Spanning Trees & maximum no of possible spanning trees for complete graphs

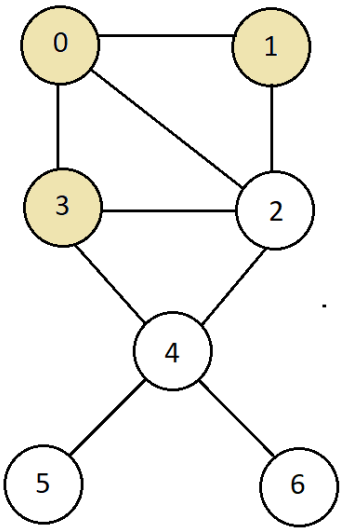
codewithharry.com/videos/data-structures-and-algorithms-in-hindi-90

In the last lecture, we finished learning both the traversal algorithms, the Breadth-First Search and the Depth First Search. We implemented both of these in C using queues and stacks respectively. Today, we'll learn what spanning trees are, and the concepts behind them, and other small topics such as connected graphs and others needed to understand spanning trees.

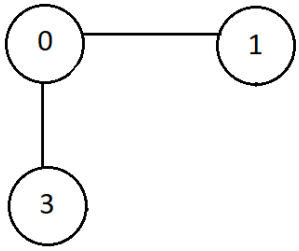
While we'll be learning more advanced topics, you will realize that applications of the algorithms we wrote BFS and DFS are not limited to just finding the traversal order of a graph, rather much more. Let's first start with what spanning trees are. But before we dig into spanning trees, we should talk about a few things you should know. First one being, a subgraph.

Subgraphs:

A subgraph of a graph G is a graph whose vertices and edges are subsets of the original graph G. It means that if we consider the graph G illustrated below, then its subgraphs could be the graph S.

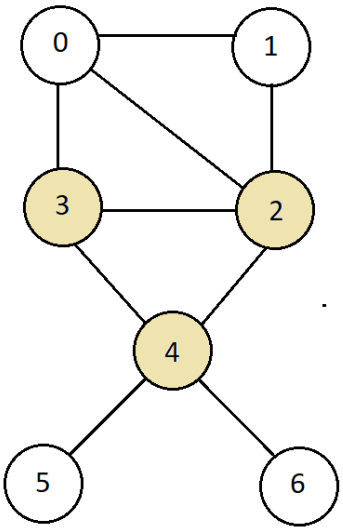


G

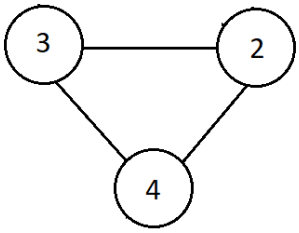


S

S is a subgraph of graph G because nodes 0, 1, and 3 form the subset of the set of nodes in G, and the edges connecting 0 to 3 and 0 to 1 also form the subset of the set of edges in G. Another subgraph of Graph G could be the one mentioned below.



G



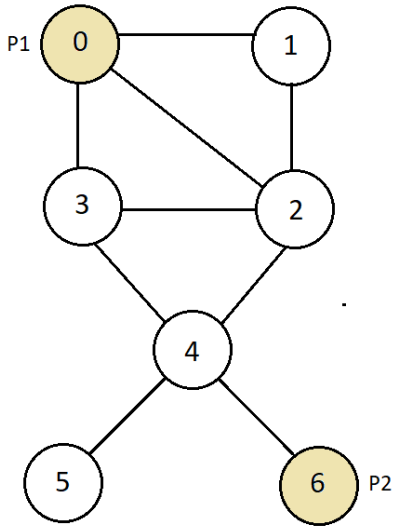
S

This is a subgraph of graph G too, because nodes 3, 2, and 4 form the subset of the set of nodes in G, and the edges connecting 3 to 4, 3 to 2, and 2 to 4 also form the subset of the set of edges in G.

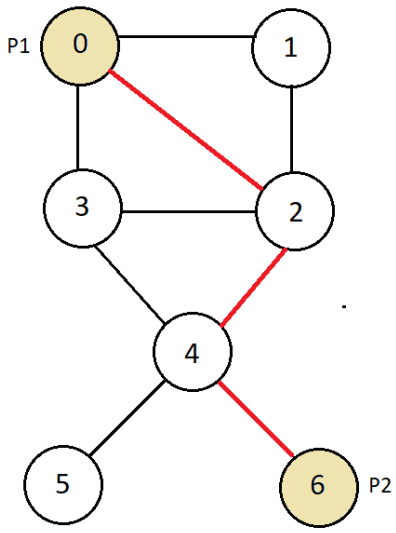
That should be enough for you to understand what subgraphs of a graph are. Moving ahead, we'll now talk about connected and complete graphs.

Connected Graphs:

A connected graph, as the word connected suggests, is a graph that is **connected** in the sense of a topological space, i.e., that 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**. Consider the graph below:



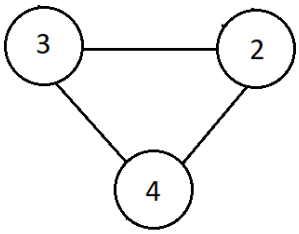
We have marked two random points P1 and P2, and we'll see if there is a path from P1 to P2. And if you could see, there is indeed a path from P1 to P2 via nodes 2 and 4.



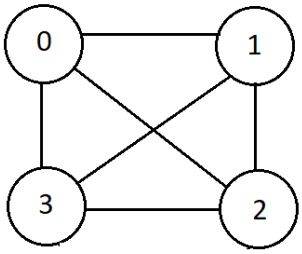
There could be a number of other ways to reach point P2 from P1, but there should exist at least one path from any point to another point for the graph to be called connected, otherwise disconnected. You should check for some other pair of vertices in the above graph.

Complete Graphs:

A complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge. Below, I have illustrated complete graphs with 3 and 4 nodes respectively.



Complete graph with 3 nodes

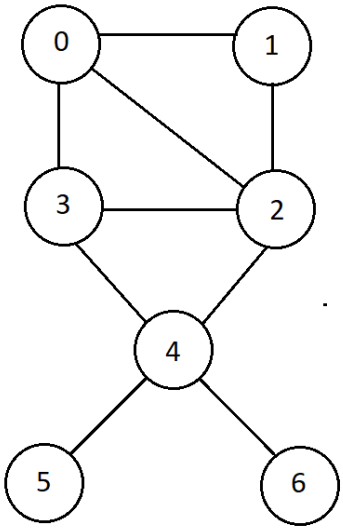


Complete graph with 4 nodes

As you can see, in any of the examples above, every pair of nodes is connected using a unique edge of their own. That is what makes a graph complete. You should consider making a complete graph with more vertices, say 5 or 7.

Note: Every complete graph is a connected graph, although every connected graph is not necessarily complete.

Quick Quiz:



1. Is the graph illustrated above connected?
2. Is the same graph complete?

Hint: Nodes 3 and 1 are not directly connected.

And now, having done subgraphs, connected and complete graphs, we are good to learn about spanning trees. Let's proceed with them.

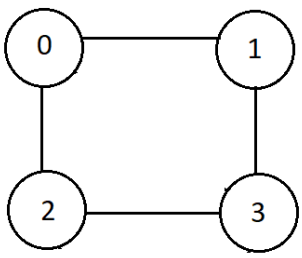
What are spanning trees?

As we learnt, a subgraph of a graph G is a graph whose vertices and edges are subsets of the original graph G. Hence, a connected subgraph 'S' of a graph G (V, E) is said to be **a spanning tree** of the graph if and only if:

1. All the vertices of G are present in S,
2. No. of edges in S should be |V|-1, where |V| represents the number of vertices.

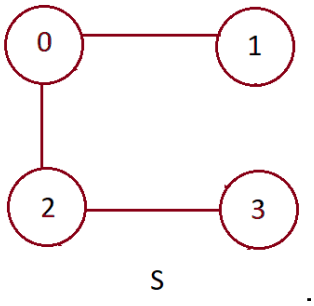
That is, for a **subgraph** of a graph to be called **a spanning tree** of that graph, it should have **all vertices of the original graph** and must have exactly **|V|-1 edges,** where |V| represents the number of vertices and the graph should be **connected**.

You might find this difficult to comprehend, but let me give you an example to help you understand. Consider a simple graph G I have illustrated below:



G

Now, consider another graph S.



Now, let's find out, step by step, if graph S is a spanning tree of graph G or not, considering nodes 1 and 3 don't have an edge in common.

Is graph S a subgraph of graph G? ✓

Yes, graph S is a subgraph of graph G. All nodes/vertices present in S are also a part of graph G, and all vertices exist in graph G too.

Is graph S connected? ✓

Yes, graph S is connected since we can go from any node to any other node via some edges in the graph.

Are all vertices of graph G present in graph S? ✓

Yes, all vertices of graph G which are vertices 0, 1, 2, and 3, are present in graph S.

Does the number of edges in graph S equal the number of vertices in graph G - 1? ✓

Yes, the number of edges in graph S equals the number of vertices in graph G - 1, since the number of vertices in graph G is 4, and the number of edges in graph S is 3.

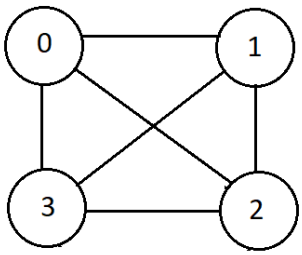
Since graph G satisfies all the above conditions, it is a spanning tree of the graph G. Therefore, whenever you are given a graph and are asked whether this graph is a spanning tree of another graph or not, you should just simply check for these four conditions, and declare it a spanning tree only if it satisfies all four conditions, and not a spanning tree even if it misses by any one of them.

And if you are asked to create a spanning tree of a graph, you must first plot all the vertices of the original graph. And then create V-1 edges which were there in the original graph and what makes your graph connected. This would be sufficient, and you can just ignore all other edges present in the original graph.

Now, you must be wondering about how many possible spanning trees are there for a graph. Well, there isn't a general formula for any random graph, but there is one for all complete graphs.

Number of Spanning trees for Complete graphs:

A complete graph has n(n-2) spanning trees where n represents the number of vertices in the graph. Consider the complete graph I have made below with 4 nodes.



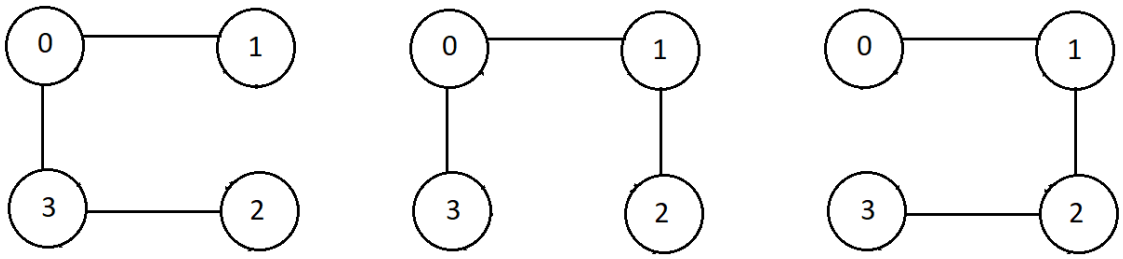
G .

This complete graph has 4 vertices, hence the number of spanning trees it can have is, 4 raised to the power (4-2), i.e., 42 which is 16.

Quick Quiz:

1. Draw any 3 spanning trees of the graph illustrated above.

Answer:



S1 S2 . S3

So, that was it for today. We learned what spanning trees are and other graph terminologies. Yet, so much more is left to the present. Next, we will learn about minimum spanning trees and spanning tree cost calculation, and more. Stay tuned.

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. Lectures on graphs have just started, and if you haven't saved this already, you just have to move on to codewithharry.com or my YouTube channel to access them. See you all there in the next lecture where we'll learn to calculate the spanning-tree cost and what minimum spanning trees are. Till then keep learning.