Chapter 7
GRAPHS

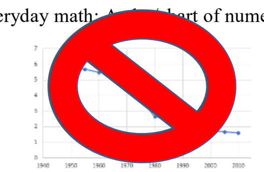
Topics

1. Graphs and Graph Models
2. Graph Terminology and Special Types of Graphs
3. Representing Graphs and Graph Isomorphism
4. Connectivity
5. Euler and Hamilton Paths
6. Shortest-Path Problems

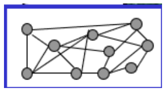
GRAPHS AND GRAPH MODELS

What are Graphs?

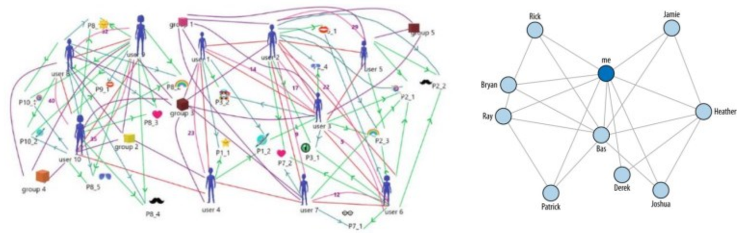
- General meaning in everyday math: A plot/chart of numerical data using a coordinate system.



- Technical meaning in discrete mathematics: A particular class of discrete structures (to be defined) that is useful for representing relations (edges) between objects (vertices) and has a convenient webby-looking graphical representation

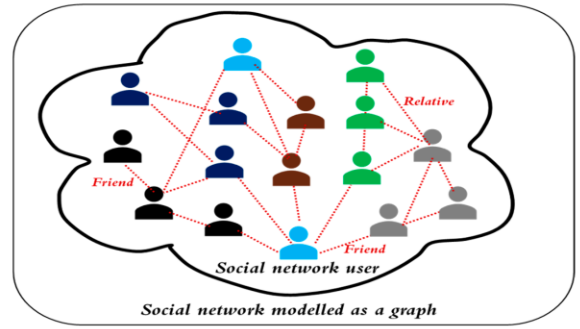


Examples of Graphs

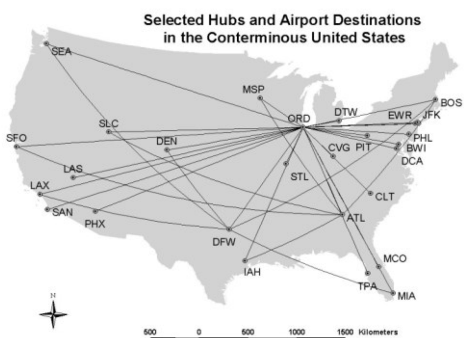


Facebook network model

Examples of Graphs



Examples of Graphs



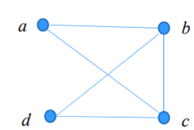
A graph showing the airline routes

Definition of a graph

Definition. A graph **G = (V, E)** consists of **V**, a nonempty set of **vertices** (or **nodes**) and **E**, a set of **edges**. Each edge has either one or two vertices associated with it, called its **endpoints**. An edge is said to connect its endpoints.

Remark. The set of vertices V of a graph G may be infinite. A graph with an infinite vertex set or an infinite number of edges is called an **infinite graph**, and in comparison, a graph with a finite vertex set and a finite edge set is called a **finite graph**.

Example.

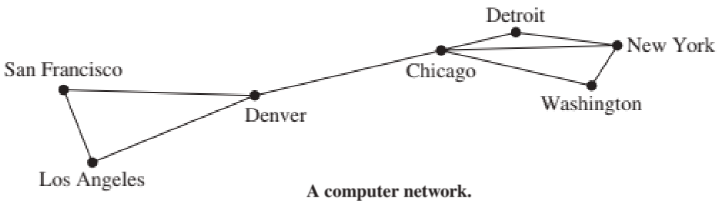


Simple Graphs

Definition. A graph in which each edge connects two different vertices and where **no two edges connect the same pair of vertices** is called a **simple graph**

Simple Graphs

Example. How many vertices? edges?



A computer network.

Simple Graphs

Example. How many vertices? edges?

Set of vertices

V =

Set of edges

E =

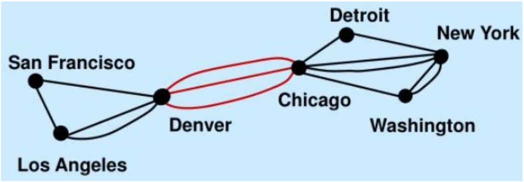
Non-simple Graphs

Definition 1. Graphs that have **2 or more edges connecting the same pair of vertices** are called **multigraphs**. When there are m different edges associated to the same unordered pair of vertices {u, v}, we also say that {u, v} is an edge of **multiplicity m**.

Definition 2. Graphs that **include loops**, and possibly 2 or more edges connecting the same pair of vertices, and in addition, an edge may connect a vertex to itself, are called **pseudographs**.

Multigraphs

Example.

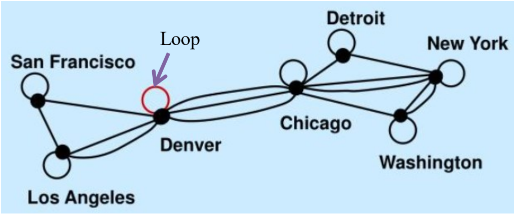


Parallel edges

A computer network with multiple links between data centers.

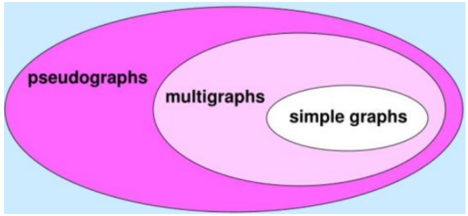
Pseudographs

Example.



A computer network with diagnostic links.

Undirected Graphs



Directed Graph

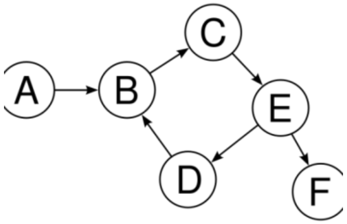
Definition. A **directed graph** (or **digraph**) (V, E) consists of a nonempty set of vertices V and a set of **directed edges** (or **arcs**) E. Each directed edge is associated with an ordered pair of vertices. The directed edge associated with the ordered pair (u, v) is said to start at u and end at v.

When a directed graph has **no loops** and has **no multiple directed edges**, it is called a **simple directed graph**. Because a simple directed graph has at most one edge associated to each ordered pair of vertices (u, v), we call **(u, v) an edge** if there is an edge associated to it in the graph.

Directed graphs that may have **multiple directed edges** from a vertex to a second (possibly the same) vertex are used to model such networks. We called such graphs **directed multigraphs**. When there are m directed edges, each associated to an ordered pair of vertices (u, v), we say that **(u, v) is an edge of multiplicity m**.

Simple Directed Graphs

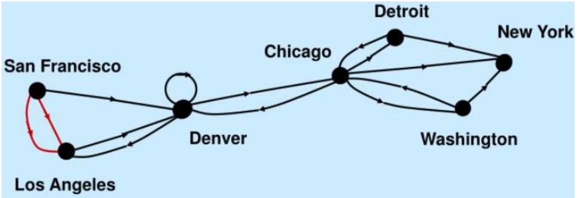
Example.



Remark. {u, v} = {v, u} BUT in directed graphs (u, v) ≠ (v, u)

Directed Multigraphs

Example. A computer network with multiple one-way links



Summary: Types of Graphs

TABLE 1 Graph Terminology.

Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

Graph Models

Graphs and graph theory can be used to model:

– Computer networks
– Social networks
– Communications networks
– Information networks
– Software design
– Transportation networks
– Biological networks

Graph Models

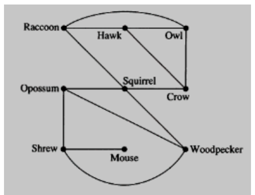


FIGURE 6 A Niche Overlap Graph.

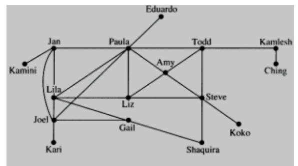
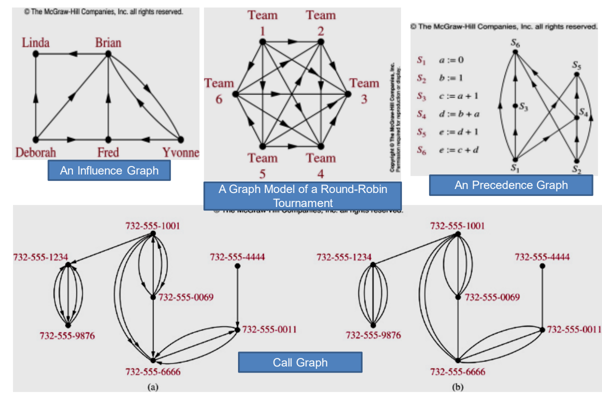


FIGURE 7 An Acquaintanceship Graph.

Niche Overlap Graph in Ecology
(sinh thái học) – Đồ thị lấn tổ

Acquaintanceship Graph
Đồ thị cho mô hình quan hệ giữa người

Graph Models



Graph Models

Example. Precedence Graphs and Concurrent Processing. Computer programs can be executed more rapidly by executing certain statements concurrently. It is important not to execute a statement that requires results of statements not yet executed. The dependence of statements on previous statements can be represented by a directed graph. Each statement is represented by a vertex, and there is an edge from one statement to a second statement if the second statement cannot be executed before the first statement. This resulting graph is called a precedence graph.

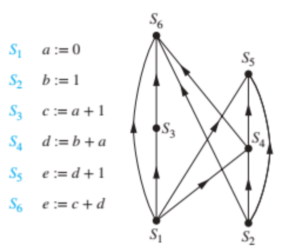
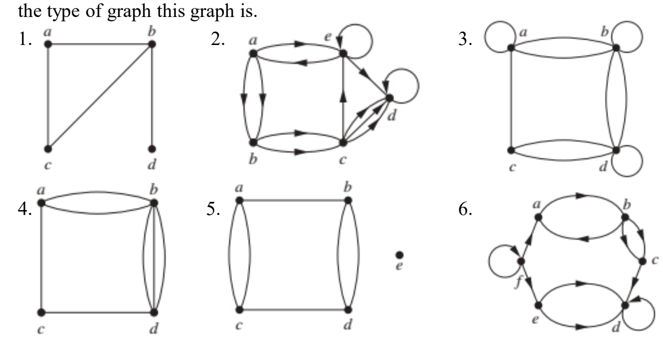


FIGURE 10 A precedence graph.

Exercises

Determine whether the graph shown has directed or undirected edges, whether it has multiple edges, and whether it has one or more loops. Use your answers to determine the type of graph this graph is.



Exercises

Construct a precedence graph for the following program:

$S_1: x := 0$
$S_2: x := x + 1$
$S_3: y := 2$
$S_4: z := y$
$S_5: x := x + 2$
$S_6: y := x + z$
$S_7: z := 4$

GRAPH TERMINOLOGY AND SPECIAL TYPES OF GRAPHS

Terminology for Undirected Graphs

Let G be an undirected graph with edge set E. Let $e \in E$ be (or map to) the pair {u, v}. Then we say:

- u, v are **adjacent**/**neighbors**/**connected**.
- Edge e is **incident** with vertices u and v.
- Edge e **connects** u and v.
- Vertices u and v are **endpoints** of edge e.

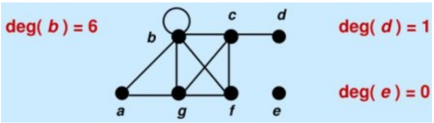
The set of all neighbors of a vertex v of $G = (V, E)$, denoted by $N(v)$, is called the *neighborhood* of v. If A is a subset of V, we denote by $N(A)$ the set of all vertices in G that are adjacent to at least one vertex in A. So, $N(A) = \bigcup_{v \in A} N(v)$.

Terminology for Undirected Graphs

Let G be an undirected graph, v ∈ V a vertex.

- The **degree** of v, **deg(v)**, is its number of incident edges. (Except that any self-loops are counted twice.)
- A vertex with **degree 0** is called **isolated**.
- A vertex of **degree 1** is called **pendant**.

Example.



Terminology for Undirected Graphs

Example. What are the degrees and what are the neighborhoods of the vertices in the graphs G and H displayed in Figure 1.

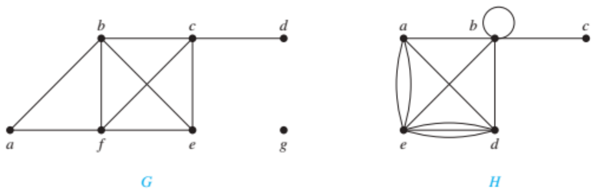
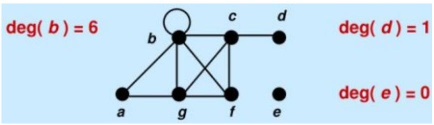


FIGURE 1 **The undirected graphs *G* and *H*.**

Terminology for Undirected Graphs

Example. Find the degree of all the other vertices: deg(a), deg(c), deg(f), deg(g), total of degrees and total number of edges



Handshaking Theorem for Undirected Graphs

THE HANDSHAKING THEOREM Let $G = (V, E)$ be an undirected graph with m edges. Then

$$2m = \sum_{v \in V} \deg(v).$$

(Note that this applies even if multiple edges and loops are present.)

Corollary. An undirected graph has an even number of vertices of odd degree.

Handshaking Theorem for Undirected Graphs

Example. How many edges are there in a graph with 10 vertices each of degree six?

Handshaking Theorem for Undirected Graphs

Example. Do there exist simple undirected graphs with 5 vertices of degrees:

a) 1, 2, 3, 3, 4 b) 1, 2, 3, 3, 3 c) 1, 2, 3, 4, 4

Terminology for Directed Graphs

Let G be a directed (possibly multi-) graph, and let e be an edge of G that is (or maps to) (u,v). Then we say:

- u is **adjacent** to v, v is adjacent from u
- e comes from u, e goes to v.
- e connects u to v, e goes from u to v
- the **initial** vertex of e is u
- the **terminal** vertex of e is v

Terminology for Directed Graphs

Let G be a directed graph, v a vertex of G.

The **in-degree** of v, $\deg^-(v)$, is the number of edges going to v.

The **out-degree** of v, $\deg^+(v)$, is the number of edges coming from v.

Remark. A loop at a vertex contributes 1 to both the in-degree and the out-degree of this vertex.

Terminology for Directed Graphs

Example. Find the in-degree and out-degree of each vertex in the graph G with directed edges shown in Figure 2.

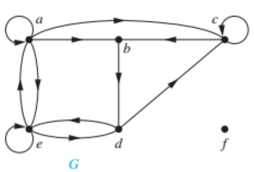


FIGURE 2 The directed graph *G*.

Handshaking Theorem for Directed Graphs

Let $G = (V, E)$ be a graph with directed edges. Then

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|.$$

Complete Graph

For any $n \in \mathbb{N}$, a **complete graph** on n vertices, $\boldsymbol{K_n}$, is a simple graph with n nodes that contains exactly one edge between each pair of distinct vertices.

Example.

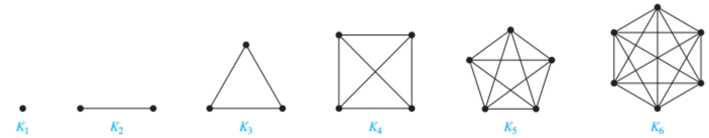


FIGURE 3 **The graphs K_n for $1 \leq n \leq 6$.**

Remark. K_n has $\frac{n(n-1)}{2} = C_n^2$ edges and n vertices. The degree of each vertex is $n - 1$

Cycles

A **cycle $\boldsymbol{C_n}$, $\boldsymbol{n \geq 3}$**, consists of n vertices $\mathrm{v}_1, \mathrm{v}_2, \dots, \mathrm{v}_\mathrm{n}$ and edges $\{\mathrm{v}_1, \mathrm{v}_2\}, \{\mathrm{v}_2, \mathrm{v}_3\}, \dots, \{\mathrm{v}_{\mathrm{n}-1}, \mathrm{v}_\mathrm{n}\}$ and $\{\mathrm{v}_\mathrm{n}, \mathrm{v}_1\}$.

Example.

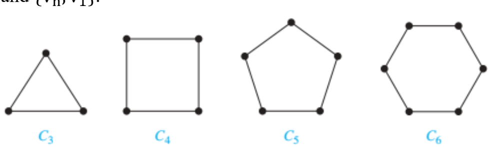


FIGURE 4 **The cycles C_3, C_4, C_5, and C_6.**

- C_n has n vertices and n edges.
- The degree of each vertex is 2.

Wheels

For any $\boldsymbol{n \geq 3}$, a **wheel $\boldsymbol{W_n}$**, is a simple graph obtained by adding an additional vertex to a cycle C_n and connecting this new vertex to each of the n vertices in C_n by new edges.

Example.

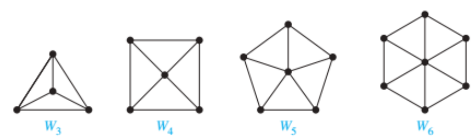


FIGURE 5 **The wheels W_3, W_4, W_5, and W_6.**

- W_n has $n + 1$ vertices and $2n$ edges.
- There are **3-degree of $\boldsymbol{n}$ vertices** and **one vertex has $\boldsymbol{n}$-degree**.

n-Cubes

***n*-Cubes** The ***n*-dimensional hypercube,** or ***n*-cube,** denoted by Q_n, is the graph that has vertices representing the 2^n bit strings of length n. Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position.

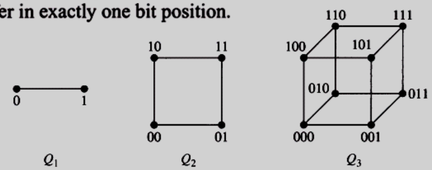


FIGURE 6 The *n*-cube Q_n for $n = 1, 2,$ and 3.

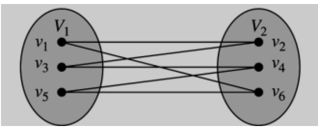
Q_n has 2^n vertices and $2^{n-1} \cdot n$ edges.

The degree of each vertex is n.

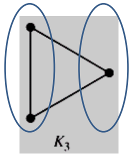
Bipartite Graphs

Definition.

A simple graph G is called *bipartite* if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2 (so that no edge in G connects either two vertices in V_1 or two vertices in V_2). When this condition holds, we call the pair (V_1, V_2) a *bipartition* of the vertex set V of G.



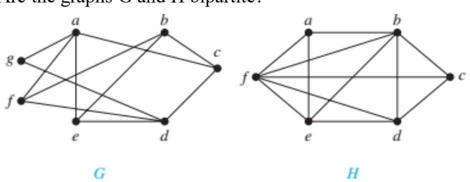
C_6 bipartite



Non-bipartite

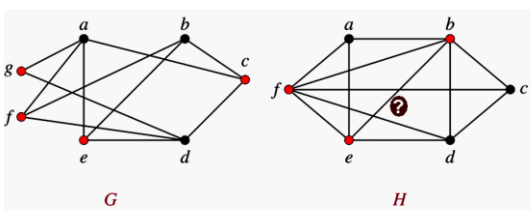
Bipartile Graphs

Example. Are the graphs G and H bipartite?



Bipartile Graphs

Theorem. A simple graph is bipartite if and only if it is possible to assign one of two different colors to each vertex of the graph so that no two adjacent vertices are assigned the same color.



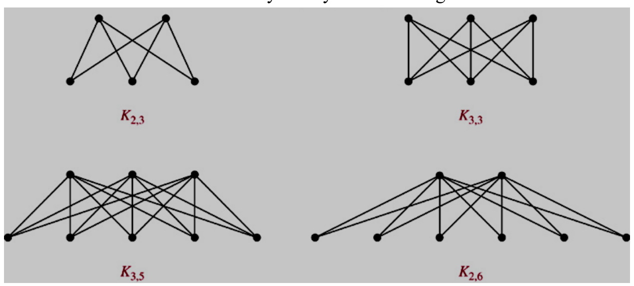
Complete Bipartite Graphs

For $m,n \in \mathbf{N}$, the *complete bipartite graph* $K_{m,n}$ is a bipartite graph where $|V_1| = m$, $|V_2| = n$, and $E = \{\{v_1,v_2\} | v_1 \in V_1 \wedge v_2 \in V_2\}$.

That is, the graph whose vertex set is divided to two disjoint subsets of m and n vertices, such that two vertices are connected if and only if they do not belong to the same subset.

Example.

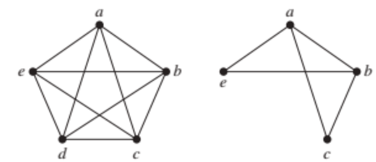
$K_{m,n}$ has $m + n$ vertices and mn edges $\deg(v) = n$, $\forall v \in V_1$, $\deg(v') = m$, $\forall v' \in V_2$



Subgraphs

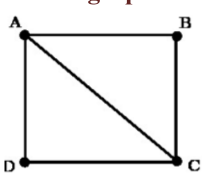
Definition. A **subgraph** of a graph G = (V, E) is a graph H = (W, F), where W ⊆ V and F ⊆ E. A subgraph H of G is a proper subgraph of G if H ≠ G.

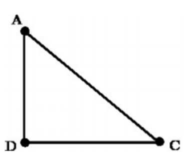
Example. A subgraph of K_5



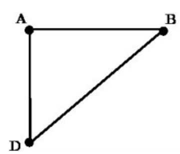
Subgraphs

Example. Given the graph G





is a subgraph of G

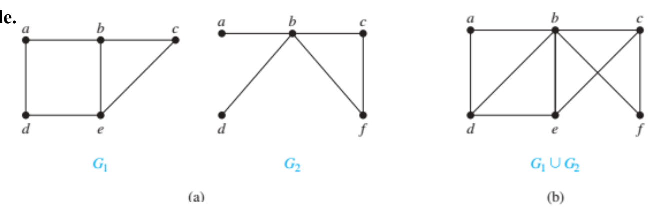


is not a subgraph of G

Graph Unions

Definition. The union of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. The union of G_1 and G_2 is denoted by $G_1 \cup G_2$.

Example.

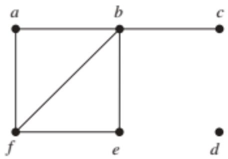


(a) The simple graphs G_1 and G_2. (b) Their union $G_1 \cup G_2$.

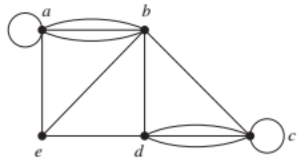
Exercises

1. Find the number of vertices, the number of edges, and the degree of each vertex in the given undirected graph. Identify all isolated and pendant vertices.

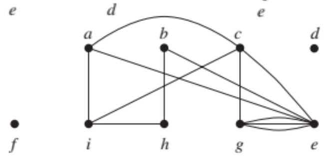
a)



b)



c)



Exercises

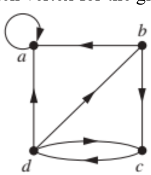
2. Find the sum of the degrees of the vertices of each graph in Exercises 1 and verify that it equals twice the number of edges in the graph.

3. Can a simple graph exist with 15 vertices each of degree five?

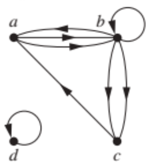
Exercises

4. Determine the number of vertices and edges and find the in-degree and out-degree of each vertex for the given directed multigraph.

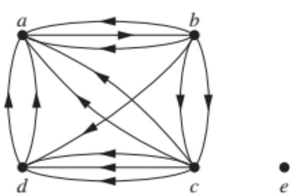
a)



b)



c)



Exercises

5. Determine the sum of the in-degrees of the vertices and the sum of the out-degrees of the vertices directly for each of the graphs in Exercise 4.

6. Draw these graphs:

a) K_7 b) $K_{1,8}$ c) $K_{4,4}$ d) C_7 e) W_7 f) Q_4

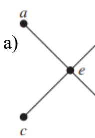
7. For which values of n are these graphs bipartite?

a) K_n b) C_n c) W_n d) Q_n

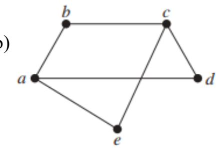
Exercises

8. Determine whether the graph is bipartite

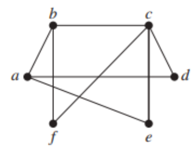
a)



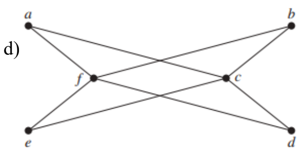
b)



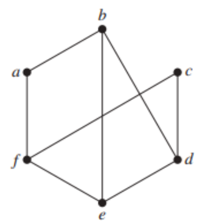
c)



d)



e)



Exercises

9. The degree sequence of a graph is the sequence of the degrees of the vertices of the graph in nonincreasing order. Find the degree sequences for each of the graphs in Exercise 8.

10. How many edges does a graph have if its degree sequence is 4, 3, 3, 2, 2? Draw such a graph.

11. Find the degree sequence of each of the following graphs.

a) K_4 b) C_4 c) W_4 d) $K_{2,3}$ e) Q_3

12. Does there exist a simple graph with 6 vertices and the degree sequence:

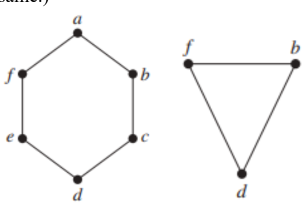
a) 5, 4, 3, 2, 1, 0 d) 3, 3, 3, 2, 2, 2

b) 2, 2, 2, 2, 2, 2 e) 5, 5, 4, 3, 2, 1

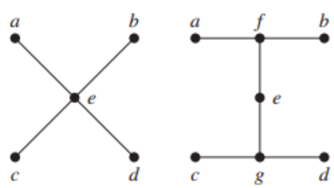
Exercises

13. Find the union of the given pair of simple graphs. (Assume edges with the same endpoints are the same.)

a)



b)

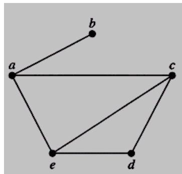


REPRESENTING GRAPHS AND GRAPH ISOMORPHISM

Adjacency Lists

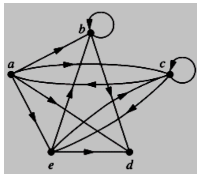
- For undirected graph: A table with 1 row per vertex, listing its adjacent vertices.
- For directed graph: 1 row per node, listing the terminal nodes of each edge incident from that node.

Example.



Vertex	Adjacent Vertices
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

FIGURE 1 A Simple Graph.



Initial Vertex	Terminal Vertices
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

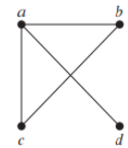
FIGURE 2 A Directed Graph.

Adjacency Matrices for Undirected Graphs

Suppose that G = (V, E) is a simple graph where |V| = n. Suppose that the vertices of G are listed arbitrarily as $v_1, v_2, \dots, v_n$. The adjacency matrix A (or A_G) of G is $A = [a_{ij}]$

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

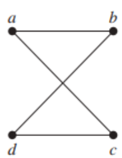
Example. The ordering of vertices is a, b, c, d.



$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Adjacency Matrices for Undirected Graphs

Example. A graph with the adjacency matrix $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ with respect to the ordering of vertices a, b, c, d is

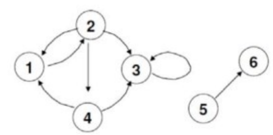


Adjacency Matrices for Directed Graphs

Suppose that G = (V, E) is a directed graph. Suppose that the vertices of G are listed arbitrarily as $v_1, v_2, \dots, v_n$. The adjacency matrix A (or A_G) of G is $A = [a_{ij}]$

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

Example.

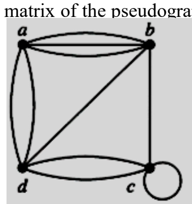


	1	2	3	4	5	6
1	0	1	0	0	0	0
2	1	0	1	1	0	0
3	0	0	1	0	0	0
4	1	0	1	0	0	0
5	0	0	0	0	0	1
6	0	0	0	0	0	0

Adjacency Matrices

- Adjacency matrices can also be used to represent graphs with loops and multiple edges.
- A loop at the vertex vi is represented by a 1 at the (i, i)th position of the matrix.
- When multiple edges connect the same pair of vertices v_i and v_j , (or if multiple loops are present at the same vertex), the (i, j)th entry equals the number of edges connecting the pair of vertices..

Example. The adjacency matrix of the pseudograph shown here using the ordering of vertices *a*, *b*, *c*, *d*.



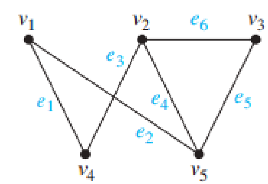
$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

Incidence Matrices

Let G = (V, E) be an undirected graph. Suppose that $v_1, v_2, \ldots, v_n$ are the vertices and $e_1, e_2, \ldots, e_m$ are the edges of G. Then the incidence matrix with respect to this ordering of V and E is the n × m matrix $M = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 & \text{when edge } e_j \text{ is incident with } v_i, \\ 0 & \text{otherwise.} \end{cases}$$

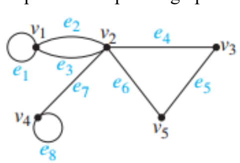
Example.



$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}.$$

Incidence Matrices

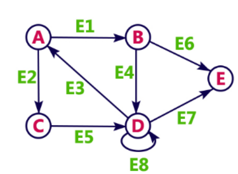
Example. Represent the pseudograph using an incidence matrix.



Incidence Matrices

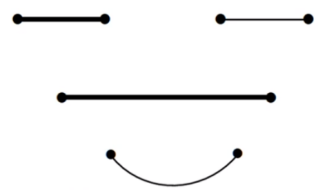
In directed graph,

- 0 is used to represent row edge which is not connected to column vertex.
- 1 is used to represent row edge which is connected as outgoing edge to column vertex.
- −1 is used to represent row edge which is connected as incoming edge to column vertex.



	E1	E2	E3	E4	E5	E6	E7	E8
A	1	1	-1	0	0	0	0	0
B	-1	0	0	1	0	1	0	0
C	0	-1	0	0	1	0	0	0
D	0	0	1	-1	-1	0	1	1
E	0	0	0	0	0	-1	-1	0

Isomorphism of Graphs



THE SAME

Isomorphism of Graphs

Definition.

The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there exists a one-to-one and onto function f from V_1 to V_2 with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1. Such a function f is called an *isomorphism*.* Two simple graphs that are not isomorphic are called *nonisomorphic*.

Isomorphism of Graphs

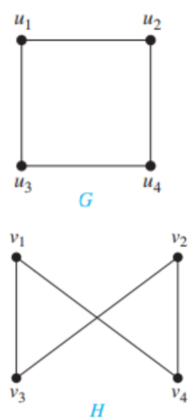
Example. The graphs G = (V, E) and H = (W, F) are isomorphic.

$f: \{u_1, u_2, u_3, u_4\} \to \{v_1, v_2, v_3, v_4\}$

$f(u_1) = v_1, f(u_2) = v_4, f(u_3) = v_3, f(u_4) = v_2$

$\{u_1, u_2\} = \{v_1, v_4\}, \{u_2, u_4\} = \{v_4, v_2\}, \{u_1, u_3\} = \{v_1, v_3\},$

$\{u_4, u_3\} = \{v_2, v_3\}$



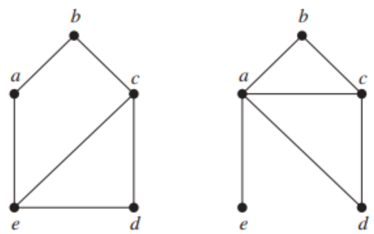
Isomorphism of Graphs

Example. Show that the graphs G = (V, E) and H = (W, F) are isomorphic.

Solution: The function f with $f(u_1) = v_1, f(u_2) = v_4, f(u_3) = v_3$, and $f(u_4) = v_2$ is a one-to-one correspondence between V and W. To see that this correspondence preserves adjacency, note that adjacent vertices in G are u_1 and u_2, u_1 and u_3, u_2 and u_4, and u_3 and u_4, and each of the pairs $f(u_1) = v_1$ and $f(u_2) = v_4$, $f(u_1) = v_1$ and $f(u_3) = v_3$, $f(u_2) = v_4$ and $f(u_4) = v_2$, and $f(u_3) = v_3$ and $f(u_4) = v_2$ consists of two adjacent vertices in H.

Isomorphism of Graphs

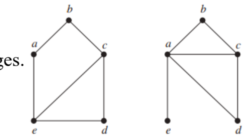
Example. Show that the graphs are not isomorphic.



G H

Isomorphism of Graphs

Example. Show that the graphs are not isomorphic.
Solution: Both G and H have five vertices and six edges.
However, H has a vertex of degree one, namely, e, whereas G has no vertices of degree one.
It follows that G and H are not isomorphic



G H

A permutation of vertices in H is similar with vertices in G ➔ Complexity: O (n!) ➔ It is often difficult to determine whether two graphs are isomorphic if number of vertices is large.

Isomorphism of Graphs

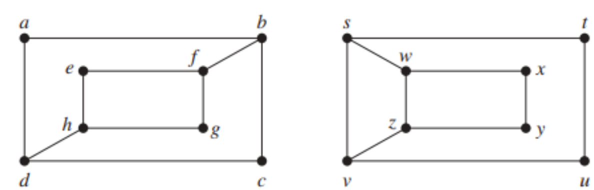
Any 2 graphs G and H will be known as isomorphism if they satisfy the following 4 conditions:

1. The same number of vertices
2. The same number of edges
3. The same degree of the vertices (deg(v) = deg[f(v)] for any node v in G)
4. f(u) is adjacent to f(v) in H for any pair of nodes u and v from G that are adjacent

Isomorphism of Graphs

Example. Determine whether the graphs shown are isomorphic.

a)

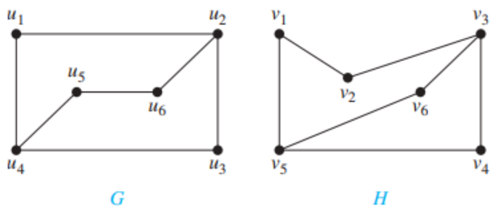


G H

Isomorphism of Graphs

Example. Determine whether the graphs shown are isomorphic.

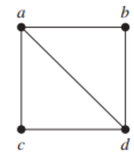
b)



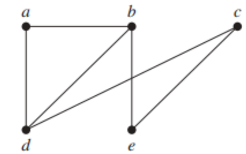
Exercises

1. Use an adjacency list and an adjacency matrix to represent the given graph.

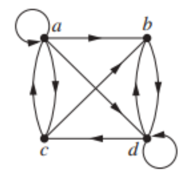
a)



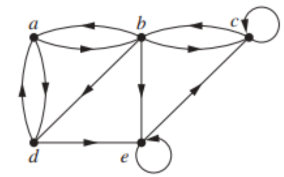
b)



c)



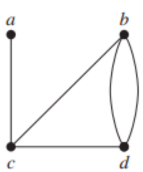
d)



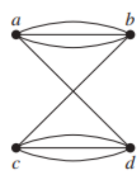
Exercises

2. Represent the given graph using an adjacency matrix

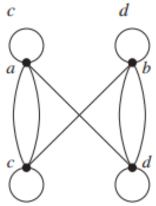
a)



b)



c)



Exercises

3. Represent each of these graphs with an adjacency matrix

a) K_4 b) $K_{1,4}$ c) $K_{2,3}$ d) C_4 e) W_4 f) Q_3

4. Draw a graph with the given adjacency matrix

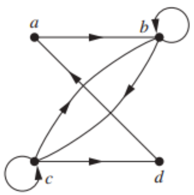
a) $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ b) $\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$ c) $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

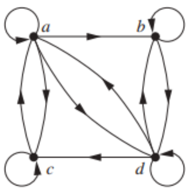
5. Draw an undirected graph represented by the given adjacency matrix

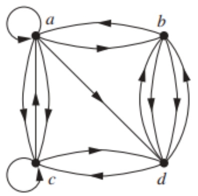
a) $\begin{bmatrix} 1 & 3 & 2 \\ 3 & 0 & 4 \\ 2 & 4 & 0 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$ c) $\begin{bmatrix} 0 & 1 & 3 & 0 & 4 \\ 1 & 2 & 1 & 3 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ 0 & 3 & 0 & 0 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$

Exercises

6. Find the adjacency matrix of the given directed multigraph with respect to the vertices listed in alphabetic order

a) 

b) 

c) 

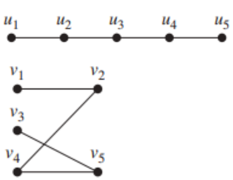
Exercises

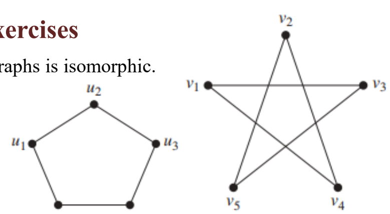
7. Draw the graph represented by the given adjacency matrix.

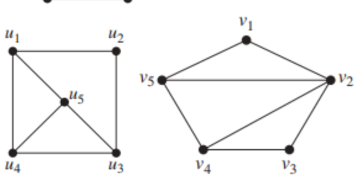
a) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix}$ c) $\begin{bmatrix} 0 & 2 & 3 & 0 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 1 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}$

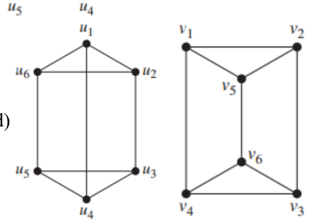
Exercises

8. Determine whether the given pair of graphs is isomorphic.

a) 

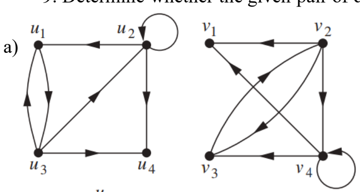
b) 

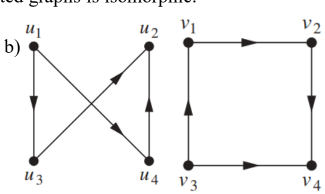
c) 

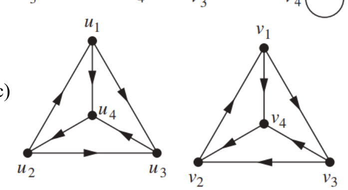
d) 

Exercises

9. Determine whether the given pair of directed graphs is isomorphic.

a) 

b) 

c) 

Exercises

10. Are the simple graphs with the following adjacency matrices isomorphic?

a) $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

b) $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

Exercises

11. Determine whether the graphs without loops with these incidence matrices are isomorphic

a) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

b) $\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$

CONNECTIVITY

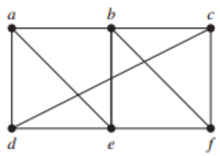
Paths in Undirected Graphs

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 $e_1, \dots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \dots, x_{n-1}, x_n = v$ of vertices such that e_i has, for i = 1, … , n, the endpoints x_{i-1} and x_i. When the graph is simple, we denote this path by its vertex sequence $x_0, x_1, \dots, x_n$ (because listing these vertices uniquely determines the path).
- The path is a circuit or circle if it begins and ends at the same vertex, that is, if u = v, and has length greater than zero.
- The path or circuit is said to pass through the vertices $x_1, x_2, \dots, x_{n-1}$ or traverse the edges $e_1, e_2, \dots, e_n$.
- A path or circuit is simple if it does not contain the same edge more than once.

Paths in Undirected Graphs

Example. In the simple graph, a, d, c, f , e is a simple path of length 4, because {a, d}, {d, c}, {c, f }, and { f, e} are all edges. However, d, e, c, a is not a path, because {e, c} is not an edge. Note that b, c, f , e, b is a circuit of length 4 because {b, c}, {c, f }, { f, e}, and {e, b} are edges, and this path begins and ends at b. The path a, b, e, d, a, b, which is of length 5, is not simple because it contains the edge {a, b} twice.

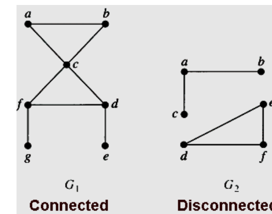


Connectedness in Undirected Graphs

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, or both, to produce a disconnected subgraph.

Example.

The graph G1 is connected, because for every pair of distinct vertices there is a path between them (the reader should verify this). However, the graph G2 is not connected. For instance, there is no path in G2 between vertices a and d.



Connectedness in Undirected Graphs

- **Theorem.** There is a simple path between every pair of distinct vertices of a connected undirected graph.
- A connected component of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G. That is, a connected component of a graph G is a maximal connected subgraph of G. A graph G that is **not connected** has two or more connected components that are disjoint and have G as their union.
- A cut vertex, or an articulation point, is a vertex that if we remove it and all the edges incident with it we will obtain a subgraph having more connected components than the original graph.
- A cut edge, or a bridge, is an edge that if we remove it we will obtain a subgraph having more connected components than the original graph.

Connectedness in Undirected Graphs

Example. The graph H is the union of three disjoint connected subgraphs H_1, H_2, and H_3. These three subgraphs are the connected components of H.

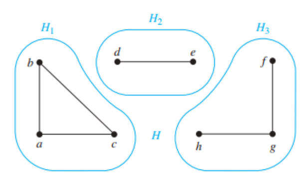
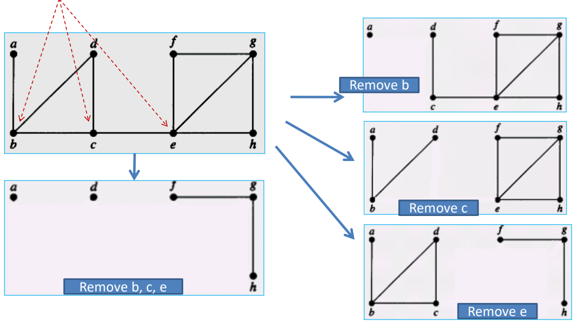


FIGURE 3 The graph *H* and its connected components H_1, H_2, and H_3.

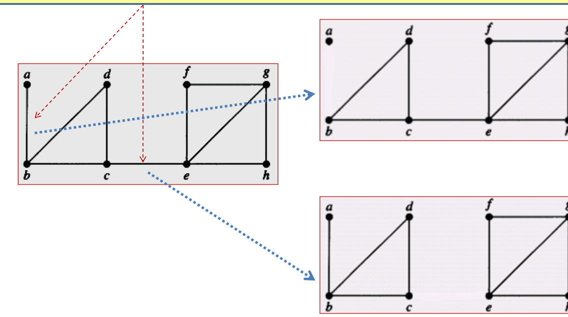
Connectedness in Undirected Graphs

Cut vertex (articulation point): It's removal will produce disconnected subgraph from original connected graph.



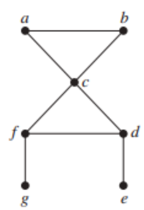
Connectedness in Undirected Graphs

Cut edge (bridge): It's removal will produce subgraphs which are more connected components (thành phần liên thông) than in the original graph



Connectedness in Undirected Graphs

Example. The cut vertices of G_1 are b, c, and e. The removal of one of these vertices (and its adjacent edges) disconnects the graph. The cut edges are {a, b} and {c, e}. Removing either one of these edges disconnects G_1.



G_1

Connectedness in Directed Graphs

Definition.

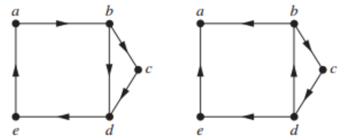
A directed graph is **strongly** connected if there is a path from a to b and from b to a whenever a and b are vertices in the graph.

A directed graph is **weakly** connected if there is a path between every two vertices in the underlying undirected graph (when the directions of the edges are disregarded).

Notes. Strongly connected implies weakly connected but **NOT vice-versa**.

Connectedness in Directed Graphs

Example. Are the directed graphs G and H strongly connected? Are they weakly connected?



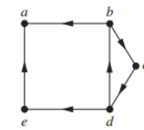
G H

Solution. G is strongly connected because there is a path between any two vertices in this directed graph. Hence, G is also weakly connected. The graph H is not strongly connected. There is no directed path from a to b in this graph. However, H is weakly connected, because there is a path between any two vertices in the underlying undirected graph of H.

Connectedness in Directed Graphs

- The subgraphs 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. Note that if a and b are two vertices in a directed graph, their strong components are either the same or disjoint.

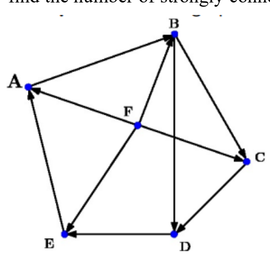
Example. The graph H has three strongly connected components, consisting of the vertex a; the vertex e; and the subgraph consisting of the vertices b, c, and d and edges (b, c), (c, d), and (d, b).



H

Connectedness in Directed Graphs

Example. Determine if the graph is strongly connected, weakly connected, and find the number of strongly connected components.

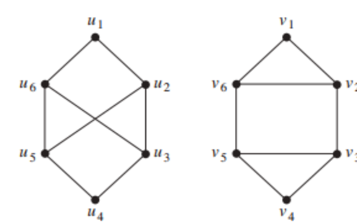


Paths and Isomorphism

- Note that connectedness, and the existence of a circuit or simple circuit of length k, where k is a positive integer greater than 2, are graph invariants with respect to isomorphism.

Example. Determine whether the graphs G and H are isomorphic.

Solution. Both G and H have six vertices and eight edges. Each has four vertices of degree three, and two vertices of degree two. So, the three invariants—number of vertices, number of edges, and degrees of vertices—all agree for the two graphs. However, H has a simple circuit of length three, namely, v1, v2, v6, v1, whereas G has no simple circuit of length three, as can be determined by inspection (all simple circuits in G have length at least four). Because the existence of a simple circuit of length three is an isomorphic invariant, G and H are not isomorphic.

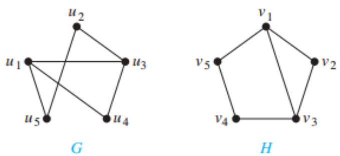


G H

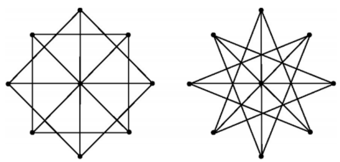
Paths and Isomorphism

Example. Using graph invariant of paths and circuits to check if the two graphs are isomorphic:

a)



b)

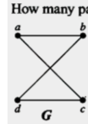


Counting Paths Between Vertices

Theorem. Let G be a graph with adjacency matrix A with respect to the ordering v_1, v_2, … , v_n of the vertices of the graph (with directed or undirected edges, with multiple edges and loops allowed). The number of different paths of length r from v_i to v_j, where r is a positive integer, equals the (i, j)th entry of A^r.

EXAMPLE 14

How many paths of length $\boxed{\text{four}}$ are there from a to d in the simple graph G?



Solution: The adjacency matrix of G (ordering the vertices as a, b, c, d) is

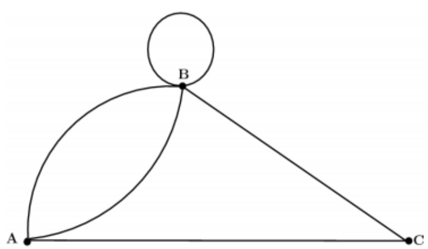
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad \mathbf{A}^{\boxed{4}} = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

Result= 8

a, b, a, b, d
a, b, a, c, d
a, b, d, b, d
a, b, d, c, d
a, c, a, b, d
a, c, a, c, d
a, c, d, b, d
a, c, d, c, d

Counting Paths Between Vertices

Example. Count the number of paths of length 3 between A and C in the graph

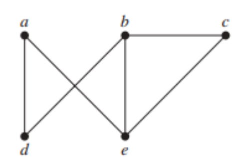


Exercises

1. Does each of these lists of vertices form a path in the following graph? Which paths are simple? Which are circuits? What are the lengths of those that are paths?

a) a, e, b, c, b   b) a, e, a, d, b, c, a

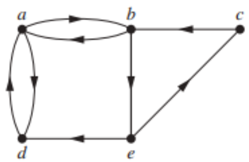
c) e, b, a, d, b, e   d) c, b, d, a, e, c



Exercises

2. Does each of these lists of vertices form a path in the following graph? Which paths are simple? Which are circuits? What are the lengths of those that are paths?

a) a, b, e, c, b　　b) a, d, a, d, a　　c) a, d, b, e, a　　d) a, b, e, c, b, d, a



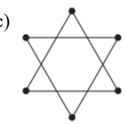
Exercises

3. Determine whether the given graph is connected.

a)

b)

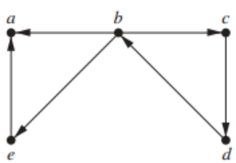
c)



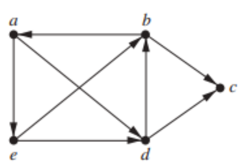
Exercises

4. Determine whether each of these graphs is strongly connected and if not, whether it is weakly connected.

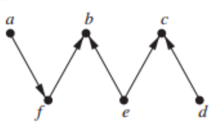
a)



b)



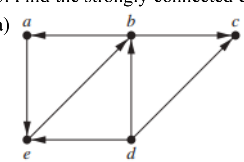
c)



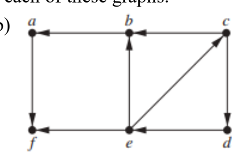
Exercises

5. Find the strongly connected components of each of these graphs.

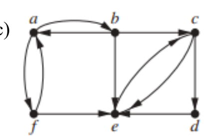
a)



b)



c)



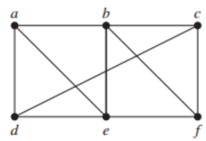
Exercises

6. Find the number of paths of length n between two different vertices in K_4 if n is

a) 2 b) 3 c) 4 d) 5

7. Given the graph. Find the number of paths between c and d in the graph of length

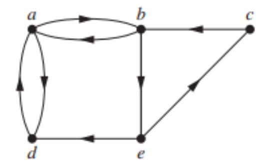
a) 2. b) 3. c) 4. d) 5. e) 6. f) 7.



Exercises

8. Find the number of paths from a to e in the directed graph of length

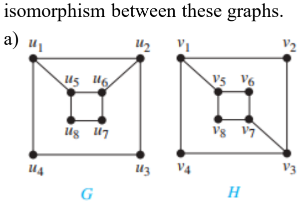
a) 2. b) 3. c) 4. d) 5. e) 6. f) 7.



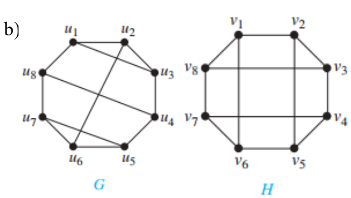
Exercises

9. Use paths either to show that these graphs are not isomorphic or to find an isomorphism between these graphs.

a)



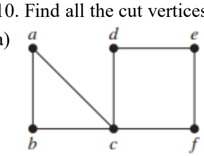
b)



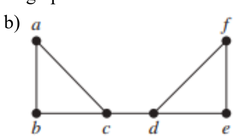
Exercises

10. Find all the cut vertices of the given graph.

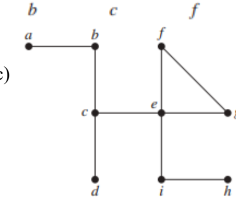
a)



b)



c)



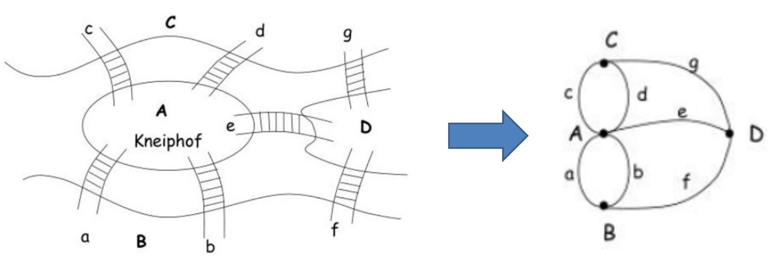
EULER AND HAMILTON PATHS

Introduction

Konigsberg Bridge Problem

- A river Pregel flows around the island Keniphof and then divides into 2.
- 4 lands areas A, B, C, D have this river on their borders. The 4 lands are connected by 7 bridges a – g.
- Determine whether it is possible to walk across all the bridges exactly once in returning back to the starting land area?

Introduction



The problem becomes: Is there a simple circuit in this multigraph that contains every edge?
→ Euler paths and circuits

Euler Paths and Circuits

Definition.

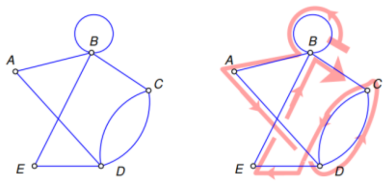
- **Euler path**: a path that travels through every **edge** of a graph once and only once.
- **Euler circuit**: a circuit that travels through every **edge** of a graph once and only once.

Notes.

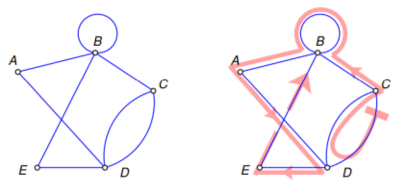
- An Euler path starts and ends at different vertices.
- An Euler circuit starts and ends at the same vertex.
- Every Euler circuit is an Euler path BUT NOT every Euler path is an Euler circuit.

Euler Paths and Circuits

Example.



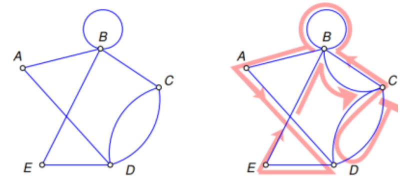
An Euler path: BBADCDEBC



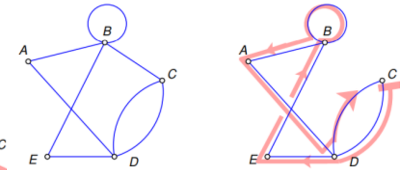
Another Euler path: CDCBBADEB

Euler Paths and Circuits

Example.



An Euler circuit: CDCBBADEBC

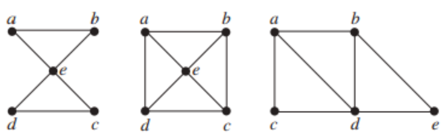


Another Euler circuit: CDEBBADC

Euler Paths and Circuits

Example.

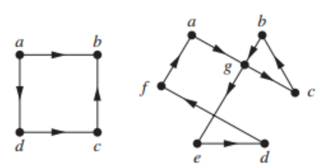
- G_1 has an Euler circuit, for example, a, e, c, d, e, b, a.
- G_2 has neither Euler circuit nor Euler path
- G_3 has an Euler path, namely, a, c, d, e, b, d, a, b but it has not an Euler circuit



G_1 G_2 G_3

Euler Paths and Circuits

Example. Which of the directed graphs have an Euler circuit? Of those that do not, which have an Euler path?



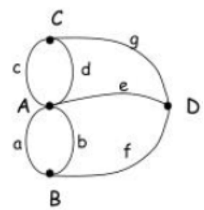
H_1 H_2 H_3

NECESSARY AND SUFFICIENT CONDITIONS FOR EULER CIRCUITS AND PATHS

Theorem.

1. A connected multigraph with at least two vertices has an Euler circuit if and only if each of its vertices has even degree.
2. A connected multigraph has an Euler path but not an Euler circuit if and only if it has exactly two vertices of odd degree.

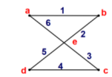
Example. The answer to the 7 bridges problem is NO



Necessary and Sufficient Conditions for Euler Circuits and Paths

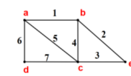
Example.

1. The graph



have $deg(a) = deg(b) = deg(c) = deg(d) = deg(e) = 2$.
So,there is an Euler circuit a, b, e, d, c, e, a.

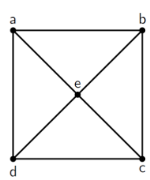
2. The graph



Have $deg(a) = deg(b) = 3; deg(c) = 4; deg(d) = deg(e) = 2$.So, there is an Euler path a, d, c, e, b, c, a, b.

Necessary and Sufficient Conditions for Euler Circuits and Paths

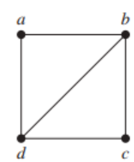
Example. The graph



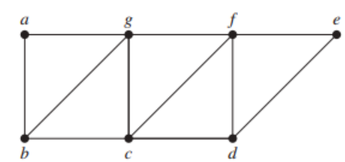
Have neither Euler circuit nor Euler path.

Necessary and Sufficient Conditions for Euler Circuits and Paths

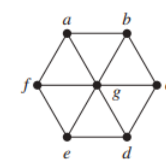
Example. Which graphs have an Euler path?



G_1



G_2



G_3

Algorithm for Finding Euler Circuits

ALGORITHM 1 Constructing Euler Circuits.

procedure *Euler*(G: connected multigraph with all vertices of even degree)
circuit := a circuit in G beginning at an arbitrarily chosen vertex with edges successively added to form a path that returns to this vertex
H := G with the edges of this circuit removed
while H has edges
  subcircuit := a circuit in H beginning at a vertex in H that also is an endpoint of an edge of *circuit*
  H := H with edges of *subcircuit* and all isolated vertices removed
  circuit := *circuit* with *subcircuit* inserted at the appropriate vertex
return *circuit* {*circuit* is an Euler circuit}

Euler paths and circuits cover every edge of a graph. These are useful in optimizing routes for applications like garbage collection, where each street (edge) only needs to be traversed once but a particular intersection (vertex) may be crossed more than once.

What about optimizing routes for applications like FedEx or UPS in package delivery? For these applications, we need to go to every house that has a delivery (vertices) but need not necessarily traverse every street (edge). For this, we need to examine **Hamilton paths and circuits**.

Hamilton Paths and Circuits

Definition.

- **Hamilton path**: a path that travels through every **vertex** of a graph one and only once.

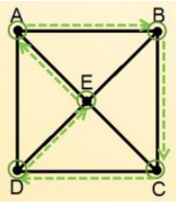
Note. In fact, every edge does not even have to be crossed.

- **Hamilton circuit**: a Hamilton path that begins and ends at the same vertex and passes through all other vertices exactly once.

Hamilton Paths and Circuits

Example.

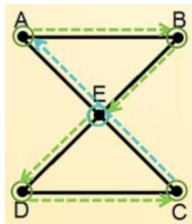
There is a Hamilton path and a Hamilton circuit, for example, A, B, C, D, E, A.



Hamilton Paths and Circuits

Example.

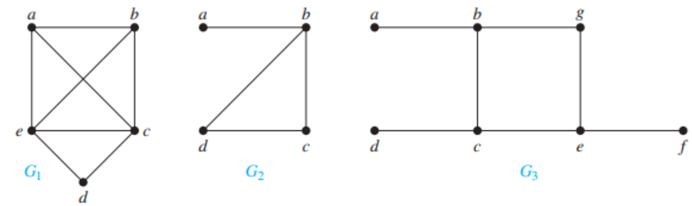
There is a Hamilton path, for example, A, B, E, D, C but NO Hamilton circuit.



Hamilton Paths and Circuits

Example.

Which of the simple graphs in Figure have a Hamilton circuit or, if not, a Hamilton path?



Conditions for The Existence of Hamilton Circuits

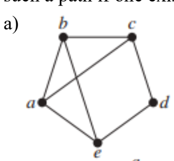
Theorem.

1. (**DIRAC'S THEOREM**) If G is a simple graph with n vertices with $n \geq 3$ such that the degree of every vertex in G is at least n/2, then G has a Hamilton circuit.
2. (**ORE'S THEOREM**) If G is a simple graph with n vertices with $n \geq 3$ such that $deg(u) + deg(v) \geq n$ for every pair of nonadjacent vertices u and v in G, then G has a Hamilton circuit.

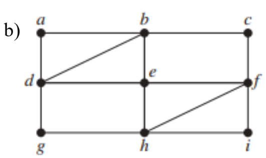
Exercises

1. Determine whether the given graph has an Euler circuit. Construct such a circuit when one exists. If no Euler circuit exists, determine whether the graph has an Euler path and construct such a path if one exists.

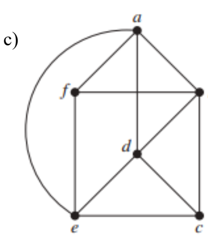
a)



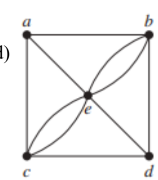
b)



c)

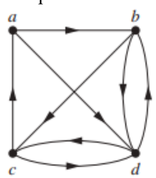


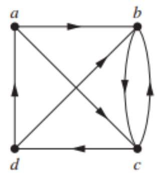
d)

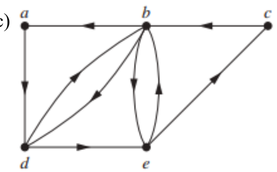


Exercises

2. Determine whether the directed graph has an Euler circuit. Construct such a circuit if one exists. If no Euler circuit exists, determine whether the graph has an Euler path and construct such a path if one exists.

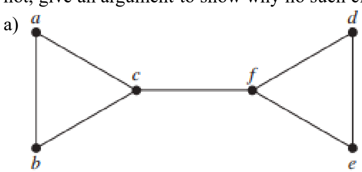
a) 

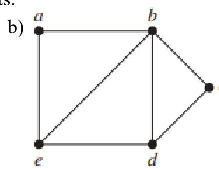
b) 

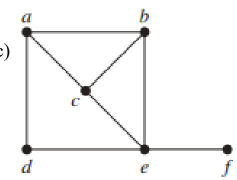
c) 

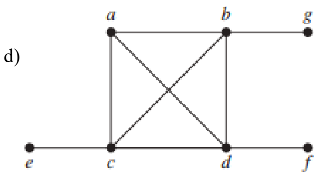
Exercises

3. Determine whether the graph has Hamilton circuit. If it does, find such a circuit. If it does not, give an argument to show why no such circuit exists.

a) 

b) 

c) 

d) 

Exercises

4. Do the graphs in Exercise 3 have a Hamilton path? If so, find such a path. If it does not, give an argument to show why no such path exists.

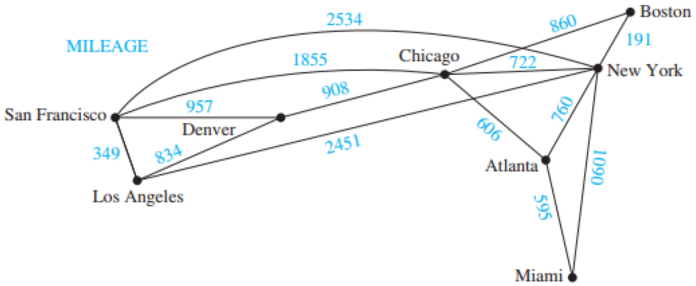
SHORTEST-PATH PROBLEMS

Weighted Graphs

- Graphs that have a number assigned to each edge are called weighted graphs or network. Each weight is a real number.
- Weight can represent distance, cost, time, etc.
- The length of a path in a weighted graph be the sum of the weights of the edges of this path.

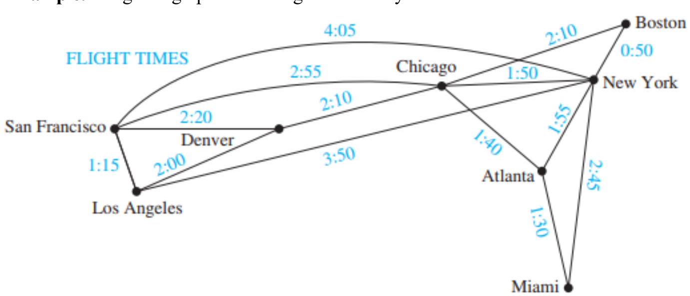
Weighted Graphs

Example. Weighted graphs modeling an airline system.



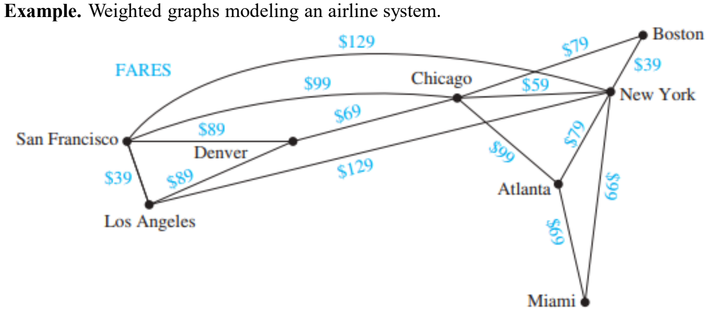
Weighted Graphs

Example. Weighted graphs modeling an airline system.



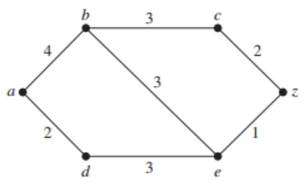
Weighted Graphs

Example. Weighted graphs modeling an airline system.



Weighted Graphs

Example. The length of the path a, b, e, z is 8.
The length of the path a, b, e, d, a is 12.



Shortest Path Problem

- **Problem.** Find the shortest path from A to Z in a weighted graph.

→ **Dijkstra's algorithm:**

- Finds the length of the shortest path from A to the first vertex.
- Finds the length of the shortest path from A to the second vertex.
- Finds the length of the shortest path from A to the third vertex.
- …
- Continue the process until Z is reached.

→ This algorithm is easily extended to find the length of the shortest path from a to all other vertices of the graph, and not just to z.

Shortest Path Algorithm Applications

Application	Vertex	Edge	Solutions
Map	Intersection	Road	Find the shortest/fastest route
Network	Router	Connection	Internet routing
Flight Agenda	Airports	Flights	Find earliest time to reach destination

Dijkstra's Algorithm

Theorem.

1. Dijkstra's algorithm finds the length of a shortest path between two vertices in a connected simple undirected weighted graph.
2. Dijkstra's algorithm uses $O(n^2)$ operations (additions and comparisons) to find the length of a shortest path between two vertices in a connected simple undirected weighted graph with n vertices.

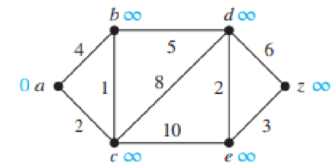
Dijkstra's Algorithm

procedure *Dijkstra*(G: weighted connected simple graph, with all weights positive)
{G has vertices $a = v_0, v_1, \ldots, v_n = z$ and lengths $w(v_i, v_j)$ where $w(v_i, v_j) = \infty$ if $\{v_i, v_j\}$ is not an edge in G}
for $i := 1$ **to** n
    $L(v_i) := \infty$
$L(a) := 0$
$S := \emptyset$
{the labels are now initialized so that the label of a is 0 and all other labels are ∞, and S is the empty set}
while $z \notin S$
    $u :=$ a vertex not in S with $L(u)$ minimal
    $S := S \cup \{u\}$
    for all vertices v not in S
        if $L(u) + w(u, v) < L(v)$ **then** $L(v) := L(u) + w(u, v)$
        {this adds a vertex to S with minimal label and updates the labels of vertices not in S}
return $L(z)$ {$L(z)$ = length of a shortest path from a to z}

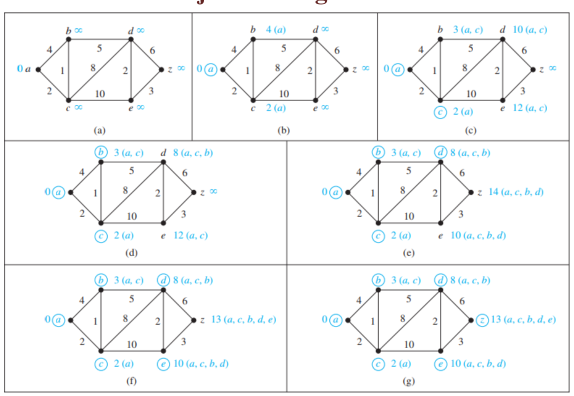
Running time: $O(n^2)$

Dijkstra's Algorithm

Example. Use Dijkstra's algorithm to find the length of a shortest path between the vertices a and z in the weighted graph.

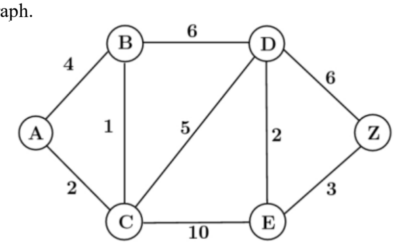


Dijkstra's Algorithm



Dijkstra's Algorithm

Problem. Use the Dijkstra's algorithm to find the path of shortest length from A to Z in a weighted graph.



The Traveling Salesperson Problem

- The traveling salesman problem is one of the classical problems in computer science.
- A traveling salesman wants to visit a number of cities and then return to his starting point. Of course he wants to save time and energy, so he wants to determine the shortest cycle for his trip.
- We can represent the cities and the distances between them by a weighted, complete, undirected graph.
- The problem then is to find the shortest cycle (of minimum total weight that visits each vertex exactly one).

The Traveling Salesperson Problem

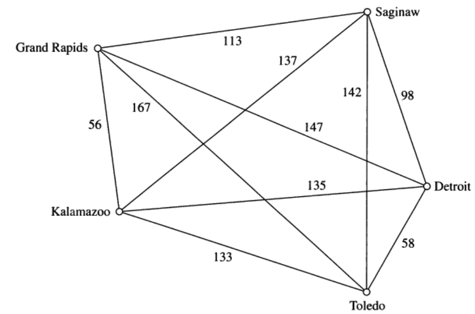


FIGURE 5 The Graph Showing the Distances between Five Cities.

Salesman starts in one city (ex. Detroit). He wants to visit 6 cities exactly once and return to his starting point (Detroit). In which order should he visit theses cities to travel the minimum total distance ?

The Traveling Salesperson Problem

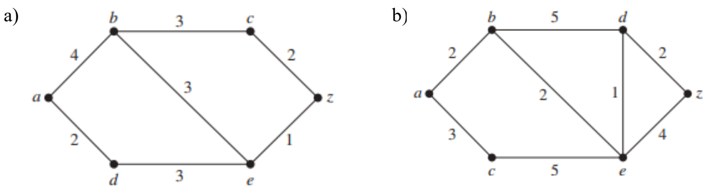
Route	*Total Distance (miles)*
Detroit–Toledo–Grand Rapids–Saginaw–Kalamazoo–Detroit	610
Detroit–Toledo–Grand Rapids–Kalamazoo–Saginaw–Detroit	516
Detroit–Toledo–Kalamazoo–Saginaw–Grand Rapids–Detroit	588
Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit	458
Detroit–Toledo–Saginaw–Kalamazoo–Grand Rapids–Detroit	540
Detroit–Toledo–Saginaw–Grand Rapids–Kalamazoo–Detroit	504
Detroit–Saginaw–Toledo–Grand Rapids–Kalamazoo–Detroit	598
Detroit–Saginaw–Toledo–Kalamazoo–Grand Rapids–Detroit	576
Detroit–Saginaw–Kalamazoo–Toledo–Grand Rapids–Detroit	682
Detroit–Saginaw–Grand Rapids–Toledo–Kalamazoo–Detroit	646
Detroit–Grand Rapids–Saginaw–Toledo–Kalamazoo–Detroit	670
Detroit–Grand Rapids–Toledo–Saginaw–Kalamazoo–Detroit	728

The Traveling Salesperson Problem

- Finding the shortest cycle is different than Dijkstra's shortest path. It is much harder too, no algorithm exists with polynomial worst-case time complexity!
- This means that for large numbers of vertices, solving the traveling salesman problem is impractical.
- The problem has theoretical importance because it represents a class of difficult problems known as NP-hard problems.
- In these cases, we can use efficient approximation algorithms that determine a path whose length may be slightly larger than the traveling salesman's path.

Exercises

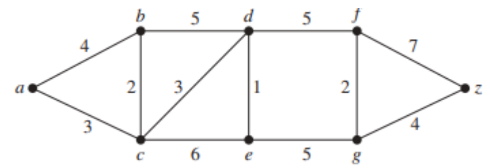
1. Find the shortest path between a and z in the weighted graph



Exercises

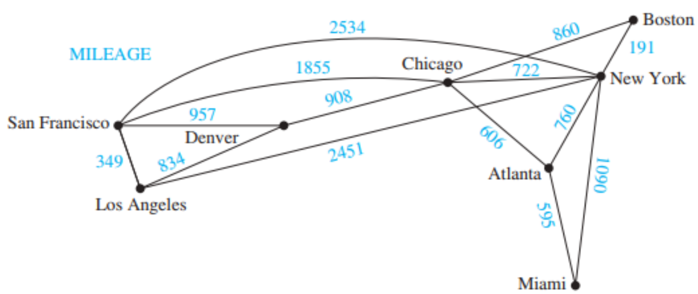
2. Find the length of a shortest path between these pairs of vertices in the weighted graph

a) a and z b) a and d c) c and f d) b and z



Exercises

3. Find a shortest path (in mileage) between Miami and Denver in the airline system



Thanks