MATH-810 Mathematical Methods for Artificial Intelligence

Introduction to Graph Theory

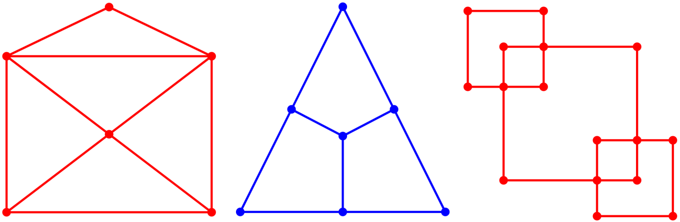
Dr. Yasir Ali

DBS&H, CEME
National University of Sciences and Technology

Islamabad, Pakistan.

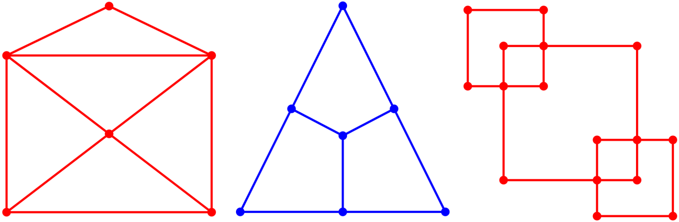
September 22, 2025

It's Puzzle Time!



- Which of these can you draw without lifting your pencil, drawing each line only once?

It's Puzzle Time!



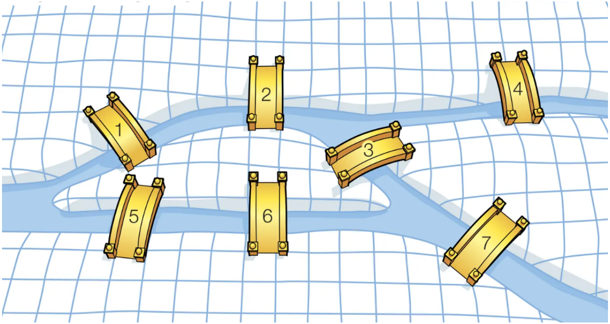
- Which of these can you draw without lifting your pencil, drawing each line only once?

- Can you start and end at the same point?

Konigsberg Bridge Problem

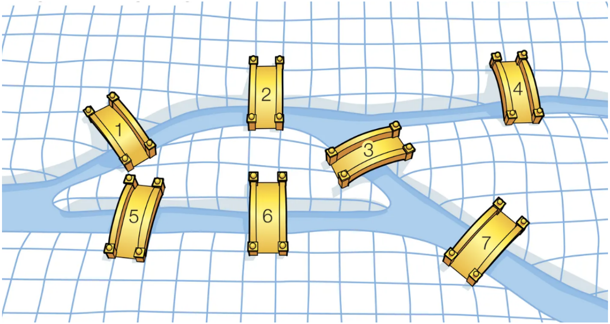
- Four sections of land
- Seven brides connecting four sections of lands

- Four sections of land
- Seven brides connecting four sections of lands



Konigsberg Bridge Problem

- Four sections of land
- Seven brides connecting four sections of lands



Challenge

Is it possible to start a walk at some location, travel across **all the bridges only once**, and **return to the same location**?

A Brief Review

- 1736- Euler solved the Konigsberg bridge problem.

A Brief Review

- 1736- Euler solved the Konigsberg bridge problem.
- 1847- G. R. Kirchhiff (1824-1887) Graph theory in electrical networks.

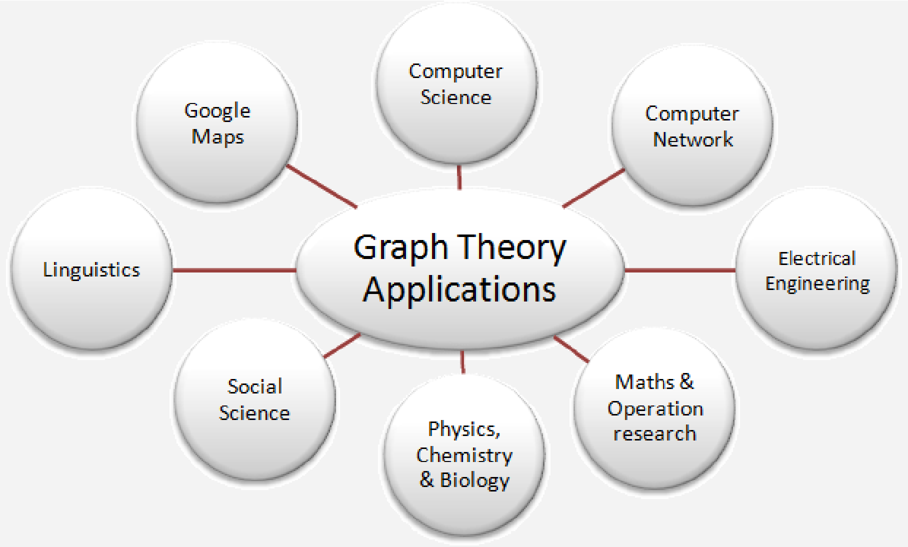
- 1736- Euler solved the Konigsberg bridge problem.
- 1847- G. R. Kirchhiff (1824-1887) Graph theory in electrical networks.
- 1857- A. Cayley (1821-1895) in Chemistry.

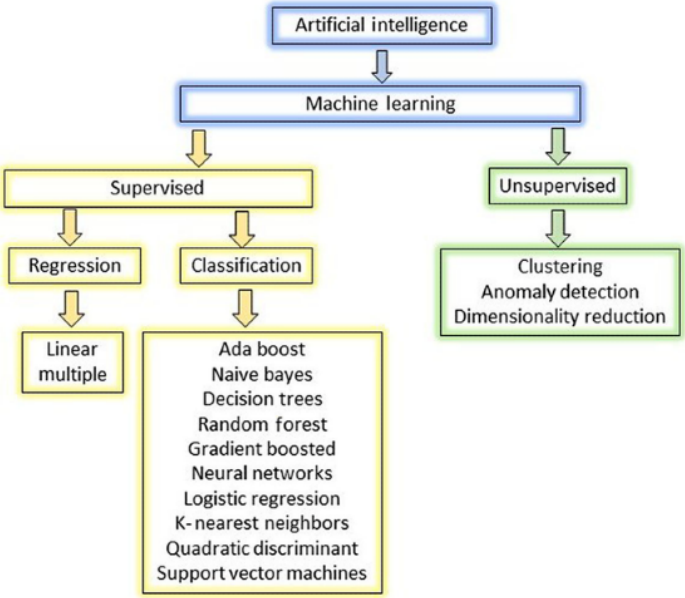
- 1736- Euler solved the Konigsberg bridge problem.
- 1847- G. R. Kirchhiff (1824-1887) Graph theory in electrical networks.
- 1857- A. Cayley (1821-1895) in Chemistry.
- 1879- Four color problem by Cayley.

A Brief Review

- 1736- Euler solved the Konigsberg bridge problem.
- 1847- G. R. Kirchhiff (1824-1887) Graph theory in electrical networks.
- 1857- A. Cayley (1821-1895) in Chemistry.
- 1879- Four color problem by Cayley.
- 1936- First book on the Graph theory, D. Konig

Konig D.,Theorie der endlichen und unendlichen Graphen,Leipzig, 1936; Chelsea, New York, 1950.


Computer Science
Google Maps
Computer Network
Linguistics
Graph Theory Applications
Electrical Engineering
Social Science
Physics, Chemistry & Biology
Maths & Operation research

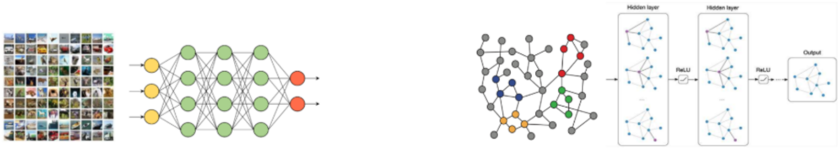


Artificial intelligence
Machine learning
Supervised
Unsupervised
Regression
Classification
Clustering
Anomaly detection
Dimensionality reduction
Linear
multiple
Ada boost
Naive bayes
Decision trees
Random forest
Gradient boosted
Neural networks
Logistic regression
K-nearest neighbors
Quadratic discriminant
Support vector machines

Are you familiar with following terms

- Graph Neural Networks (GNNs)
- Graph Convolutional Network (GCN)
- Graph Attention Network (GAT)
- Graph Recurrent Network (GRN)

J. Zhou et al., "Graph neural networks: A review of methods and applications," AI Open, vol. 1, pp. 57-81, 2020, doi: 10.1016/j.aiopen.2021.01.001.



Conventional AI:
Independent training samples

Graph AI:
Corelated training samples

Figure 1: Comparing Conventional AI (left) with Graph AI (right).

> Graph Machine Learning uses machine learning techniques to process graph data that can be used for predictive, modeling, and analytics tasks.

Graph ML covers many subtopics as well:

- Graph Embedding
- Graph Neural Networks
- Knowledge Graphs
- Influence Maximization
- Disease Outbreak Detection
- Social Network Analysis, etc

With the booming ML, Graph ML has been shown to achieve better performance than the traditional graph mining approach, in many predictive and analytics tasks.

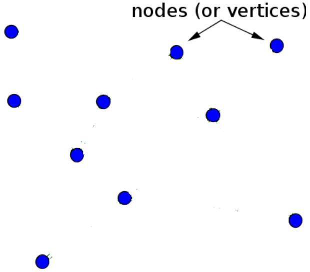
Graphs are a kind of data structure which models a set of objects (nodes) and their relationships (edges)

Mathematically

A graph $G = (V, E)$ consists of two sets V and E,

V a nonempty set of vertices (or nodes)

E a set of edges.



$G = (V, E)$

$v_i \in V$

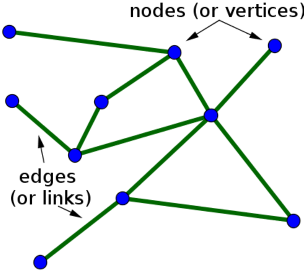
Graphs are a kind of data structure which models a set of objects (nodes) and their relationships (edges)

Mathematically

A graph $G = (V, E)$ consists of two sets V and E,

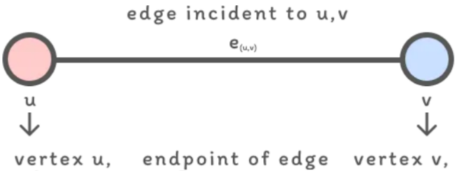
V a nonempty set of vertices (or nodes)

E a set of edges.



$G = (V, E)$

$e_k \in E$



Endpoints

Each edge has two vertices associated with it, called its endpoints.

vertices u and v are endpoints of the edge e

Incident

An edge is said to incident its endpoints

the edge e is incidents on vertices u and v

Neighbors

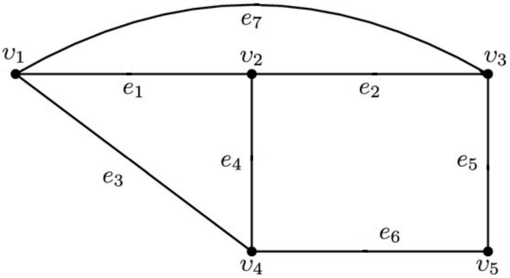
Two vertices are neighbors if there is an edge incident on those vertices.

vertices u and v are neighbors as there is an edge e is incidents on vertices u and v

Order and Size of a Graph

Order of G is the number of vertices of G that is, $|V|$

Size of G is the number of edges in G, that is, $|E|$



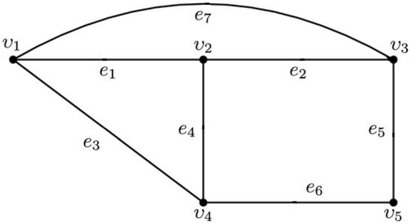
- v_i represents the vertices of a graph, here $i = 1, 2, 3, 4, 5$

- e_k represents the edges of a graph, $k = 1, 2, \cdots, 7$

Let $G = (V, E)$ be a graph and v a vertex of G.

The degree of v is the number of edges that are incident on v

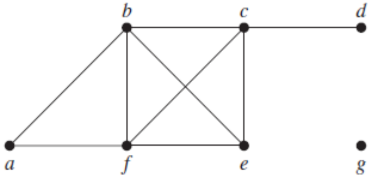
Degree of v is denoted by $deg(v)$



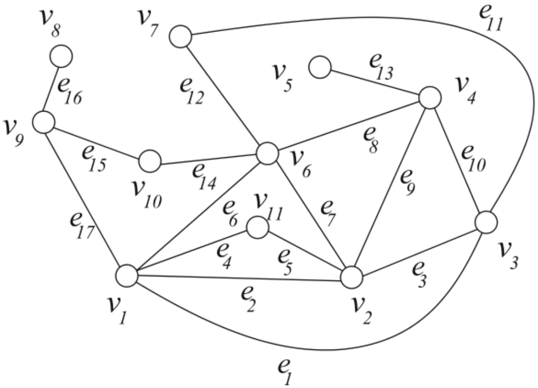
$deg(v_1) = 3 \quad deg(v_2) = 3$
$deg(v_3) = 3 \quad deg(v_4) = 3$
$deg(v_5) = 2$

The total degree of G is the sum of the degrees of all the vertices of G.

Degree



- A vertex is **pendant** if and only if it has degree one.
- An isolated vertex is the one with zero degree.



Simple Graph of order 11 and size 17

Find

- Degree of all vertices
- Neighbors of all vertices

The Handshake Theorem

If G is any graph, then the sum of the degrees of all the vertices of G equals twice the number of edges of G.

For $G(V, E)$ where $|V| = n$ and $|E| = m$ then

$$\sum_{i=1}^{n} deg(v_i) = 2|E|$$

Total Degree of a Graph?

The total degree of a graph is even.

The number of odd-degree vertices in any graph is always even.

Subgraph of a graph G

A graph H is said to be a subgraph of a graph G if, and only if,

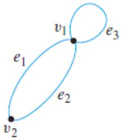
Subgraph of a graph G

A graph H is said to be a subgraph of a graph G if, and only if, every vertex in H is also a vertex in G,

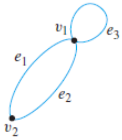
Subgraph of a graph G

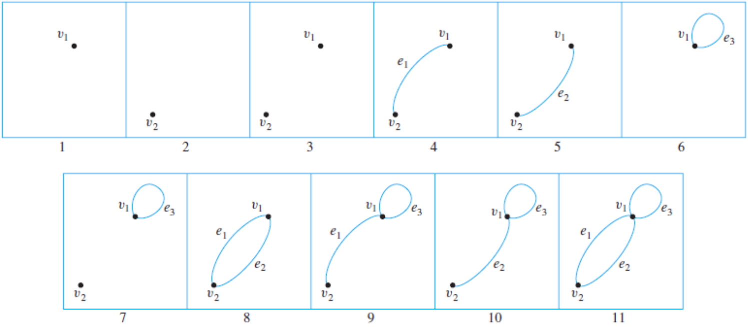
A graph H is said to be a subgraph of a graph G if, and only if,
every vertex in H is also a vertex in G,
every edge in H is also an edge in G,
and every edge in H has the same endpoints as it has in G.

Subgraph



Subgraph





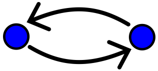
A directed graph (or digraph) (V, E) consists of a nonempty set of vertices V and a set of **directed edges** (or arcs) E.
Each directed edge is associated with an ordered pair of vertices. The directed edge associated with the ordered pair (u, v) is said to *start at u* and *end at v*.

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*.

Difference between $\{u, v\}$ and (u, v)

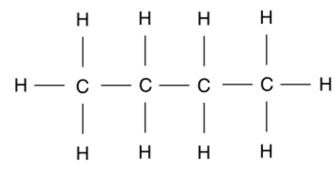
A directed graph (or digraph) (V, E) consists of a nonempty set of vertices V and a set of **directed edges** (or arcs) E.
Each directed edge is associated with an ordered pair of vertices. The directed edge associated with the ordered pair (u, v) is said to *start at u* and *end at v*.

Difference between $\{u, v\}$ and (u, v)



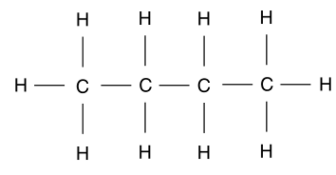
Graph Terminology.			
Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

Draw a graph of C_4H_{10}



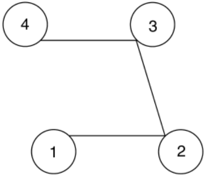
Represent the graph $G = \{\{1,2,3,4\},\{x,4\} : |x-4| \leq 1\}$

Draw a graph of C_4H_{10}

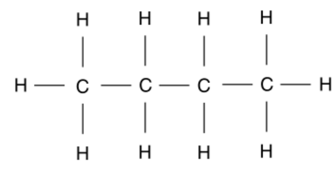


Represent the graph $G = \{\{1,2,3,4\}, \{x,4\} : |x-4| \leq 1\}$

$$V = \{1,2,3,4\}$$
$$E = \{\{1,1\},\{1,2\},\{2,2\},\{2,3\},\{3,3\},\{3,4\}\}$$



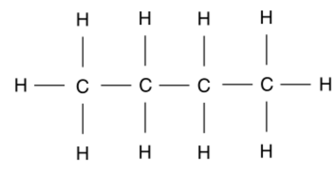
Draw a graph of C_4H_{10}



Represent the graph $G = \{\{1,2,3,4\}, \{x,4\} : |x-4| \leq 1\}$

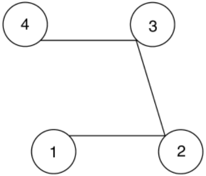
$$V = \{1,2,3,4\}$$
$$E = \{\{1,1\},\{1,2\},\{2,2\},\{2,3\},\{3,3\},\{3,4\}\}$$

Draw a graph of C_4H_{10}



Represent the graph $G = \{\{1,2,3,4\}, \{x,4\} : |x-4| \leq 1\}$

$$V = \{1,2,3,4\}$$
$$E = \{\{1,1\}, \{1,2\}, \{2,2\}, \{2,3\}, \{3,3\}, \{3,4\}\}$$



Minimum degree and maximum degree:

For a graph $G = (V, E)$, we introduce the following symbols:

$\delta(G) =$ Minimum of all the degrees of the vertices of a graph G.

$\Delta(G) =$ Maximum of all the degrees of the vertices of a graph G.

Thus

$$\delta(G) = \min\{\deg(v) : v \in V\}$$

$$\Delta(G) = \max\{\deg(v) : v \in V\}$$

$\Delta(G) \leq n - 1$

Maximum degree of any vertex in a simple graph can not be greater than $n - 1$

Degree Sequence of G

Let G be a graph with $V = \{v_1, v_2, v_3, \ldots, v_n\}$ as the vertex set. Also let $d_i = \deg(v_i)$, then the sequence $(d_1, d_2, \ldots, d_n)$ in any order is called Degree Sequence of G

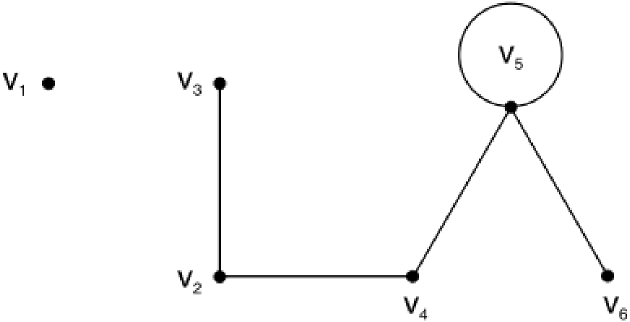
(i) The vertices of a graph G, are ordered so that the degree sequence is monotonically increasing so that

$$\delta(G) = d_1 \leq d_2 \leq d_3 \leq \cdots \leq d_n = \Delta(G)$$

(ii) Two graphs with the same degree sequence are said to be *degree equivalent.*

(iii) It is customary to denote the degree sequence in power notation. If $(2, 2, 2, 3, 3, 4, 5, 5, 6)$ is the degree sequence of a graph G, then it is represented in power notation as $2^3, 3^2, 4^1, 5^2, 6^1$, the degree set being $\{1, 2, 3\}$.

Write degree sequence for the following graph



A sequence $d = (d_1, d_2, \cdots, d_n)$ is graphic, if there is a simple non-directed graph with the degree sequence d.

Show that the following sequences are graphic or not:

$$(i) \quad (2, 3, 4, 5, 6, 7), \qquad (ii) \quad (2, 2, 4)$$

k-regular Graph

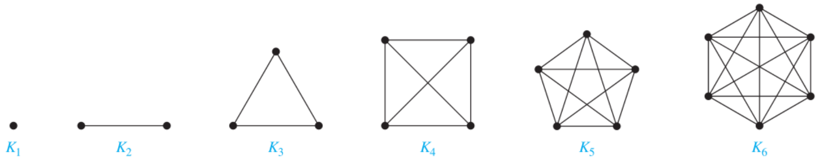
A graph G is said to be k-regular, if every vertex of G has degree k

$$\delta(G) = \Delta(G) = k$$

Complete Graph K_n

A complete graph on n vertices is a simple graph that contains exactly one edge between each pair of distinct vertices.

A $n-1$-regular graph on n vertices



Complete Bipartite Graph

Let m and n be positive integers. A complete bipartite graph on (m, n) vertices, denoted $K_{m,n}$,

Complete Bipartite Graph

Let m and n be positive integers. A complete bipartite graph on (m, n) vertices, denoted $K_{m,n}$, is a simple graph with distinct vertices $v_1, v_2, ..., v_m$ and $w_1, w_2, ..., w_n$ that satisfies the following properties: For all $i, k = 1, 2, ..., m$ and for all $j, l = 1, 2, ..., n$

Complete Bipartite Graph

Let m and n be positive integers. A complete bipartite graph on (m, n) vertices, denoted $K_{m,n}$, is a simple graph with distinct vertices $v_1, v_2, ..., v_m$ and $w_1, w_2, ..., w_n$ that satisfies the following properties: For all $i, k = 1, 2, ..., m$ and for all $j, l = 1, 2, ..., n$

1. There is an edge from each vertex v_i to each vertex w_j.

Complete Bipartite Graph

Let m and n be positive integers. A complete bipartite graph on (m, n) vertices, denoted $K_{m,n}$, is a simple graph with distinct vertices $v_1, v_2, ..., v_m$ and $w_1, w_2, ..., w_n$ that satisfies the following properties: For all $i, k = 1, 2, ..., m$ and for all $j, l = 1, 2, ..., n$

1. There is an edge from each vertex v_i to each vertex w_j.
2. There is no edge from any vertex v_i to any other vertex v_k.

Complete Bipartite Graph

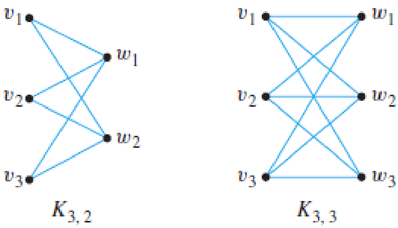
Let m and n be positive integers. A complete bipartite graph on (m, n) vertices, denoted $K_{m,n}$, is a simple graph with distinct vertices $v_1, v_2, ..., v_m$ and $w_1, w_2, ..., w_n$ that satisfies the following properties: For all $i, k = 1, 2, ..., m$ and for all $j, l = 1, 2, ..., n$

1. There is an edge from each vertex v_i to each vertex w_j.
2. There is no edge from any vertex v_i to any other vertex v_k.
3. There is no edge from any vertex w_j to any other vertex w_l.

Let m and n be positive integers. A complete bipartite graph on (m, n) vertices, denoted $K_{m,n}$, is a simple graph with distinct vertices $v_1, v_2, \ldots, v_m$ and $w_1, w_2, \ldots, w_n$ that satisfies the following properties: For all $i, k = 1, 2, \ldots, m$ and for all $j, l = 1, 2, \ldots, n$

1. There is an edge from each vertex v_i to each vertex w_j.
2. There is no edge from any vertex v_i to any other vertex v_k.
3. There is no edge from any vertex w_j to any other vertex w_l.

The complete bipartite graphs $K_{3,2}$ and $K_{3,3}$ are



Adjacency Lists

This list specify all the vertices that are ***adjacent*** to each vertex of the graph.

This list specify all the vertices that are ***adjacent*** to each vertex of the graph.

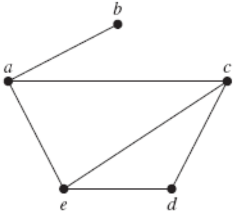


TABLE An Adjacency List for a Simple Graph.

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

Matrix Representation of Graphs

1. Adjacency Matrix
2. Incidence Matrix

Matrix Representation of a Graph

Suppose that $G = (V, E)$ is a simple graph with order n and size m.

Adjacency Matrix

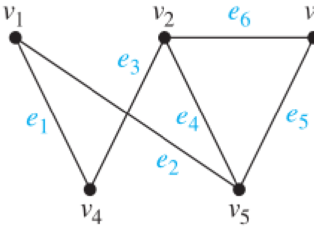
The adjacency matrix A (or A_G) is an $n \times n$ matrix with $A = [a_{ij}]$

$$a_{ij} = \begin{cases} 1 & \text{if } v_i v_j \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Incidence Matrix

The incidence matrix is an $n \times m$ matrix $M = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 & \text{when } v_i \in e_j \\ 0 & \text{otherwise.} \end{cases}$$

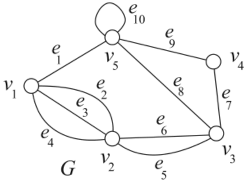


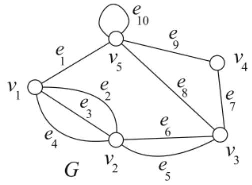
Adjacency Matrix A:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}_{n \times n}$$

Incidence Matrix M:

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}_{n \times m}$$


e_{10}
e_9
e_1
v_5
v_4
v_1
e_2
e_8
e_7
e_3
e_6
e_4
v_2
v_3
G
e_5



$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 3 & 0 & 0 & 1 \\ 3 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

$A(G)$

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

$M(G)$

Walk, Trail, Path and Cycle

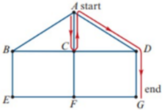
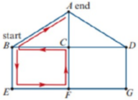
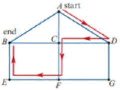
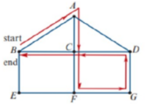
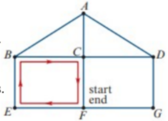
- In a graph $G = (V, E)$, a ***walk*** is a sequence $v_1, v_2, \ldots, v_k$ of k vertices such that each (v_j, v_{j+1}) for $1 \leq j \leq k-1$ is an edge in E.

- In a graph $G = (V, E)$, a ***walk*** is a sequence $v_1, v_2, \dots, v_k$ of k vertices such that each (v_j, v_{j+1}) for $1 \leq j \leq k-1$ is an edge in E.
- A walk with no repetition of edges is called a ***trail***.

- In a graph $G = (V, E)$, a ***walk*** is a sequence $v_1, v_2, \ldots, v_k$ of k vertices such that each (v_j, v_{j+1}) for $1 \leq j \leq k-1$ is an edge in E.
- A walk with no repetition of edges is called a ***trail***.
- A ***path*** is trail with all distinct vertices.
  A path
  $$S = v_1, v_2, \ldots, v_k$$
  is called a path from v_1 to v_k or a v_1-v_k path. Here v_1 and v_k are called ***initial*** and ***terminal*** vertices of the path S, respectively.

Walk, Trail, Path and Cycle

- In a graph $G = (V, E)$, a ***walk*** is a sequence $v_1, v_2, \ldots, v_k$ of k vertices such that each (v_j, v_{j+1}) for $1 \leq j \leq k-1$ is an edge in E.
- A walk with no repetition of edges is called a ***trail***.
- A ***path*** is trail with all distinct vertices.
  A path
  $$S = v_1, v_2, \ldots, v_k$$
  is called a path from v_1 to v_k or a v_1-v_k path. Here v_1 and v_k are called ***initial*** and ***terminal*** vertices of the path S, respectively.
- A ***cycle or simple circuit*** is a trail of the form of $v_1, v_2, \ldots, v_k, v_1$ with $k \geq 3$ such that $v_i \neq v_j$ for $1 \leq i < j \leq k$.

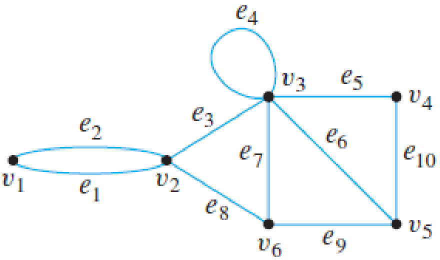
WALK	A walk is a sequence of edges, linking successive vertices in a graph. A walk starts at one vertex and follows any route to finish at another vertex. A–C–A–D–G.	
TRAIL	A trail is a walk with no repeated edges. B–E–F–C–B–A. (note that vertex B is repeated)	
PATH	A path is a walk with no repeated edges and no repeated vertices. A–D–C–F–E–B.	
CIRCUIT	A circuit is a trail (no repeated edges) that starts and ends at the same vertex. Circuits are also called closed trails. A–C–F–G–D–C–B–A. Note: There are no repeated edges in this circuit, but one vertex, C, is repeated. The start and end vertices are also repeated because of the definition of a circuit.	
CYCLE	A cycle is a path (no repeated edges, no repeated vertices) that starts and ends at the same vertex. The start and end vertex is an exception to repeated vertices. Cycles are also called closed paths. F–E–B–C–F. Note: There are no repeated edges and no repeated vertices in this cycle, except for the start and end vertices.	

Walk, Trail, Path and Cycle

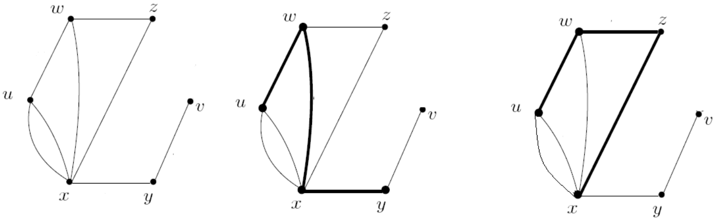
	Repeated Edge?	Repeated Vertex?	Starts and Ends at Same Point?	Must Contain at Least One Edge?
Walk	allowed	allowed	allowed	no
Trail	no	allowed	allowed	no
Path	no	no	no	no
Closed walk	allowed	allowed	yes	no
Circuit	no	allowed	yes	yes
Simple circuit	no	first and last only	yes	yes

In the graph below, determine which of the following walks are trails, paths, circuits, or simple circuits.

a. $v_1 e_1 v_2 e_3 v_3 e_4 v_3 e_5 v_4$ b. $e_1 e_3 e_5 e_5 e_6$ c. $v_2 v_3 v_4 v_5 v_3 v_6 v_2$
d. $v_2 v_3 v_4 v_5 v_6 v_2$ e. $v_1 e_1 v_2 e_1 v_1$ f. v_1



Path: A path from vertex v_0 to vertex v_k is sequence of vertices $< v_0, v_1, v_2, \cdots, v_k >$ such that $(v_{i-1}, v_i) \in E$ for $i = 1, 2, \cdots, k$
$V = \{u, v, w, x, y, z\}$



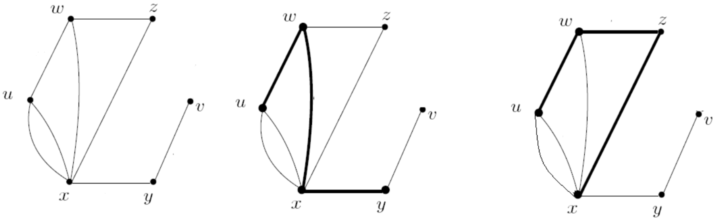
path

$P_1 :< u, w, x, y >$, $\{u, w\}, \{w, x\}, \{x, y\} \in E$

Path and Cycle

Path: A path from vertex v_0 to vertex v_k is sequence of vertices $< v_0, v_1, v_2, \cdots, v_k >$ such that $(v_{i-1}, v_i) \in E$ for $i = 1, 2, \cdots, k$

$V = \{u, v, w, x, y, z\}$

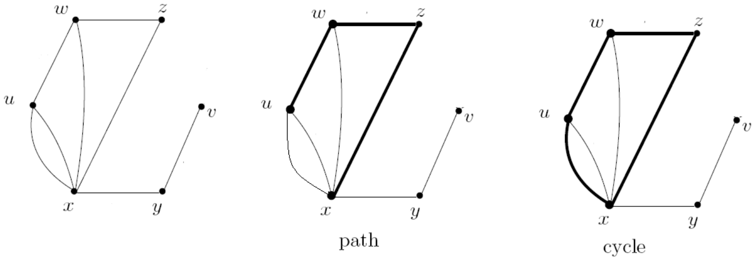


path

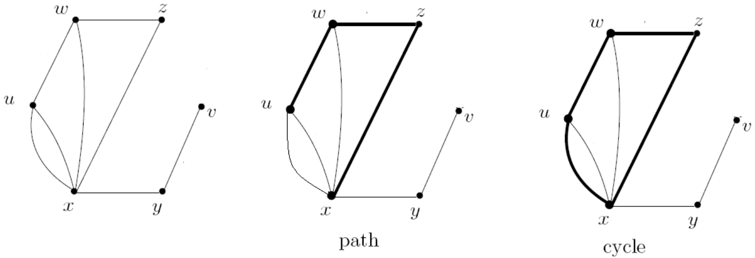
$P_1 :< u, w, x, y >$, $\{u, w\}, \{w, x\}, \{x, y\} \in E$

$P_2 :< u, w, z, x >$, $\{u, w\}, \{w, z\}, \{z, x\} \in E$

Cycle: If starting and end vertices of a path are same.



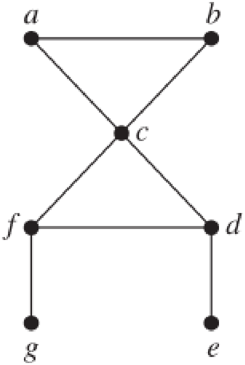
Cycle: If starting and end vertices of a path are same.



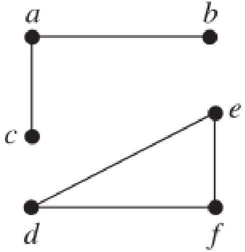
$P_2' :< u, w, z, x, u >, \{u, w\}, \{w, z\}, \{z, x\}, \{x, u\} \in E$

Connected Graph

A graph G is connected if there is a path between each pair of vertices in G. Otherwise, G is called a disconnected graph



G_1

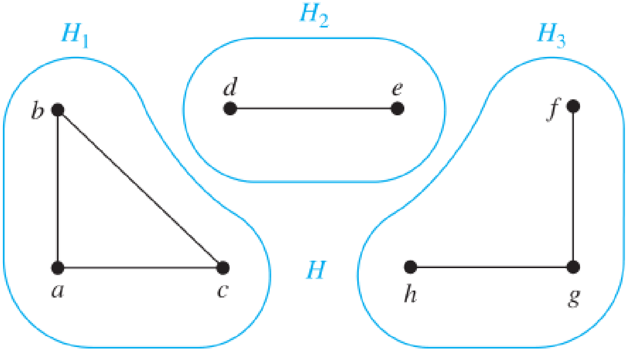


G_2

Connected Component of a Graph G

A connected component of 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.



Connected Components H_1, H_2, and H_3.

Eulerian and Hamiltonian Circuit

Eulerian Path:

- This path visits every **edge** of the graph exactly once.
- It does not necessarily start and end at the same vertex.

Hamiltonian Path:

- A path that visits each **vertex** exactly once.
- It does not necessarily start and end at the same vertex.

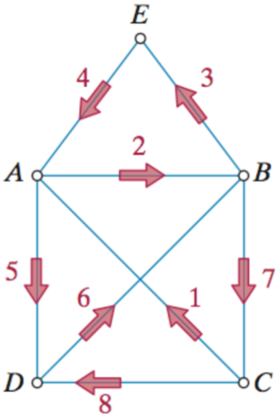
Properties of Eulerian Graph

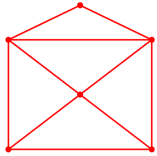
The following statements are equivalent for a connected a graph G.

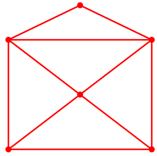
1. G is Eulerian.
2. Every point of G has even degree.
3. The set of edges of G can be partitioned into cycles.

(i) If a graph G has more thanwo vertices of odd degree, then there can be no Euler path in G.

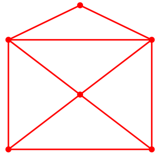
(ii) If G is connected graph and has exactly two vertices of odd degree, there is an Euler path in G. Any Euler path in graph G must begin at vertex of odd degree and end at the other.



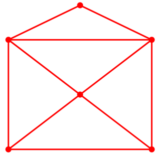




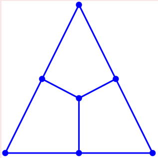
- Two odd degree vertices

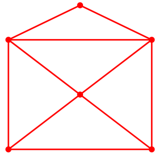


- Two odd degree vertices
- One can visit every edge exactly once.
- Starting and ending vertices will not be the same.

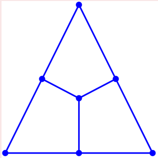


- Two odd degree vertices
- One can visit every edge exactly once.
- Starting and ending vertices will not be the same.

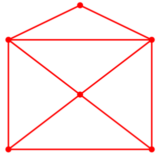




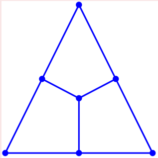
- Two odd degree vertices
- One can visit every edge exactly once.
- Starting and ending vertices will not be the same.



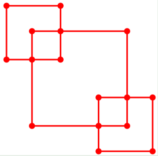
- More than two odd degree vertices
- One can not visit every edge exactly once.



- Two odd degree vertices
- One can visit every edge exactly once.
- Starting and ending vertices will not be the same.



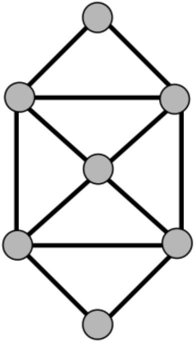
- More than two odd degree vertices
- One can not visit every edge exactly once.

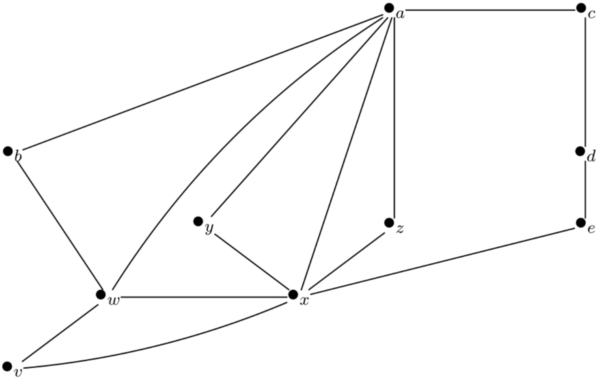


- No odd degree vertices
- One can visit every edge exactly once.
- Starting and ending vertices will be the same.

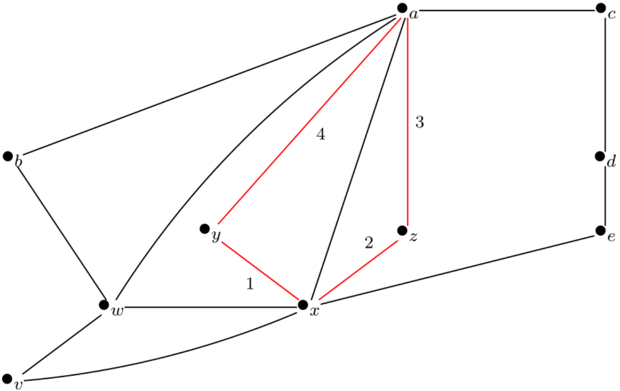
Finding Euler Circuits: DFS and then Splice

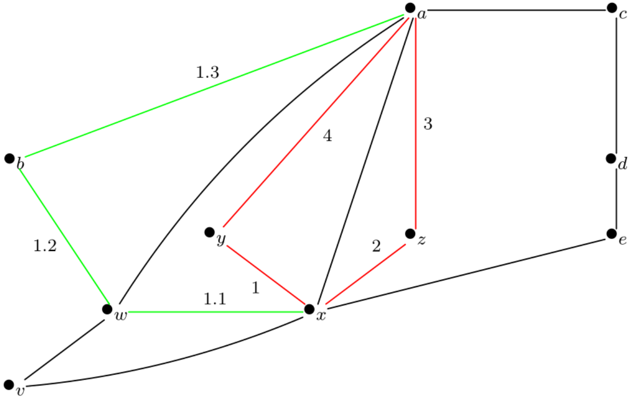
- Given a graph $G = (V,E)$, find an Euler circuit in G
  - Can check if one exists in $O(|V|)$ time (check degrees)
- Basic Euler Circuit Algorithm:
  1. Do a depth-first search (DFS) from a vertex until you are back at this vertex
  2. Pick a vertex on this path with an unused edge and repeat 1.
  3. Splice all these paths into an Euler circuit
- Running time = $O(|V| + |E|)$




a
c
b
d
y
z
e
w
x
v

First just find a cycle, starting at vertex y



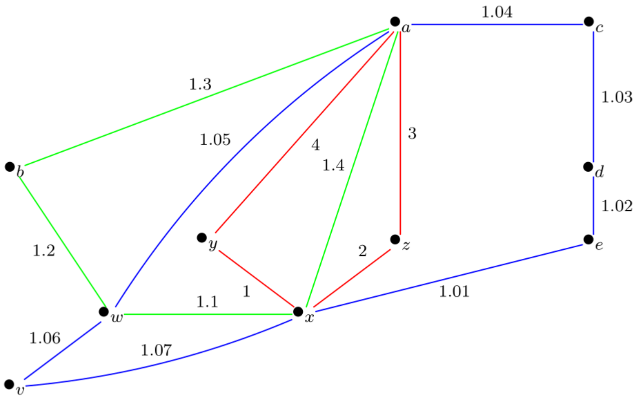


Our circuit is now described by the list of edges we have labeled

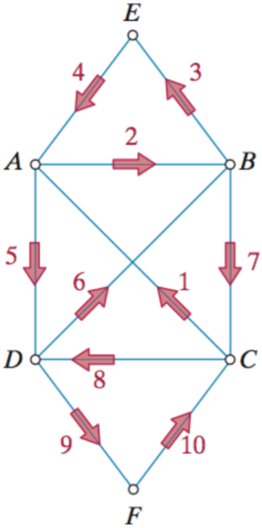
$$(1, 1.1, 1.2, 1.3, 1.4, 2, 3, 4)$$

or by the list of vertices:

$$(y, x, w, b, a, x, z, a, y).$$



$$(y, x, e, d, c, a, w, v, x, w, b, a, x, z, a, y)$$


E
4
3
2
A
B
5
7
6
1
D
C
8
9
10
F