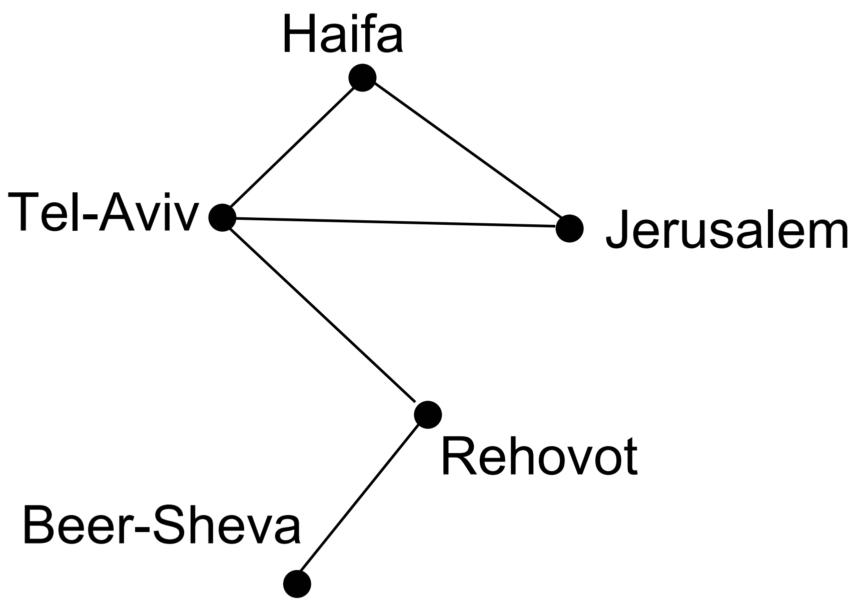
Lesson 12:

Graphs

- Basic Terminology
- Types of Graphs
- Graph Isomorphism
- Graph Connectivity
- Euler & Hamiltonian Paths



Chapter 8.1-8.5

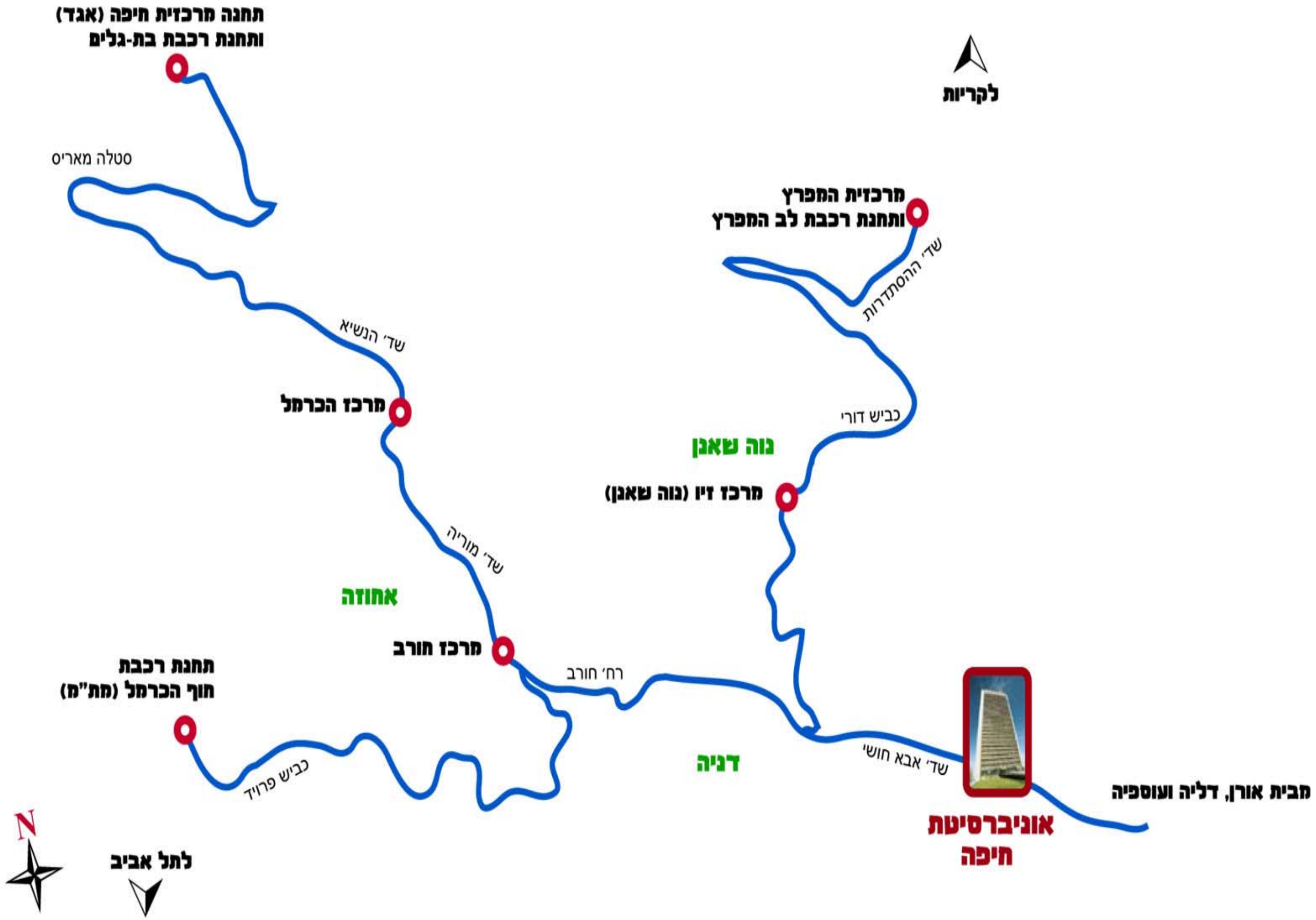
Graphs

A Graph is a discrete structure composed of vertices and connecting edges.

Graphs represent data and problems and can be used to analyze and solve problems.

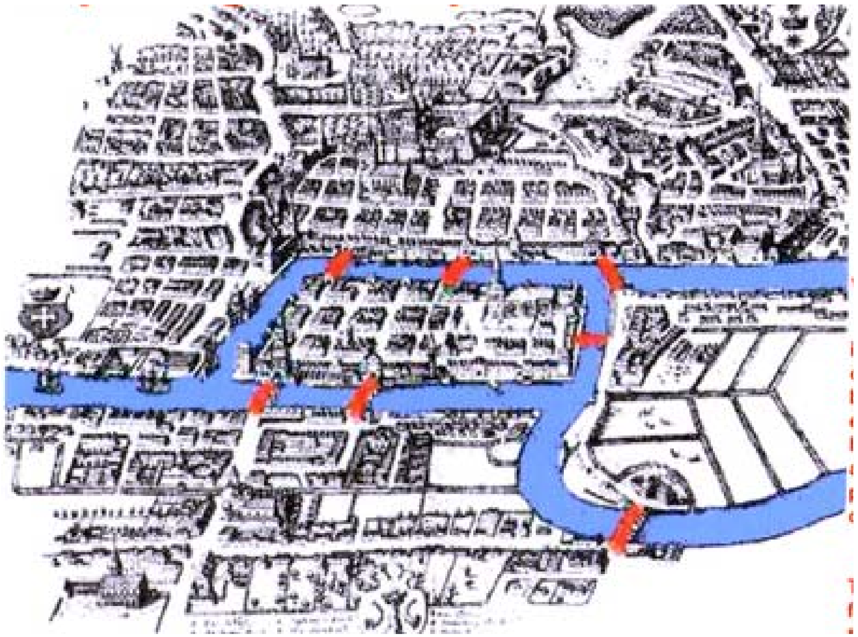
- Road and city Maps
- Communication/computer Networks
- Molecular Structures
- Relations between people/objects.
- WWW

Graphs


תחנה מרכזית חיפה (אגד)
ותחנת רכבת בת-גלים
לקריות
סטלה מאריס
מרכזית המפרץ
ותחנת רכבת לב המפרץ
שד' ההסתדרות
שד' הנשיא
מרכז הכרמל
כביש דורי
נוה שאנן
מרכז זיו (נוה שאנן)
שד' מוריה
אחוזה
מרכז חורב
רח' חורב
תחנת רכבת
חוף הכרמל (מת"מ)
דניה
שד' אבא חושי
מבית אורן, דליה ועוספיה
אוניברסיטת
חיפה
כביש פרויד
N
לתל אביב

Graphs

In the city of Königsberg Prussia (now Kaliningrad , Russia)

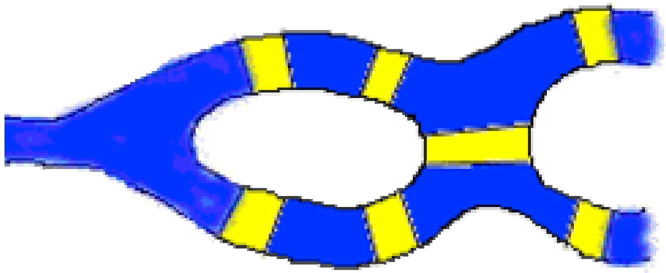


The city is divided into 4 parts with 7 interconnecting bridges.

Can one tour the city and cross all bridges exactly once and return to the starting position?

Graphs

The Seven Bridges of Königsberg.



Leonard Euler (1707 - 1783)

Types of Graphs

Definition:

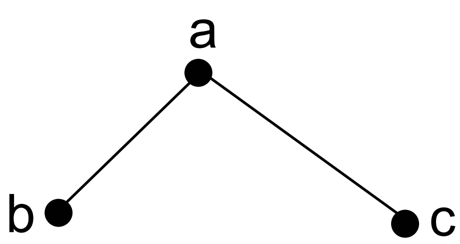
An *Undirected Graph* (גרף לא מכוון) $G = (V,E)$ consists of a nonempty finite set V and a set E of unordered pairs of elements from V.

The elements of set V are called *Vertices* (קודקודים).
The elements of Set E are called *Edges* (צלעות).

An Edge in an undirected graph is denoted $\{v1,v2\}$.

Types of Graphs

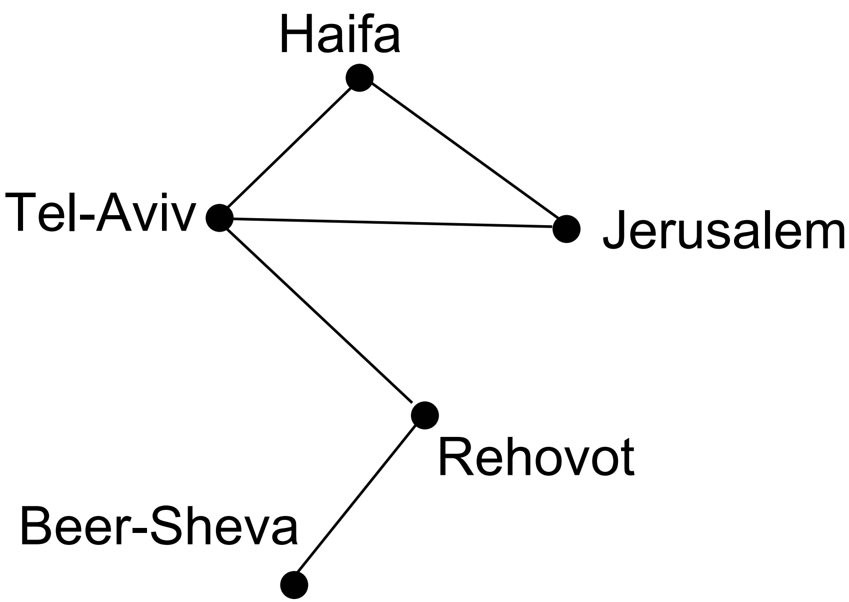
<u>Example</u>: Undirected Graph



$$G = (\underbrace{\{a,b,c\}}_{V}, \underbrace{\{ \{a,b\},\{a,c\} \}}_{E})$$

Types of Graphs

<u>Example</u>: Undirected Graph



Two-Way Communication Network

Road/Street Map

Types of Graphs

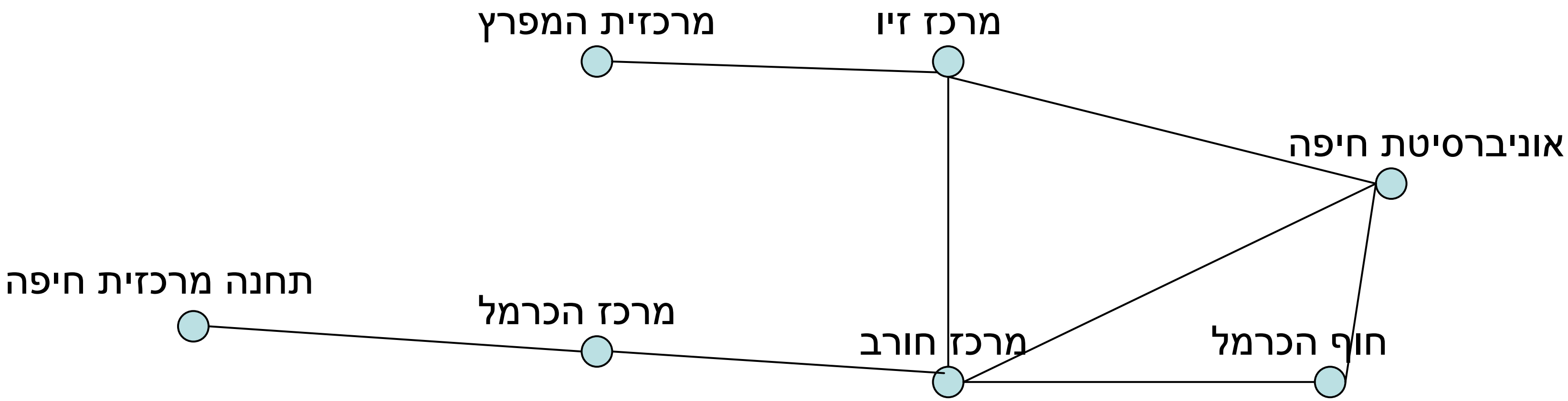
Example: Undirected Graph

V (קודקודים):

1. אוניברסיטת חיפה
2. תחנה מרכזית חיפה
3. מרכזית המפרץ
4. מרכז זיו
5. מרכז חורב
6. מרכז הכרמל
7. חוף הכרמל

E (צלעות) :

1. {אוניברסיטת חיפה , חוף הכרמל}
2. {אוניברסיטת חיפה , מרכז זיו}
3. {אוניברסיטת חיפה , מרכז חורב}
4. {מרכז זיו, מרכזית המפרץ}
5. {מרכז חורב, חוף הכרמל}
6. {מרכז חורב, מרכז הכרמל}
7. {מרכז הכרמל, תחנה מרכזית חיפה}



Types of Graphs

<u>Definition</u>:

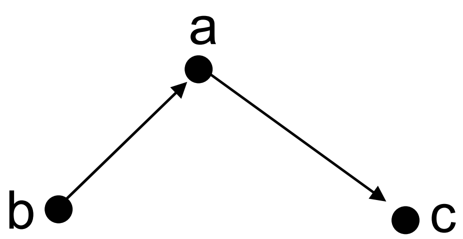
An *Directed Graph* (גרף מכוון) G = (V,E) consists of a nonempty finite set V and a set E of *ordered* pairs of elements from V.

The elements of set V are called *Vertices* (קודקודים).
The elements of Set E are called *Edges* (קשתות).

An Edge in an undirected graph is denoted (v1,v2).

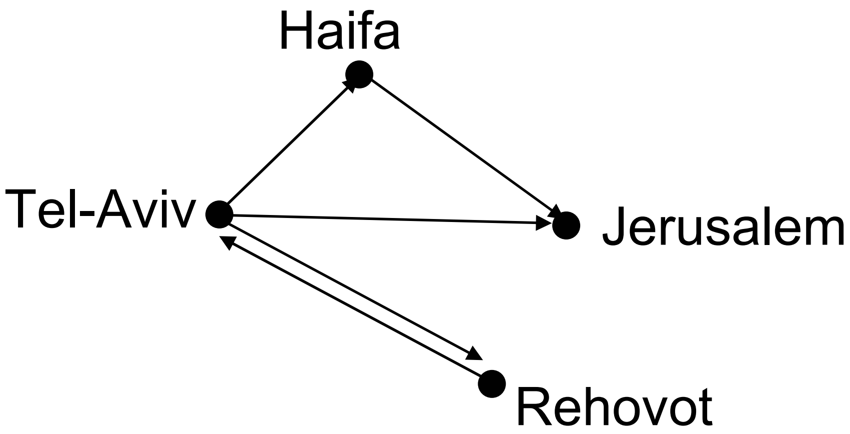
Types of Graphs

<u>Example</u>: Undirected Graph



$$G = (\underbrace{\{a,b,c\}}_{V}, \underbrace{\{\ (b,a),(a,c)\ \}}_{E})$$

<u>Example</u>: Undirected Graph



One-Way Communication Network

One way Road/Street Map

Types of Graphs

Definition:

An *Simple Graph* (גרף פשוט) G = (V,E) is a graph with

1) No 'loops' (i.e. E does not contain edges of the form (x,x) with $x \in V$).

2) There is at most one edge between every 2 vertices.

Types of Graphs

<u>Definition</u>:

An *Multigraph* (מולטי גרף) is a graph with more than one edge between 2 vertices.

G = (V,E) and a function f (or G = (V,E,f)).

V = Set of vertices

E = set of names of edges (e.g. e1,e2...)

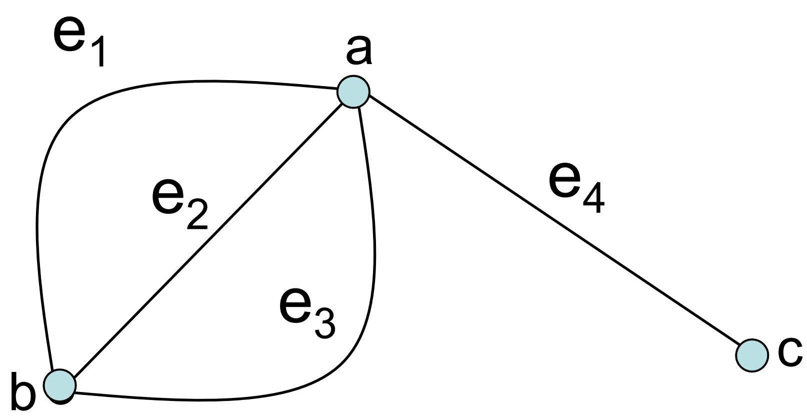
f = a function $f: E \rightarrow V \times V / \{\{v,v\} \mid v \in V\}$

which defined the vertices connected by the edge.

If $f(e_1) = f(e_2)$ then e_1 and e_2 are called *Parallel Edges* (קשתות מקבילות)

Types of Graphs

Example: Multigraph



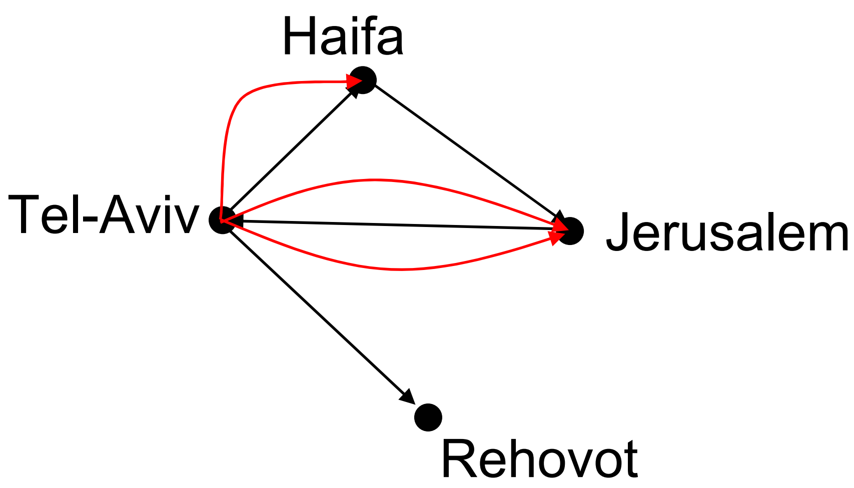
$$G = (\underbrace{\{a,b,c\}}_{V}, \underbrace{\{e_1, e_2, e_3, e_4\}}_{E}, f)$$

$f(e_1) = \{a,b\}$ $f(e_3) = \{a,b\}$

$f(e_2) = \{a,b\}$ $f(e_4) = \{a,c\}$

Types of Graphs

<u>Example</u>: Multigraph



Two-Way Communication Network

Road/Street Map

Types of Graphs

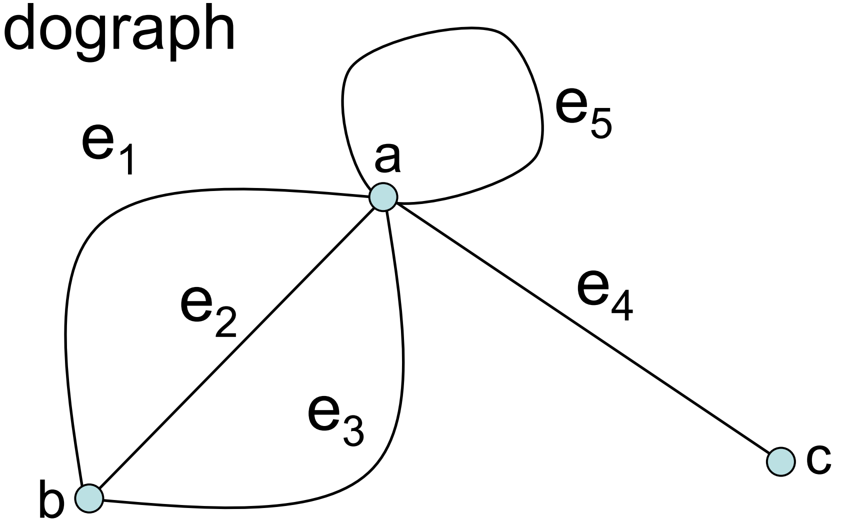
Definition:

A *Pseudograph* (פסאודו גרף) is a multigraph $G = (V,E,f)$ that may contain 'loops'.

i.e. E may contain edges $f(e) = \{v,v\} \quad v \in V$.

Types of Graphs

Example: Pseudograph



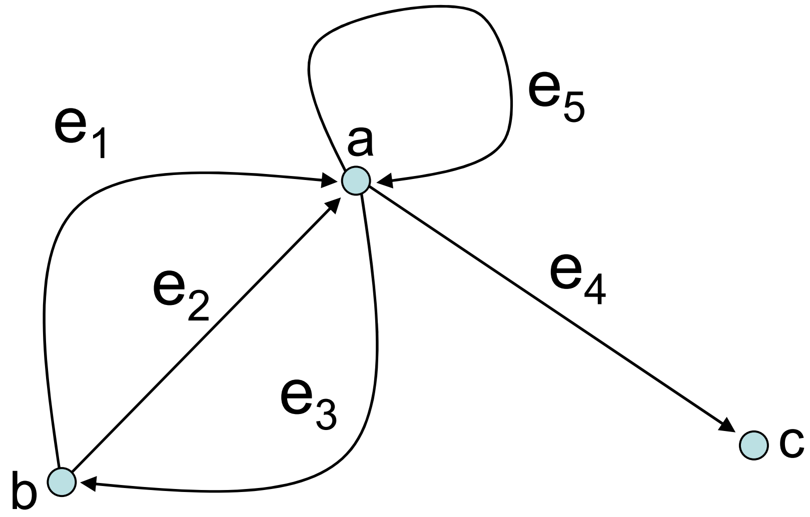
$$G = (\underbrace{\{a,b,c\}}_{V}, \underbrace{\{ e_1, e_2, e_3, e_4, e_5 \}}_{E}, f)$$

$f(e_1) = \{a,b\}$ $f(e_3) = \{a,b\}$ $f(e_5) = \{a,a\}$

$f(e_2) = \{a,b\}$ $f(e_4) = \{a,c\}$

Types of Graphs

Example: Directed pseudograph



$$G = (\underbrace{\{a,b,c\}}_{V},\ \underbrace{\{ e_1, e_2, e_3, e_4, e_5 \}}_{E},\ f)$$

$f(e_1) = (b,a)$ $f(e_3) = (a,b)$ $f(e_5) = (a,a)$

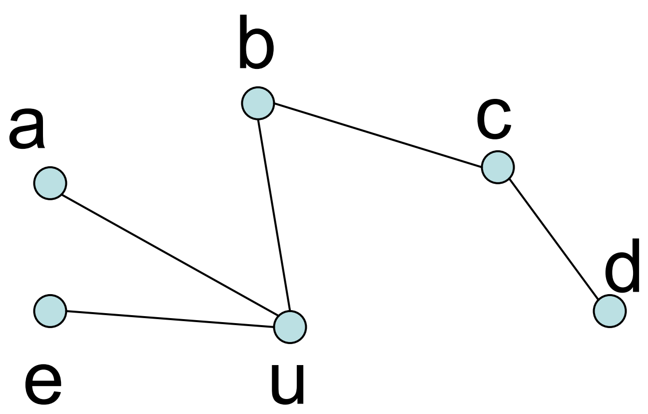
$f(e_2) = (b,a)$ $f(e_4) = (a,c)$

Terminology of Graphs

<u>Definition</u>:

Two vertices u,v in an undirected graph G={V,E} are called *adjacent* (שכנים) if there is an edge between u and v: $\{u,v\} \in E$.

Denote by $\mu(a)$ the set of all vertices adjacent to a.



u is adjacent to b.

u is not adjacent to c.

$\mu(u) = \{a,b,e\}$

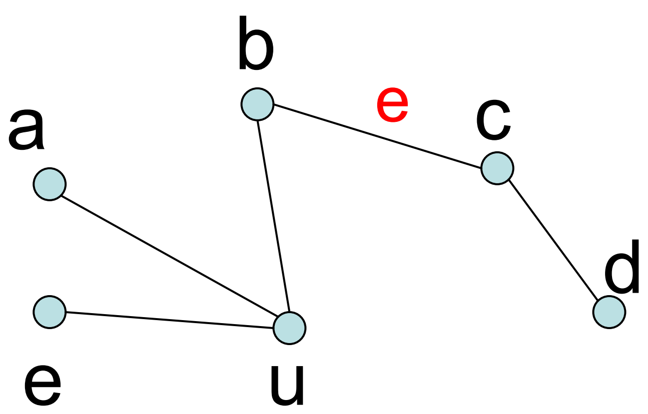
Terminology of Graphs

<u>Definition</u>:

Let e ={u,v} be an edge in an undirected graph.

Then, e *connects* (מקשר) between u and v.

u and v are the *endpoints* (נקודות קצה) of e.



e = {b,c}

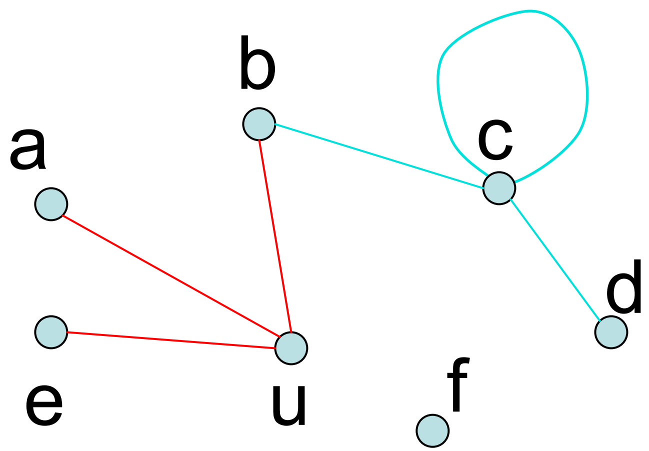
e connects between b and c.

b and c are the endpoints of e.

Terminology of Graphs

<u>Definition</u>:

The *degree* (דרגה) of a vertex v in an undirected graph is the number of edges that connect with v (a loop edge counts as 2). $\deg(v) = |\mu(v)|$



deg (u) = 3

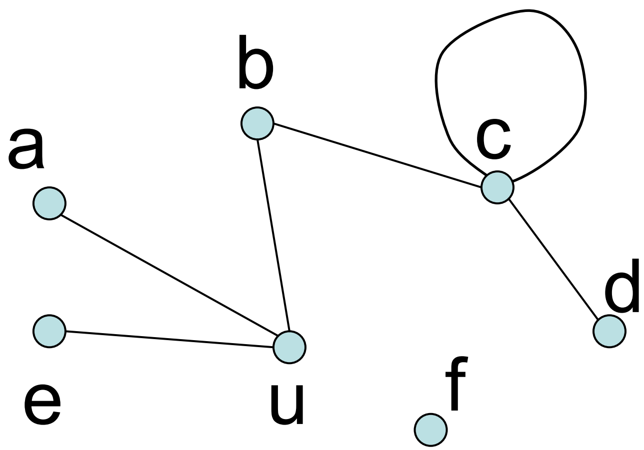
deg (c) = 4

deg (f) = 0

Terminology of Graphs

A vertex with deg(v) = 0 is called *isolated* (מבודד)

A vertex with deg(v) = 1 is called *pendant* (תלוי)



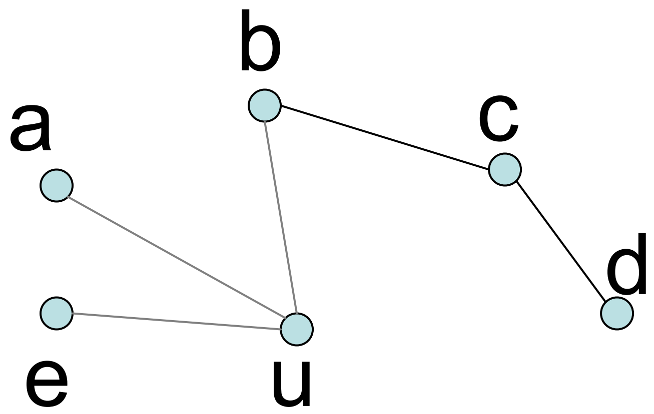
f is isolated

a,d,e are pendant

Terminology of Graphs

Theorem: The Handshaking Theorem

In an undirected graph $G = \{V,E\}$: $2*|E| = \sum_{v \in V} deg(v)$



$\sum deg(v) = 10$

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

Terminology of Graphs

<u>Definition</u>:

Let e = (u,v) be an edge in a **directed** graph.

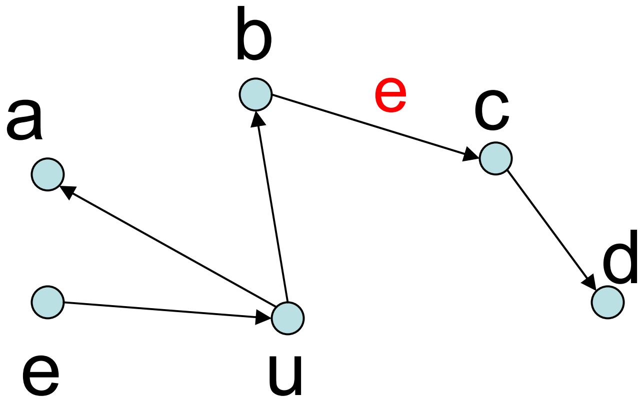
Then, u is *adjacent to* (שכן ל-) v

and v is *adjacent from* (שכן מ-) u.

and v is u and v are the *endpoints* (נקודות קצה) of e.

Edge e *from* u *to* v (יוצאת\מתחילה ב-u ונכנסת\מסתיימת ב-v).

u is the *initial vertex* (מקור), v is the *end/terminal* (יעד)



e = (b,c)

e connects from b to c.

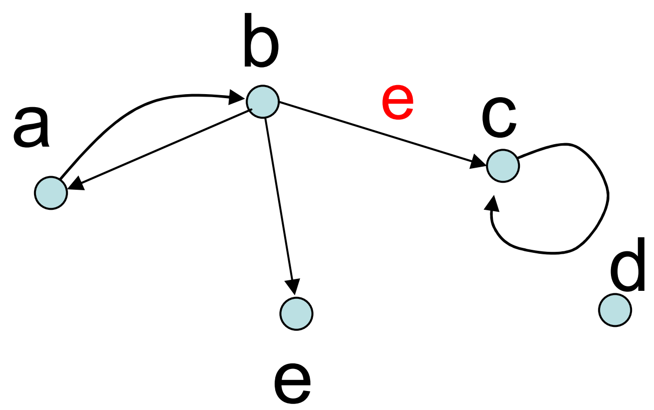
Terminology of Graphs

<u>Definition</u>:

In a directed graph $G = \{V,E\}$,

The *in-degree* (דרגת כניסה) of a vertex v $\deg^-(v)$ equals the number of edges with v as its terminal.

The *out-degree* (דרגת יציאה) of a vertex v $\deg^+(v)$ equals the number of edges with v as its initial vertex.



$\deg^-(a) = 1$ $\deg^+(a) = 1$

$\deg^-(b) = 1$ $\deg^+(b) = 3$

$\deg^-(c) = 2$ $\deg^+(c) = 1$

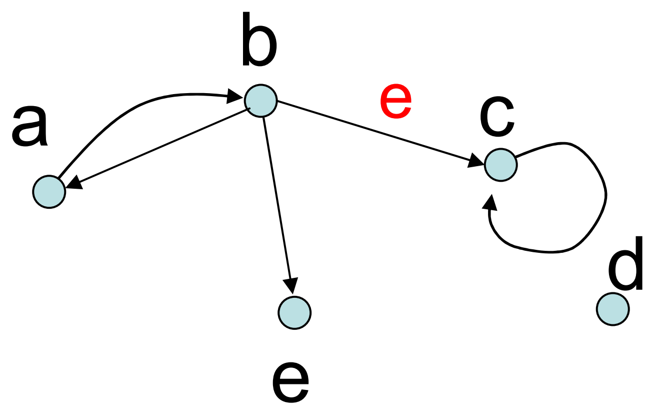
$\deg^-(e) = 1$ $\deg^+(e) = 0$

Terminology of Graphs

<u>Definition</u>:

In a directed graph G = {V,E},

The *degree* (דרגה) of a vertex v is the sum of the in-degree and out-degree. $deg(a) = deg^{-}(a)+deg^{+}(A)$

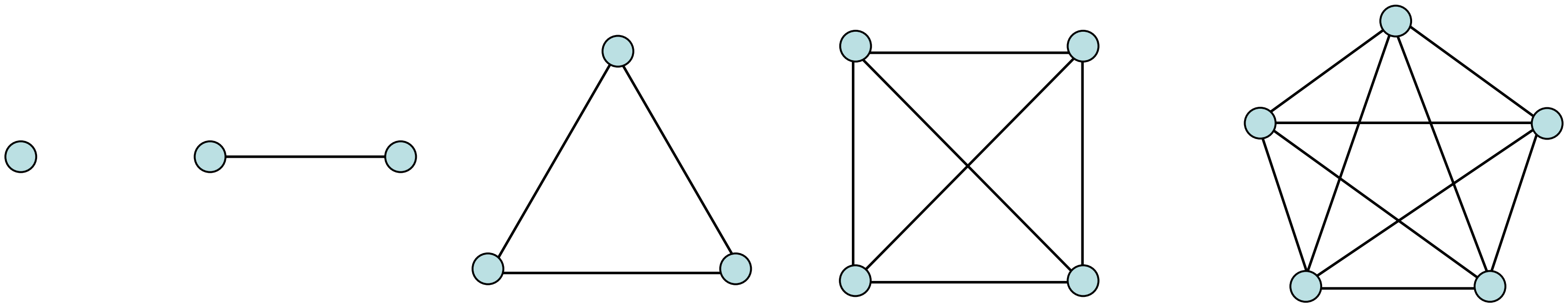


$deg(b) =$

$deg^{-}(b) + deg+(b) = 1+3 = 4$

Examples of Graphs

Complete Graph (גרף מלא) with n vertices K_n is a simple graph with an edge between every two vertices.

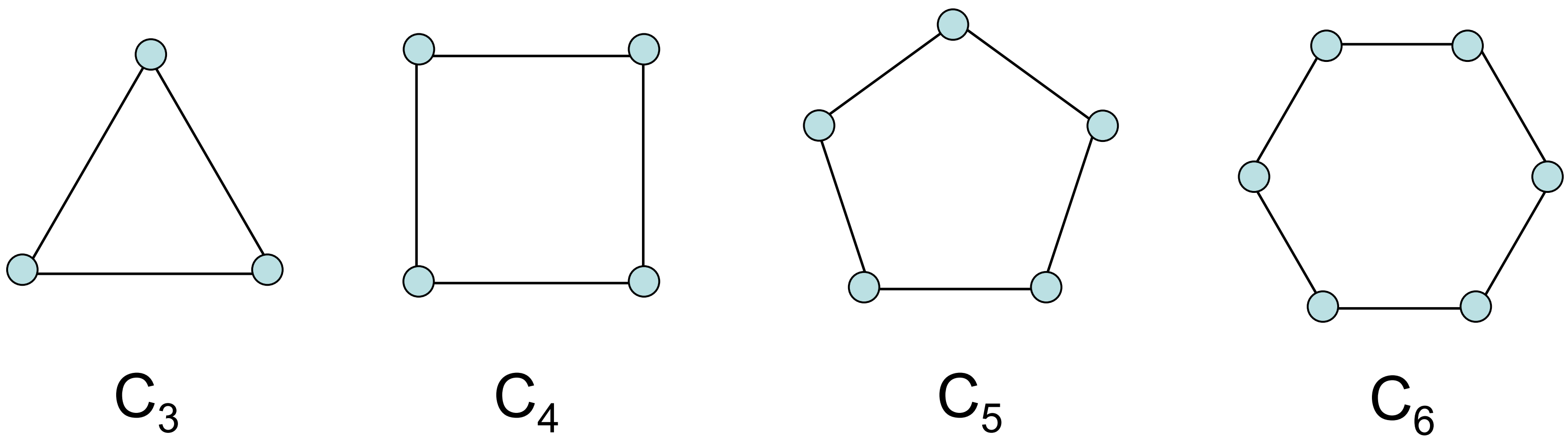


K_1 K_2 K_3 K_4 K_5

Examples of Graphs

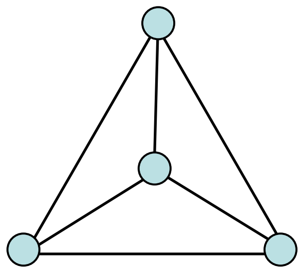
A *Cycle* (גרף מעגלי) with n vertices $n \geq 3$ C_n is a simple graph with vertices: $\{v_1, v_2, \ldots, v_n\}$ and edges:

$$\{ \{v_1,v_2\}, \{v_2,v_3\}, \ldots, \{v_{n-1},v_n\}, \{v_n,v_1\} \}.$$

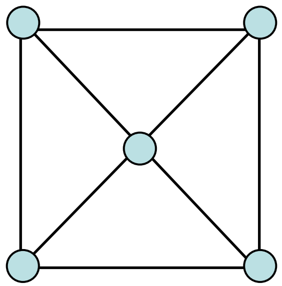


Examples of Graphs

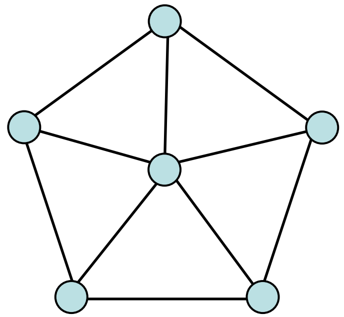
A ***Wheel*** (גרף גלגל) with n vertices $n \geq 3$ W_n is a Cycle with an additional vertex connected to all other vertices.



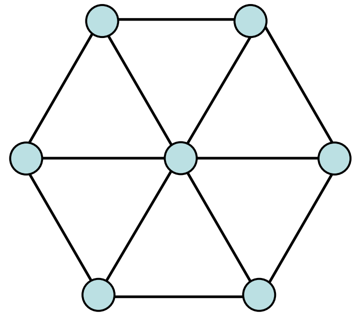
W_3



W_4



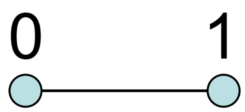
W_5



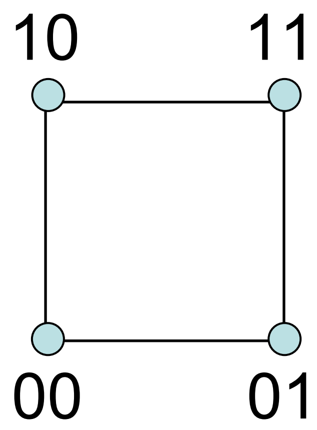
W_6

Examples of Graphs

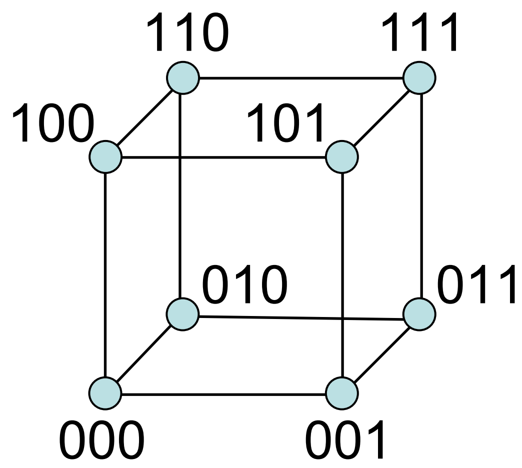
A *n-Cube* (גרף קובי) with n vertices Q_n is the graph whose vertices represent the 2^n Bit-strings of length n. Two vertices are adjacent iff their bit-strings differ by exactly 1 bit.



Q_1



Q_2



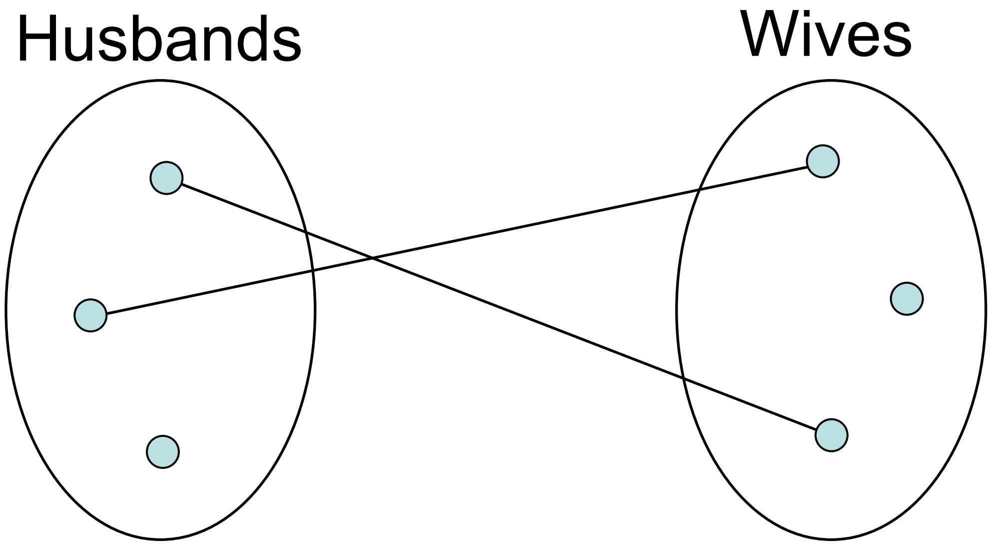
Q_3

Bipartite Graphs

G = {V,E}

Vertices are people in a village.

An edge connects between married partners.

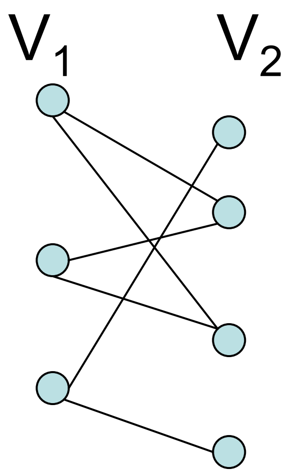


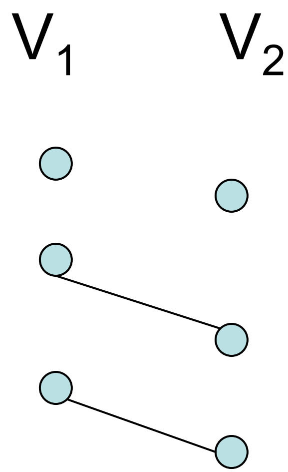
Vertices can be divided into 2 sets with edges only between sets.

Bipartite Graphs

<u>Definition</u>:

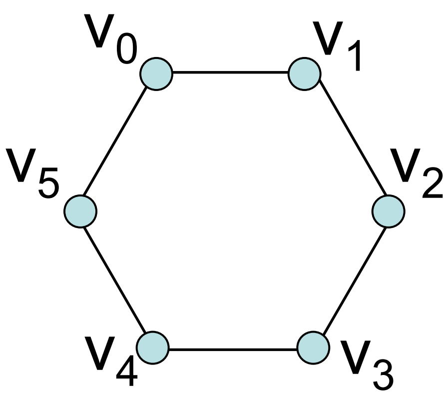
A simple graph $G=\{V,E\}$ is called *Bipartite* (גרף דו-צדדי) if V can be partitioned into disjoint sets V_1, V_2 s.t. every edge in E connects between a vertex in V_1 and a vertex in V_2. (No edges connect between vertices in V_i)





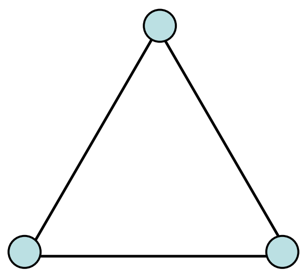
Bipartite Graphs

Examples:



C_6 is bipartite

$V = \{v_0,v_2,v_4\} \cup \{v_1,v_3,v_5\}$

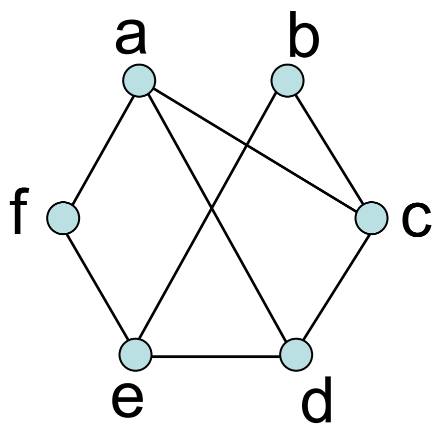


K_3 is **not** bipartite

Every partition will have a subset with 2 vertices, in which, since the graph is complete, there will be an edge.

Bipartite Graphs

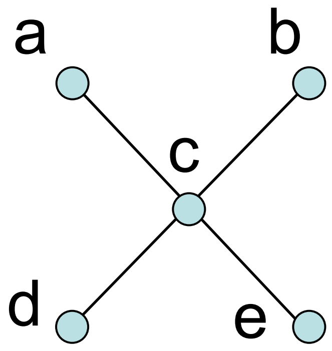
Examples:



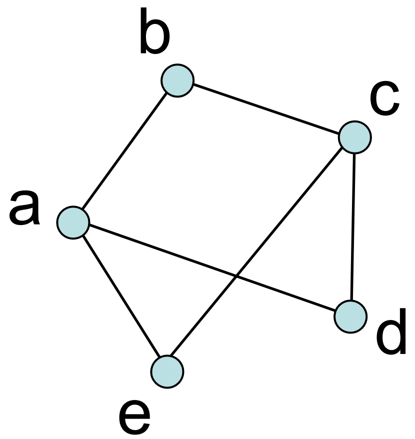
Bipartite ? No: {a,b,c} forms a K_3

Bipartite Graphs

<u>Examples:</u>



Bipartite ? Yes: $V = \{c\} \cup \{a,b,d,e\}$

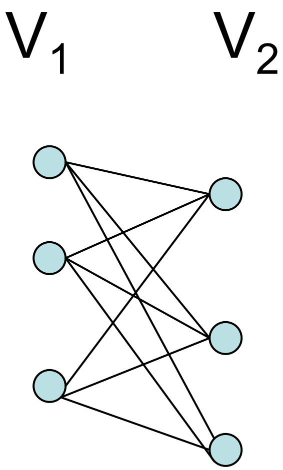


Bipartite ? Yes: $V = \{b,e,d\} \cup \{a,c\}$

Bipartite Graphs

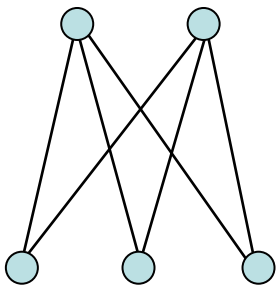
<u>Definition</u>:

A graph G={V,E} is a Complete *Bipartite* Graph $K_{n,m}$ (גרף דו-צדדי שלם) if V ca be partitioned into V_1 and V_2 $|V_1| = n$ $|V_2| = m$ and an edge is in the graph <u>**iff**</u> it connects a vertex in V_1 with a vertex in V_2.

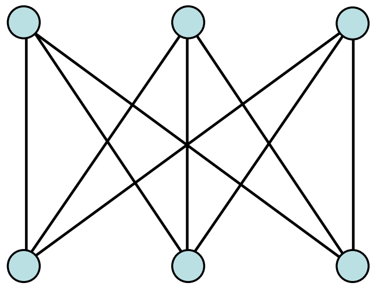


Bipartite Graphs

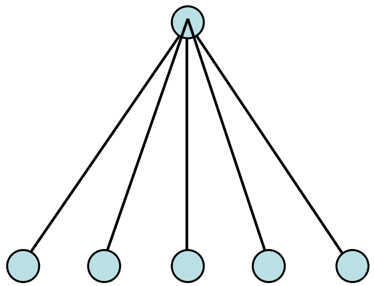
<u>Examples:</u>



$K_{2,3}$



$K_{3,3}$



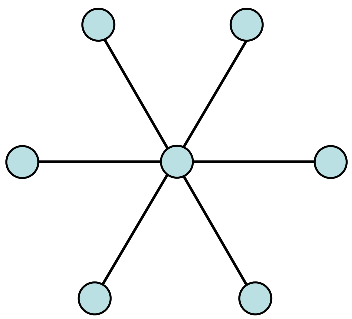
$K_{1,5}$

Examples of Graphs

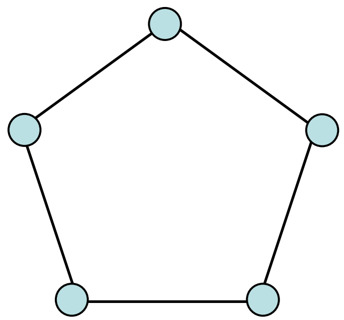
Local Area Networks (LAN) -

a network connecting computers and peripherals.

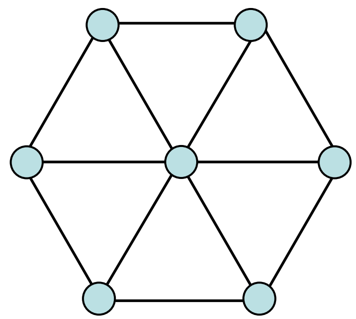
Often there is a server with which all others communicate.



$K_{1,n}$



C_n



W_n

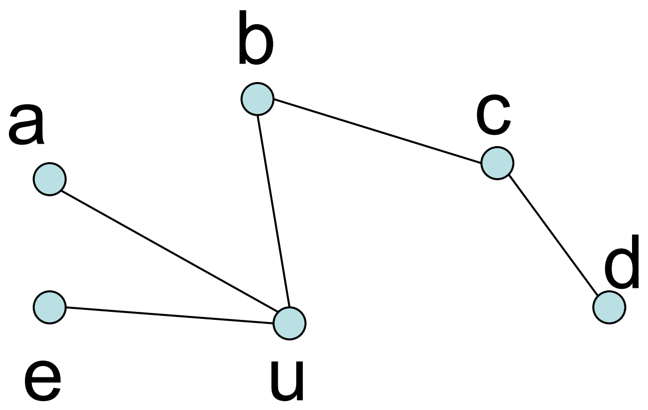
Creating New Graphs

<u>Definition</u>:

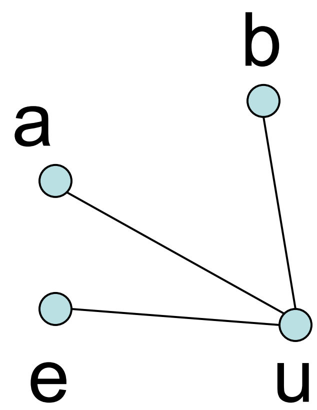
A *subgraph* (תת-גרף) of a graph G={V,E} is a graph

$G' = (V', E')$ s.t. $V' \subseteq V$ and $E' \subseteq E$.

G



Subgraph of G

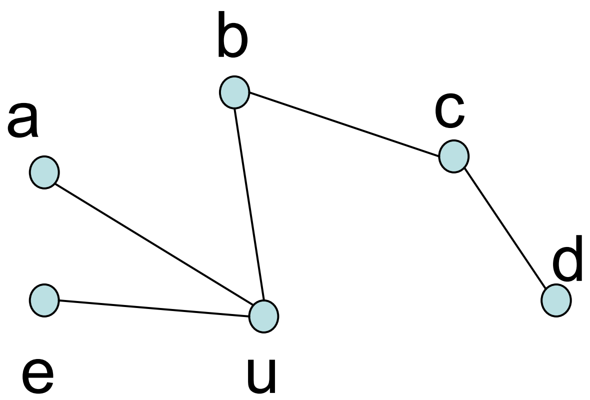


Creating New Graphs

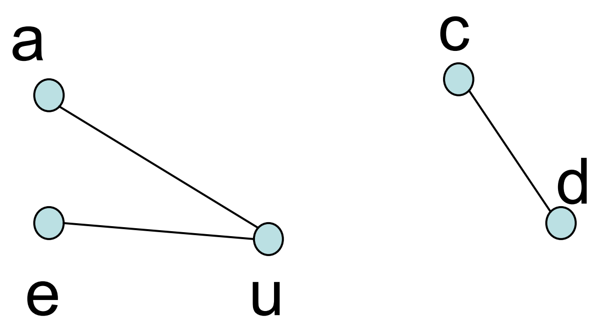
Definition:

A subgraph of a graph G={V,E} can be obtained by deleting a set of vertices and all their connecting edges. G\S where $S \subseteq V$

G



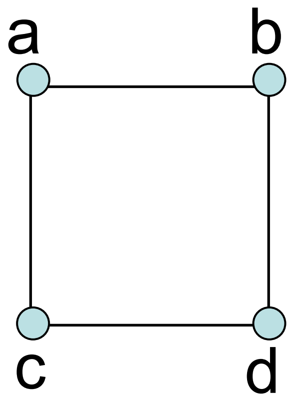
G \ {b} = Subgraph of G



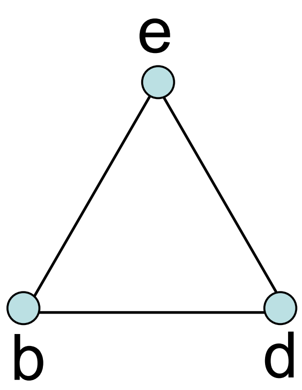
Creating New Graphs

<u>Definition</u>:

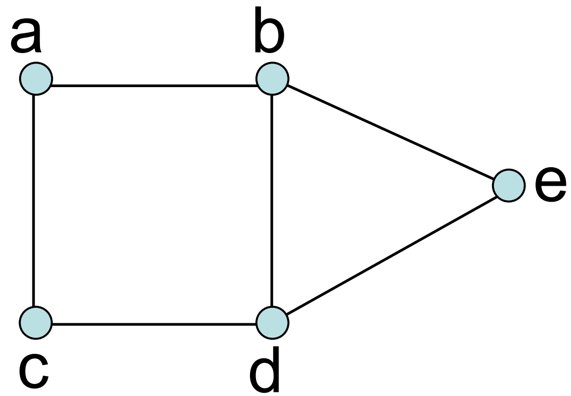
The union of 2 graphs G1={V1,E1} and G2 = {V2,E2} is the graph $G1 \cup G2 = (V1 \cup V2, E1 \cup E2)$.



G_1



G_2



$G_1 \cup G_2$

Representing Graphs

$G = (V,E)$ $\longleftrightarrow$ $R = \{ (a,b) \mid \{a,b\} \in E \}$

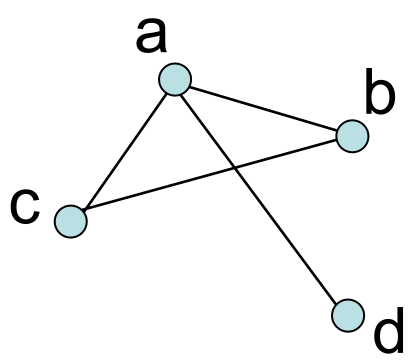
Adjacency Matrix of G — Matrix representation of R

$$\begin{array}{cc} & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

$$\text{Entry } (i,j) = \begin{cases} 1 \text{ if } \{v_i, v_j\} \in E \\ 0 \text{ otherwise} \end{cases}$$

Representing Graphs

<u>Example:</u>



$$
\begin{array}{c|c}
 & \begin{array}{cccc} a & b & c & d \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}
\end{array}
$$

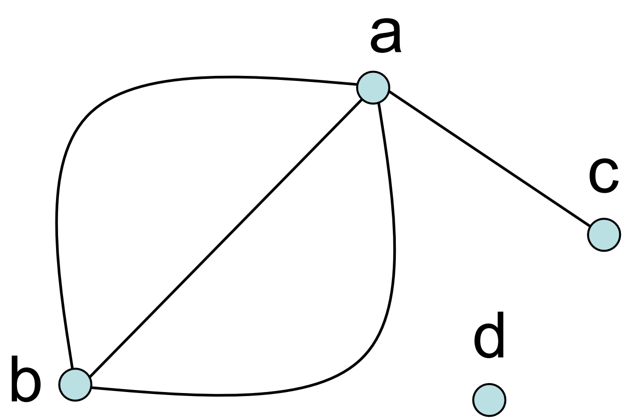
Representing Graphs

Undirected Graphs	⟶	Symmetric Adjacency matrix
Directed Graphs	⟶	Not necessarily Symmetric
Simple Graphs	⟶	0 on the diagonal
Multigraphs	⟶	Matrix entries can be > 1
Pseudo Graphs	⟶	diagonal may be > 0

Representing Graphs

<u>Example:</u>

Multigraph

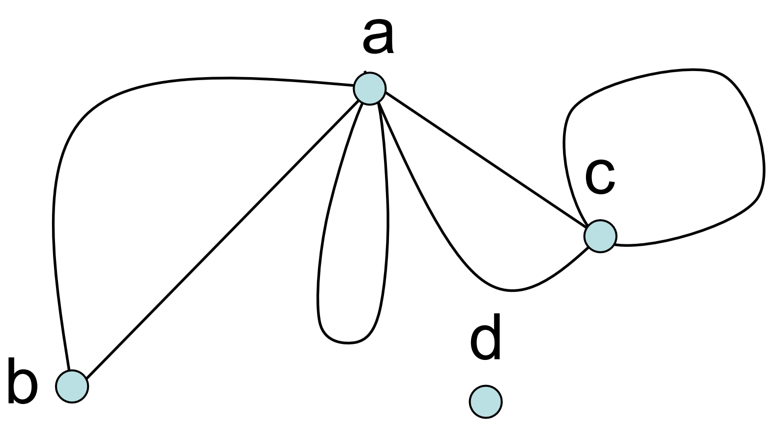


$$\begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 0 & 3 & 1 & 0 \\ 3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Representing Graphs

<u>Example:</u>

Pseudograph

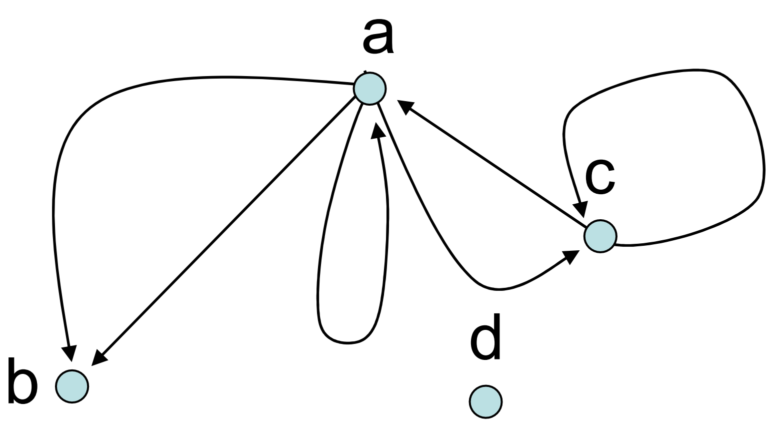


$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 1 & 2 & 2 & 0 \\ 2 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Representing Graphs

<u>Example:</u>

Directed Pseudograph



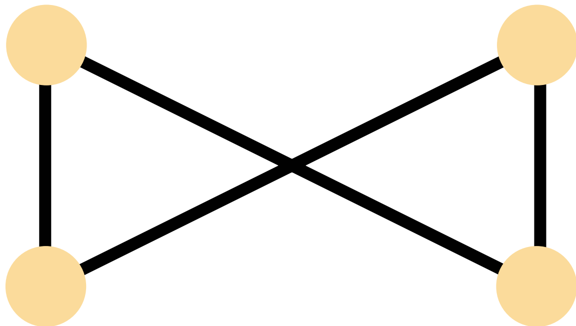
$$\begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Entry (i,j) = 1 if (i,j) $\in$ E

Graph Isomorphism

Molecule - stable state

Molecule - reactive state



Same Graph ?

Graph Isomorphism

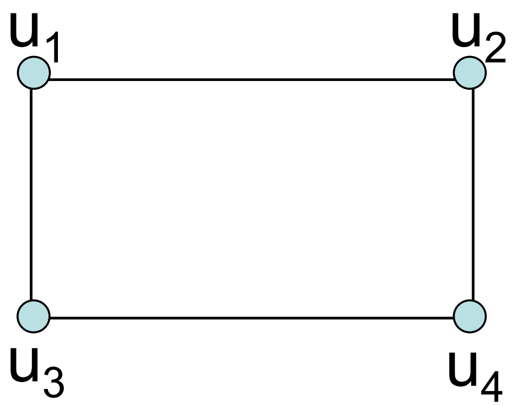
Definition:

Two simple graphs $G_1 = (V_1,E_1)$, $G_2 = (V_2,E_2)$ are *Isomorphic* (איזומורפים) if there is a 1:1 and onto function from V_1 to V_2 s.t. a,b are adjacent in G_1 **iff** f(a), f(b) are adjacent in G_2.

Function f is called an *Isomorphism*.

Graph Isomorphism

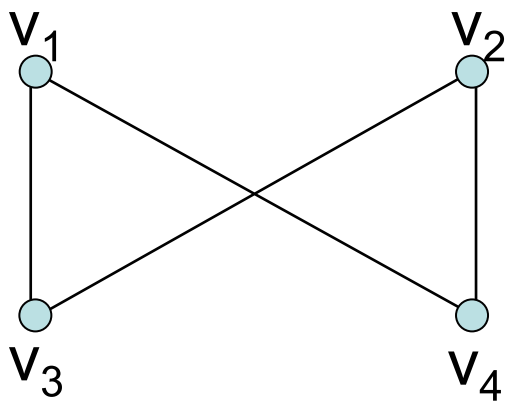
<u>Example:</u>



$G_1 = (U,E_1)$

$f : U \rightarrow V$

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



$G_2 = (V,E_2)$

f is 1:1 and onto

f is adj. preserving

Graph Isomorphism

Proving Isomorphism of graphs: HARD!

n! possible 1:1 and onto functions between V_1 and V_2 ($|V_i|=n$)

Contradicting Isomorphism may be easier.

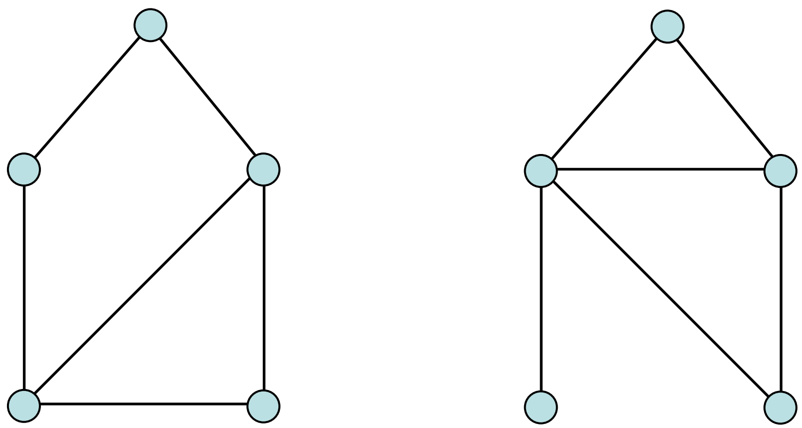
Invariant properties under isomorphism:

- number of vertices $|V_1| = |V_2|$
- number of edges $|E_1| = |E_2|$
- vertex degree $deg(v_i) = deg(f(v_i))$

Graph Isomorphism

<u>Example:</u>

G1=(V1,E1)   G2=(V2,E2)

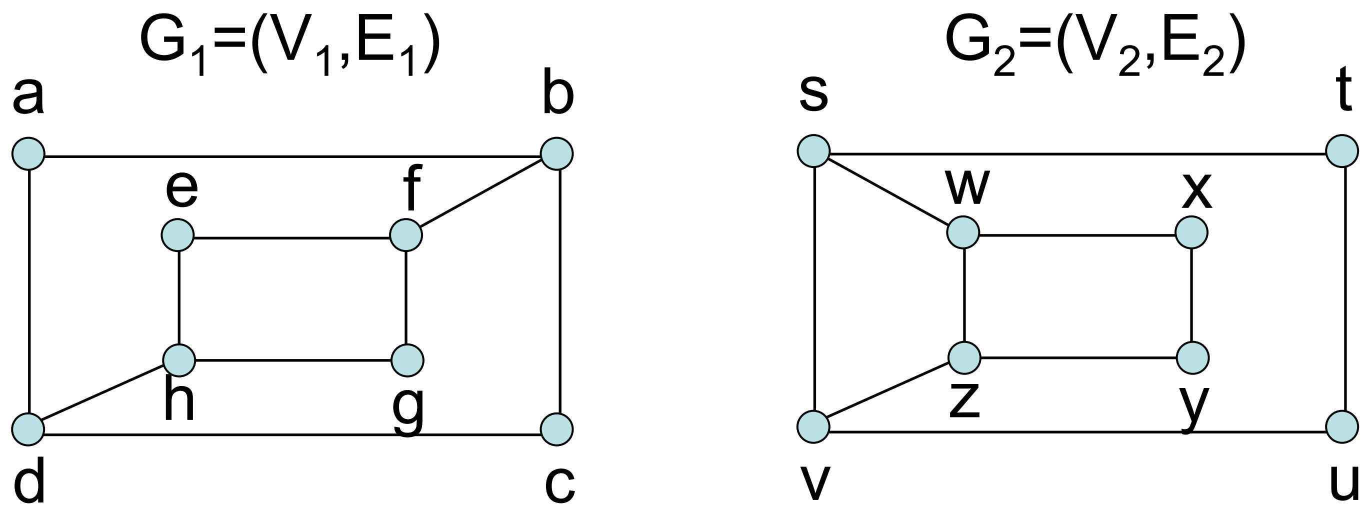


G1 and G2 are **NOT** Isomporphic. |V1|=|V2|, |E1|=|E2|

but G2 has a vertex with deg = 1, and G1 does not.

Graph Isomorphism

Example:



$|V_1|=|V_2| = 8$

$|E_1|=|E_2| = 10$

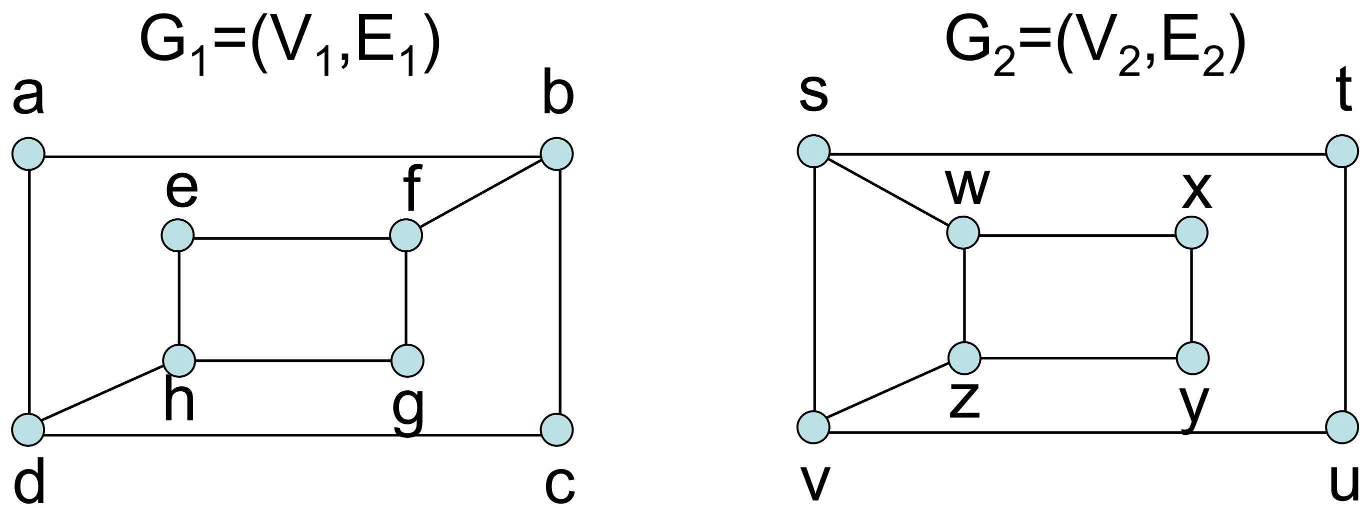
4 vertices of deg = 4

4 vertices of deg = 3

Are G_1 and G_2 Isomprphic ?

Graph Isomorphism

<u>Example Cont:</u>



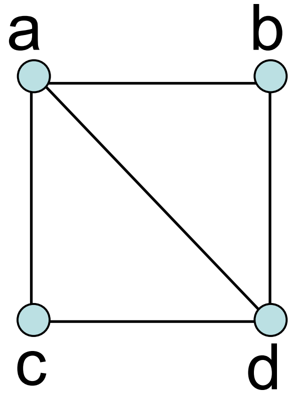
Answer: G_1 and G_2 are NOT Isomorphic !

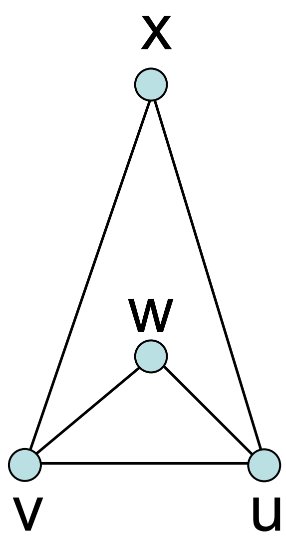
Proof: deg(a) = 2 so $f(a) \in \{t,u,x,y\}$

$\mu(a) = \{b,d\}$ and deg(b)=deg(d)=3.

However t,u,x,y do not have 2 neighbors of deg=3.

Graph Isomorphism





f(a) = u

f(b) = x

f(c) = w

f(d) = v

$$
\begin{array}{c}
\\ a \\ b \\ c \\ d
\end{array}
\begin{array}{c}
\begin{array}{cccc} a & b & c & d \end{array} \\
\begin{pmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0
\end{pmatrix}
\end{array}
\qquad
\begin{array}{c}
\\ u \\ v \\ w \\ x
\end{array}
\begin{array}{c}
\begin{array}{cccc} u & v & w & x \end{array} \\
\begin{pmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0
\end{pmatrix}
\end{array}
$$

Graph Isomorphism

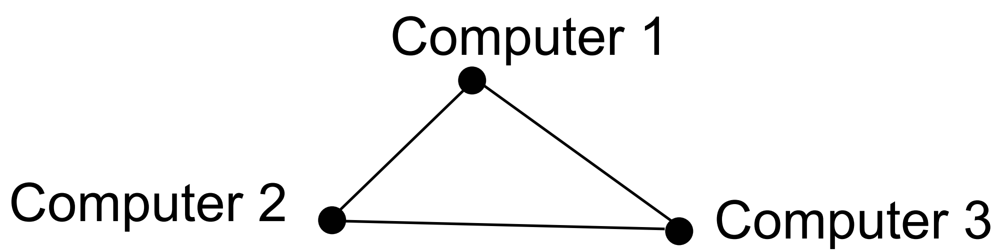
Isomorphism = permutation of the adjacency matrix.

$$
\begin{array}{c|c}
 & \begin{array}{cccc} u & v & w & x \end{array} \\
\begin{array}{c} u \\ v \\ w \\ x \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}
\end{array}
\quad \longrightarrow \quad
\begin{array}{c|c}
 & \begin{array}{cccc} u & v & w & x \end{array} \\
\begin{array}{c} f(a)=u \\ f(b)=x \\ f(c)=w \\ f(d)=v \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}
\end{array}
$$

$$
\begin{array}{c|c}
 & \begin{array}{cccc} f(a) & f(b) & f(c) & f(d) \\ u & v & w & x \end{array} \\
\begin{array}{c} f(a)=u \\ f(b)=x \\ f(c)=w \\ f(d)=v \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}
\end{array}
\quad = \quad
\begin{array}{c|c}
 & \begin{array}{cccc} a & b & c & d \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}
\end{array}
$$

Graph Connectivity

- Are two computers in a network connected?
- What are all the paths that an email may take to reach its destination?
- What is the shortest/fastest connection between two computers?



Is there a "path" between 2 vertices in a graph?

Are two vertices in a graph "connected" via edges?

Graph Connectivity

<u>Definition</u>:

A *path* (מסלול) of length n from vertex u to vertex v in a graph G={V,E} is a sequence of n edges in E:

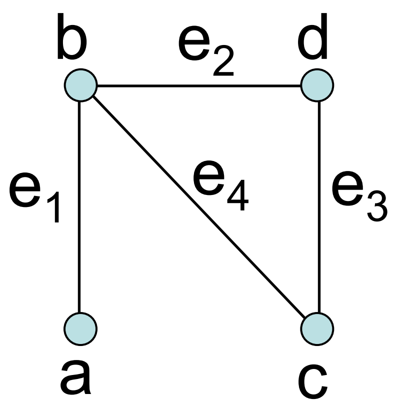
$e_1=\{x_0,x_1\}$, $e_2=\{x_1,x_2\}$, ... , $e_n=\{x_{n-1},x_n\}$

s.t. $x_0=u$ and $x_1=v$.

In a simple graph the path can be represented by a sequence of n+1 vertices: $x_0=u, x_1, \ldots, x_n=v$

Graph Connectivity

<u>Example:</u>



There is a path between a and d:

of length 2: e1,e2 $\equiv$ a,b,d

of length 3: e1,e4,e3 $\equiv$ a,b,c,d

of length 4: e1,e4,e4,e2 $\equiv$ a,b,c,b,d

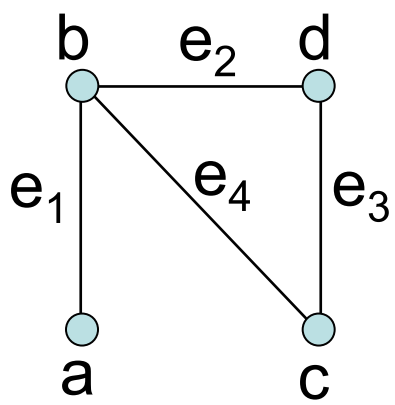
a,c,d - is not a path

A path of length 0: __ $\equiv$ a

Graph Connectivity

<u>Definition</u>:

A *circuit* (מעגל) in a graph is a path from a node to itself.



circuit from b to b of length 3:

e2,e3,e4 $\equiv$ b,d,c,b

e4,e3,e2 $\equiv$ b,c,d,b

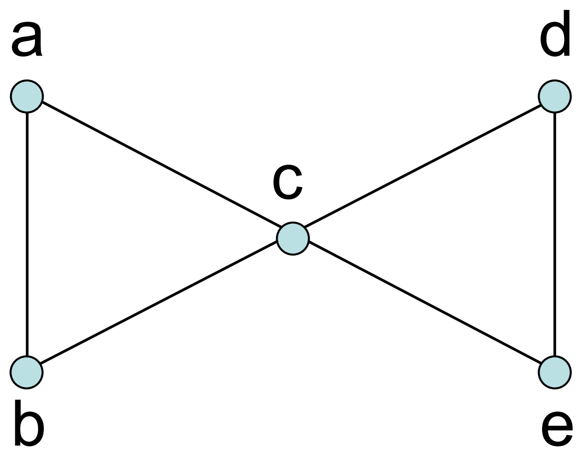
circuit from b to b of length 6:

e2,e3,e4,e2,e3,e4 $\equiv$ b,d,c,b,d,c,b

Graph Connectivity

<u>Definition</u>:

A path or circuit in a graph is called *simple* (פשוט) if it does not pass through a vertex more than once. (There is no ‘loop’ in the path).



abcd - simple path

abca - simple circuit

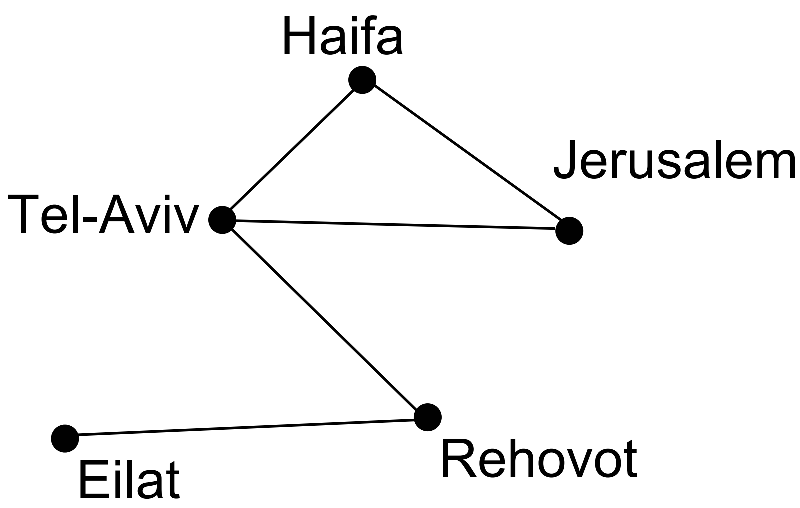
abcdecb - not simple

contains a ‘loop’

Graph Connectivity

<u>Definition</u>:

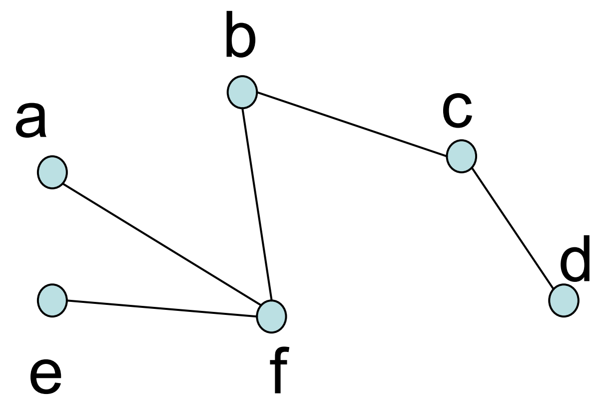
An undirected graph is called *connected* (קשיר) if there is a path between every two distinct vertices.



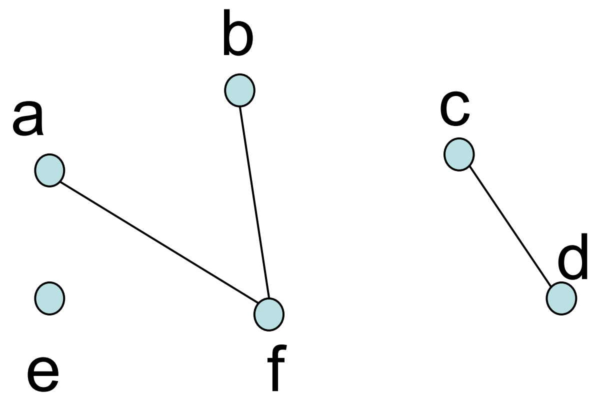
Are every 2 cities connected?

Graph Connectivity

<u>Example:</u>



Connected



Not Connected

Graph Connectivity

Theorem: In a connected undirected graph there is a simple path between every 2 distinct vertices.

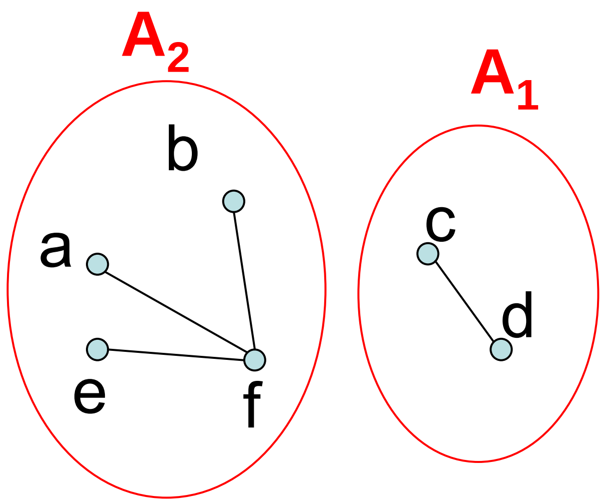
Proof: assume a, b are vertices. Since G is connected there is a path between a and b. Let $x_0,\ldots,x_n$ be the shortest of these paths, then it is necessarily simple. If not, then there is a vertex that appears twice in the path: $x_i = x_j$. But then $x_0,\ldots x_i,x_{j+1},\ldots,x_n$ is a path from a to b which is shorter - contradiction!

Graph Connectivity

G = (V,E) - an undirected graph.

Define a relation R on V : aRb iff there is a path from a to b.

Is R an equivalence relation?

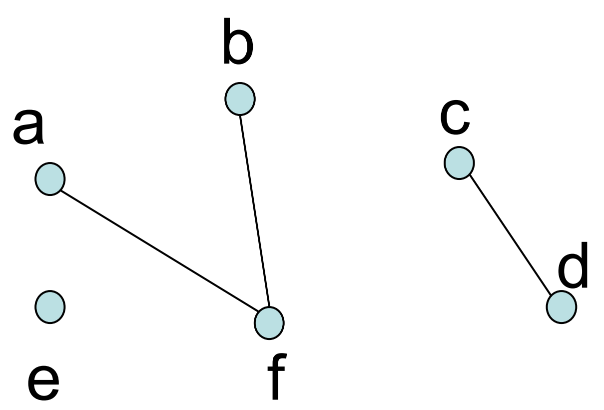


R induces a partition on V.
Every equivalence class is a connected subgraph.

Graph Connectivity

<u>Definition</u>:

A graph that is not connected is a union of two or more disjoint connected subgraphs. These subgraphs are called *connected components* (רכיבי קשירות).



Not Connected - 3 connected components

Graph Connectivity - Directed Graphs

Definition:

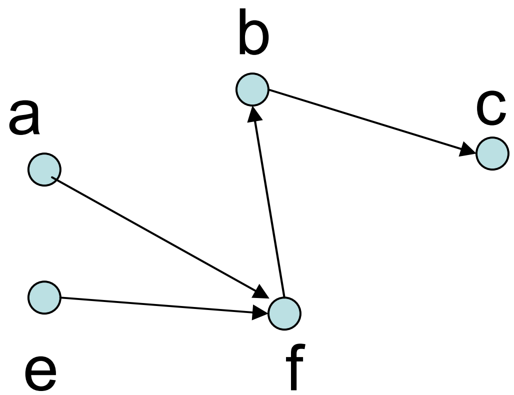
A *path/circuit* (מסלול/מעגל) of length n from vertex u to vertex v in a **directed** graph G={V,E} is a sequence of n edges in E:

$e_1=(x_0,x_1)$, $e_2=(x_1,x_2)$, ... , $e_n=(x_{n-1},x_n)$

s.t. $x_0=u$ and $x_1=v$.

Graph Connectivity - Directed Graphs

Example:



There is a path from a to b

There is **no** path from b to a

There is a path from c to f in the underlying undirected graph.

Graph Connectivity - Directed Graphs

Definition:

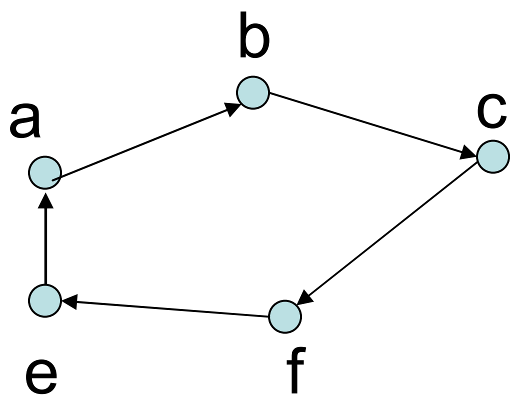
A directed graph is called *strongly connected* (קשיר חזק) if there is a path between every two distinct vertices.

Definition:

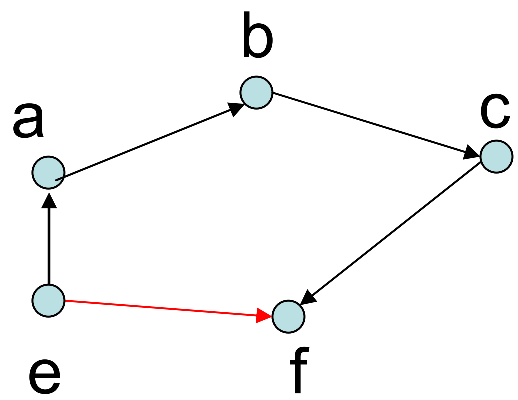
A directed graph is called *weakly connected* (קשיר חלש) if there is a path between every two distinct vertices in the underlying undirected graph.

Graph Connectivity - Directed Graphs

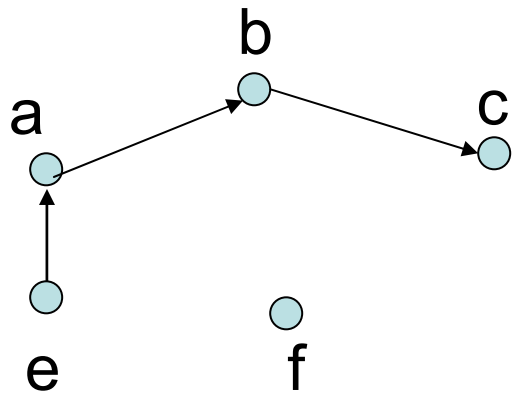
Examples:



Strongly Connected



Weakly Connected



Not Connected

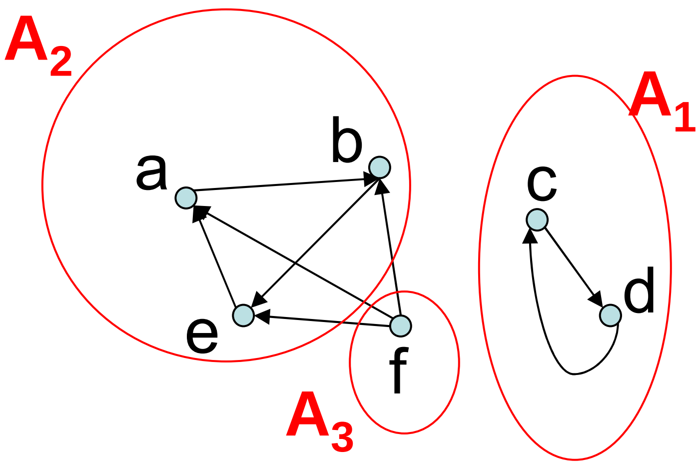
Graph Connectivity - Directed Graphs

G = (V,E) - a directed graph.

Define a relation R on V :

aRb iff there are paths from a to b and from b to a.

Is R an equivalence relation?



R induces a partition on V.

Every equivalence class is

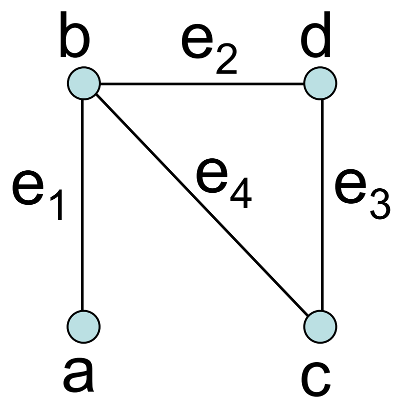
a strongly connected subgraph.

Called a *connected component.*

Graph Connectivity

How many paths of length n exist between 2 vertices in a graph?

Example:



Between a and b there are 3 paths of length 3:

abcb abab abdb

Graph Connectivity

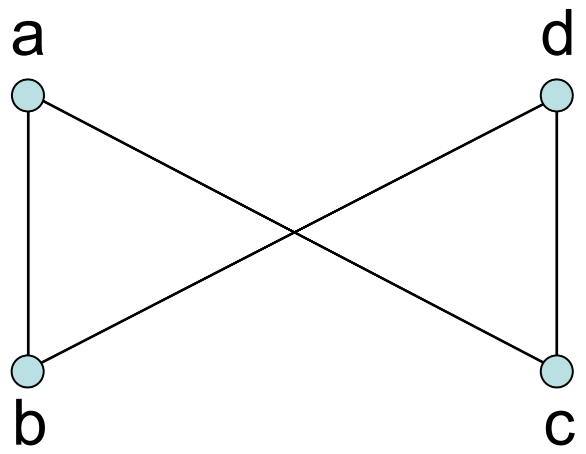
In Relations: There is a path of length n between a and b iff $(a,b) \in R^n$.

Theorem: Let G be a graph with adjacency matrix M. The number of different paths of length n between vertex i and vertex j equals the (i,j)-th entry of M^n.

Proof: By induction on n.

Graph Connectivity

<u>Example:</u> How many paths of length 4 exist from a to d ?



$$M = \begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

$$M^4 = \begin{pmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{pmatrix}$$

There are 8 paths from a to d.

abdbd abdcd ababd abacd

acdbd acdcd acabd acacd

Graph Connectivity

Questions:

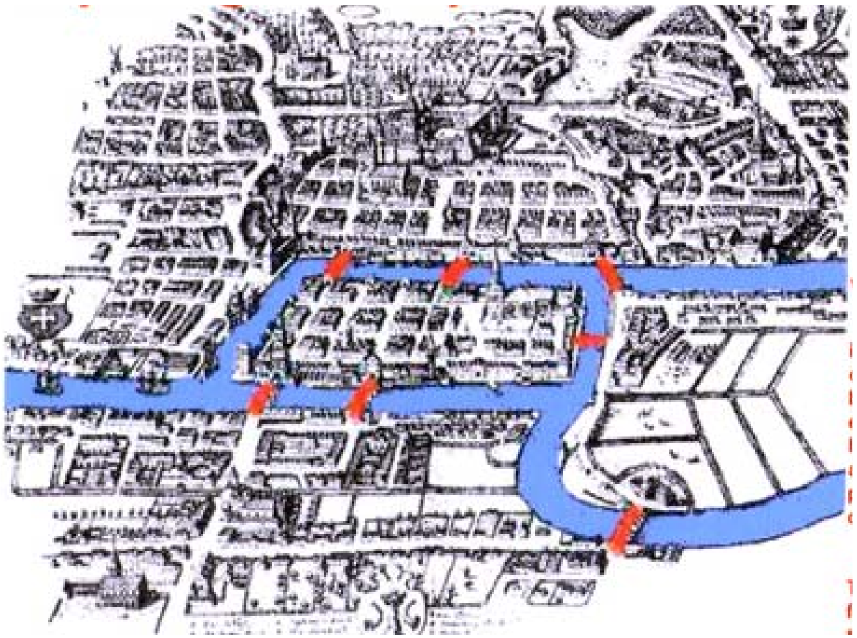
Describe an algorithm to find the length of the shortest path between vertices i and j.

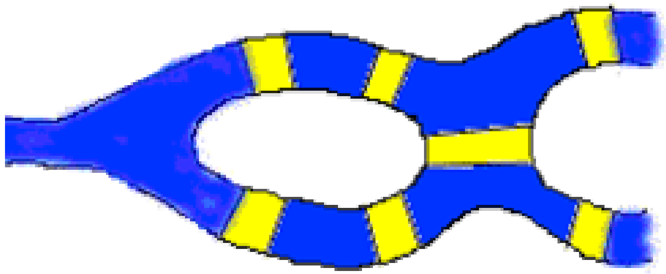
Describe an algorithm to determine whether a graph is connected.

Euler & Hamilton paths

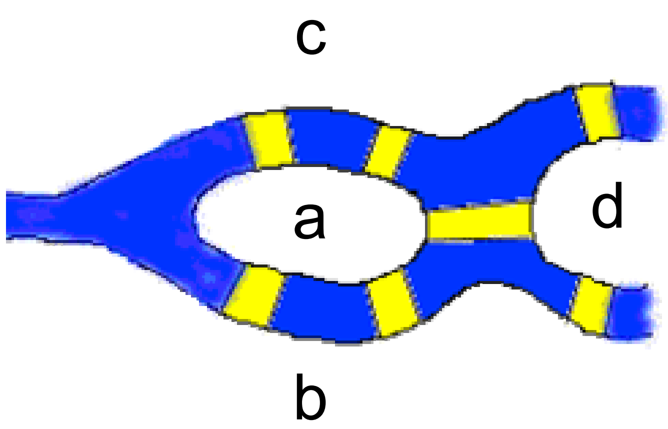
Is there a circuit that contains all edges?

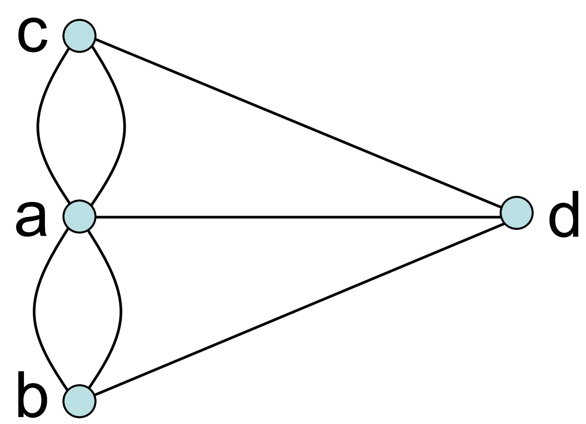
Is there a circuit that contains all vertices?





Euler & Hamilton paths





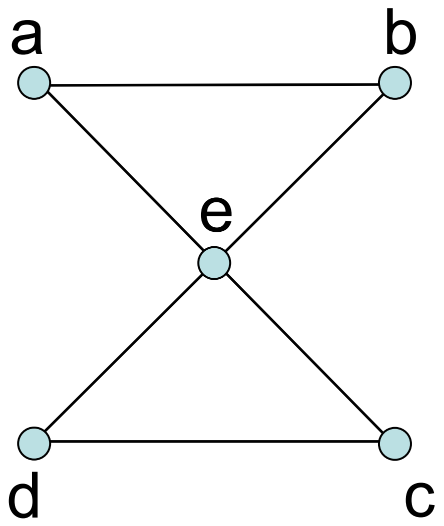
<u>Definition</u>:

An *Euler Path* (מסלול אוילר) in multigraph G is a path that contains every edge of G.

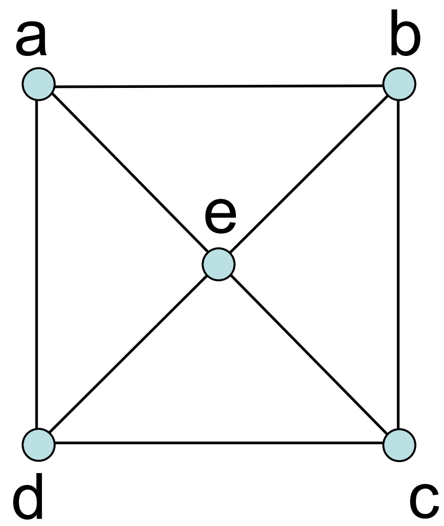
An *Euler Circuit* (מעגל אוילר) in multigraph G is a path that contains every edge of G.

Euler & Hamilton paths

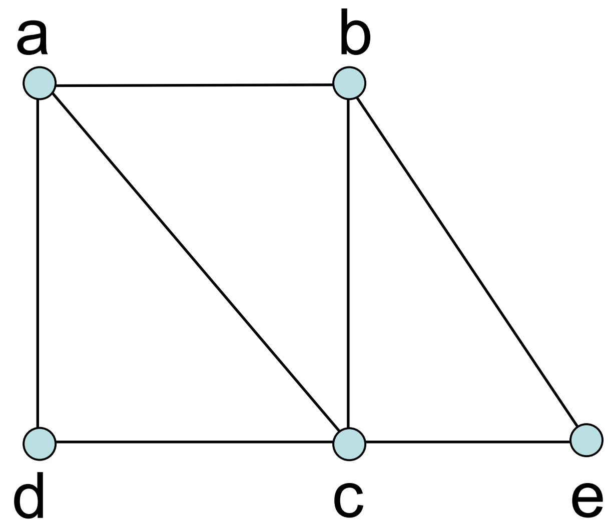
Examples:



Euler Path - no
Euler Circuit - yes
abedcea



Euler Path - no
Euler Circuit - no



Euler Path - yes
Euler Circuit - no
adcabceb

Euler & Hamilton paths

Theorem: A connected multigraph G has an Euler Circuit IFF all vertices have even degrees.

Proof: By Construction.

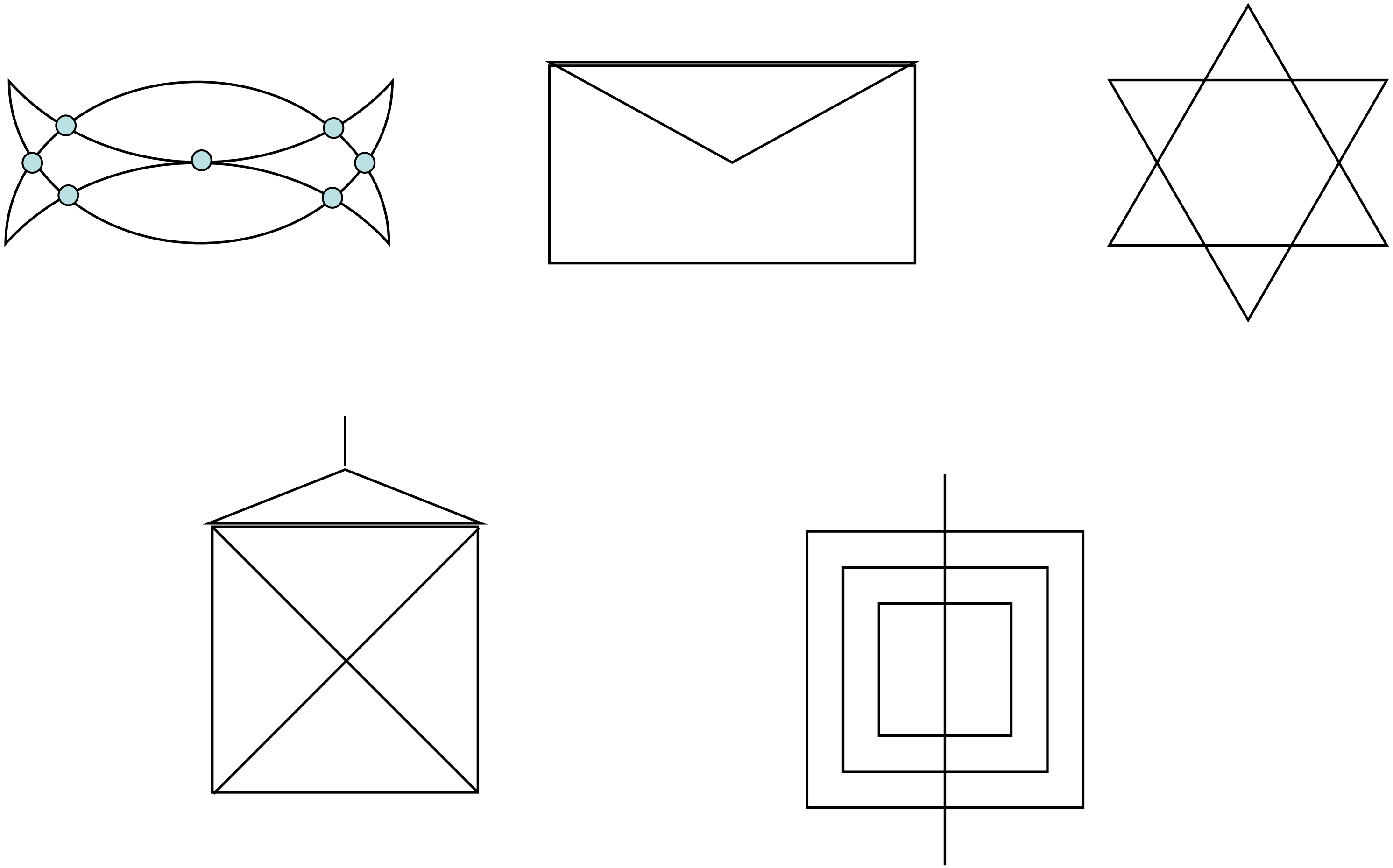
Euler & Hamilton paths

Theorem: A connected multigraph G has an Euler Path IFF exactly 2 vertices have an odd degree.

Proof: By Construction.

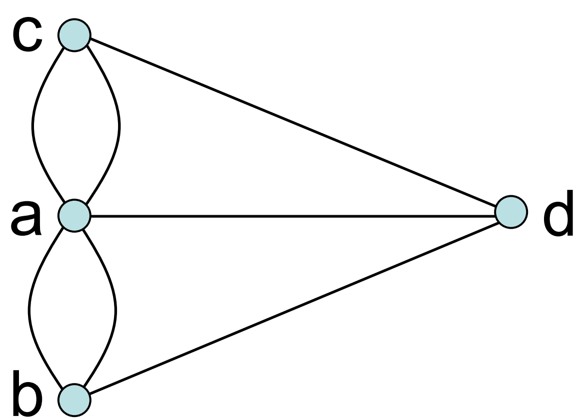
Euler & Hamilton paths

<u>Examples:</u>



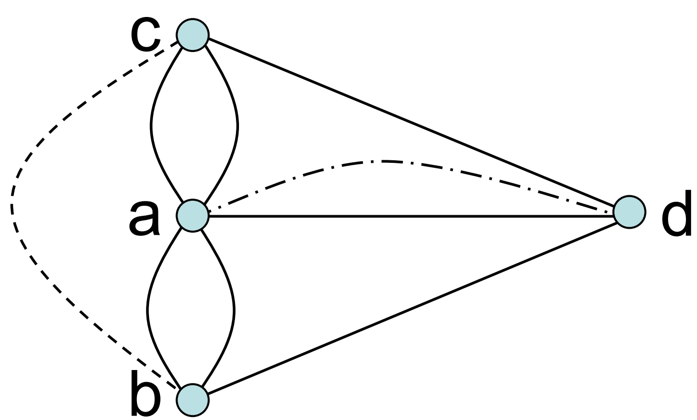
Euler & Hamilton paths

The Seven Bridges of Königsberg ?



no Euler Path

no Euler Circuit



Euler Path

Euler Circuit