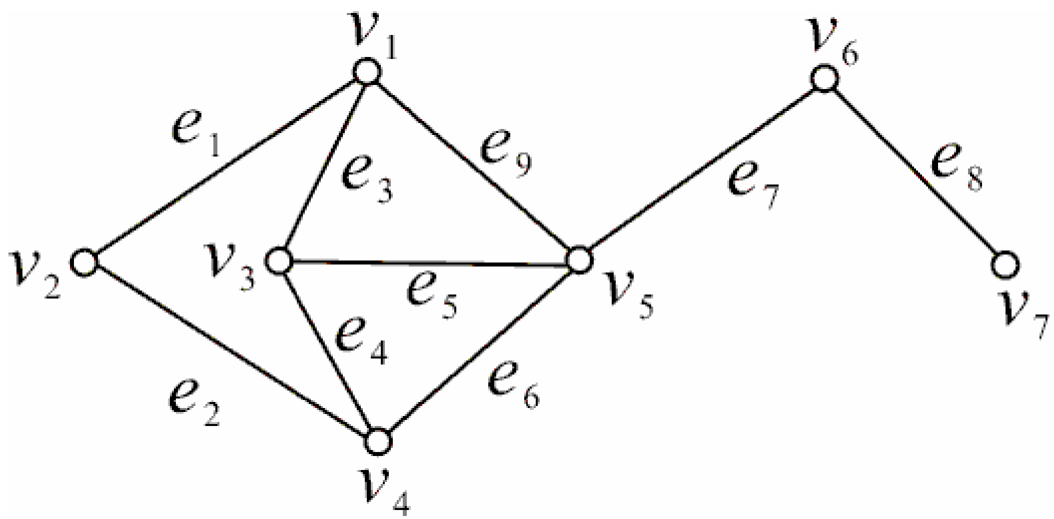
Paths

- Definition
- Let n be a nonnegative integer and G an undirected graph. A path of length n from u to v in G is a sequence of n edges e_1,… e_n of G for which there exists a sequence x_0=u, x_1, … x_n =v of vertices such that $e_i$has the endpoints $x_{i-1}$and x_i, when the graph is simple, we denote this path by its vertex sequence.

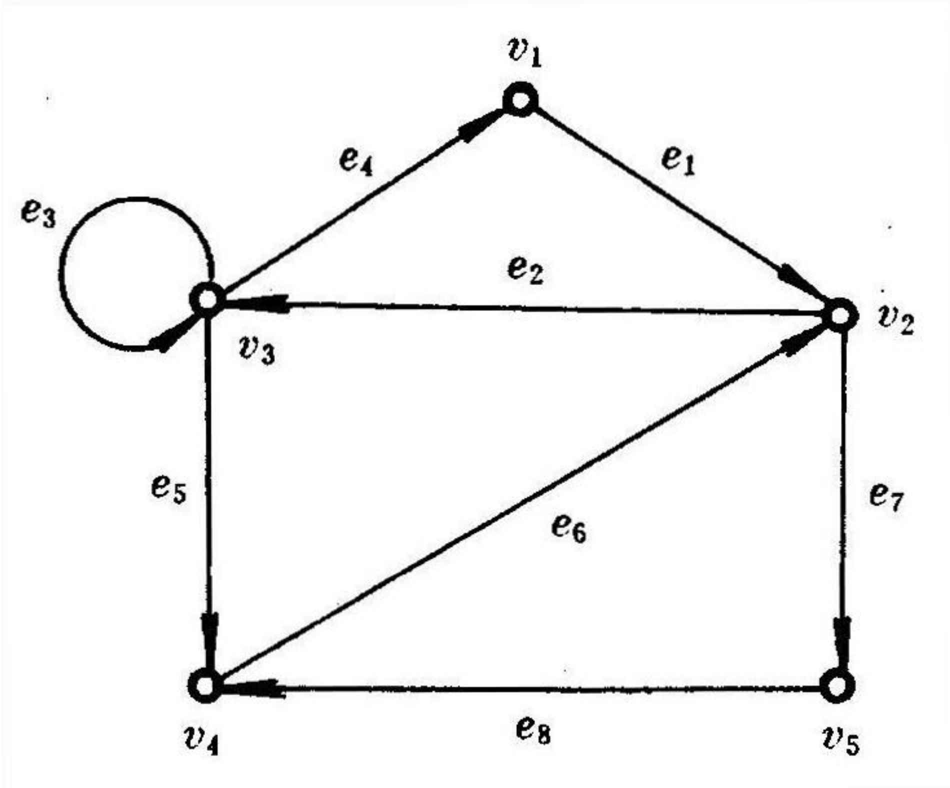
  circuit ---- begins and ends at the same vertex
- Simple path ---- does not contain the same edge more than once.
- length of a path ---- the number of edges on the path

Example



- Definition
- Let n be a nonnegative integer and G a directed graph. A path from a to b in G is a sequence of edges(x_0, x_1), (x_1, x_2),…(x_{n-1}, x_n) , where x_0=a, x_n =b. that is, a sequence of edges where the terminal vertex of an edge is the same as the initial vertex in the next edge in the path.
- Using $e_1, e_2 \ldots e_n$ to denote path.
- A path can pass through a vertex more than once, can traverse an edge more than once.

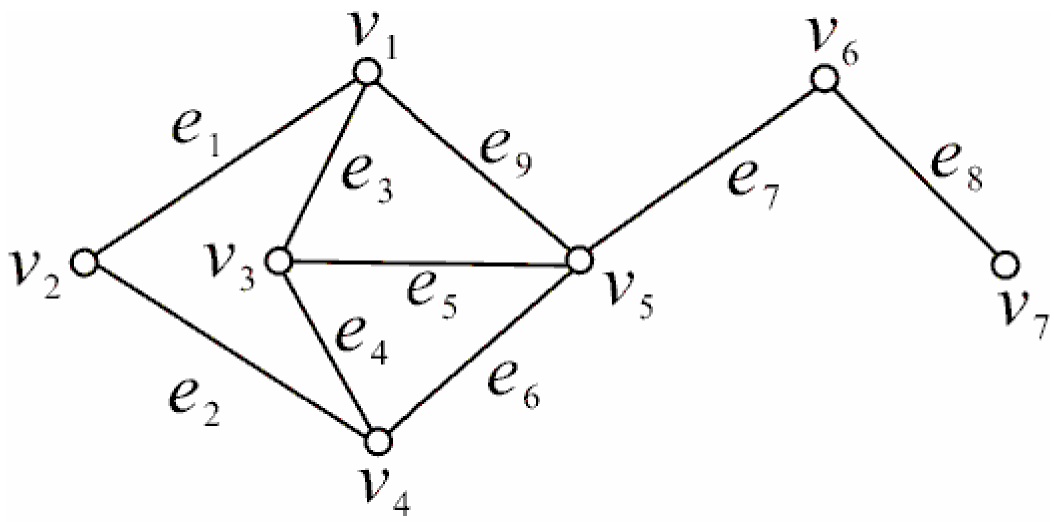
Example



Connectedness

- Definition:
- An undirected graph is called connected if there is a path between every pair of distinct vertices of the graph. An undirected graph that is not connected is called disconnected. We say that we disconnect a graph when we remove vertices or edges, to produce a disconnected subgraph.

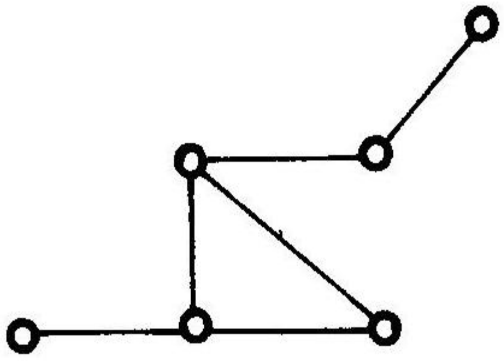
Example



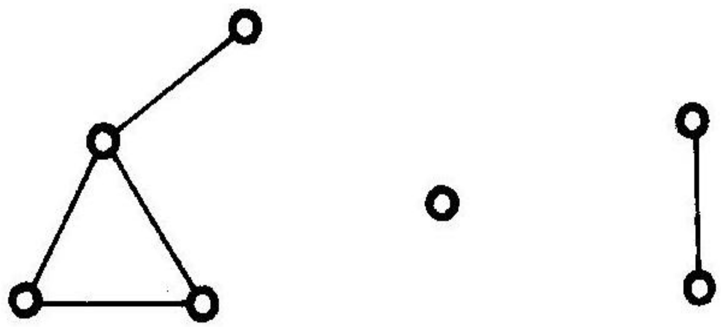
- Theorem
- There is a simple path between every pair of distinct vertices of a connected undirected graph.

Connected components

- A connected component of a graph G is a connected graph of G that is not a proper subgraph of another subgraph of G. That is, a connected component of a graph G is a maximal connected graph of G.
- A graph that is not connected has two or more connected component that are disjoint and have G as their union.



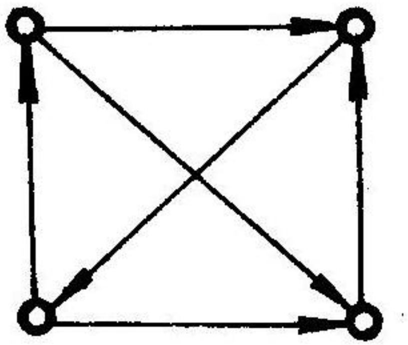
(a)



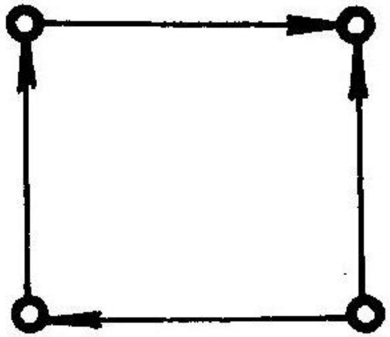
(b)

Connectedness

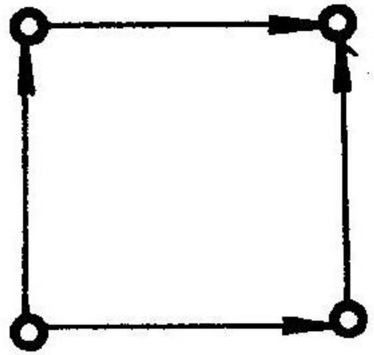
- Definition:
- A directed graph is strongly connected if there is a path from a to b and from b to a whenever a and b are vertices in the graph.
- A directed graph is weakly connected if there is a path between every two vertices in the underlying undirected graph.
- G is strongly connected if and only if there is a circuit passing through every vertex.



(a)



(b)



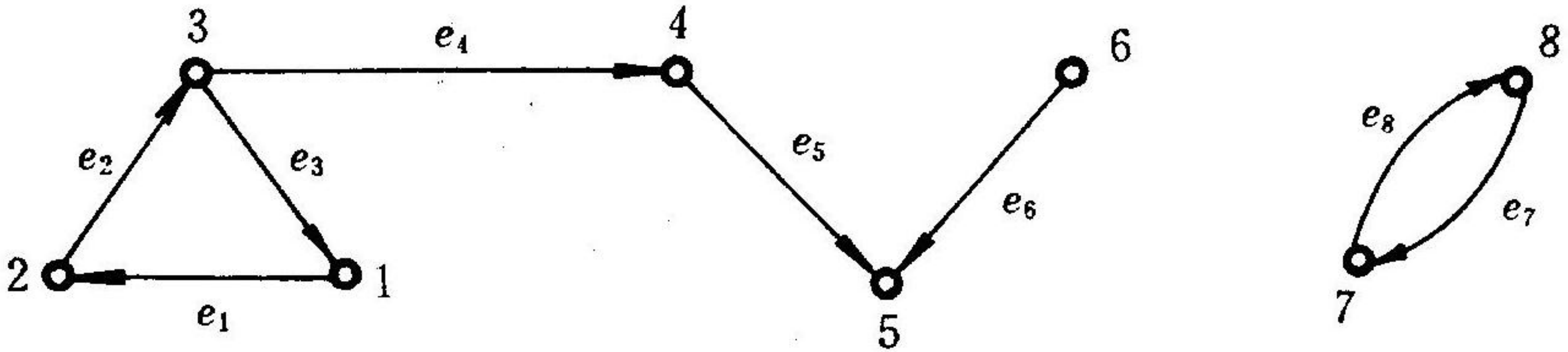
(c)

Strong components

- The subgraph of a directed graph G that are strongly connected but not contained in larger strongly connected subgraphs, that is, the maximal strongly connected subgraphs, are called the strongly connected components or strong components of G.

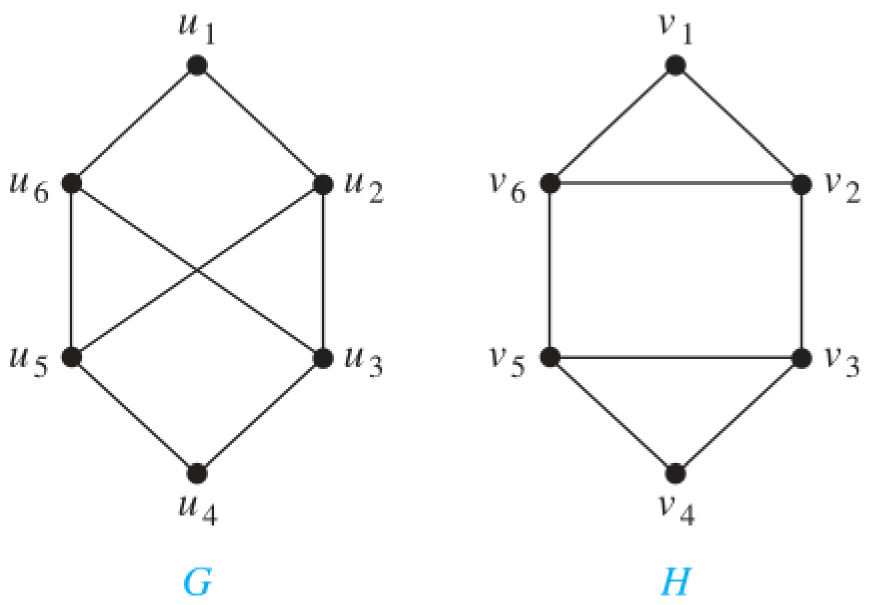
Strong components

- $\{ \langle \{1,2,3\},\{e_1,e_2,e_3\} \rangle, \langle \{4\},\varphi \rangle, \langle \{5\},\varphi \rangle, \langle \{6\},\varphi \rangle, \langle \{7,8\},\{e_7,e_8\} \rangle \}$



Path and Isomorphism

- Path can be used to determine whether two graphs are isomorphic or not
- the existence of a simple circuit of length k



H has a simple circuit of length three, namely, v1, v2, v6, v1, whereas G has no simple circuit of length three

Counting paths

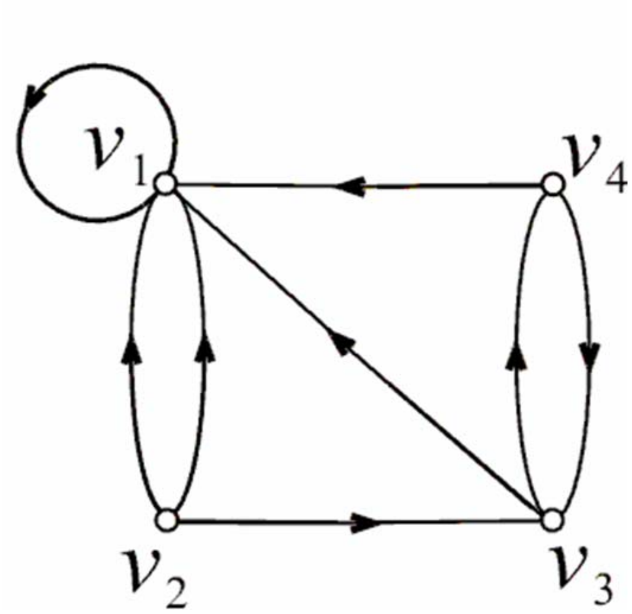
Theorem   Let G be a graph with adjacency matrix A with respect to the ordering v1, v2, … , vn of the vertices of the graph (with directed or undirected edges, with multiple edges and loops al-lowed). The number of different paths of length r from vi to vj, where r is a positive integer,equals the (i, j)th entry of A^r.

$$b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj},$$

Example

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \qquad A^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \\ 3 & 0 & 1 & 0 \end{bmatrix} \qquad A^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 5 & 0 & 0 & 1 \\ 4 & 0 & 1 & 0 \\ 4 & 0 & 0 & 1 \end{bmatrix}$$



The number of different paths of length 1 is 8, the circuit is 1.

The number of different paths of length 2 is 11, the circuit is 3.

The number of different paths of length 3 is 14, the circuit is 1.

The number of different paths of length 4 is 17, the circuit is 3.

Shortest-path problems

- In many applications, each edge of a graph has an associated numerical value, called a weight. The weight of an edge is often referred to as the "cost" of the edge.
- In applications, the weight may be a measure of distance, time, cost etc.
- Usually, the edge weights are nonnegative integers.

Shortest Paths

- shortest-path: Given two vertices A and B, there are more than one paths from A to B. The path with minimum cost is called shortest path.
- There are several different algorithms to find a shortest path.

Dijkstra's Algorithm

- We assume that there is a path from the source vertex v0 to every other vertex in the graph.
- Let S be the set of vertices whose minimum distance from the source vertex has been found. Initially S contains only the source vertex.
- The algorithm is iterative, adding one vertex to S at each step.
- We maintain an array D such that for each vertex v, D[v] is the minimum distance from the source vertex to v via vertices that are already in S.
- Every subpath is the shortest path in this whole shortest path.

- STEPS:

1. let S={$\mathbf{v_0}$}, compute D[i] for each vertex vi as following:

$$D[i] \begin{cases} 0 & \text{if } i=0 \\ w_{0i} & \text{if } i\neq 0\text{, and } \langle v_0, v_i\rangle \text{ is an edge, } w_{si} \text{ is the weight} \\ \infty & \text{if } i\neq 0\text{, and } \langle v_0, v_i\rangle \text{ is not an edge} \end{cases}$$

2. choose the vertex v_j such that:

   D[j] = min { D[k] | $v_k \notin$ S }

   then the v_j is starting of the next shortest path and D[j] is its cost.

STEPS:

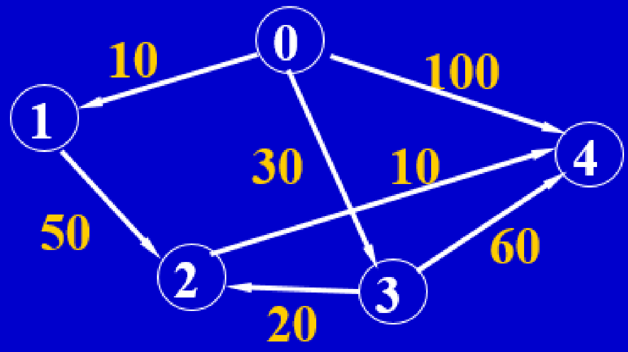
3. Place $\mathbf{v_j}$ in S. That is

$$S = S \cup \{\mathbf{v_j}\}$$

4. For each $\mathbf{v_k} \notin S$, modify the D[k]:

$$D[k] = \min \{\ D[k],\ \ D[j]+weight(<vj,vk>)\}$$

5. Repeat 2---4 until all vertices have been added in S.



Steps	S	D[0]	D[1]	D[2]	D[3]	D[4]
begin	{ 0 }	0	10	∞	30	100
1	{ 0, 1 }	0	10	60	30	100
2	{ 0, 1, 3 }	0	10	50	30	90
3	{ 0, 1, 3, 2 }	0	10	50	30	60
4	{ 0, 1, 3, 2, 4 }	0	10	50	30	60

homework

- 10.4 P724

  3, 4, 5,11, 14(a)(b),19(b)(c), 20,
- 10.6 p751

  2, 17