning trees of T 2, this is another spanning tree of the graph G. You can see that this is also a sub graph of the graph G, and it is a tree and it includes all the vertices of the graph G that is why this is this is a spanning tree.

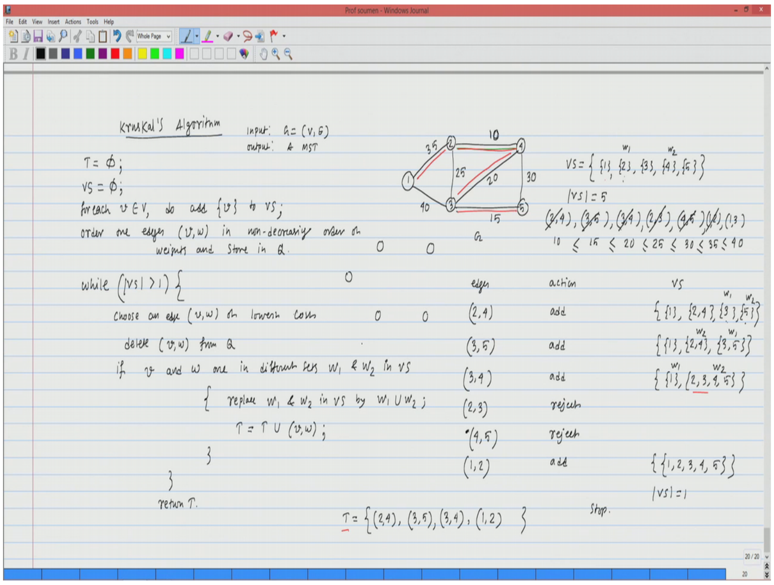
Now, what is the cost of this tree? The cost of this tree this spanning street spanning tree T 1 is just the sum of edge cost. So, this edge has cost 3 or weight 3, this edge has weight 2 this has edge has weight 3. So, the cost of T 1 is 3 plus 3 plus 2 that is 8. And the cost of this tree this spanning tree is 3 plus 2 plus 2. So, this is the cost of T 2 is equal to 7. And you can this is a small graph. So, you can check that T 2 is the unique minimum spanning tree of G. So that means, you can not find another spanning tree of weight less than 7.

So, let me formally say the definition of minimum spanning tree. This is also denoted by MST, a minimum spanning tree of G is a this spanning tree of G for which the some of edge costs is minimum. So, we understood the definition of minimum spanning tree, just in general let me tell this spanning word is used. So, when I say a spanning cycle or Hamiltonian cycle it is a cycle which includes all the vertices of the graph.

When I say spanning path between 2 vertices u and v that mean it is a path between u and v which includes all the vertices of the graph. Sometimes we also use the word spanning sub graph. Spanning sub graph is a sub graph of a graph G which includes all the vertices of the graph G. So, the other way of understanding spanning tree is that it is a spanning sub graph which spanning sub graph, means it is a it is include all the vertices of the graph and it is a tree if the spanning sub graph is a tree then it is call spanning trees. So, over all the spanning means it is it is span or it includes all the vertices of the graph original graph.

Now what we will do is that we will learn a technique to find a minimum spanning tree in a given weighted graph and for that we learned here Kruskal algorithm to find a minimum spanning tree in a given graph, given weighted graph.

(Refer Slide Time: 10:10)



Kruskal algorithm, and I use a an example to illustrate this algorithm also. So, this is the weighted graph that I will be using to illustrate Kruskal algorithm. 2 4 these are the level for the vertices is not quiet, this is 3 5 and vertex 1 is here. And the cost of this edge is 35 cost of the edge 2 4 is 10. The cost of the edge 3 4 is 20. Cost of the edge 2 3 is 25 and cost of 1 3 is 40. Cost of 3 5 is 15. And the cost of 4 5 is 30 ok.

So, this is the graph weighted graph that I will be using to illustrate the algorithm. The basic idea is to you try to get a spanning tree of this graph, such that you are using the edges of lower weights. So, that is the basic idea. So, let me just write down formally the algorithm, it is very intuitively very clear that why this works. So, T is the set of 3 edges; that means, the edges which will be included in the spanning tree. So, initially this the set T is empty T is fee. And then I will be using another set called vs, vs is the set of all vertices basically. So, that also initially empty. And then I construct vs so for each for each v belongs to the vertex set. So, input to this algorithm is input to this algorithm is a is a graph or weighted graph G V E and output of this algorithm is a minimum spanning tree.

Sometimes the minimum spanning tree is not unique; if there are 2 edges or 2 or more edges with the same weight in such cases. For each vertex v in the vertex set capital V do add a single ton set v to vs. So, vs is a set of initially. So, what I will illustrate parallely here. So, vs for this given graph is consists of this singleton sets 1 2, 2 is corresponds to second vertex 3 4 5 will.

Now what we do is that we order the edges say v w in non-decreasing, in non decreasing order of weights and store somewhere. Store in Q generally this is store in the priority Q, but this is a data structure which is which I am not assuming that you know data structure. So, you store them somewhere. So, so you arrange the edges in non decreasing order of weights; that means, this is what I want to say. So, the edge which is having the lowest weight that is 2 4, 2 4 is the edge which is having that weight 10 and this is the minimum. So, this will come first and then the edge 3 5, 3 5 has weight 15. So, 15 is less than sorry 10 is less than or equal to 15.

The next edge is 3 4 has weight 20 the next is 2 3, 2 3 has weight 25. Next edge is 4 5 which has weight 30, and then we have 1 2 which has weight 35 of and then we have 1 3 which has weight 40. So, all the edges will be stored in this order. And then you are sort of ready with the basic setups, and then you run this while loop while the cardinality of vs is greater than 1. So, at this moment my cardinality of vs is equal to 5, because there are 5 sets. So, it will go inside the loop what we will do what it will do is that it will choose it will choose an edge v w of lowest cost; that means, it will first choose the edge 24 which is having the lowest cost 10. And then what we will do is that it will delete that edge from Q. So, these are these are the edges which are stored in Q. So, you delete v w from Q.

So, after the first iteration the edge 2 4 will be deleted from the Q. Now what will do here is that if v and w are in different sets, w 1 and w 2. So, that is w 1 w 2 corresponds to this sets here 1 2 3 4 5 are in different sets in vs. Then you do the following, what you do is that you replace w 1 and w 2 in vs by w 1 union w 2 and you update your T. If this is true I will just explain this one T union. So, T is a set of 3 has basically all the edges which will be considered in the spanning tree or minimum spanning tree. So, you include this edge in T that v w is included. And that is all and at the end you return T. So, once you are outside this loop you return the T which is the set of tree edges ok.

Now let me explain this one. So, here the edges the first I will consider the edge 2 4, because I am inside this loop here because my carnality of vs is 5 which is greater than 1. So, I choose an edge v w of lowest cost that is 24, sorry 2 comma 4 for me. And then I will delete 2 4 from the from Q and then I will check whether 2 and 4 are in 2 different sets s 2 is here this is my w 1 at this moment. And 4 is in a different set. So, this is that in 2 different sets w 1 and w 2.

Since this is true what I will do is that I will revise my vs, I revise my v s. so, this 2 sets will be replaced by it is union. So, my new vs set is now consists of 1 2 4 3 and 5. I hope that this is clear, and the action here is I write is that you add this edge in T. So, add means I am updating my T. So, the T initially it was null. So, add means I include 2 4 in my 3. And I will include more edges slowly and again you see that now the cardinality of vs is 4 which is greater than 1.

So, I choose an edge which is of lower cost. So, I will choose cut 3 5, and I delete it from the Q. Now I am dealing with the edge 3 5 see I am always trying to use an edge of lower cost first. I have used 2 4 and that is included in the 3 next time using 3 5 and I will see whether 3 5 can be can also be included in the spanning tree. So, 3 5 now 3 is in one set and 5 is in a different set. So, they are in 2 different sets. So, this is true. So, I will add this edge in the in T and what I have to make some update in vs. So, this is my v one sorry this is my w 1 and this is my w 2 at this moment.

So now my new v will be 1 2 4 3 5 I will take their union to update my vs. so, the next edge. So, the cardinality of vs is 3 which is greater than 1. So, I have to take on the next edge 3 4 and I will delete it from the Q. So, 3 4 is. So, 3 4 3 is here and 4 is here. So, this is my w 1 now this is my w 2. So, they are in 2 different sets. So, I will add this in my tree set of 3 edges. So, 3 4 edge will be added. And what I will have to do is that I have to update my vs also. So, the new vs will be one and union of w 1 w 2 that is 2 3 4 5. Well, still the cardinality of vs is 2. So I can go inside this while loop vs is still the cardinality of vs is 2 which is greater than 1.

So, I will choose an edge of lower cost 2 3 2 3 and now I will check whether 2 and 3 are in 2 different sets that is not true there in the same set. So, I will reject this one I will reject this edge. In fact, you know you can see the other way of thinking is that 2 3 is creating a cycle with the existing edges that is why we reject 2 3 let me just tell in that respect also. So, till now I have 2 4 in my 3. So, 2 4 is in the I use that color. So, 2 4 is there in my preset. And then 3 5 3 5 is in and it is in the in T. And 3 4 is also a 3 edge 3 4 is 3 edge.

Now, if I include 2 3 also that will create a cycle, but I am looking for a spanning tree. So, that is why we reject 2 3, this is another way of thing which is same as 2 3 is in the same set; that means, if we add to 3 there will be a cycle in the existing tree. So, that is why we reject this one. And next still the cardinality of vs. so, will not update vs also the cardinality of vs is still 2. So, we go for the next edge then and then that is 4 5. And again I see that 4 and 5 both are in the same set which is equivalent to say that 4 5 is creating a cycle in the existing tree. So, we will reject will reject it. So, there is no update on vs the cardinality of vs is still 2. So, we are inside the loop next we consider 1 2 delete 1 2 from Q.

Now one and 2 are in 2 different set this is my w 1 this is my w 2 there in 2 different sets. So, I add them. I will add this edge in my set of 3 edges 1 2 will be considered and here you can see that 1 2 is not creating any cycle which is same as one and 2 are in 2 different sets. So, I will update my vs now my new vs is this one, 1 2 3 4 5. And now I can see that my vs is having only one set. So, the cardinality of vs at this moment is equal to 1.

So, I am sort of outside this I am out of this while loop. So, I have to stop here. So, I will not check with 1 3 anymore. So, I will stop here and I will return this tree. And as you can see that this tree what is the weight of this tree this tree is 1 2 is added at the end. So, this is the tree this is the spanning tree shown by the red lines. And the cost of this tree you can compute and you can check that this is a minimum spanning tree. You cannot find any tree which is which is a of lower cost then the tree given by the Kruskal algorithm.

Just you know there are several ways of thinking the same algorithm. So, instead of thinking them there that vs is a different sets, one can also consider vs as you start with few parties this isolated vertices basically. So, the singleton set here is equivalent to all the vertices are initially isolated vertices.

And then you add 2 4 first that is equivalent to you join these 2 vertices, these 2 isolated vertices. And next is 3 5 which is equivalent to you join these 2 vertices 3 and 5 2 4 and this is one. And then 3 4; that means, you are joining these 2 vertices 3 and 4 and finally 1 2. That means, this is the spanning tree basically and there are different way of representing or thinking this Kruskal's algorithm.

Well, So I hope that you understood Kruskal algorithm and how to find a minimum spanning tree a given a weighted graph.

Thank you very much.