

Theory: Spanning trees

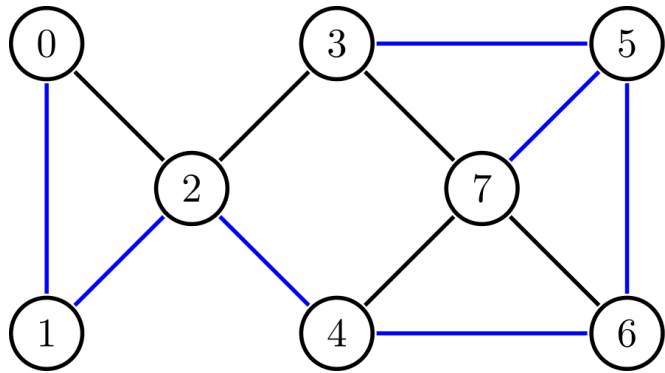
🕒 22 minutes 6 / 6 problems solved

Start practicing

438 users solved this topic. Latest completion was about 9 hours ago.

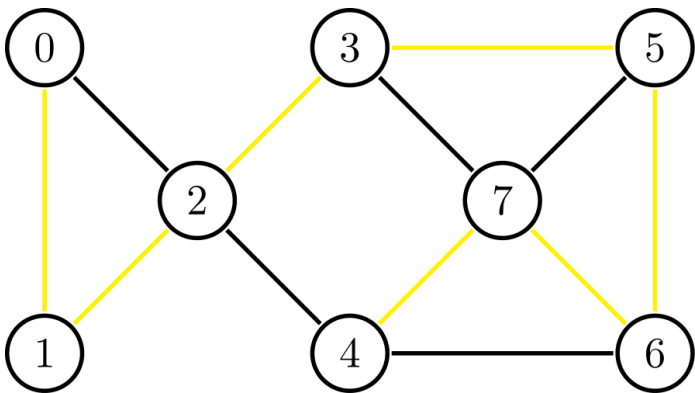
§1. A spanning tree in unweighted graphs

Recall that a **tree** is a connected graph without cycles. A **spanning tree** of a connected unweighted graph G is a subgraph of G that is a tree that connects (spans) all the nodes of G . Consider the following example:



A subgraph consisting of blue edges is a tree since it is connected and contains no cycles. Also, it covers all 8 nodes of the graph. Therefore, it is a spanning tree for this graph.

Note that there may exist several spanning trees in a graph. Given below is an example of another spanning tree for the above graph (shown in yellow):



For any spanning tree, all properties of trees are fulfilled. In particular, if a graph contains n nodes, any spanning tree of the graph consists of $n - 1$ edges. You can see that the above graph has 8 nodes and both spanning trees have 7 edges.

Another important property of trees is that removing any edge makes the tree disconnected. At this point, a spanning tree can be thought of as a **minimally connected subgraph**.

There exist several problems where spanning trees can be useful. For example, assume that the above graph represents a network, where the nodes are computers and the edges are potential links between them. The goal is to select the minimum number of links such that all the computers are connected. For this problem, a spanning tree gives us a desirable solution since it connects all the nodes of the graph and is minimum (removing any edge from a tree makes the tree disconnected).

§2. A spanning tree in weighted graphs

In a weighted graph, a spanning tree is the same as in an unweighted, but it also has a **weight** equal to the **sum of the edge weights** of the tree. Let's see the following example:

Current topic:

✓ [Spanning trees](#) ...

Topic depends on:

✓ [Weighted graph](#) ...

✓ [Tree](#) ...

Topic is required for:

✓ [Prim's algorithm](#) ...

✓ [Kruskal's algorithm](#) ...

Table of contents:

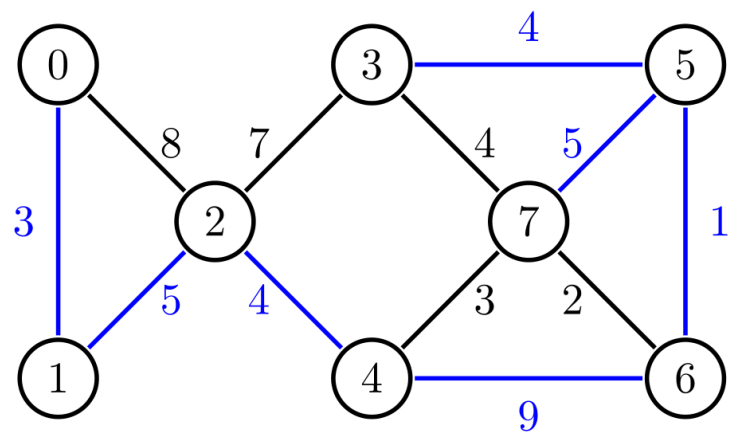
[1 Spanning trees](#)

[§1. A spanning tree in unweighted graphs](#)

[§2. A spanning tree in weighted graphs](#)

[§3. Algorithms for finding a spanning tree](#)

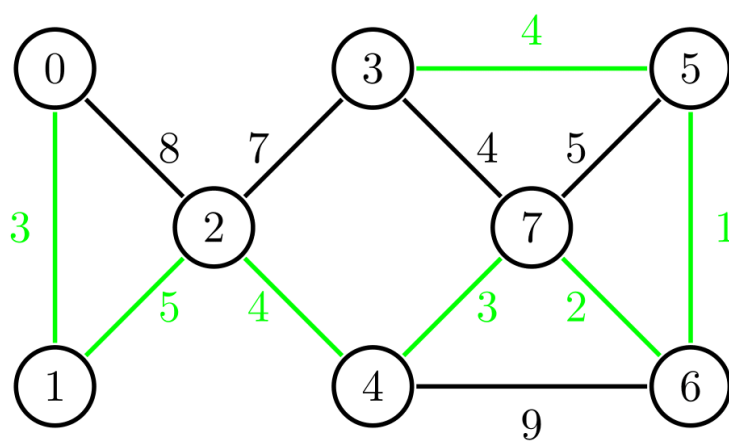
[Feedback & Comments](#)



The blue edges of the above graph form a spanning tree which weight is equal to

$3 + 5 + 4 + 9 + 1 + 5 + 4 = 31$ If we again consider the graph as a computer network, then the weight of each edge can be thought of as a maintenance cost of the corresponding link. If we need to find a set of links connecting all the computers, it is reasonable to minimize the total cost of the links.

In a weighted connected graph, a spanning tree having the smallest weight among all other spanning trees is called a **minimum spanning tree** (shortly, **MST**). For the above graph, the spanning tree consisting of the blue edges is not minimum, since there exists another spanning tree in the graph with a smaller weight:



The total weight of the green spanning tree is $3 + 5 + 4 + 3 + 2 + 1 + 4 = 22$, which is less than the cost of the blue one. Also, the green spanning tree is minimum, since there is no other spanning tree in the graph having a smaller weight.

One may prove that if all edge weights of a graph are unique then there exists only one minimum spanning tree in the graph. Otherwise, the graph may contain several spanning trees.

Note that all the above graphs are connected. Graphs consisting of several connected components don't have a spanning tree since nodes from different components cannot be connected. In such a case, we may find a spanning tree in each connected component. A union of all these spanning trees will be a **spanning forest** of the graph.

§3. Algorithms for finding a spanning tree

In unweighted graphs, a spanning tree can be found using either a depth-first search or breadth-first search. As for weighted graphs, two widely used approaches for the construction of a minimum spanning tree are Prim's and Kruskal's algorithms. All of them will be covered in the following lessons.

Report a typo

38 users liked this theory. didn't like it. What about you?



Start practicing

