

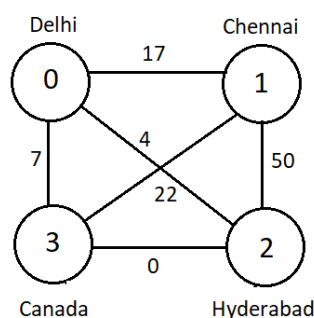
Calculating Spanning tree Cost & Minimum Spanning Tree

codewithharry.com/videos/data-structures-and-algorithms-in-hindi-91

In the last lecture, we introduced to you the concept of a spanning tree. We discussed the conditions required for a graph to be called a spanning tree for another graph and several other terminologies related to graphs. Today, we'll delve deep into spanning trees and see their applications as well as learn what minimum cost spanning trees are.

For easy understanding of things, I'll walk you all through an instance where there is a subject named *Prem*, and he's desperate to meet his beloved who is residing currently at someplace whose location he is not aware of. Although he has good options for places to visit and there are different routes to see all those places. *Prem* is not sort of a rich brat. He would definitely love to save money while he travels through places.

Consider the graph given below and the places are shown in the graph are the ones *Prem* has to visit. He wants to travel through all the places and at the same time for minimum possible expenditure. Each two places are connected through a route and every route would have some travel cost as well.



Prem, therefore, needs to come up with an algorithm that will help him find his beloved by wandering through all possible locations and at the least possible cost. And this is where finding a minimum cost spanning tree helps.

Let's give ourselves a quick revision of a few things:

Connected Graphs:

A connected graph is a graph where there is a path from any point to any other point in the graph. And the graph which is not connected is said to be disconnected.

Complete Graphs:

A complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge.

Spanning trees:

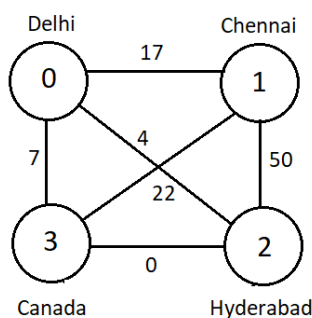
We learned that any graph S which satisfies below mentioned four conditions would be considered a spanning tree for another graph G . Those four conditions were:

1. Graph S is a subgraph of graph G .
2. Graph S is a connected graph.
3. All vertices of graph G are present in graph S .
4. The number of edges in graph S equals the number of vertices in graph G - 1.

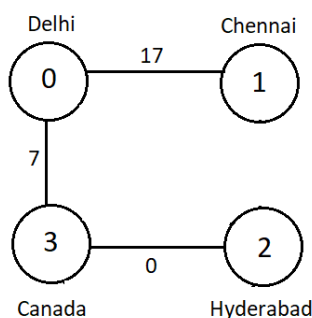
Having done the revision, we'll now look to proceed with what we mean by the cost of a spanning tree.

Cost of a Spanning Tree:

The cost of the spanning tree is the sum of the weights of all the edges in the tree. Now, consider the example of *Prem* we took at the beginning, and the graph he was to create a spanning tree of.

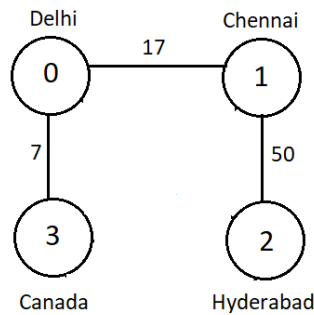


Now, suppose we created a spanning tree out of it which I've illustrated below.



S_1

What would be the cost of this spanning tree? Yes, $17+7+0$ which is, 24. Let this cost for the spanning tree S_1 , be $Cost_1$. Suppose another spanning tree of the same graph be:



S2

This spanning tree has a cost worth $17+7+50$, which is 74, and which is definitely greater than 24 what we calculated earlier for S1. And therefore, *Prem* would definitely not want to incur some cost greater than 24. Let's move ahead now and see what a minimum spanning tree is.

Minimum spanning tree:

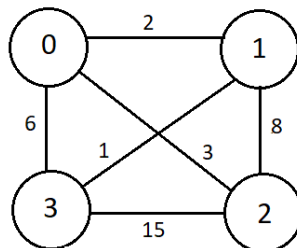
A Minimum Spanning tree, abbreviated as MST, is the spanning tree having the minimum cost. A graph can have a number of spanning trees all having some cost, say S1, S2, S3, S4, and S5 having cost 100, 104, 500, 400, and 10 respectively, but the one incurring the minimum cost is called the minimum spanning tree. Here, S5 having cost 10 would be the minimum spanning tree.

Applications of Minimum Spanning Tree:

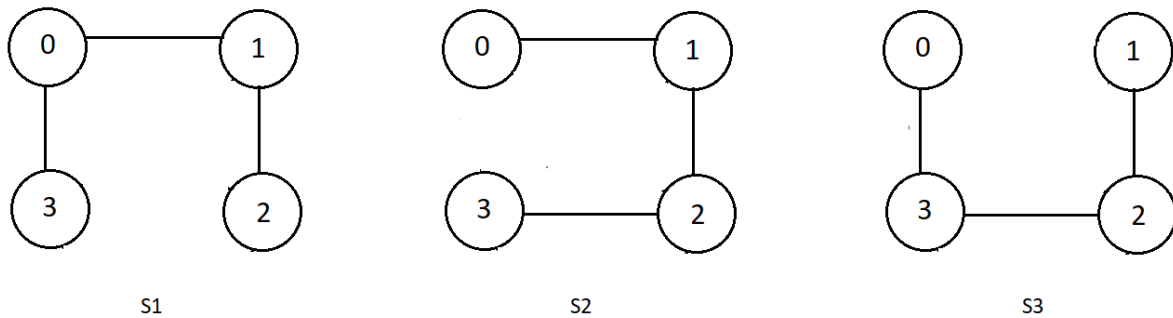
Applications of a minimum spanning tree must have gotten somewhat clear from the example of *Prem*, where we want to traverse all the nodes, which means the graph remains connected but it must have only bare minimum costing edges. So, this was a basic application of a minimum spanning tree. Things will gradually get clearer. Let's take some graphs and do some exercises on them ourselves.

Exercise:

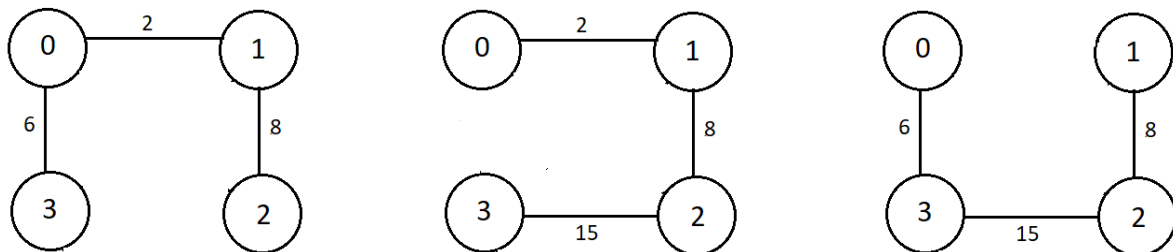
Find the cost of any 3 spanning trees of the graph given below.



Consider the given graph be G , and our first step would be to create three spanning trees S_1 , S_2 , and S_3 of the graph G which we have already learned in the last class.



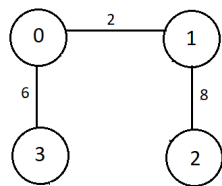
Then, mention weights to all the edges.



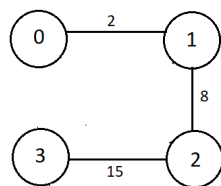
Now, let C_1 , C_2 , and C_3 be the cost of spanning trees S_1 , S_2 , and S_3 respectively. Therefore, C_1 is $(6+2+8) = 16$, C_2 is $(2+8+15) = 25$ and C_3 is $(6+15+8) = 29$. Hence the cost of these spanning trees is 16, 25, 29 respectively. One thing we can infer from this is, C_1 is less than C_2 , and C_2 is less than C_3 , i.e., $C_1 < C_2 < C_3$. As a result, C_1 is preferred to both C_2 and C_3 , and C_2 is preferred to C_3 .

Find the minimum spanning tree of the same graph we considered in the previous exercise.

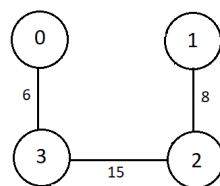
Since graph G is a complete graph and it has a total of four nodes, the total number of spanning trees possible for graph G would be 4 raised to $(4-2)$, which is 42, 16. So, the first step would be to create all the 16 possible spanning trees. All of them are illustrated below along with the edge weights.



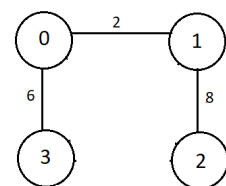
S1



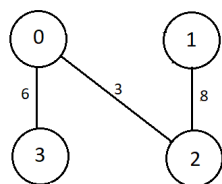
S2



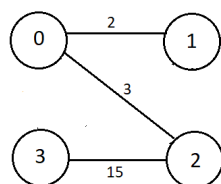
S3



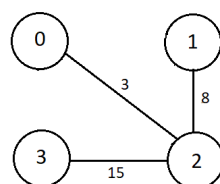
S4



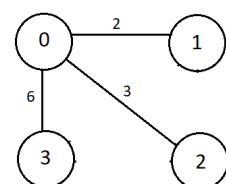
S5



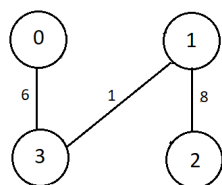
S6



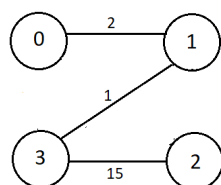
S7



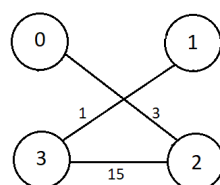
S8



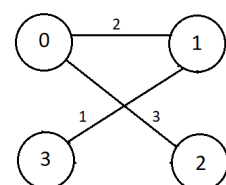
S9



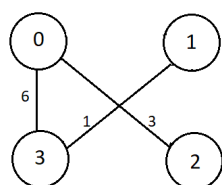
S10



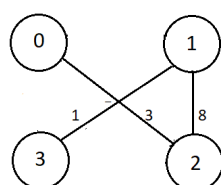
S11



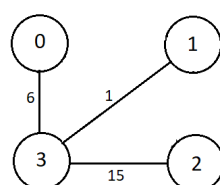
S12



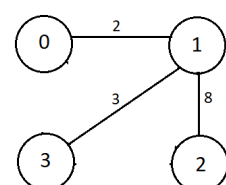
S13



S14



S15



S16

So, these were all the possible spanning trees we had. The rest I'll leave to you. You first have to count the cost of all these spanning trees, and then choose the minimum out of them. Tell me in the comments if you did all of them. I'll be more than happy to know if you could.

That was all for today. I hope things are coming together for you. We'll do more of these lectures. Next, we'll discuss Prim's algorithm, one of the reasons I chose *Prem* as the subject.:) This is to automate the process of finding the minimum spanning trees of graphs. Another algorithm is Kruskal's. Just stay, and you'll get to learn all of it.

Thank you for being with me throughout the session. I hope you enjoyed it. If you appreciate my work, please let your friends know about this channel too. You haven't saved this graph playlist already, you just have to move on to [codewithharry.com](https://www.codewithharry.com) or my

YouTube channel to access them. See you all there in the next lecture where we'll start with the Prim's algorithm. Till then keep learning.