Graph

Graph

A *graph* G consists of two things:

(i) A set $V = V(G)$ whose elements are called *vertices*, *points*, or *nodes* of G.
(ii) A set $E = E(G)$ of unordered pairs of distinct vertices called *edges* of G.

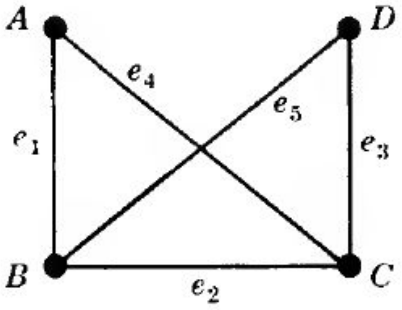
We denote such a graph by $G(V, E)$ when we want to emphasize the two parts of G.

Vertex and Edge

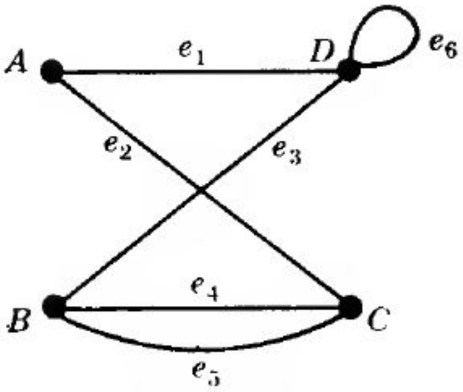
(i) V consists of vertices A, B, C, D.

(ii) E consists of edges $e_1 = \{A, B\}$, $e_2 = \{B, C\}$, $e_3 = \{C, D\}$, $e_4 = \{A, C\}$, $e_5 = \{B, D\}$.

In fact, we will usually denote a graph by drawing its diagram rather than explicitly listing its vertices and edges.



(a) Graph



(b) Multigraph

Degree of vertex

Degree of a Vertex

The *degree* of a vertex v in a graph G, written deg (v), is equal to the number of edges in G which contain v, that is, which are incident on v. Since each edge is counted twice in counting the degrees of the vertices of G, we have the following simple but important result.

Theorem 8.1: The sum of the degrees of the vertices of a graph G is equal to twice the number of edges in G.

Consider, for example, the graph in Fig. 8-5(a). We have

$$\deg(A) = 2, \quad \deg(B) = 3, \quad \deg(C) = 3, \quad \deg(D) = 2.$$

Subgraph

Subgraphs

Consider a graph $G = G(V, E)$. A graph $H = H(V', E')$ is called a *subgraph* of G if the vertices and edges of H are contained in the vertices and edges of G, that is, if $V' \subseteq V$ and $E' \subseteq E$. In particular:

(i) A subgraph $H(V', E')$ of $G(V, E)$ is called the subgraph *induced* by its vertices V' if its edge set E' contains all edges in G whose endpoints belong to vertices in H.

(ii) If v is a vertex in G, then $G - v$ is the subgraph of G obtained by deleting v from G and deleting all edges in G which contain v.

(iii) If e is an edge in G, then $G - e$ is the subgraph of G obtained by simply deleting the edge e from G.

Isomorphic Graphs

Graphs $G(V, E)$ and $G(V^*, E^*)$ are said to be *isomorphic* if there exists a one-to-one correspondence $f: V \rightarrow V^*$ such that $\{u, v\}$ is an edge of G if and only if $\{f(u), f(v)\}$ is an edge of G^*. Normally, we do not distinguish between isomorphic graphs (even though their diagrams may "look different"). Figure 8-6 gives ten graphs pictured as letters. We note that A and R are isomorphic graphs. Also, F and T are isomorphic graphs, K and X are isomorphic graphs and M, S, V, and Z are isomorphic graphs.

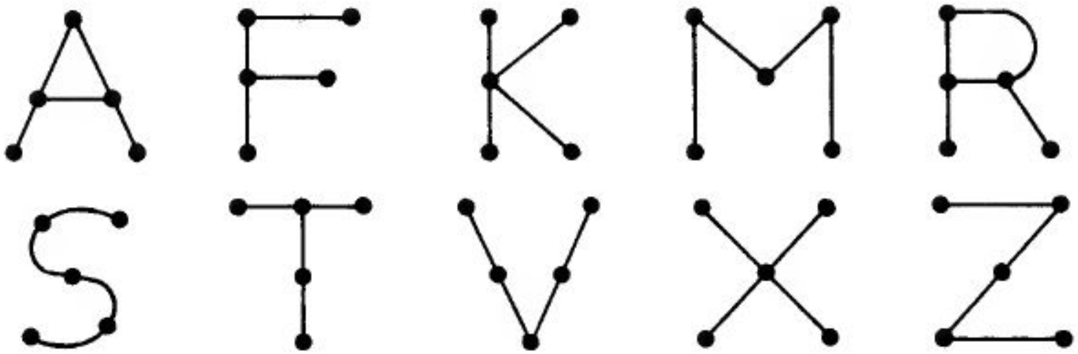


Fig. 8-6

Homeomorphic Graphs

Given any graph G, we can obtain a new graph by dividing an edge of G with additional vertices. Two graphs G and G^* are said to *homeomorphic* if they can be obtained from the same graph or isomorphic graphs by this method. The graphs (a) and (b) in Fig. 8-7 are not isomorphic, but they are homeomorphic since they can be obtained from the graph (c) by adding appropriate vertices.

Homomorphic graph

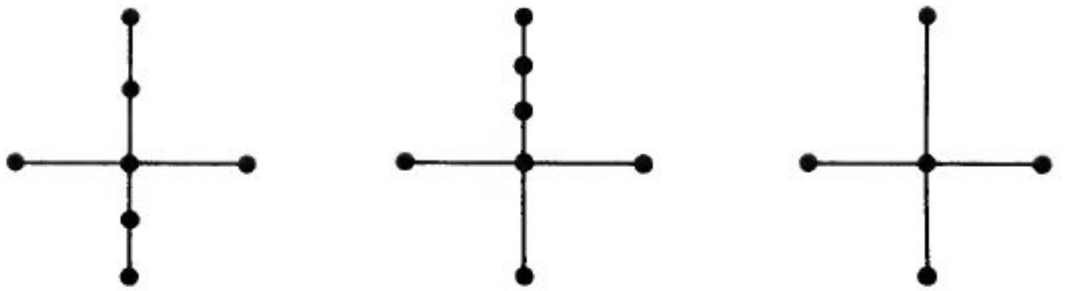
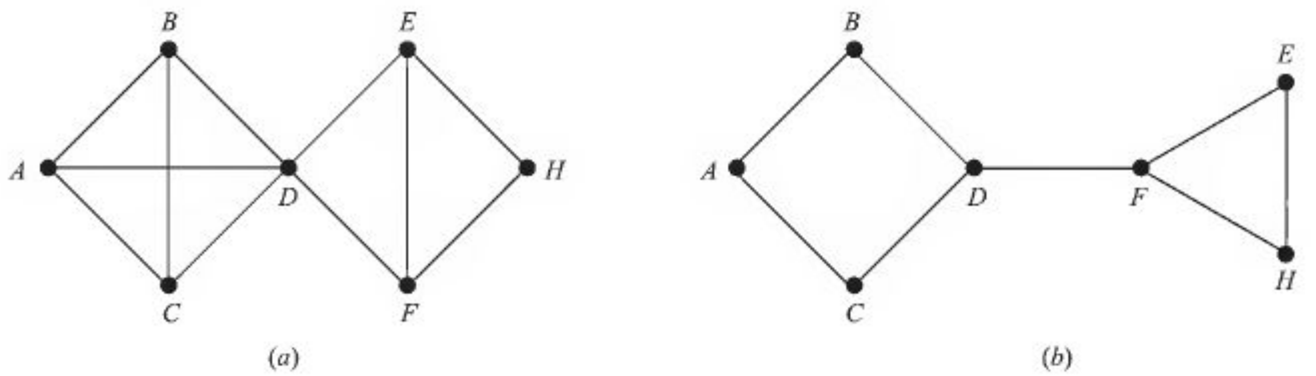
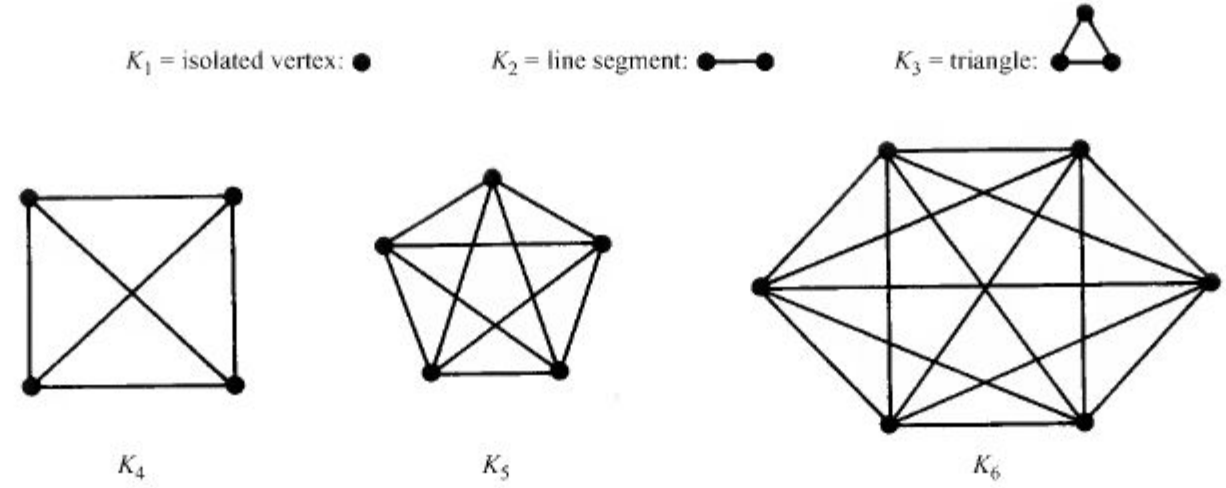

(a) (b) (c)

Fig. 8-7

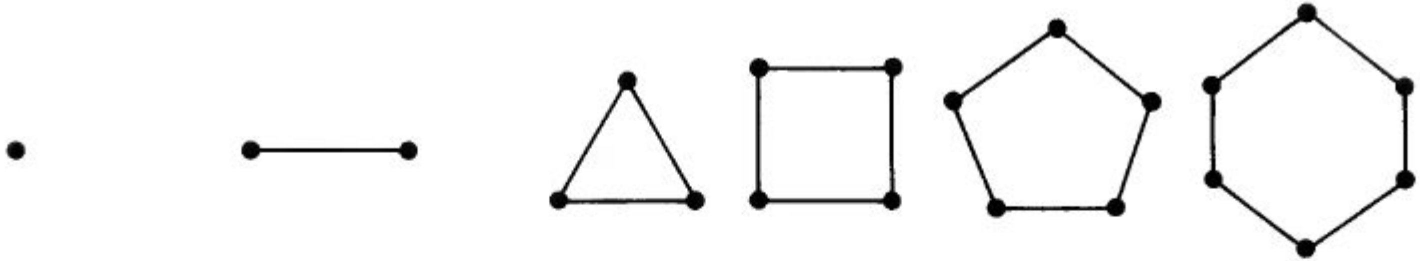
Cut point and Bridge



Complete Graph



Regular graph



(i) 0-regular (ii) 1-regular (iii) 2-regular

Bipartite graph

A graph G is said to be bipartite if its vertices V can be partitioned into two subsets M and N such that each

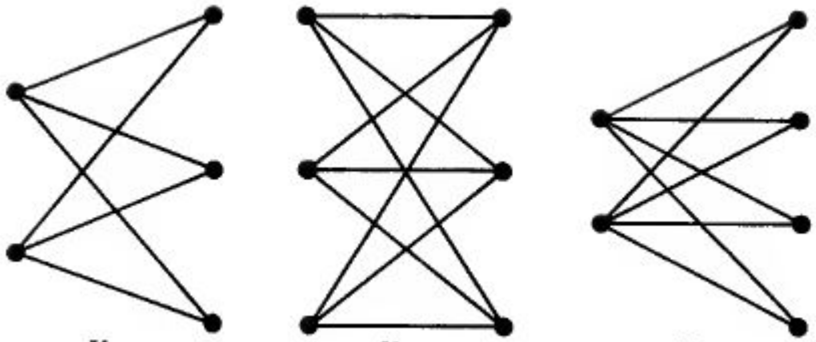
edge of G connects a vertex of M to a vertex of N. By a complete bipartite graph, we mean that each vertex of

M is connected to each vertex of N\ this graph is denoted by Km n where m is the number of vertices in M and

n is the number of vertices in N, and, for standardization, we will assume m < n. Figure 8-16 shows the graphs

K2 3 , K3 3 , and K2 4 , Clearly the graph Km has mn edges.

Bipartite graph



$K_{2,3}$ $K_{3,3}$ $K_{2,4}$

Tree

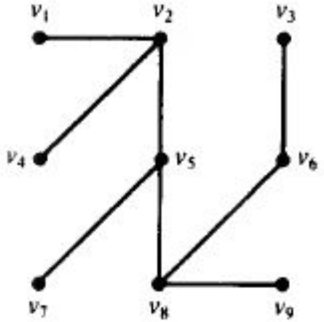
Let G be a graph with $n > 1$ vertices. Then the following are equivalent:

(i) G is a tree.

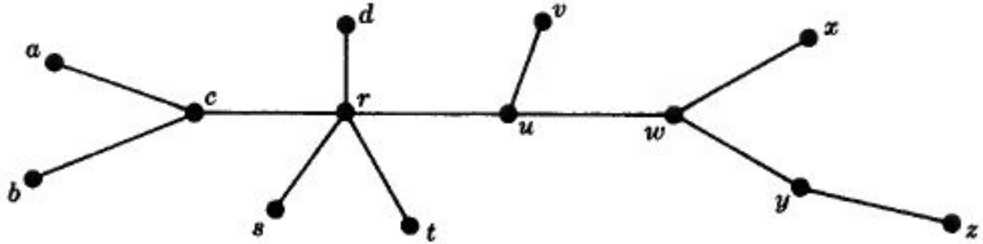
(ii) G is a cycle-free and has $n — 1$ edges.

(iii) G is connected and has $n — 1$ edges.

Tree



(a)



(b)

Spanning tree

A subgraph T of a connected graph G is called a spanning tree of G if T is a tree and T includes all the

vertices of G. Figure 8-18 shows a connected graph G and spanning trees 7\, T2 , and T3 of G.

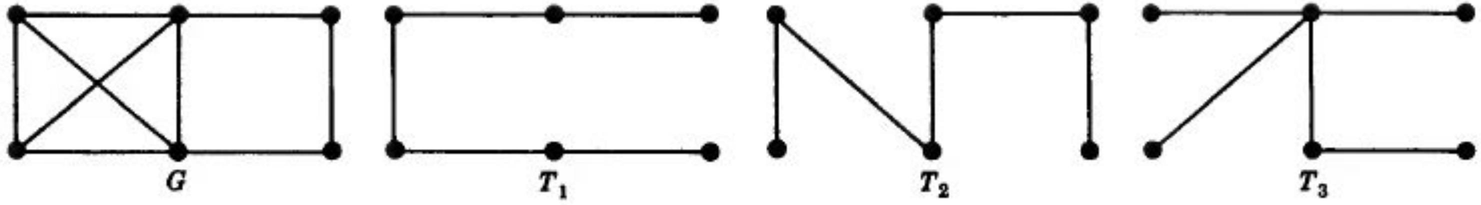


Fig. 8-18

Minimum spanning tree algorithm

Algorithm 8.2: The input is a connected weighted graph G with n vertices.

Step 1. Arrange the edges of G in the order of decreasing weights.

Step 2. Proceeding sequentially, delete each edge that does not disconnect the graph until $n - 1$ edges remain.

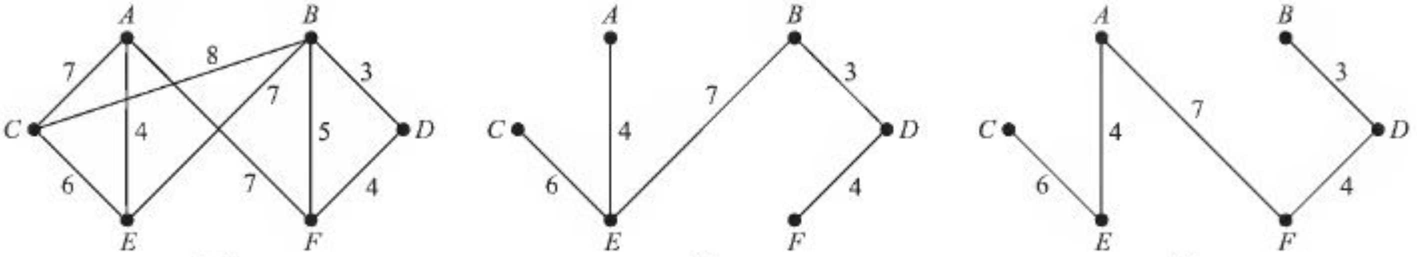
Step 3. Exit.

Algorithm 8.3 (Kruskal): The input is a connected weighted graph G with n vertices.

Step 1. Arrange the edges of G in order of increasing weights.

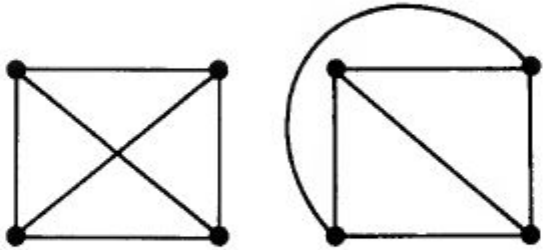
Step 2. Starting only with the vertices of G and proceeding sequentially, add each edge which does not result in a cycle until $n - 1$ edges are added.

Step 3. Exit.



(a) (b) (c)

Planar graph


(a) (b)