

**General Properties of Spanning Tree**

We now understand that one graph can have more than one spanning tree. Following are a few properties of the spanning tree connected to graph G −

* A connected graph G can have more than one spanning tree.
* All possible spanning trees of graph G, have the same number of edges and vertices.
* The spanning tree does not have any cycle (loops).
* Removing one edge from the spanning tree will make the graph disconnected, i.e. the spanning tree is **minimally connected**.
* Adding one edge to the spanning tree will create a circuit or loop, i.e. the spanning tree is **maximally acyclic**.
* Always undirected (as tree is always undirected),also the corresponding graph must be undirected.

## Application of Spanning Tree

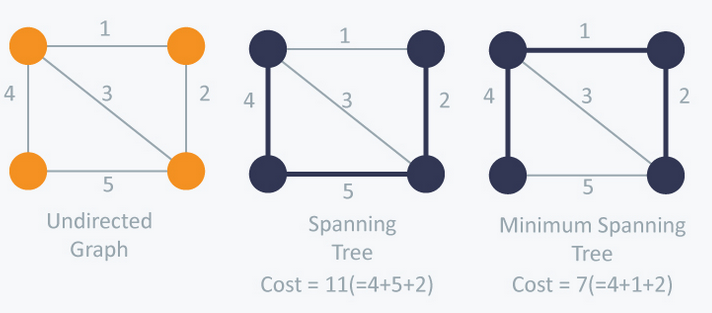
Spanning tree is basically used to find a minimum path to connect all nodes in a graph. Common application of spanning trees are −

* **Civil Network Planning**
* **Computer Network Routing Protocol**
* **Cluster Analysis**

Let us understand this through a small example. Consider, city network as a huge graph and now plans to deploy telephone lines in such a way that in minimum lines we can connect to all city nodes. This is where the spanning tree comes into picture.

* A **minimum spanning tree** (MST) or **minimum** weight **spanning tree** is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the **minimum** possible total edge weight.

In real-world situations, this weight can be measured as distance, congestion, traffic load or any arbitrary value denoted to the edges.



There are two famous algorithms for finding the Minimum Spanning Tree:

# Kruskal’s Algorithm

Kruskal’s Algorithm builds the spanning tree by adding edges one by one into a growing spanning tree. Kruskal's algorithm follows greedy approach as in each iteration it finds an edge which has least weight and add it to the growing spanning tree.

**Algorithm Steps:**

* Sort the graph edges with respect to their weights.
* Start adding edges to the MST from the edge with the smallest weight until the edge of the largest weight.
* Only add edges which doesn't form a cycle
* Prim's algorithm is significantly faster when you've got a really dense graph with many more edges than vertices.

==============

In an [undirected graph](https://en.wikipedia.org/wiki/Undirected_graph) *G*, two [*vertices*](https://en.wikipedia.org/wiki/Vertex_(graph_theory)) *u* and *v* are called connected if *G* contains a [path](https://en.wikipedia.org/wiki/Path_(graph_theory)) from *u* to *v*. Otherwise, they are called *disconnected*. If the two vertices are additionally connected by a path of length 1, i.e. by a single edge, the vertices are called *adjacent*. A [graph](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)) is said to be connected if every pair of vertices in the graph is connected.