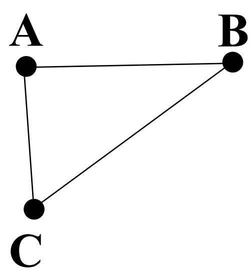
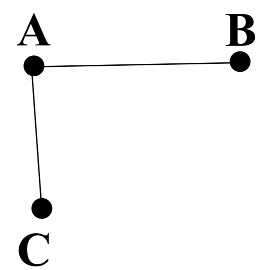
Tree:

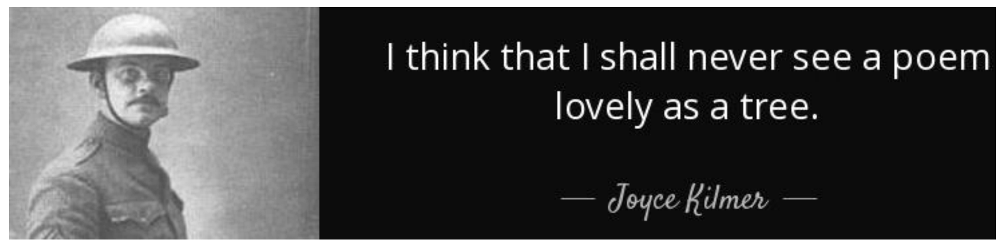
It's a connected graph that is efficiently connected, i.e. there are no unnecessary edges to retain connectedness of the vertices.



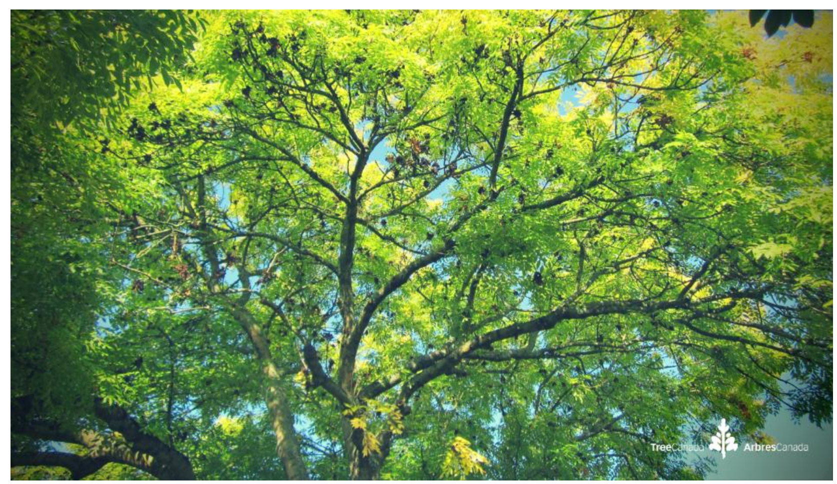


Properties of Trees:

1. It contains no circuits.



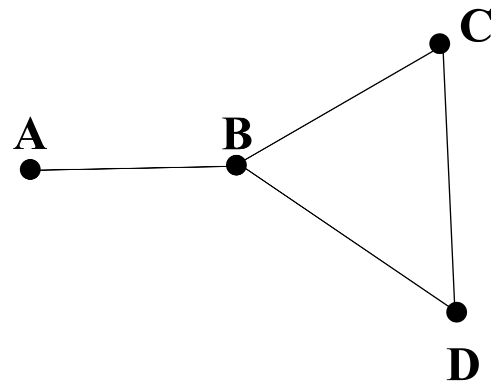
2. There is exactly one path joining any two vertices.

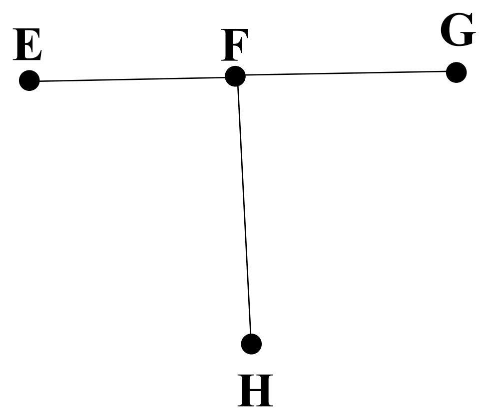


3. Every edge is a bridge.

4. A tree with *n* vertices has (*n*-1) edges.

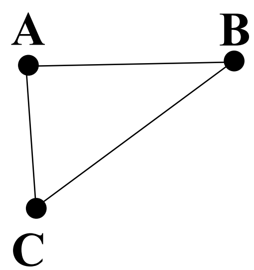
A connected graph that satisfies any one of these four properties must be a tree.


A
B
C
D

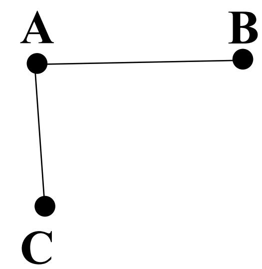

E
F
G
H

Subgraph:

It's a graph that consists of some of the vertices and edges of an original graph.



Original Graph

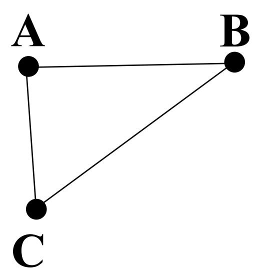


Subgraph

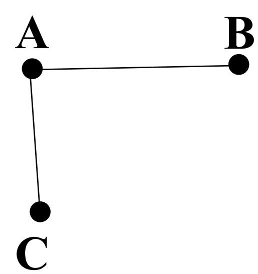
Subgraph

Spanning Tree:

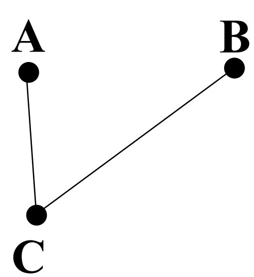
It's a subgraph of a connected graph that contains all of the original vertices and is a tree. The process of producing a spanning tree is called pruning.



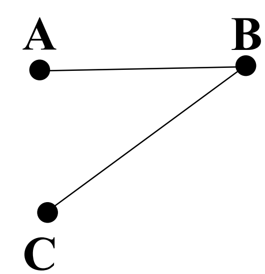
Original Graph



Spanning Tree



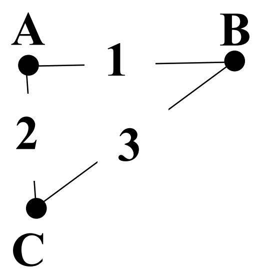
Spanning Tree



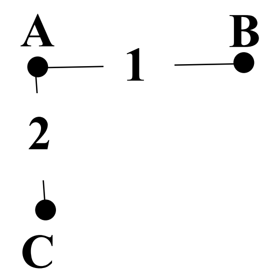
Spanning Tree

Minimum Spanning Tree:

It's a spanning tree for a weighted connected graph that has the smallest possible weight.



Original Weighted Graph



Minimum Spanning Tree

There is an algorithm for finding a minimum spanning tree.

Kruskal's Algorithm:

1. Find an edge with the smallest weight, and include it.

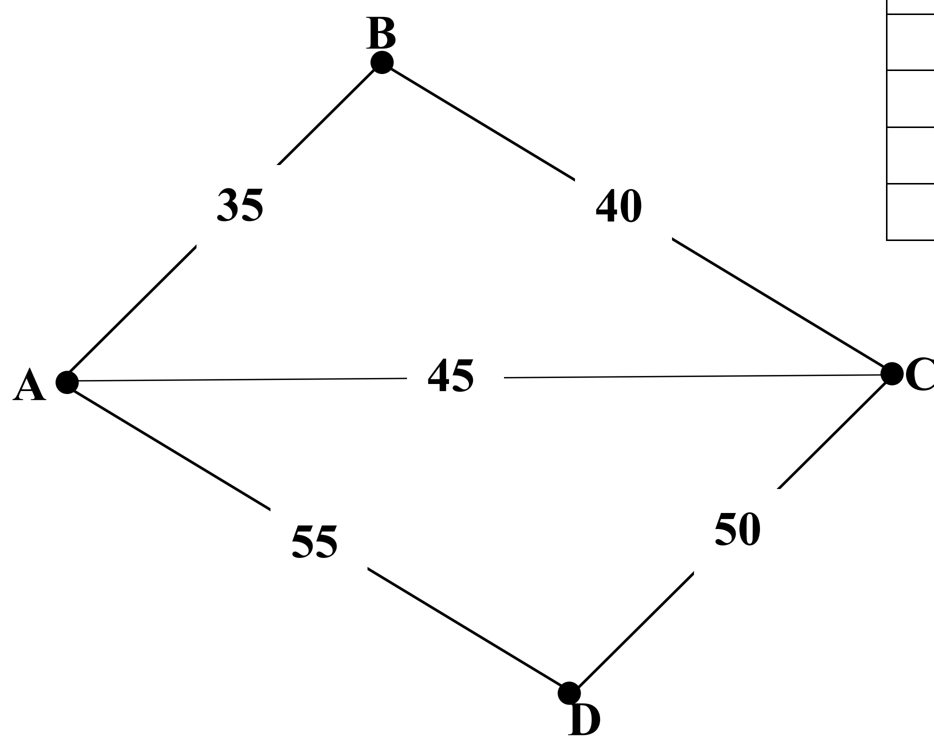
2. Find an edge with the next-smallest weight, and include it if it doesn't produce a circuit.

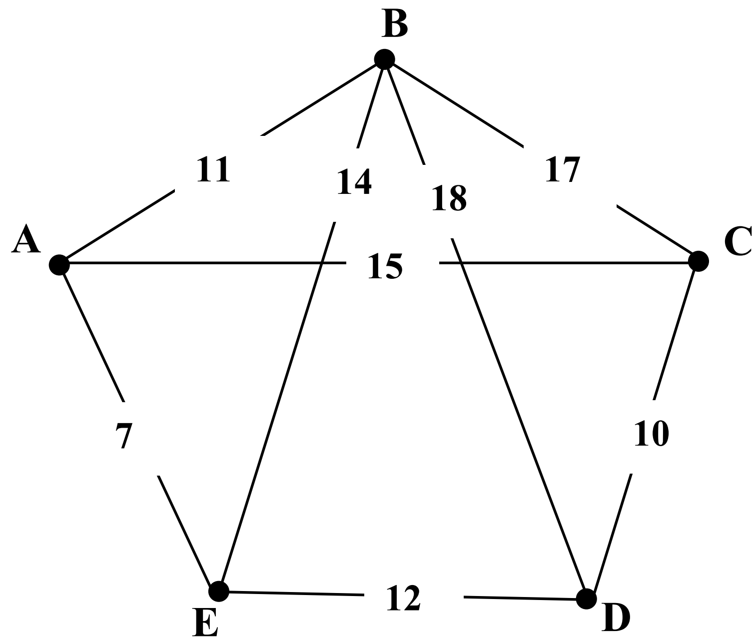
3. Find an edge with the next-smallest weight, and include it if it doesn't produce a circuit.

4. Continue the process until all the vertices are included and connected with no circuits.

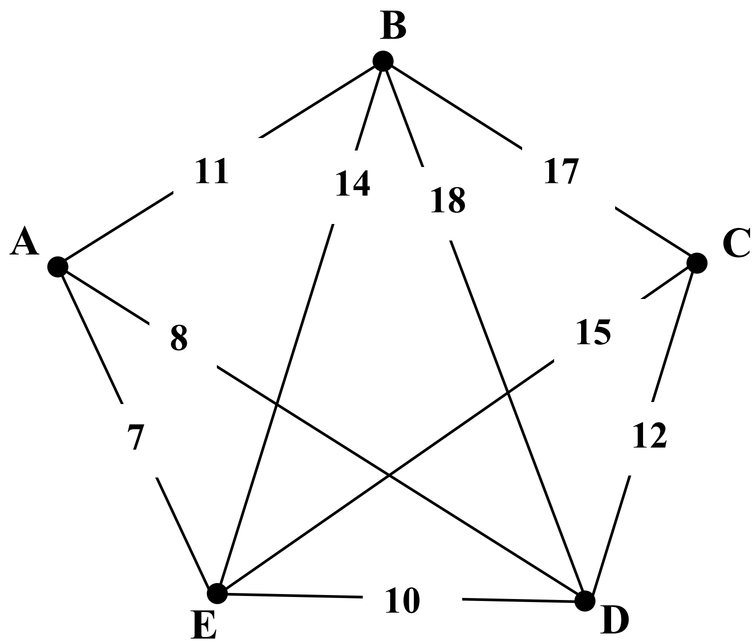
Apply Kruskal's Algorithm to the following weighted graphs.

Weights	[redacted]
35	
40	
45	
50	
55	





Weights	[redacted]
7	
10	
11	
12	
14	
15	
17	
18	



Weights	[redacted]
7	
8	
10	
11	
12	
14	
15	
17	
18	