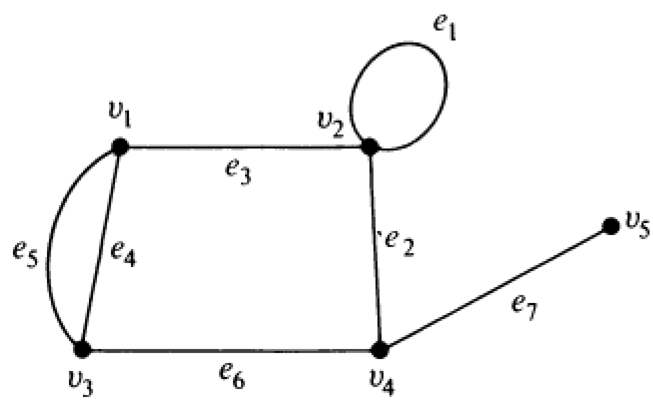
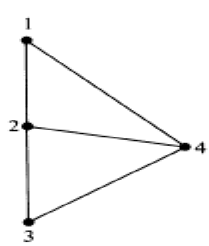
What is a Graph?

- A graph $G = (V, E)$ consists of a set of objects $V = \{v_1, v_2, \ldots\}$ called vertices, and another set $E = \{e_1, e_2, \ldots\}$, whose elements are called edges, such that each edge e_k is identified with an unordered pair (v_i, v_j) of vertices. The vertices v_i, v_j associated with edge e_k are called the end vertices of e_k.
- The most common representation of a graph is by means of a diagram, in which the vertices are represented as points and each edge as a line segment joining its end vertices. Often this diagram itself is referred to as the graph.

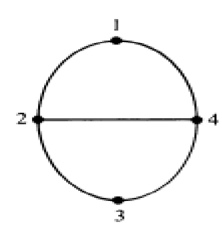
  Eg: Graph with five vertices and seven edges.



- Observe that this definition permits an edge to be associated with a vertex pair (v_i, v_i). Such an edge having the same vertex as both its end vertices is called a self-loop.
  Eg: Edge e_1 is a self loop.
- Also note that the definition allows more than one edge associated with a given pair of vertices, **Eg:** Edges e_4 and e_5.(Such edges are referred to as parallel edges.)
- A graph that has neither self-loops nor parallel edges is called a **simple graph**.
- It should also be noted that, in drawing a graph, it is immaterial whether the lines are drawn straight or curved, long or short: what is important is the incidence between the edges and vertices.
  Eg: For example, the two graphs drawn in Figs. (a) and (b) are the same, because incidence between edges and vertices is the same in both cases.



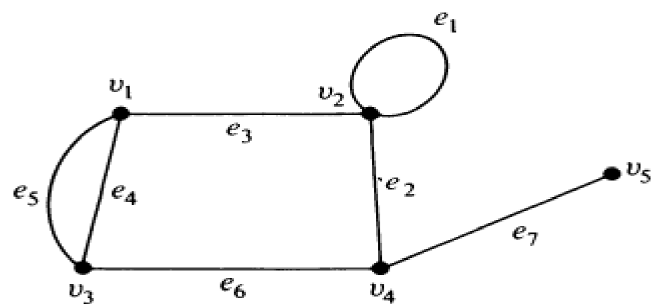
(a)



(b)

INCIDENCE AND DEGREE

- When a vertex v_i is an end vertex of some edge e_j, then, v_i and e_j are said to be incident with (on or to) each other.
  Eg: for example, edges e_2, e_6, and e_7 are incident with vertex v_4



- Two nonparallel edges are said to be adjacent if they are incident on a **common vertex**.
  Eg: For example, e_2 and e_7 are adjacent.
- Similarly, two vertices are said to be adjacent if they are the end vertices of the **same edge**.
  Eg: Here, v_4 and v_5 are adjacent, but v_1 and v_4 are not.
- The number of edges incident on a vertex v_i with self-loops counted twice, is called the degree, $d(v_i)$, of vertex v_i.
  Eg: for example, $d(v_1) = d(v_3) = d(v_4) = 3$, $d(v_2) =4$, and $d(v_5) =1$.
- Let us now consider a graph G with e edges and n vertices $v_1, v_2, \ldots, v_n$. Since each edge contributes two degrees, the sum of the degrees of all vertices in G is twice the number of edges in G.

$$\sum_{i=1}^{n} d(v_i) = 2e$$

  Eg: In the graph, $d(v_1) + d(v_2) + d(v_3) + d(v_4) + d(v_5) = 3+4+3+3+1 = 14 =$ 2* no. of edges.

- **Theorem:** The number of vertices of odd degree in a graph is always even.

Note:

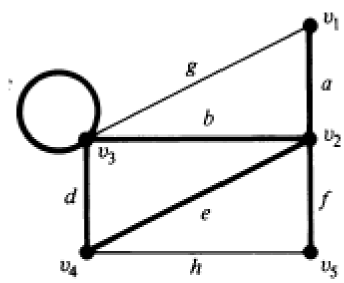
(1) the maximum degree of any vertex in a **simple graph** with n vertices is (n-1).

(2) the maximum number of edges in a **simple graph** with n vertices is n(n - 1)/2.

Walks, Paths, and Circuits

- A walk is defined as a finite alternating sequence of vertices and edges, beginning and ending with vertices, such that each edge is incident with the vertices preceding and following it.
- No edge appears (is covered or traversed) more than once in a walk. A vertex, however, may appear more than once.

Eg: For the given below graph, v_1 a v_2 b v_3 c v_3 d v_4 e v_2 f v_5 is a walk (shown with heavy lines).



- Vertices with which a walk begins and ends are called its terminal vertices. Eg: Vertices v_1 and v_5 are the terminal vertices of the walk shown.
- It is possible for a walk to begin and end at the same vertex. Such a walk is called a closed walk. A walk that is not closed (i.e., the terminal vertices are distinct) is called an open walk (as given here).
- An open walk in which no vertex appears more than once is called a **path**.

  Eg: v_1 a v_2 b v_3 d v_4 is a path, whereas v1 a v2 b v3 c v3 d v4 e v2 f v5 is not a path.

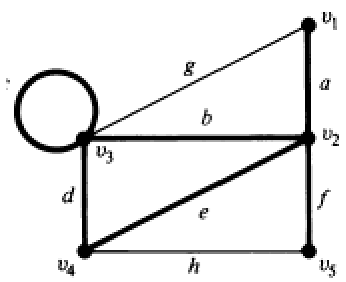
- The number of edges in a path is called the length of a path.
- A closed walk in which no vertex (except the initial and the final vertex) appears more than once is called a **circuit**. That is, a circuit is a closed, non-intersecting walk.

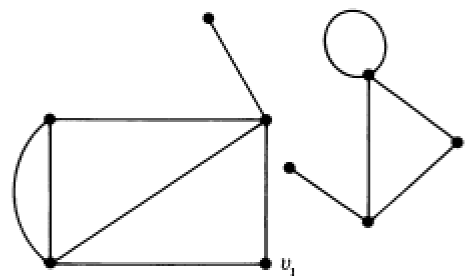
  Eg: v2 b v3 d v4 e v2 is a circuit.

CONNECTED GRAPHS, DISCONNECTED GRAPHS, AND COMPONENTS

- A graph G is said to be connected if there is at least one path between every pair of vertices in G. Otherwise, G is disconnected.

  Eg:





- It is easy to see that a disconnected graph consists of two or more connected graphs. Each of these connected subgraphs is called a component.

TREES

A tree is a connected graph without any circuits.

It follows immediately from the definition that a **tree has to be a simple graph**, that is, having neither a self-loop nor parallel edges (because they both form circuits).

SOME PROPERTIES OF TREES

(1) There is one and only one path between every pair of vertices in a tree, T.
(2) If in a graph G there is one and only one path between every pair of vertices, G is a tree.
(3) A tree with n vertices has n - 1 edges.
(4) Any connected graph with n vertices and n - 1 edges is a tree.
(5) A graph is a tree if and only if it is minimally connected.
(6) A graph G with n vertices, n - 1 edges, and no circuits is connected.

Note: The above properties(theorems) can be summarized by saying that the following are five different but **equivalent definitions** of a tree.

1. G is **connected** and is **circuitless,** or

2. G is connected and has **n - 1 edges**, or

3. G is circuitless and has n - 1 edges, or

4. There is exactly one path between every pair of vertices in G, or

5. G is a minimally connected graph.

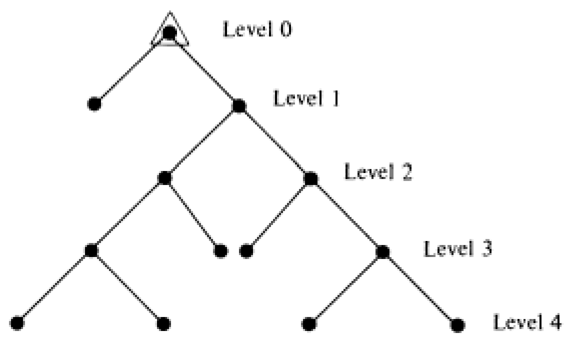
ROOTED AND BINARY TREES

- A tree in which one vertex (called the root) is distinguished from all the others is called a rooted tree. (In a diagram of a rooted tree, the root is generally marked distinctly)
- Generally, the term tree means trees without any root. However, for emphasis they are sometimes called **free trees** (or nonrooted trees) to differentiate them from the rooted kind.

Binary Trees

- A special class of **rooted trees,** called binary rooted trees, is of particular interest, since they are extensively used in the study of computer search methods, binary identification problems, and variable-length binary codes.
- A binary tree is defined as a tree in which there is exactly one vertex of degree two, and each of the remaining vertices is of degree one or three.
  (Obviously, we are talking about trees with three or more vertices.)
  Since the vertex of degree **two is distinct from all other vertice**s, this vertex serves as a root.

  Eg: A 13-vertex, 4-level binary tree.



<u>Two properties of binary trees follow directly from the definition:</u>

1. The number of vertices n in a binary tree is always odd.

Proof: This is because there is exactly one vertex of even degree, and the remaining n - 1 vertices are of odd degrees. Since we know that the number of vertices of odd degrees is even, (n – 1) is even. Hence n is odd.

2. Let p be the number of pendant vertices in a binary tree T. Then (n - p – 1) is the number of vertices of degree three. Therefore, the number of edges in T equals:

$$(p + 3(n - p - 1) + 2) = 2(n - 1)$$

$$\Rightarrow p = (n + 1)/2$$

$$\Rightarrow p - 1 = n - p$$

"the number of **internal vertices*** in a binary tree is one less than the number of pendant vertices."

(* A nonpendant vertex in a tree is called an internal vertex.)