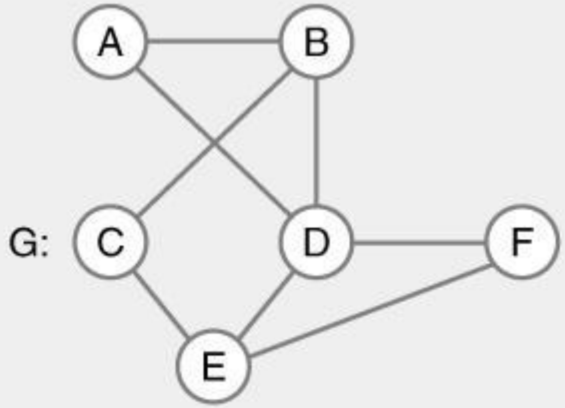
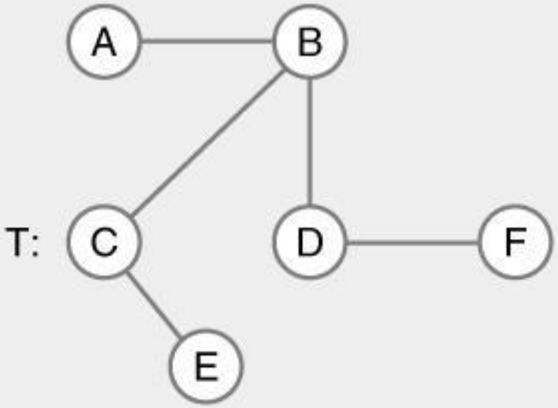
Spanning Tree

- **Set of edges connecting all nodes in graph**
  - **need N-1 edges for N nodes**
  - **no cycles, can be thought of as a tree**
- **Can build tree during traversal**



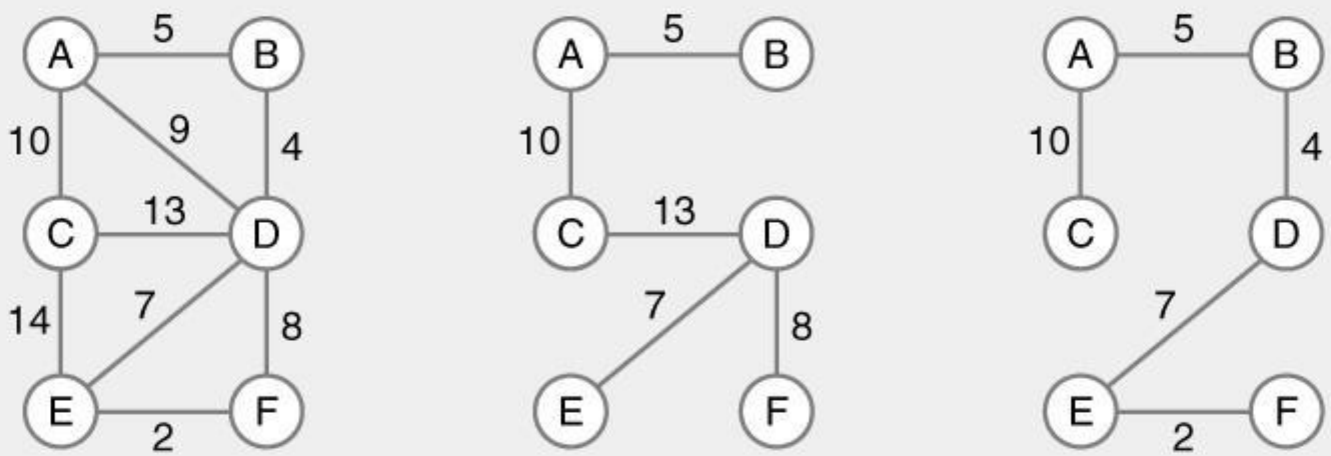
(a) Graph G



(b) Spanning tree T of graph G

Minimum Spanning Tree (MST)

- **Spanning tree with minimum total edge weight**



(a) Graph G

(b) A spanning tree of cost C = 43

(c) A minimum spanning tree of cost C = 28

Minimum Spanning Tree (MST)

- **Possible to have multiple MSTs**
  - **Different spanning trees with same weight**

- **Example applications**
  - **Minimize length of telephone lines for neighborhood**
  - **Minimize distance of airplane routes serving cities**

Algorithms for Finding MST

- **Three well known algorithms**

1. **Borůvka's algorithm [1926]**
   - **For constructing efficient electricity network**
   - **Rediscovered by Sollin in 1960s**
2. **Prim's algorithm [1957]**
   - **First discovered by Vojtěch Jarník in 1930**
   - **Similar to Djikstra's algorithm**
3. **Kruskal's algorithm [1956]**
   - **By Prof. Clyde Kruskal's uncle**

MST – Kruskal's Algorithm

sort edges by weight (from least to most)

tree = ∅

for each edge (X,Y) in order

    if it does not create a cycle

        add (X,Y) to tree

        stop when tree has N–1 edges

Keeps track of

- lightest edge remaining
- whether adding edge to MST creates cycle

Optimal solution computed with greedy algorithm

MST – Kruskal's Algorithm Example