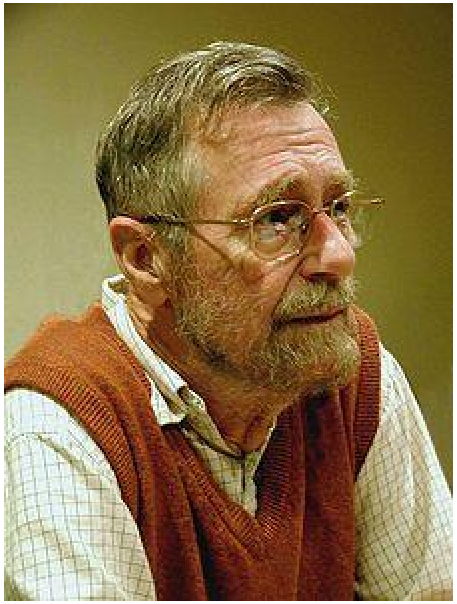
Chapter 8 Graph Theory 图论

Lu Han

hl@bupt.edu.cn

8.4 A Shortest-Path Algorithm 最短路径算法

Due to **Edsger W. Dijkstra (艾兹格·迪科斯彻)**, Dutch computer scientist born in 1930.



Edsger W. Dijkstra (1930–2002) was born in The Netherlands. He was an early proponent of programming as a science. So dedicated to programming was he that when he was married in 1957, he listed his profession as a programmer. However, the Dutch authorities said that there was no such profession, and he had to change the entry to “theoretical physicist.” He won the prestigious Turing Award in 1972.

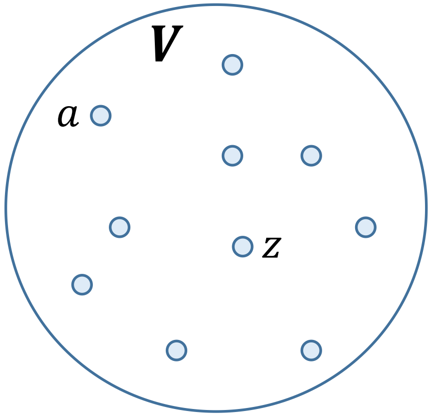
Dijkstra's algorithm (狄克斯特拉算法) finds the length of the shortest path from a single vertex to any other vertex in a connected weighted graph.

Dijkstra's algorithm (狄克斯特拉算法)

The given graph G is **connected, weighted graph**. Assume that **the weights are positive numbers**. We want to find a shortest path from vertex a to vertex z.

Dijkstra's algorithm (狄克斯特拉算法)

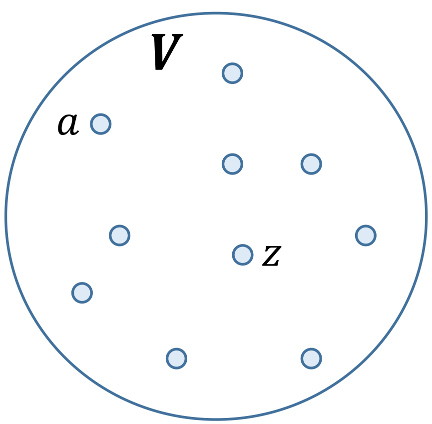
The given graph G is **connected, weighted graph**. Assume that **the weights are positive numbers**. We want to find a shortest path from vertex a to vertex z.



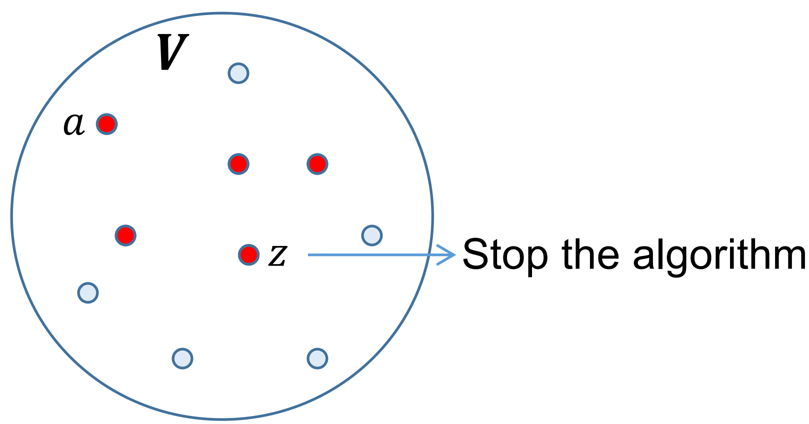
Initialization. Temporarily labelled each vertex $v \in V$ with a value $L(v)$.

Dijkstra's algorithm (狄克斯特拉算法)

The given graph G is **connected, weighted graph**. Assume that **the weights are positive numbers**. We want to find a shortest path from vertex a to vertex z.



Initialization. Temporarily labelled each vertex $v \in V$ with a value $L(v)$.



Each iteration changes the status of one temporairly labelled vertex from temporary to permanent. Update the label of some related vertices.

Dijkstra's algorithm (狄克斯特拉算法)

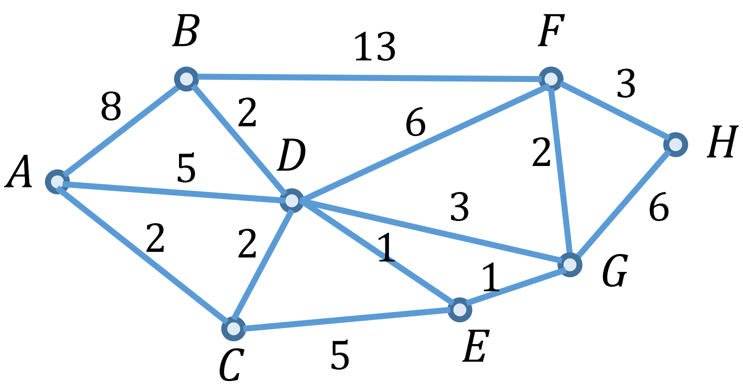
Input: A connected, weighted graph in which all weights are positive; vertices a and z.

Output: $L(z)$, the length of a shortest path from a to z.

procedure $dijkstra(w, a, z, L)$
  $L(a) := 0$
  for each node $x \neq a$ **do**
  $L(x) := \infty$
 T:= set of all nodes
 //T is the set of vertices whose shortest
 // distance from a has not been found
 while $z \in T$ **do**
   chose $v \in T$ with minimum $L(v)$
   $T := T - \{v\}$
   for each $x \in T$ adjacent to v **do**
    $L(x) := min\{L(x),\ L(v) + w(v, x)\}$
  end
 end dijkstra

Dijkstra's algorithm (狄克斯特拉算法)

Exercise Which vertex has the largest shortest path starting from vertex A?

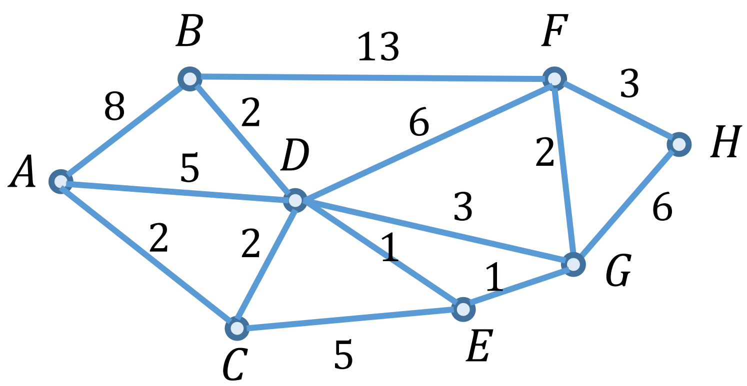



```
procedure dijkstra(w, a, z, L)
    L(a): = 0
    for each node x≠a do
    L(x): = ∞
  T:= set of all nodes
  //T is the set of vertices whose shortest
  // distance from a has not been found
  while z ∈ T do
      choose v ∈ T with minimum L(v)
      T: = T − {v}
      for each x ∈ T adjacent to v do
         L(x): = min {L(x), L(v) + w(v, x)}
     end
  end dijkstra
```

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								

Dijkstra's algorithm (狄克斯特拉算法)

Exercise Which vertex has the largest shortest path starting from vertex A?




```
procedure dijkstra(w, a, z, L)
    L(a) := 0
    for each node x≠a do
    L(x) := ∞
  T:= set of all nodes
  //T is the set of vertices whose shortest
  // distance from a has not been found
  while z ∈ T do
      choose v ∈ T with minimum L(v)
      T: = T − {v}
      for each x ∈ T adjacent to v do
         L(x): = min {L(x), L(v) + w(v,x)}
      end
  end dijkstra
```

	A	B	C	D	E	F	G	H
1	**0**	∞	∞	∞	∞	∞	∞	∞
2		8_A	$\mathbf{2_A}$	5_A	∞	∞	∞	∞
3		8_A		$\mathbf{4_C}$	7_c	∞	∞	∞
4		6_D			$\mathbf{5_D}$	10_D	7_D	∞
5		$\mathbf{6_D}$				10_D	6_E	∞
6						10_D	$\mathbf{6_E}$	∞
7						$\mathbf{8_G}$		12_G
8								$\mathbf{11_F}$

8.4 A Shortest-Path Algorithm 最短路径算法

Dijkstra's Shortest-Path Algorithm

Theorem 8.4.3 Dijkstra's shortest-path algorithm correctly finds the length of a shortest path form a to z.

8.4 A Shortest-Path Algorithm 最短路径算法

Dijkstra’s Shorest-Path Algorithm

Theorem 8.4.3 Dijkstra’s shortest-path algorithm correctly finds the length of a shortest path form a to z.

In addition to circle a vertex, we will also label it with the name of the vertex from which it was labeled.

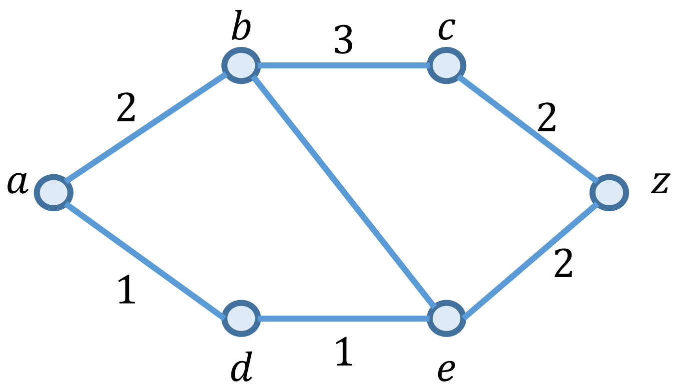
8.4 A Shortest-Path Algorithm 最短路径算法

Dijkstra’s Shortest-Path Algorithm

Theorem 8.4.3 Dijkstra’s shortest-path algorithm correctly finds the length of a shortest path form a to z.

In addition to circle a vertex, we will also label it with the name of the vertex from which it was labeled.

Example 8.4.4



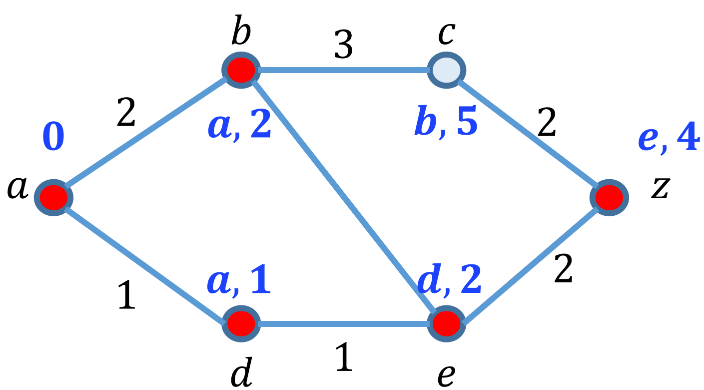
8.4 A Shortest-Path Algorithm 最短路径算法

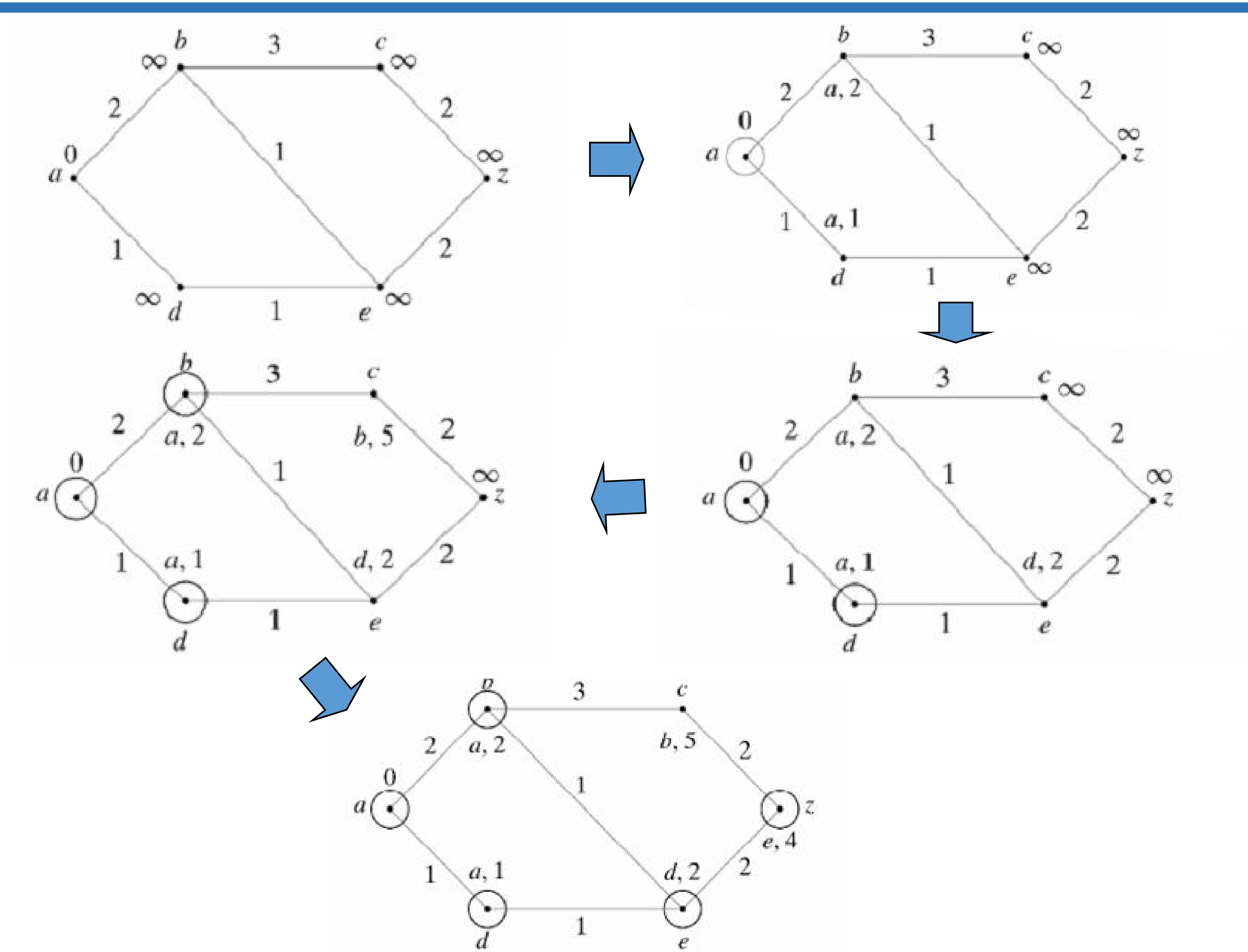
Dijkstra's Shorest-Path Algorithm

Theorem 8.4.3 Dijkstra's shorest-path algorithm correctly finds the length of a shortest path form a to z.

In addition to circling a vertex, we will also label it with the name of the vertex from which it was labeled.

Example 8.4.4




a b c d e z
0 ∞ ∞ ∞ ∞ ∞
2 3 2 1 1 1 2 2
a, 2
a, 1
b, 5
d, 2
e, 4

8.4 A Shortest-Path Algorithm 最短路径算法

Dijkstra's Shorest-Path Algorithm

Theorem 8.4.3 Dijkstra's shorest-path algorithm correctly finds the length of a shortest path form a to z.

Let P be a shortest path from a to z.

We want to prove that

(i) $L(z) \geq$ length of P

(ii) $L(z) \leq$ length of P

8.4 A Shortest-Path Algorithm 最短路径算法

Dijkstra's Shortest-Path Algorithm

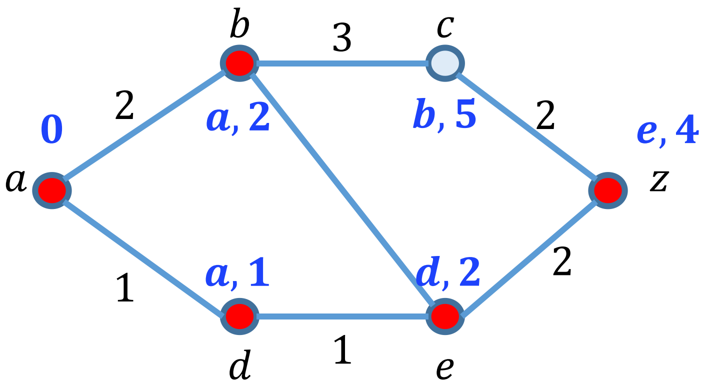
Theorem 8.4.3 Dijkstra's shorest-path algorithm correctly finds the length of a shortest path form a to z.

Let P be a shortest path from a to z.

We want to prove that

(i) $L(z) \geq$ length of P

(ii) $L(z) \leq$ length of P



8.4 A Shortest-Path Algorithm 最短路径算法

Dijkstra's Shorest-Path Algorithm

Theorem 8.4.3 Dijkstra's shorest-path algorithm correctly finds the length of a shortest path form a to z.

Let P be a shortest path from a to z.

We want to prove that

(i) $L(z) \geq$ length of P

(ii) $L(z) \leq$ length of P

8.4 A Shortest-Path Algorithm 最短路径算法

Dijkstra's Shorest-Path Algorithm

Theorem 8.4.3 Dijkstra's shorest-path algorithm correctly finds the length of a shortest path form a to z.

Let P be a shortest path from a to z. (ii) $L(z) \leq$ length of P

Proof We use mathematical induction on i to prove that the ith time we choose a vertex v with minimum $L(v)$, $L(v)$ is the length of a shortest path from a to v.

8.4 A Shortest-Path Algorithm 最短路径算法

Dijkstra's Shorest-Path Algorithm

Theorem 8.4.3 Dijkstra's shorest-path algorithm correctly finds the length of a shortest path form a to z.

Let P be a shortest path from a to z. (ii) $L(z) \leq$ length of P

Proof We use mathematical induction on i to prove that the ith time we choose a vertex v with minimum $L(v)$, $L(v)$ is the length of a shortest path from a to v.

Basis Step ($i = 1$)

Inductive Step ($k < i$)
If there is a path from a to w whose length is less than $L(v)$, then w is not in T.

8.4 A Shortest-Path Algorithm 最短路径算法

Modify Dijkstra's shortest-path algorithm so that it accepts a weighted graph that is not necessarily connected. At termination, what is $L(z)$ if there is no path from a to z?

8.4 A Shortest-Path Algorithm 最短路径算法

True or false? When a connected, weighted graph and vertices a and z are input to the following algorithm, it returns the length of a shortest path from a to z.

Algorithm 8.4.6

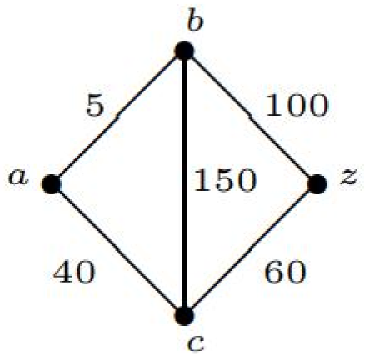
```
algor(w, a, z) {
    length = 0
    v = a
    T = set of all vertices
    while (v¬ = z) {
        T = T − {v}
        choose x ∈ T with minimum w(v, x)
        length = length + w(v, x)
        v = x
    }
    return length
}
```

8.4 A Shortest-Path Algorithm 最短路径算法

True or false? When a connected, weighted graph and vertices a and z are input to the following algorithm, it returns the length of a shortest path from a to z.

Algorithm 8.4.6

```
algor(w, a, z) {
    length = 0
    v = a
    T = set of all vertices
    while (v¬ = z) {
        T = T − {v}
        choose x ∈ T with minimum w(v, x)
        length = length + w(v, x)
        v = x
    }
    return length
}
```

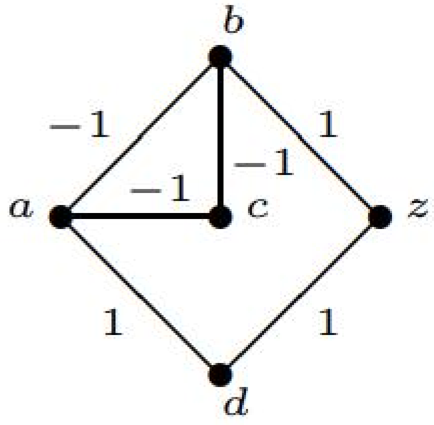


8.4 A Shortest-Path Algorithm 最短路径算法

True or false? Dijkstra's shortest-path algorithm finds the length of a shortest path in a connected, weighted graph even if some weights are negative. If true, prove it; otherwise, provide a counterexample.

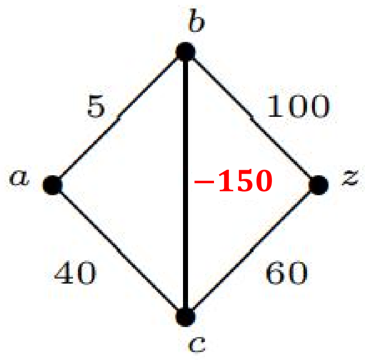
8.4 A Shortest-Path Algorithm 最短路径算法

True or false? Dijkstra's shortest-path algorithm finds the length of a shortest path in a connected, weighted graph even if some weights are negative. If true, prove it; otherwise, provide a counterexample.



8.4 A Shortest-Path Algorithm 最短路径算法

True or false? Dijkstra’s shortest-path algorithm finds the length of a shortest path in a connected, weighted graph even if some weights are negative. If true, prove it; otherwise, provide a counterexample.



8.5 Representations of Graphs 图的表示

adjacent & incident？

8.5 Representations of Graphs 图的表示

adjacent & incident？

- Two vertices are called **adjacent** if they are connected by an edge.
- A vertex and an edge are called **incident**, if the vertex is one of the two vertices the edge connects.

8.5 Representations of Graphs 图的表示

Adjacency Matrix (邻接矩阵)

adjacent & incident?

- Two vertices are called **adjacent** if they are connected by an edge.
- A vertex and an edge are called **incident**, if the vertex is one of the two vertices the edge connects.

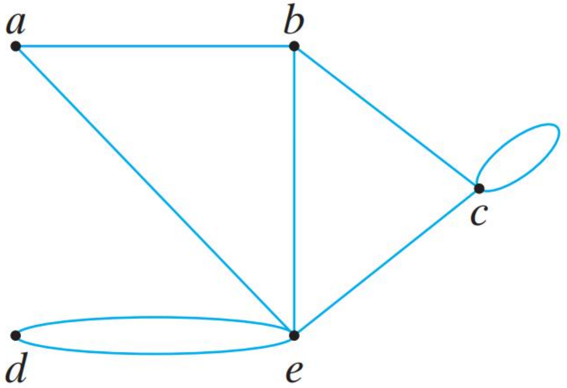
8.5 Representations of Graphs 图的表示

Adjacency Matrix (邻接矩阵)

- Select an ordering of the vertices.
- Label the rows and columns of a matrix with the ordered vertices.
- The entry in this matrix in row i, column j:

  if $i \neq j$, is the number of edges incident on i and j;

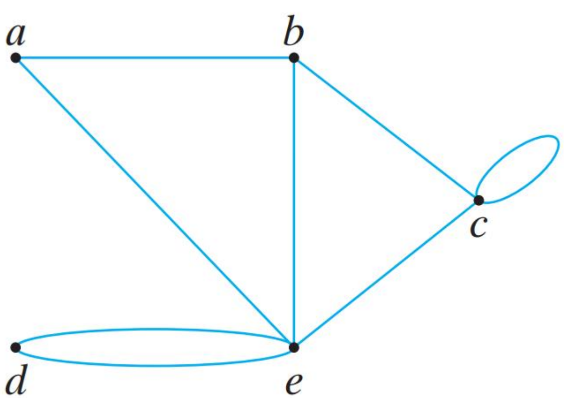
  if $i = j$, is twice the number of loops incident on i.



8.5 Representations of Graphs 图的表示

Adjacency Matrix (邻接矩阵)

- Select an ordering of the vertices.
- Label the rows and columns of a matrix with the ordered vertices.
- The entry in this matrix in row i, column j:
  if $i \neq j$, is the number of edges incident on i and j;
  if $i = j$, is twice the number of loops incident on i.



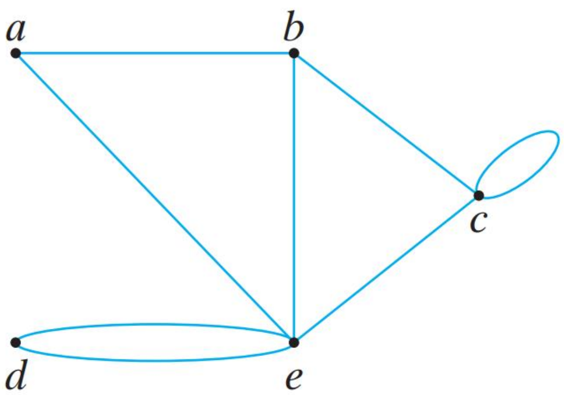
$$\begin{array}{c c} & \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 \\ 1 & 1 & 1 & 2 & 0 \end{pmatrix} \end{array}$$

The degree of a vertex v in a graph G can be obtained by
...

8.5 Representations of Graphs 图的表示

Adjacency Matrix (邻接矩阵)

- Select an ordering of the vertices.
- Label the rows and columns of a matrix with the ordered vertices.
- The entry in this matrix in row i, column j:
  if $i \neq j$, is the number of edges incident on i and j;
  if $i = j$, is twice the number of loops incident on i.



$$
\begin{array}{c c}
 & \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 \\ 1 & 1 & 1 & 2 & 0 \end{pmatrix}
\end{array}
$$

The degree of a vertex v in a graph G can be obtained by summing row v or column v in G's adjacency matrix.

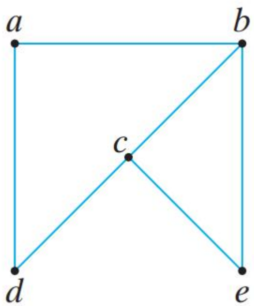
8.5 Representations of Graphs 图的表示

Adjacency Matrix (邻接矩阵)

If A is the adjacency matrix of a simple graph, the ijth entry of A^2 is equal to the number of paths of length 2 from vertex i to vertex j.

8.5 Representations of Graphs 图的表示

Adjacency Matrix (邻接矩阵)



$$A^2 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 2 & 0 & 2 & 0 & 1 \\ 0 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 0 & 1 \\ 0 & 2 & 0 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 \end{pmatrix} \end{array}$$

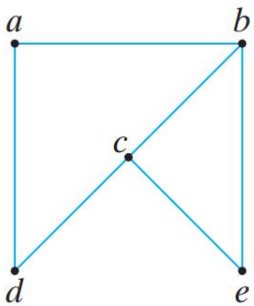
If A is the adjacency matrix of a simple graph, the ijth entry of A^2 is equal to the number of paths of length 2 from vertex i to vertex j.

8.5 Representations of Graphs 图的表示

Adjacency Matrix (邻接矩阵)

True or False?

The entry on the main diagonal of A^2 give the degrees of the vertices.



$$A^2 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 2 & 0 & 2 & 0 & 1 \\ 0 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 0 & 1 \\ 0 & 2 & 0 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 \end{pmatrix} \end{array}$$

If A is the adjacency matrix of a simple graph, the ijth entry of A^2 is equal to the number of paths of length 2 from vertex i to vertex j.

8.5 Representations of Graphs 图的表示

Adjacency Matrix (邻接矩阵)

True or False?

The entry on the main diagonal of A^2 give the degrees of the vertices (when the graph is a simple graph).

If A is the adjacency matrix of a simple graph, the ijth entry of A^2 is equal to the number of paths of length 2 from vertex i to vertex j.

8.5 Representations of Graphs 图的表示

Incidence Matrix (关联矩阵)

adjacent & incident？

- Two vertices are called **adjacent** if they are connected by an edge.
- A vertex and an edge are called **incident**, if the vertex is one of the two vertices the edge connects.

8.5 Representations of Graphs 图的表示

Incidence Matrix (关联矩阵)

- Label the rows with the vertices. Label the columns with the edges.
- If e is incident on v, the entry for row v and column e is 1, otherwise it is 0.

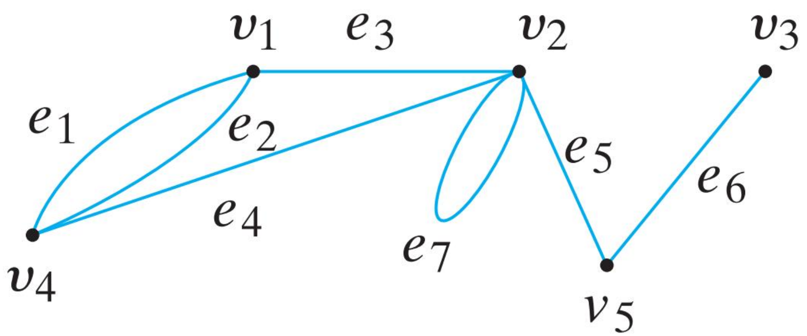
adjacent & incident？

- Two vertices are called **adjacent** if they are connected by an edge.
- A vertex and an edge are called **incident**, if the vertex is one of the two vertices the edge connects.

8.5 Representations of Graphs 图的表示

Incidence Matrix (关联矩阵)

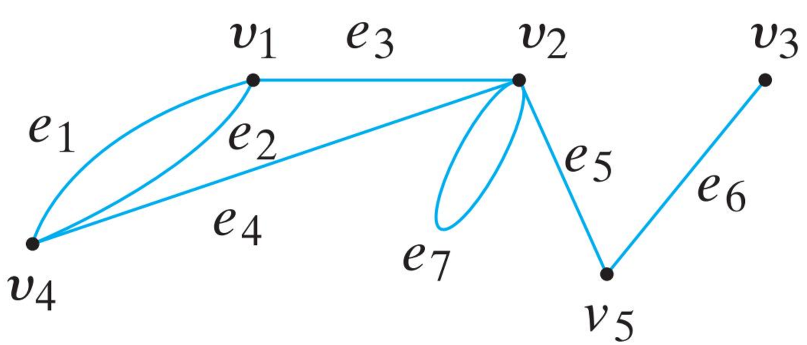
- Label the rows with the vertices. Label the columns with the edges.
- If e is incident on v, the entry for row v and column e is 1, otherwise it is 0.



8.5 Representations of Graphs 图的表示

Incidence Matrix (关联矩阵)

- Label the rows with the vertices. Label the columns with the edges.
- If e is incident on v, the entry for row v and column e is 1, otherwise it is 0.



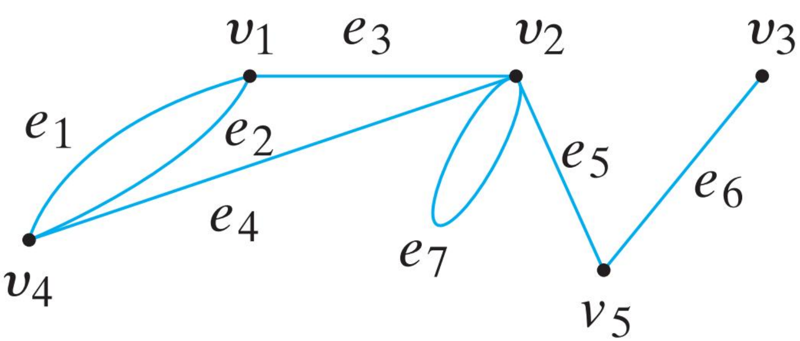
$$
\begin{array}{c c}
 & \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} &
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}
\end{array}
$$

True or False? In a graph
each column has two 1’s.
and the sum of a row gives the degree of the vertex identified with that row.

8.5 Representations of Graphs 图的表示

Incidence Matrix (关联矩阵)

- Label the rows with the vertices. Label the columns with the edges.
- If e is incident on v, the entry for row v and column e is 1, otherwise it is 0.

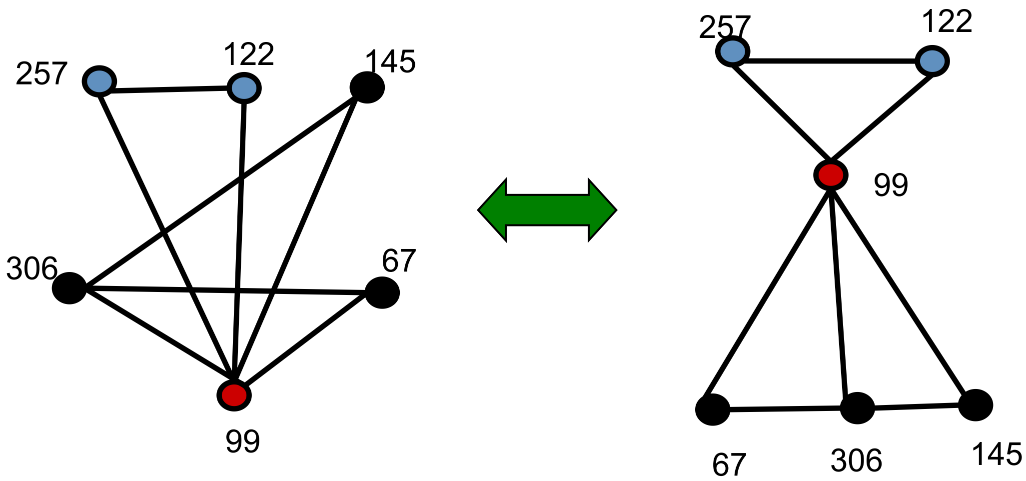


$$
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\begin{array}{c}
\begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}
\end{array}
$$

True or False? In a graph without loops
each column has two 1's.
and the sum of a row gives the degree of the vertex identified with that row.

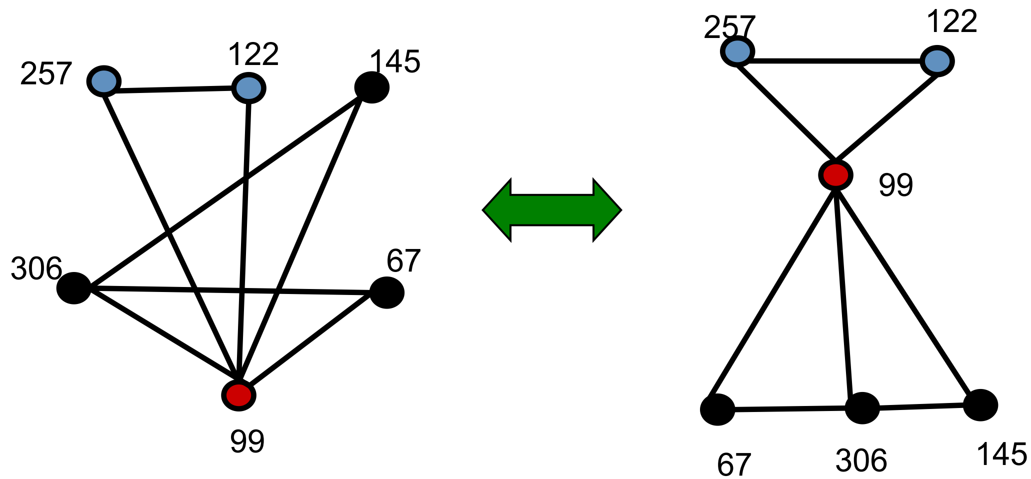
8.6 Isomorphisms of Graphs 图的同构

Same graph?



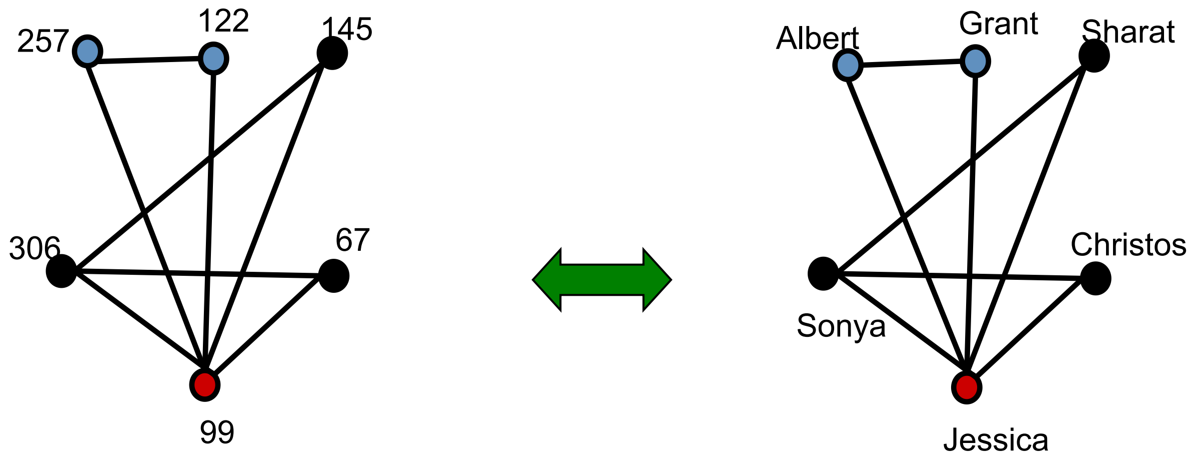
8.6 Isomorphisms of Graphs 图的同构

Same graph
(different *drawings*)



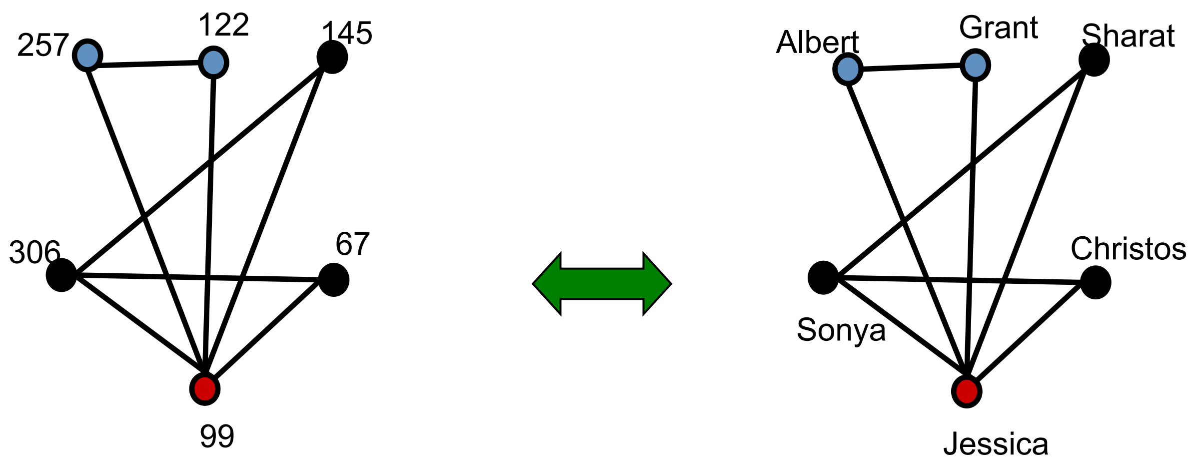
8.6 Isomorphisms of Graphs 图的同构

Same graph?



8.6 Isomorphisms of Graphs 图的同构

Same graph
(different *labels*)



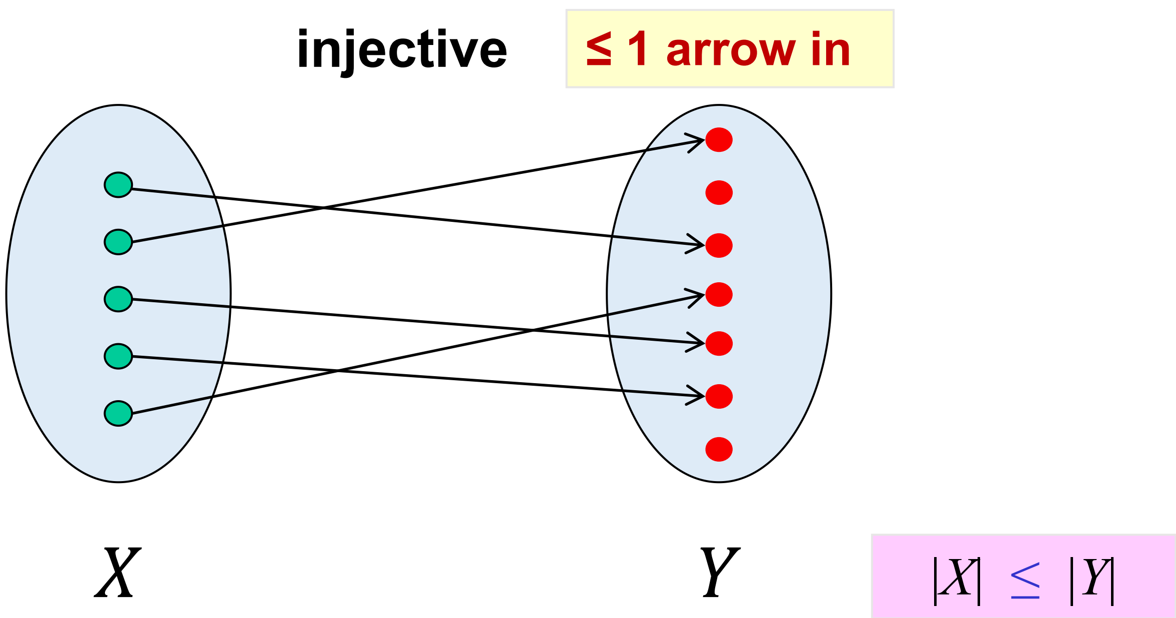
8.6 Isomorphisms of Graphs 图的同构

Definition 8.6.1 G_1 and G_2 are **isomorphic (同构的)** if there exist a one-to-one, onto functions f from the vertices of G_1 to the vertices of G_2 and a one-to-one, onto function g from the edges of G_1 to the edges of G_2, so that an edge e is incident on v and w in G_1 if and only if the edge $g(e)$ is incident on $f(v)$ and $f(w)$ in G_2. The pair of functions f and g is called an **isomorphism of G_1 onto G_2 (G_1到G_2上的同构映射)**.

one-to-one, onto function？

3.1 Functions 函数

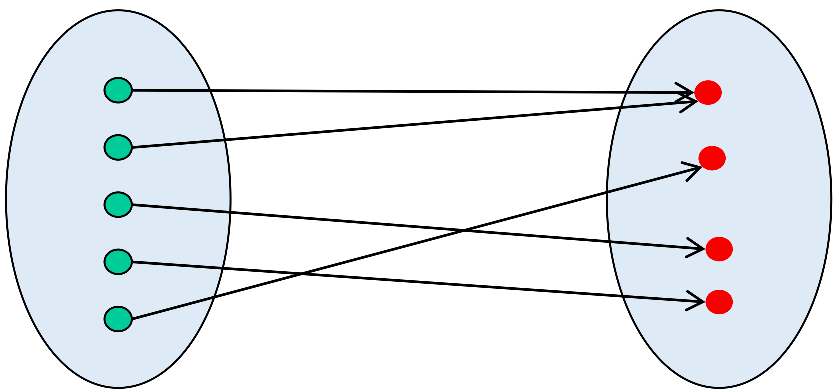
Definition 3.1.21 A function f from X to Y is said to be **one-to-one (or injective) (单射的)** if **for all $x_1, x_2 \in X$, if $f(x_1) = f(x_2)$ then $x_1 = x_2$**.



3.1 Functions 函数

Definition 3.1.29 A function f from X to Y is said to be **onto** Y **(or surjective) (满射的)** if **for every $y \in Y$, there exists $x \in X$ such that $f(x) = y$**.

surjective **$\geq$ 1 arrow in**



X Y

$|X| \geq |Y|$

8.6 Isomorphisms of Graphs 图的同构

Definition 8.6.1 G_1 and G_2 are **isomorphic (同构的)** if there exist a one-to-one, onto functions f from the vertices of G_1 to the vertices of G_2 and a one-to-one, onto function g from the edges of G_1 to the edges of G_2, so that an edge e is incident on v and w in G_1 if and only if the edge $g(e)$ is incident on $f(v)$ and $f(w)$ in G_2. The pair of functions f and g is called an **isomorphism of G_1 onto G_2 (G_1到G_2上的同构映射)**.

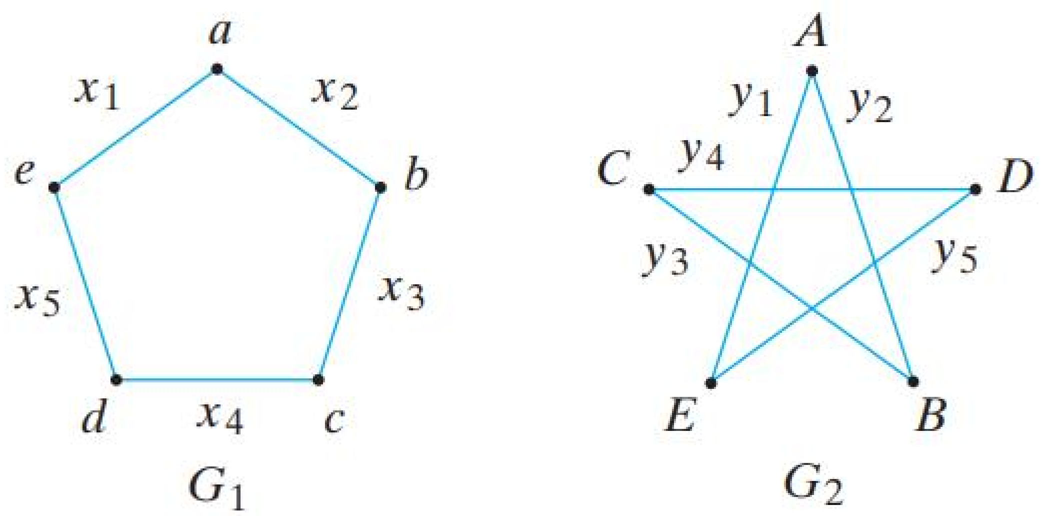
one-to-one, onto function？

Definition 3.1.21 A function f from X to Y is said to be **one-to-one (or injective) (单射的)** if **for all $x_1, x_2 \in X$, if $f(x_1) = f(x_2)$ then $x_1 = x_2$**.

Definition 3.1.29 A function f from X to Y is said to be **onto Y (or surjective) (满射的)** if **for every $y \in Y$, there exists $x \in X$ such that $f(x) = y$**.

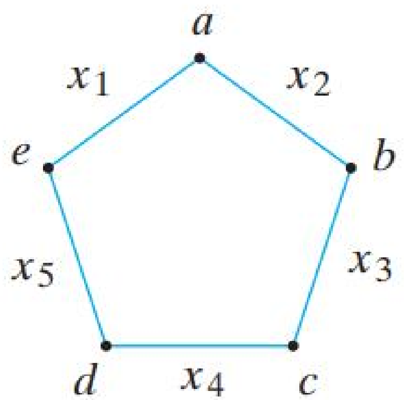
8.6 Isomorphisms of Graphs 图的同构

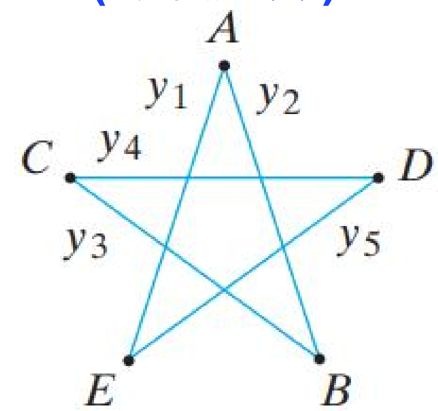
Definition 8.6.1 G_1 and G_2 are **isomorphic (同构的)** if there exist a one-to-one, onto functions f from the vertices of G_1 to the vertices of G_2 and a one-to-one, onto function g from the edges of G_1 to the edges of G_2, so that an edge e is incident on v and w in G_1 if and only if the edge $g(e)$ is incident on $f(v)$ and $f(w)$ in G_2. The pair of functions f and g is called an **isomorphism of G_1 onto G_2 (G_1到G_2上的同构映射)**.



8.6 Isomorphisms of Graphs 图的同构

Theorem 8.6.4 Graphs G_1 and G_2 are isomorphic if and only if for some ordering of their vertices, their **adjacency matrices (邻接矩阵)** are equal.





$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

8.6 Isomorphisms of Graphs 图的同构

Theorem 8.6.4 Graphs G_1 and G_2 are isomorphic if and only if for some ordering of their vertices, their **adjacency matrices (邻接矩阵)** are equal.

↓

incidence matrices (关联矩阵)
for some ordering of their vertices and edges

8.6 Isomorphisms of Graphs 图的同构

Theorem 8.6.4 Graphs G_1 and G_2 are isomorphic if and only if for some ordering of their vertices, their **adjacency matrices (邻接矩阵)** are equal.

Corollary 8.6.5 Let G_1 and G_2 be **simple graphs**. The following are equivalent:
(a) G_1 and G_2 are isomorphic.
(b) There is a one-to-one, onto function f from the vertex set of G_1 to the vertex set of G_2 satisfying the following: Vertices v and w are adjacent in G_1 if and only if the vertices $f(v)$ and $f(w)$ are adjacent in G_2.

8.6 Isomorphisms of Graphs 图的同构

How to prove that two simple graphs Graphs G_1 and G_2 are not isomorphic?

Find a property of G_1 that G_2 does not have but that G_2 would have if G_1 and G_2 were isomorphic. Such a property is called an **invariant (不变量)**.

A property P is an invariant if whenever G_1 and G_2 are isomorphic graphs:

If G_1 has property P, G_2 also has property P.

8.6 Isomorphisms of Graphs 图的同构

How to prove that two simple graphs Graphs G_1 and G_2 are not isomorphic?

Find a property of G_1 that G_2 does not have but that G_2 would have if G_1 and G_2 were isomorphic. Such a property is called an **invariant (不变量)**.

A property P is an invariant if whenever G_1 and G_2 are isomorphic graphs:

If G_1 has property P, G_2 also has property P.

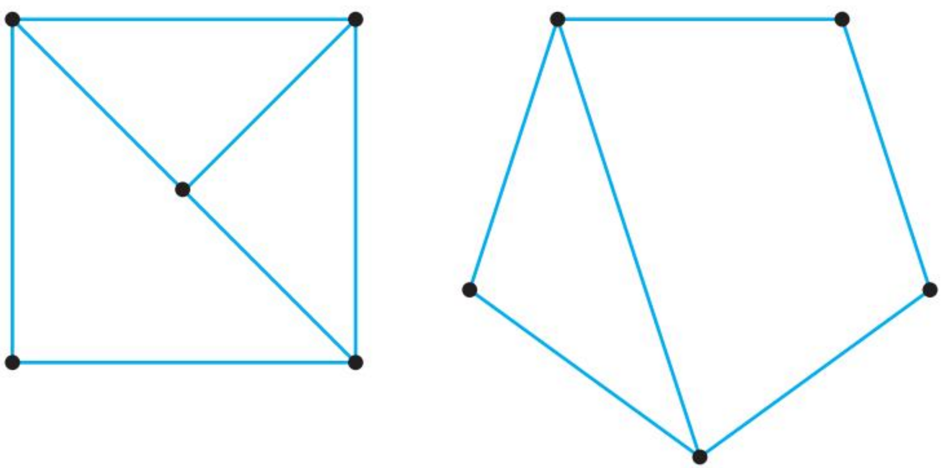
- If G_1 and G_2 are isomorphic, then G_1 and G_2 have the same number of edges and the same number of vertices.
- If k is a positive integer, "has a vertex of degree k" is an invariant.
- If l is a positive integer, "has a simple cycle of length l" is an invariant.

8.6 Isomorphisms of Graphs 图的同构

How to prove that two simple graphs Graphs G_1 and G_2 are not isomorphic?

Find a property of G_1 that G_2 does not have but that G_2 would have if G_1 and G_2 were isomorphic. Such a property is called an **invariant (不变量)**.

Example 8.6.7

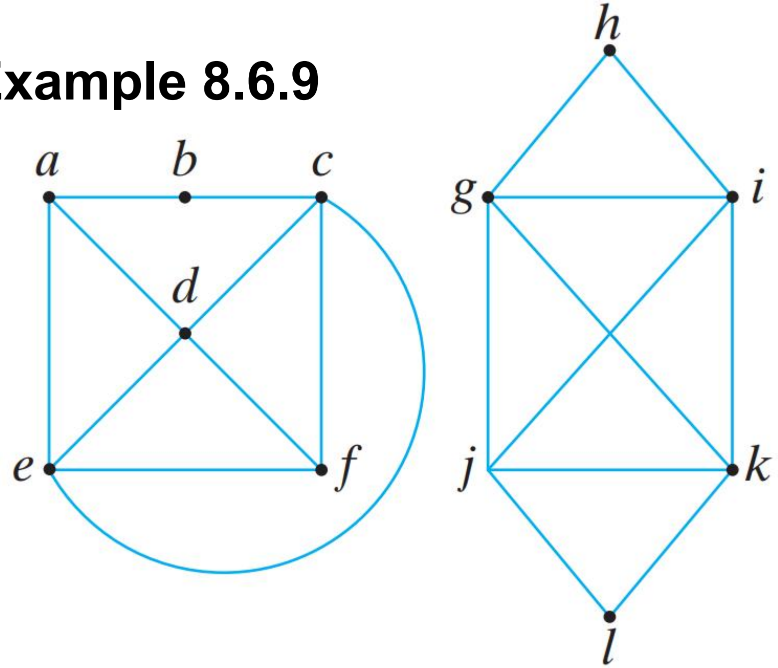


8.6 Isomorphisms of Graphs 图的同构

How to prove that two simple graphs Graphs G_1 and G_2 are not isomorphic?

Find a property of G_1 that G_2 does not have but that G_2 would have if G_1 and G_2 were isomorphic. Such a property is called an **invariant (不变量)**.

Example 8.6.9

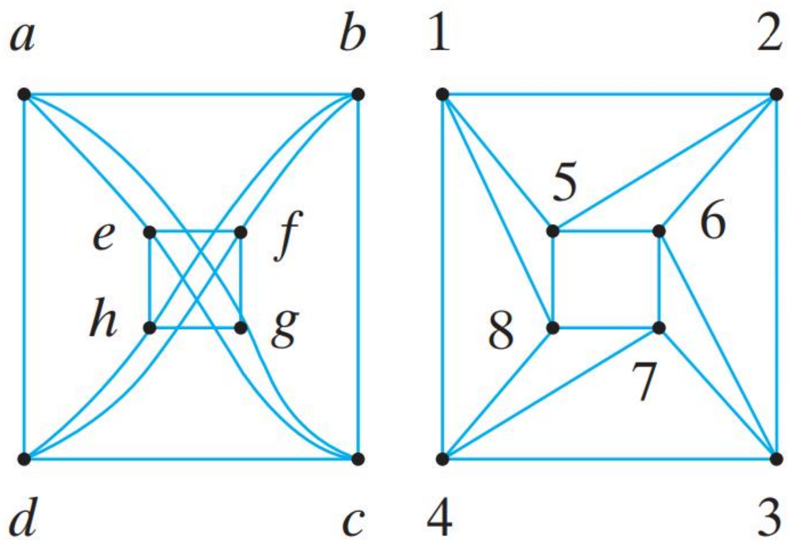


8.6 Isomorphisms of Graphs 图的同构

How to prove that two simple graphs Graphs G_1 and G_2 are not isomorphic?

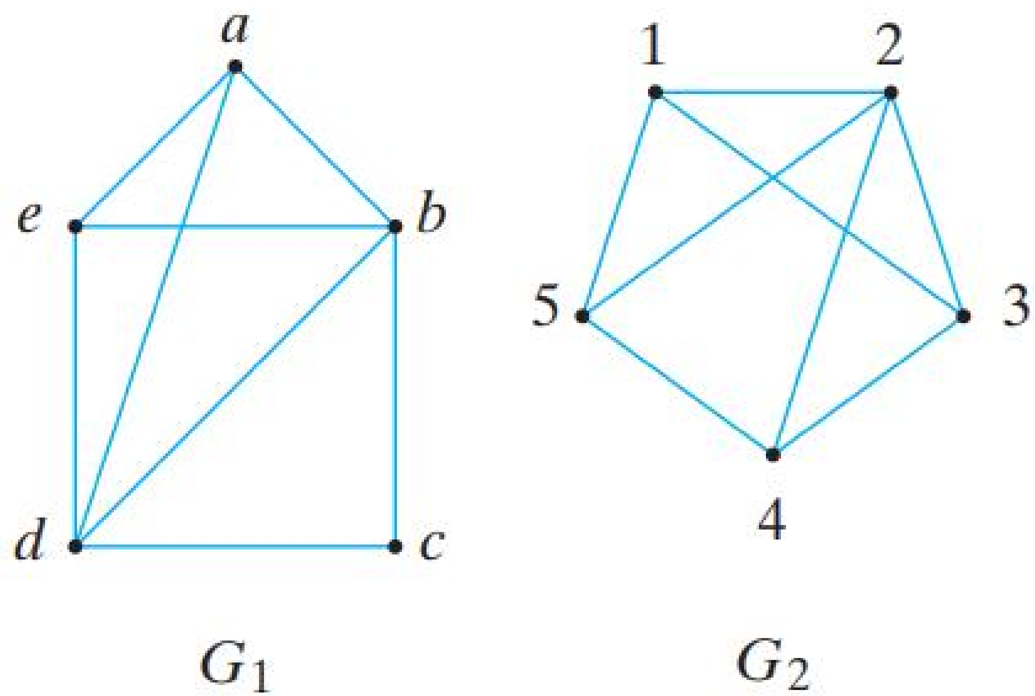
Find a property of G_1 that G_2 does not have but that G_2 would have if G_1 and G_2 were isomorphic. Such a property is called an **invariant (不变量)**.

Example 8.6.10



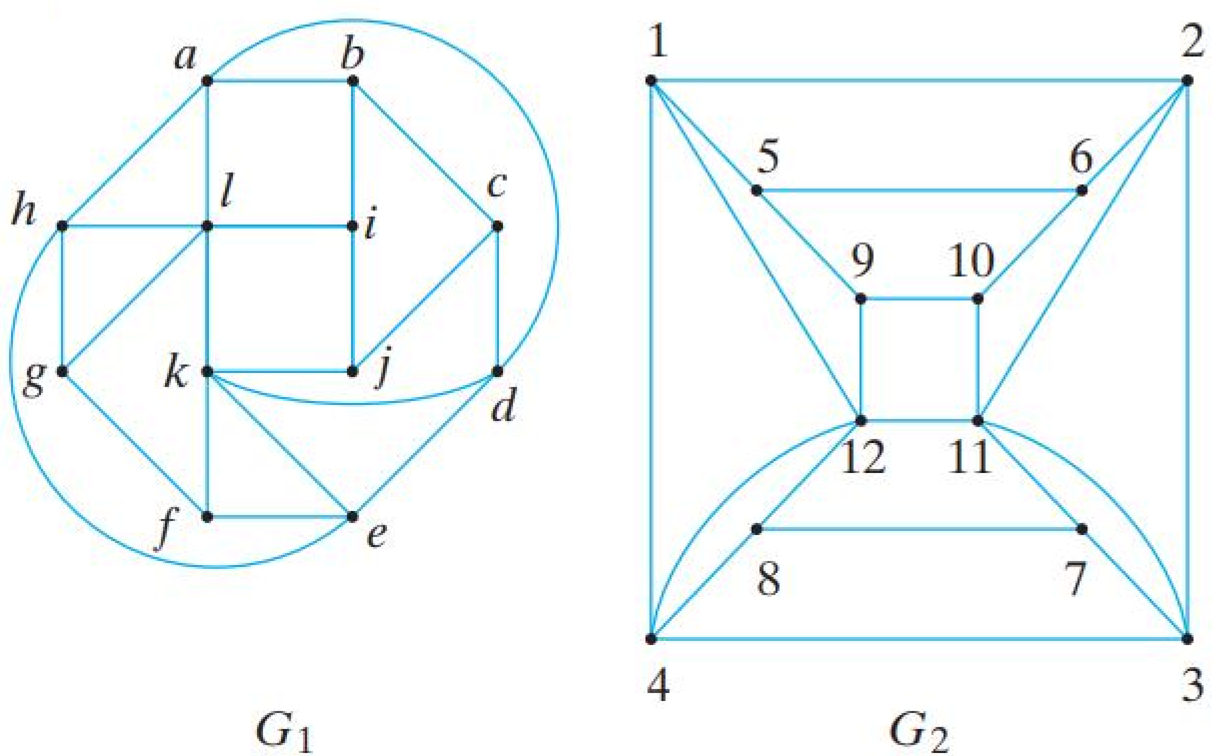
8.6 Isomorphisms of Graphs 图的同构

Determine whether the graphs G_1 and G_2 are isomorphic.



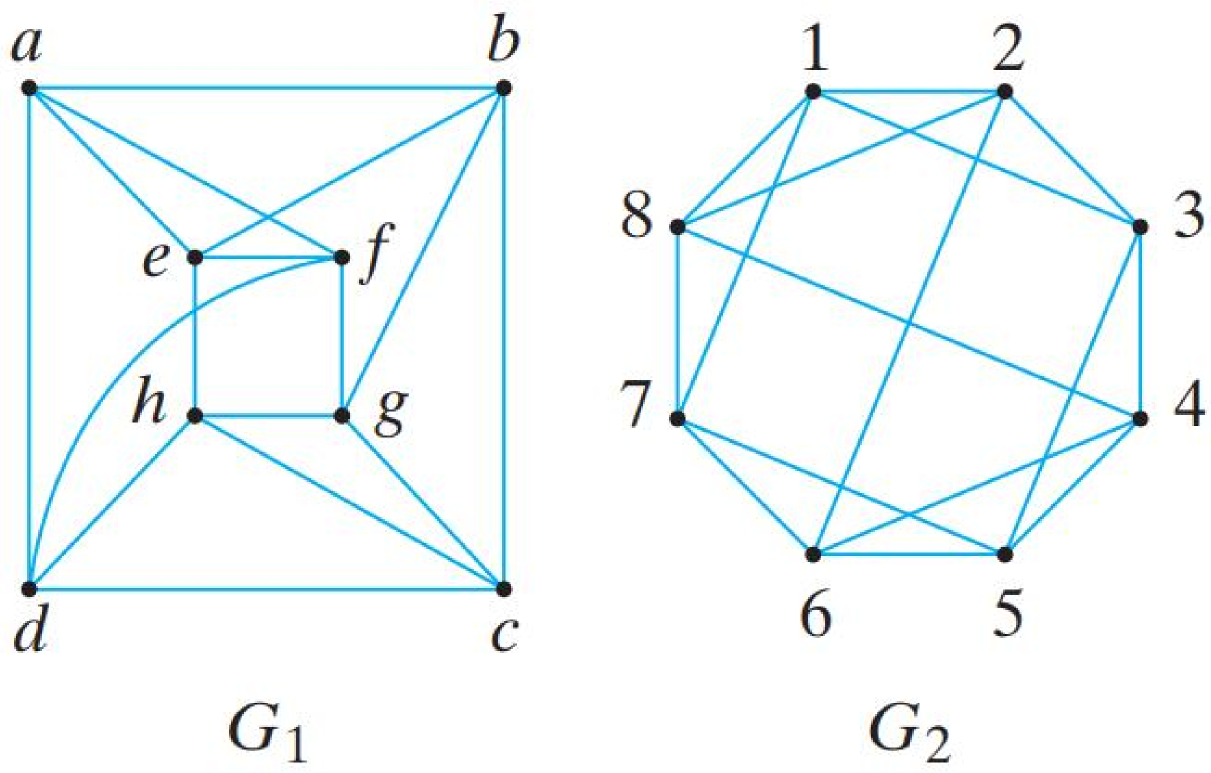
8.6 Isomorphisms of Graphs 图的同构

Determine whether the graphs G_1 and G_2 are isomorphic.



8.6 Isomorphisms of Graphs 图的同构

Determine whether the graphs G_1 and G_2 are isomorphic.



8.6 Isomorphisms of Graphs 图的同构

Determine whether the graphs G_1 and G_2 are isomorphic.