GRAPHS

Consider a situation that

Person A wants to visit Dhaka Universit(y)
(located at
NSU

So, he/she decides to create a map that would include routes.

Nodes ≡ Vertices

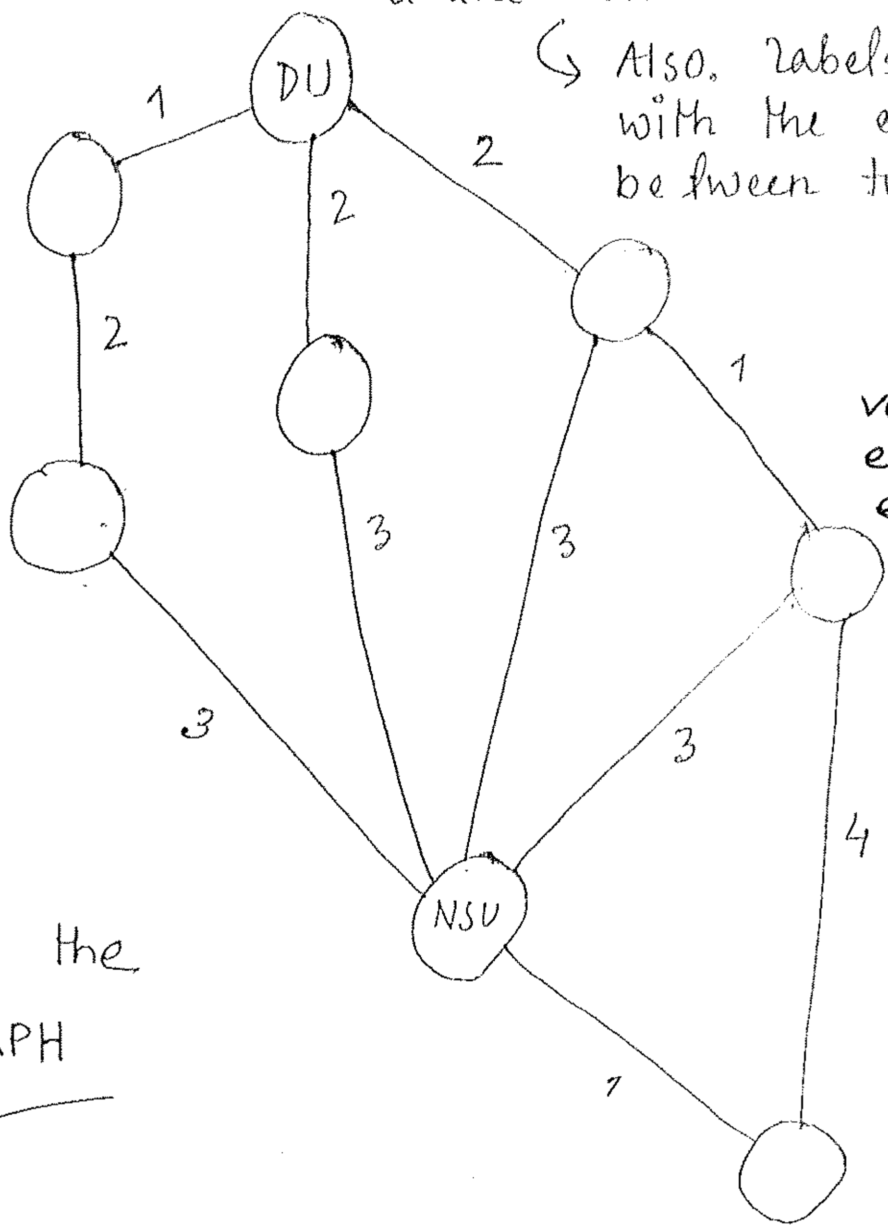
Lines ≡ edges

Considers different stoppage places as nodes

Measures the distance between different nodes.

Finally, connect nodes by drawing a line between the two nodes.

Also, labels each line with the exact distance between two nodes



* Here, numeric values next to each ~~weights~~ edge make the graph a weighted graph.

This is the GRAPH

4.1 Definition

* By definition, a graph $G = (\overset{\text{vertices}}{V}, \overset{\text{Edges}}{E})$ consists of a nonempty set of vertices (or, nodes) and a set of Edges (E).

* Each edge has either $\underbrace{\text{one or two vertices}}_{\text{endpoints}}$, a ociated with it.

* Each edge connects its endpoints

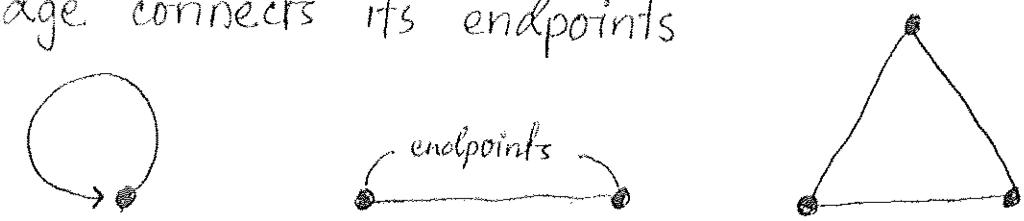


Fig. example

Example :

Suppose, a network is made up of data centers and communication link between all the computers. Now, if the location of the data centers is represented as points, the network can be represented as a Graph

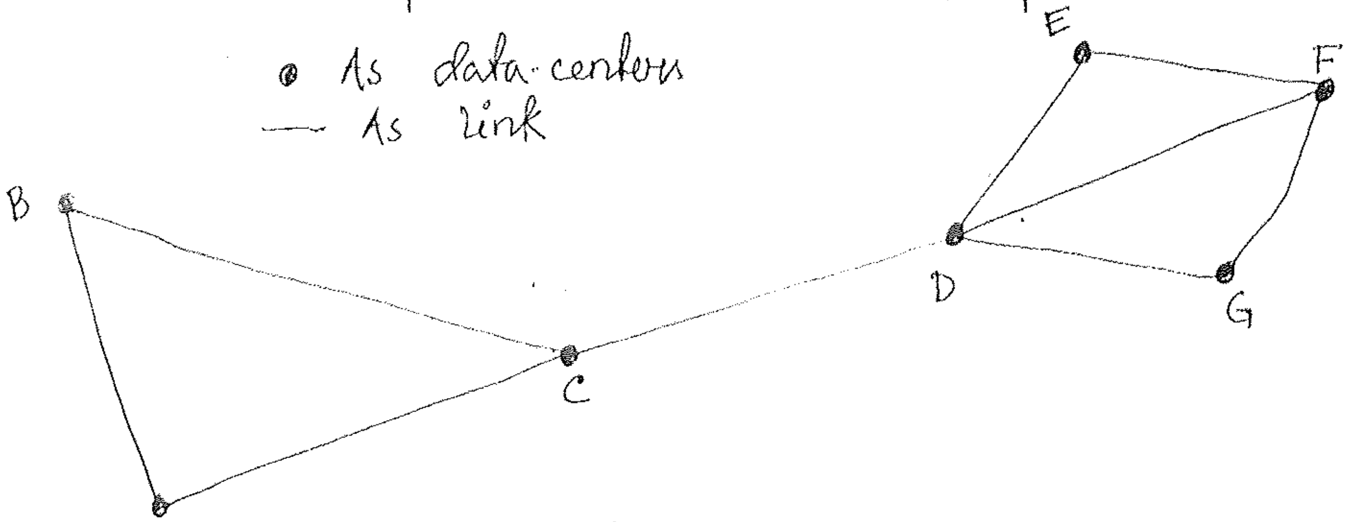


Fig. Network as a graph

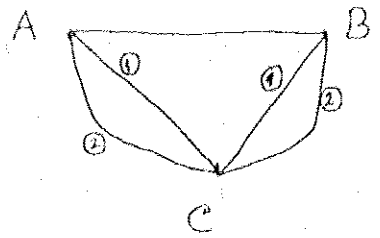
Finite Vs. Infinite Graph

A graph with finite number of vertices is known as Finite graph.

A graph with infinite number of vertices is infinite graph.

Multigraphs

In multigraphs, multiple edges may connect same two vertices.



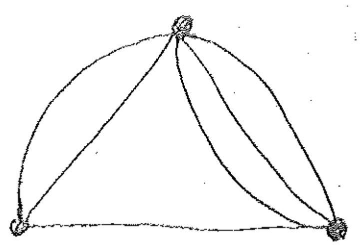
Here, multiple edges connect A and B
B and C

When m different edges connect vertices A and B, then we can say that

$\{A, B\}$ is an edge of multiplicity m

In above graph multiplicity m is two for both $\{A, C\}$ and $\{B, C\}$

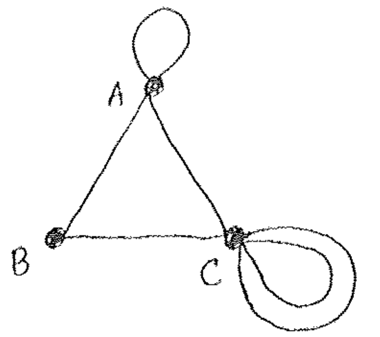
Precisely, multiple edges between nodes are permitted or required in multigraphs.



Loops

It is the edge of a graph which joins a vertex to itself.

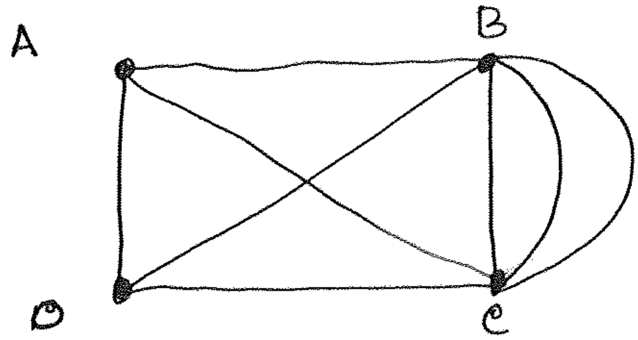
It is also known as self-loop



Edge multiplicity :

This is often measured for a multigraph. Edge multiplicity is calculated for end vertex of multigraph. It is also used to measure the graph multiplicity.

Definition : Edge multiplicity of a given vertex in a multigraph would be the number of edges that share the end vertex.



So, edge multiplicity of node/vertex

A : 3

B : 3

C : 5

D : 5

Maximum edge multiplicity in such graph is known as graph multiplicity.

Simple Graph

In a simple graph, each edge of the graph two different vertices, and no two edges connect the same pair of vertices.

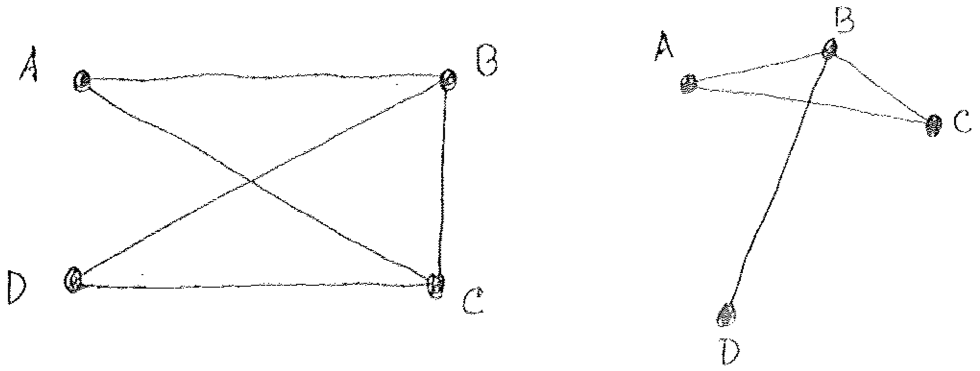
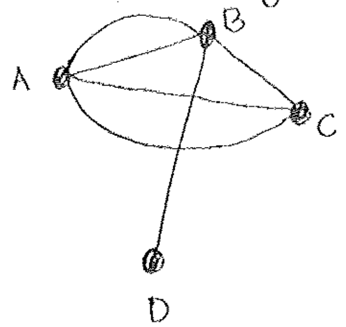
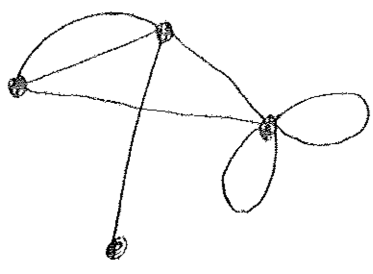


Fig. Simple Graph

Example of non-simple graph



✓ This graph has multiple edges.

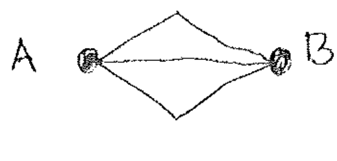


✓ This graph has multiple edges and loops.

Loops Edge

Multiple edges are two or more edges connecting the same two vertices.

Number of edges is denoted as degree "d"



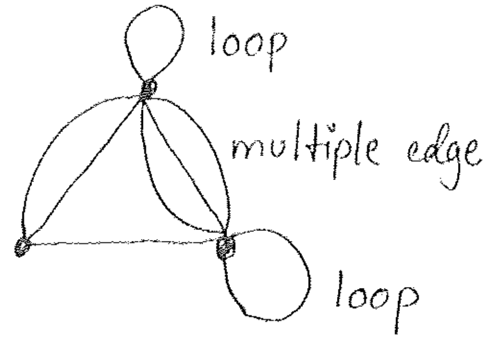
Here, $d = 3$

Pseudograph

In a pseudograph,

- Graph loops &
- Multiple Edges

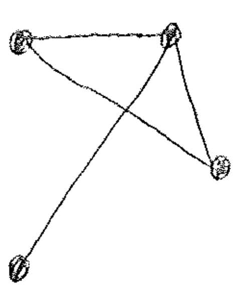
are permitted.



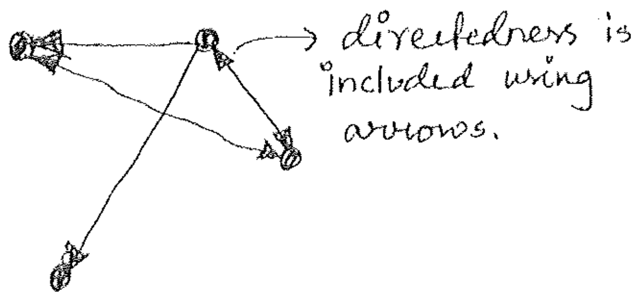
Directedness of a graph :

Based on the direction of an edge between two nodes, a graph can be subdivided into two main subclasses;

a. Undirected graph
b. Directed graph



Undirected graph



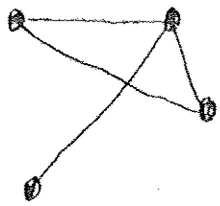
Directed graph

Labeled and Unlabeled graph

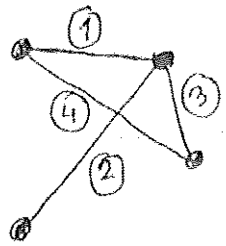
In any graph, the edges and vertices, both, may be assigned specific values, labels or colors.

↓ Generates

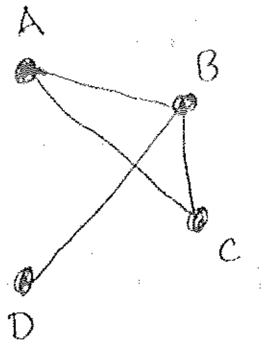
Labeled Graph



Unlabeled graph



Edge-labeled graph

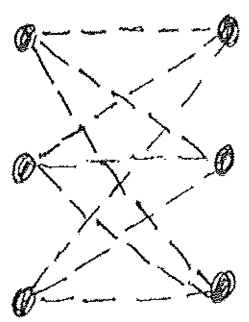


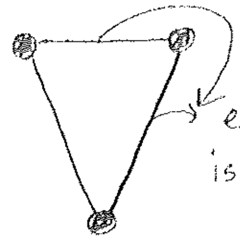
Vertex-Labeled graph

Colors can be assigned to vertices or edges — this is known as edge coloring and graph coloring

Vertex coloring :

It is an assignment of colors or labels to each vertex such that no edge connects two nodes of similar color





edge color is different

Edge coloring:

Adjacent edges ~~to~~ receive differen colors.

Application of Graphs & Graph Theory

✓ Computer networks

Here, nodes may be computers, or may be data centers

✓ Influence Graph: Suppose we want to study group behavior of a group of people. It is common that certain people can infuence the thinking of other people. In such care we can use directed graph.

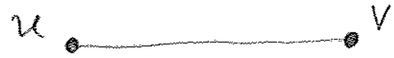
Also, we can apply graph concept in analyzing the interactions of people in social network

GRAPH TERMINOLOGY

Definition : Two vertices u and v are undirected graph G are called adjacent in G if u and v are end points of an edge of G.

If 'e' is associated with $\{u, v\}$, the edge e is called incident with the vertices u and v

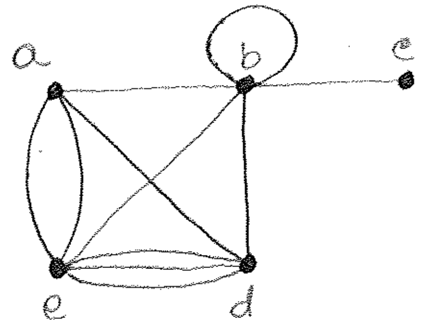
u, v are endpoints of an edge associated with $\{u, v\}$.



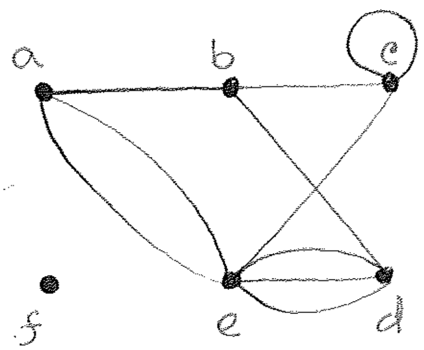
Definition : Degree of a vertex

Degree of a vertex in an undirected graph is the number of edges incident with it

A loop at a vertex contributes twice to the degree of that vertex.



deg (a) = 4
deg (b) = 6
deg (c) = 1
deg (d) = 5
deg (e) = 6



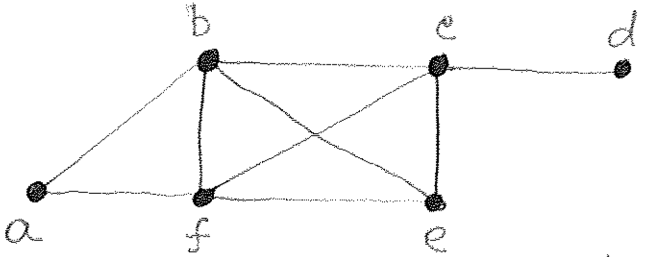
deg (a) = 3
deg (b) = 3
deg (c) = 4
deg (e) = 6
deg (d) = 4
deg (f) = 0

HANDSHAKING THEOREM

Given $G = (V, E)$ is an undirected graph, then

$$2e = \sum_{u \in V} \deg(u),$$ where "e" is the number of edges.

Consider the below graph —



Total edges

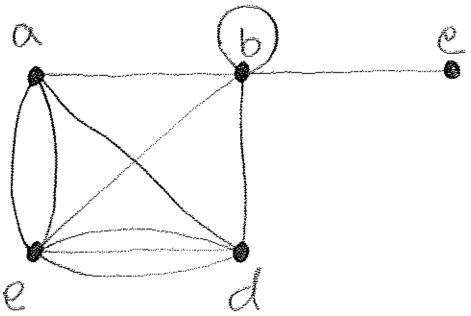
$e = 9$

So, $2e = 9 \times 2 = 18$

$\deg(a) = 2$	$\deg(d) = 1$
$\deg(b) = 4$	$\deg(e) = 3$
$\deg(c) = 4$	$\deg(f) = 4$
10	8

So, $2e = 18$, $\sum_{u \in V} \deg(u) = 10 + 8 = 18$

Another example —



$\deg(a) = 4$
$\deg(b) = 6$
$\deg(c) = 1$
$\deg(d) = 5$
$\deg(e) = 6$

22

edge number

$e = 11$

So, $2e = 22$

So,

$$\sum_{u \in V} \deg(u) = 2e$$

Question :

How many edges are there in a graph with 10 vertices each of degree 6?

Answer :

From the Handshaking Theorem, we know

$$2e = \sum_{u \in V} \deg(u)$$

Here, total degree is $6 \times 10 = 60$

$$\sum_{u \in V} \deg(u)$$

So, $2e = 60$

$$\Rightarrow e = 60/2 = 30$$

number of edges

Observation

$2e \equiv$ Even number by definition

implies that

Sum of the degrees of vertices of an undirected graph is even

Theorem :

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

田 THEOREM

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

Proof :
Let V_1 and V_2 be the set of vertices of even degree and set of vertices of odd degree respectively.

So, $$2e = \sum_{u \in V} \deg(u) = \sum_{u \in V_1} \deg(u) + \sum_{u \in V_2} \deg(u)$$

We know

even + even = even ... ... (1)

odd + odd = even ... ... (2)

even + odd = odd ... ... (3)

Now, $2e$ is even. so, $\sum_{u \in V} \deg(u)$ is even.

As. V_1 is the set of vertices with even degree, summing up all the degrees generate another even number. Because,

even + even = even

so, $\sum_{u \in V_1} \deg(u) \equiv$ even

$\downarrow$ implies that

$\sum_{u \in V_2} \deg(u)$ must be even

As known, V_2 is the set of nodes with odd degrees

From eqⁿ (3), it is evident that

odd + odd = even

$\downarrow$ implies that

For $\sum_{u \in V_2} deg(u)$ to be even,

we must have even number of vertices with odd degree: That is

$$\sum_{u \in V_2} deg(u) = even = odd + odd + odd + odd + odd + odd + \vdots$$

So, An undirected graph has an even number of vertices of odd degree

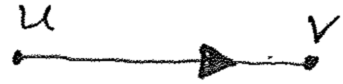
Proved

Directed Graph

A directed graph (V, E), also known as digraph, consists of a nonempty set of vertices V and a set of directed edges E.

Each directed edge is associated with an ordered pair of vertices.

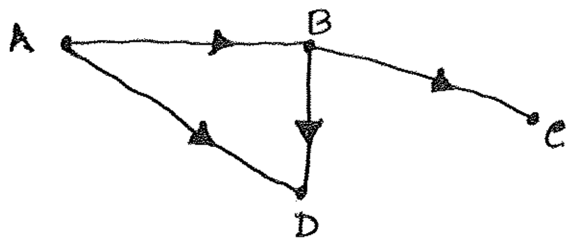
The directed edge associated with the ordered pair (u, v) starts at u and ends at v.



Directed edge

Simple directed graph

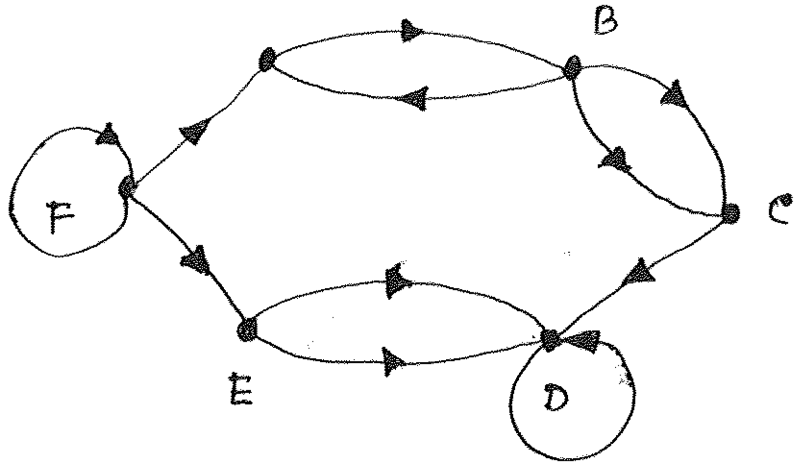
It doesn't have any loop. Also, no multiple directed edges exist in this graph.



Directed multigraphs

- have multiple directed edges from a vertex to a second vertex.
- If there are m directed edges from u to v, we can say that (u, v) is an edge of multiplicity m.

⊞ Example : Directed Multigraph



Mixed Graph :

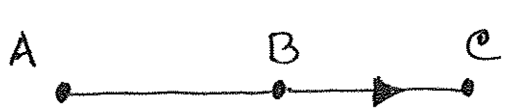
A graph that has both directed and undirected edges is called a mixed graph.

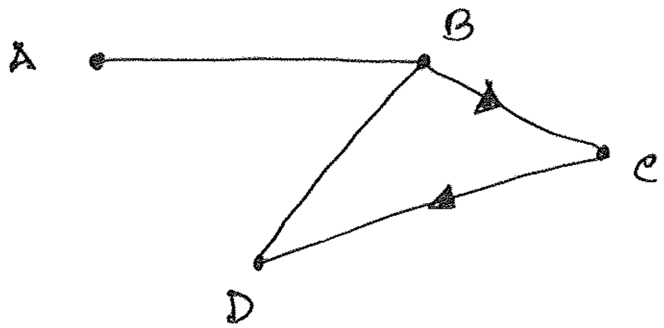
That is —

Mixed graph has
- a set of undirected edges
- a set of directed edges
- a set of vertices

$G = (V, E, A)$

Example :





Graph Types : Table

Type	Edges	Multiple edges Allowed ?	Loops
Simple graph	Undirected	NO	NO
Multigraph	"	YES	NO
Pseudograph	"	YES	YES
Simple directed graph	Directed	NO	NO
Directed multigraph	Directed	YES	YES
Mixed graph	Directed and Undirected	YES	YES

田 Initial vs. Terminal vertex

Given (u, v) is an edge

$\downarrow$ implies that

u is initial vertex
v is terminal or end vertex

田 In-degree vs. Out-degree

In directed graphs

In-degree of a vertex v, is denoted as $deg^-(v)$, means the number of edges with v as the terminal vertex.

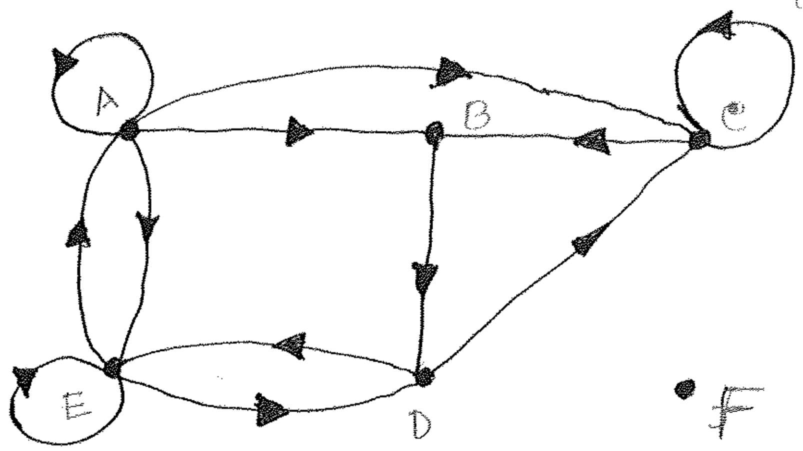
Out-degree of a vertex v, denoted as $deg^+(v)$, means the number of edges with v as the initial vertex.

田 Loop in a directed graph contributes 1 to both the in-degree and out-degree of the vertex

Example : In-degree / Out-degree

In-degree → number of edges coming towards a node

Out-degree → number of edges going out of a node



So,

$deg^-(A) = 2$	$deg^-(B) = 2$	$deg^-(C) = 3$
$deg^+(A) = 4$	$deg^+(B) = 1$	$deg^+(C) = 2$

$deg^-(D) = 2$	$deg^-(E) = 3$	$deg^-(F) = 0$
$deg^+(D) = 2$	$deg^+(E) = 3$	$deg^+(F) = 0$

Given a graph $G = (V, E)$, then

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

For instance, in the above graph

$$\sum_{u \in V} deg^-(u) = 12$$

$$\sum_{u \in V} deg^+(u) = 12$$

Total edges = 12

Special simple graphs

Below are a few of the graphs that often arise in many applications:

Complete Graph:

- exactly one edge between a pair of distinct vertices
- denoted as K_n — complete graph on n vertices.

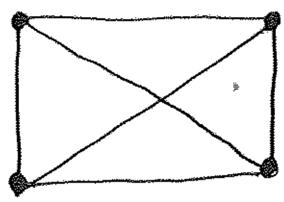
$n=1$, K_1

K_2

K_3

$$^3C_2 = \frac{3!}{2!} = 3$$

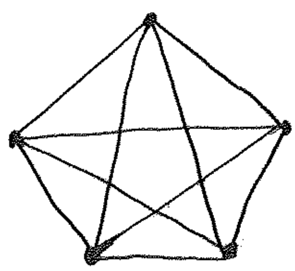
K_4



Number of distinct pair

$$^4C_2 = \frac{4!}{2!(4-2)!} = \frac{4!}{2!\,2!} = 6$$

K_5



$$^5C_2 = \frac{5!}{3!\,2!} = \frac{20}{2} = 10$$

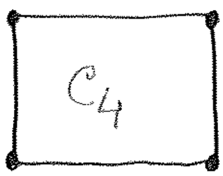
Cycles (simple graph)

- Requires at least 3 nodes/vertices
- Denoted as C_n. for $n \geq 3$
- should have vertices $1, 2, 3, \ldots\ldots n$ ~~ed~~ and edges $\{1,2\}, \{2,3\}, \ldots \{n-1, n\}$ and $\{n, 1\}$

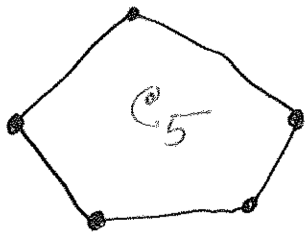
So,

$C_3 \equiv$ Cycle consists of three nodes

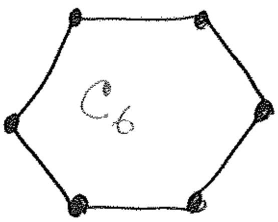
$C_4 \equiv$



$C_5 \equiv$



$C_6 \equiv$

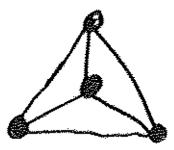


Cycles can be transformed into otherforms of graphs by adding another vertex ~~to~~ and connect this new vertex to each of the n vertices with a new edge. ↳ Known as wheels

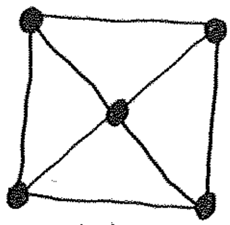
Wheels

- Denoted by W_n, where $n \geq 3$.
- Wheels are extended form of cycles.

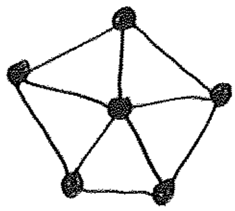
  add a new ~~vet~~ vertex and connect each of the existing vertex with the newly added vertex using new edges.



W_3



W_4



W_5

n-Cubes

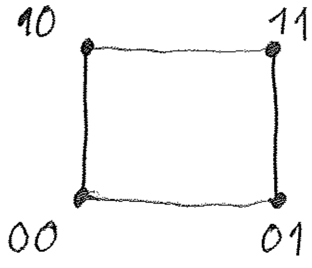
n-dimensional hypercube (n-cube) → Denoted as Q_n

- Graph with vertecies representing 2^n bit-strings of length n.

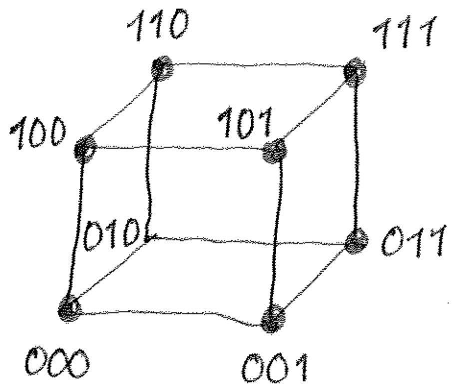
Let's say that bit-string length is 2, then total vertices would be $2^2 = 4$.
string length is 2

00
01
10
11

Two vertices are adjacent if and only if the bit-strings they represent differ exactly in one position. So, Q_2 becomes —

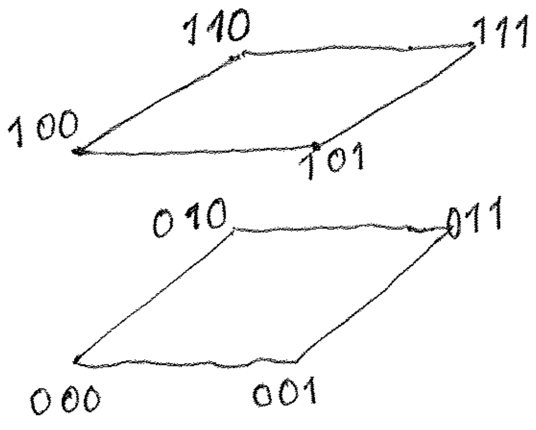


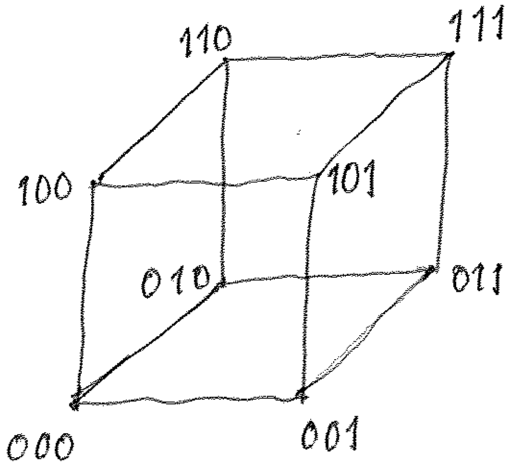
For instance Q_3 could be formed as



Construct $(n+1)$-cube Q_{n+1}

- Take two copies of Q_n
- Label each vertices of $\left[\begin{array}{l} \text{one } Q_n \text{ with "0"} \\ \text{other } Q_n \text{ with "1"} \end{array}\right]$ as the prefix
- Add edges connecting two vertices that have labels differing only in the first bit.





So, number of vertices 2^n
number of edges $2^{n-1} \times n$

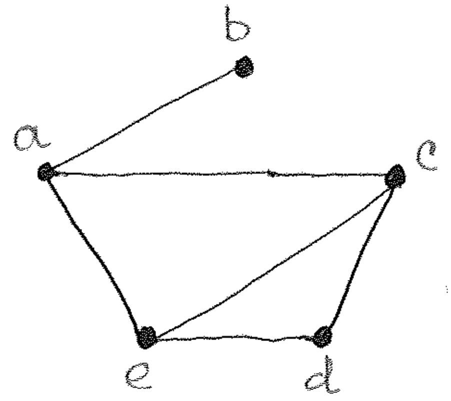
Two vertices are adjacent if and only if symbols differ in exactly one bit

REPRESENTING GRAPH

There are many ways to represent graphs and one of the ways to do that is to use adjacency list.

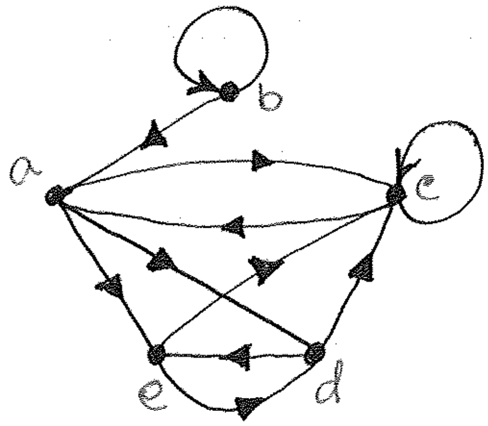
Adjacency list :

- Specifies the vertices that are adjacent to each vertex of the graph.



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

Show the adjacency list for a directed graph as shown below :



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

Adjacency List : Representation of graphs using list of edges, or by adjacency lists, often becomes cumbersome. Instead, matrices are used to simplify the computation.

Two matrix approaches commonly used are :

1. Based on adjacency of vertices
2. Incidence of vertices and edges

Suppose, $G = (V, E)$ is a simple graph, where $|V| = n$, and the vertices of G are listed arbitrarily as $1, 2, \ldots n$,

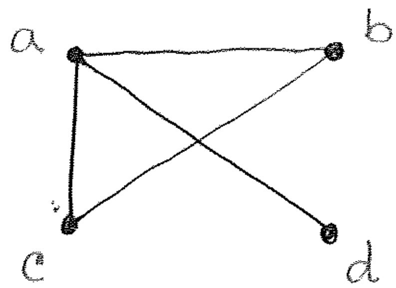
then, the adjacency matrix A of G with respect to the listing of vertices, is "$n \times n$" zero-one matrix, where

$(i, j)^{th}$ entry 1 means i and j are adjacent

$(i, j)^{th}$ entry 0 means i and j are not adjacent.

That is, $$a_{ij} = \begin{cases} 1 & \text{if } \{i, j\} \text{ is an edge of } G \\ 0 & \text{otherwise} \end{cases}$$

Example :



Let's list the nodes as

a : 1
b : 2
c : 3
d : 4

So, 4×4 will be the dimension of the matrix

So, $$A = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Here, the adjacency matrix A is for the simple graph.

does not contain loop

multiple edges are not allowed

$A^T \equiv$ Transpose of A

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

So, $$A^T = A$$

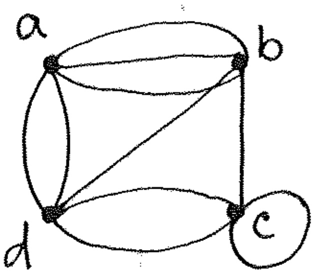
That is, adjacency matrix for simple undirected graph is symmetric

Adjacency matrix can also be used to represent undirected graphs with loops and with multiple edges. The matrix entries are decided as follows:

(i) Loop at a vertex a_i is represented as 1 in the corresponding position a_{ii}

(ii) Multiple edges between a pair of node is denoted with exact number of edges

For instance, consider the pseudograph as below



The adjacency matrix for the node order a, b, c, d is

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

Here, cc entry is 1

	a	b	c	d
a	aa	ab	ac	ad
b	ba	bb	bc	bd
c	ca	cb	cc	cd
d	da	db	dc	dd

Interestingly, For each node order the adjacency matrix will be different.

Also, if $A = [a_{ij}]$ $\forall i, \forall j$ then the adjacency matrix elements follows the specific order of the elements initially fixed. That is:

Node :	a	b	c	d
index :	1	2	3	4

so, ~~node~~ a_{23} denotes edge for b & c

Adjacency matrix for directed graph

Suppose, the ordering of the nodes are 1, 2, ...n
$G = (V, E)$ is the directed graph. Then the adjacency matrix

has 1 in $(i,j)^{th}$ position if there's an edge from i^{th} node to j^{th} node.

That is,

$$a_{ij} = \begin{cases} 1 & \text{for } \{i, j\} \text{ is an edge of } G \\ 0 & \text{Otherwise} \end{cases}$$

As it may happen that $(j,i)^{th}$ position may not have 1 because of the fact that there's no edge from j^{th} node i^{th} node, the adjacency matrix is not necessarily a symmetric matrix.

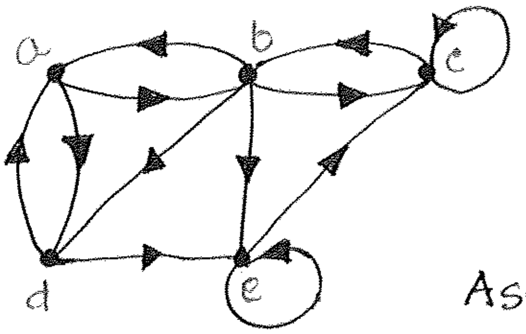
- Adjacency matrix can be used to represent directed graph with multiple edges.

  However, the adjacency matrix for directed multigraph is not zero-one matrix. Instead, for directed graph with multiple edges, the matrix entry

  a_{ij} equals the number of edges that are from ~~to~~ i to j. That is, if there are two edges from i towards j exist, the a_{ij} location of the adjacency matrix would be $a_{ij} = 2$.

Example

Find the adjacency matrix of the below graph



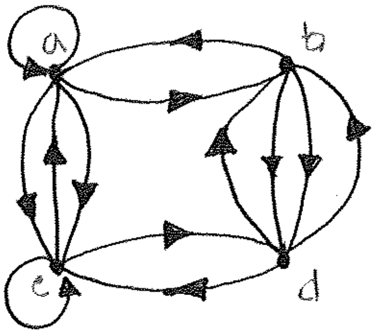
Here, we have 5 nodes, so we have an adjacency matrix of dimension 5×5

Assume that the order of the nodes a, b, c, d, e

So, the adjacency matrix for the given order of the vertices:

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

Adjacency matrix for the below directed graph



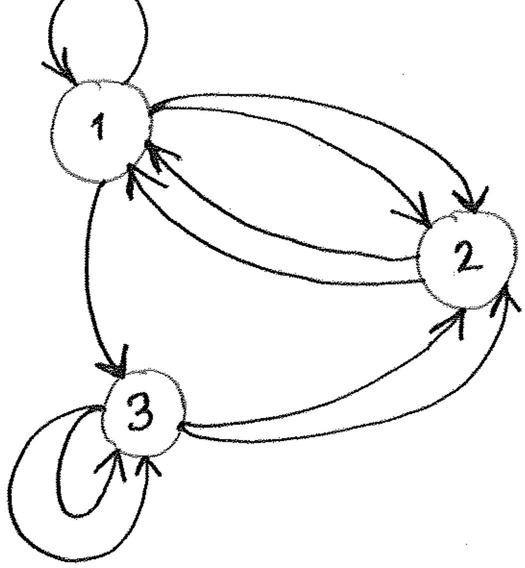
The given graph is a directed pseudograph as it contains both loop and multiple edges between a pair of vertices.

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

Example

Given the adjacency matrix, draw the corresponding graph

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

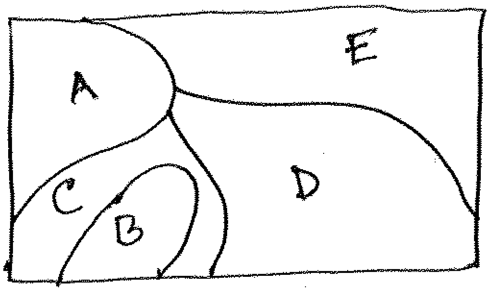


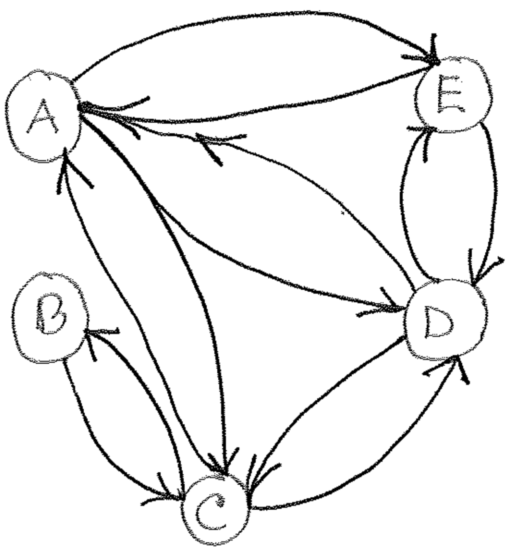
As $a_{31} \neq a_{13}$

$a_{32} \neq a_{23}$

we can conclude that it is for directed graph

Draw the directed graph considering that an edge exists between two different areas if and only if they Share common boundaries.





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

Trade-offs : Adjacency List and Adjacency Matrices

An optimal choice between the two approaches is dependent on the type of graph we analyze.

For sparse graph, where we have relatively few edges among the possible set of edges, we prefer adjacency list. Because,

if the degree of each vertex is not exceeding a constant "c" that is considerably smaller than n, Total vertices

then, each adjacency list contains "c" or fewer number of nodes.

as the degree does not exceed "c"

So, The maximum amount of items will be in all lists must be less than or equal to "cn"

whereas, the adjacency matrix of the graph with n vertices has n^2 entries.

However,

For dense graph where the number of edges are significantly large, the adjacency matrix is preferred.

For instance, a graph that contains more than half of the possible edges.

Incidence Matrices

It is another alternative approach to represent a graph. Consider that $G=(V,E)$ is a graph representing an undirected graph. Let,

vertices : $1, 2, 3, \ldots \ldots n$

edges : $e_1, e_2, e_3 \ldots \ldots m$

Then, the incidence matrix with respect to the aforementioned ordering is the matrix M with dimension

$$M_{n \times m}$$

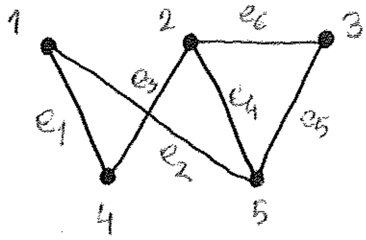
where,

$m_{ij} \equiv$ entry at the i^{th} row and j^{th} column

$\equiv$ 1 when edge e_j is incident with node i

0 otherwise

Example :

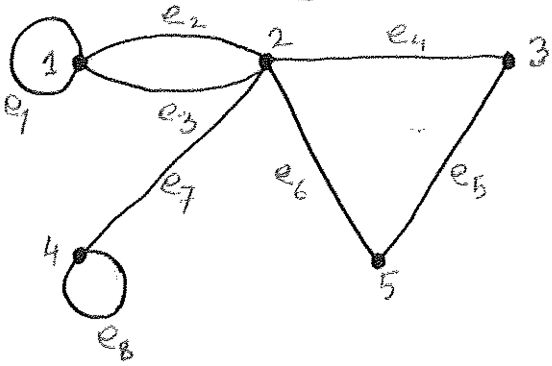


There are 1, 2, 3, 4, 5 vertices

edges: $e_1, e_2, e_3, e_4, e_5, e_6$

$$\begin{array}{c|c} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Incidence matrices can also be used to represent multiple edges and loops.



- multiple edges $\equiv$ columns with identical entries.
- loops $\equiv$ columns with exactly one entry equal to 1

The incidence matrix becomes:

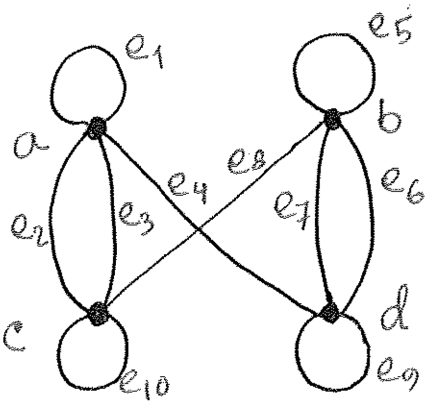
$$\begin{array}{c c}
 & \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}
\end{array}$$

For loop at node 1

For multiple edges

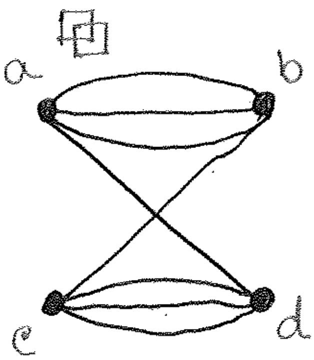
Check e_2, e_3 columns, they are identical, and they represent the two edges drawn between vertices 1 and 2.

Example:



$$\begin{array}{c c}
 & \begin{array}{cccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}
\end{array}$$

It's a pseudograph



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

Incidence matrix for directed graph

For the directed graph, the inciden matrix approach considers different numeric values for the pair of nodes that are connected by an edge. Precisely,

if node pair (u, v) has an edge connecting them, according to the directed graph definition, the edge

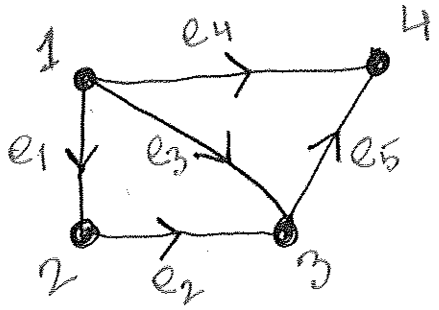
starts at u

ends at v

+1 for the node v in the matrix.

−1 for the node in the matrix for the corresponding edge.

Example:



So, the incidence matrix becomes:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} -1 & 0 & -1 & -1 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Graph Density: Graph density is defined as the ratio between the number of edges and the total possible edges.

Suppose, a graph has "n" nodes, so, total possibe edges would be for undirected graph

$$\binom{n}{2} = {}^{n}C_{2} = \frac{n!}{(n-2)!\,2!}$$

$$= \frac{n(n-1)\,(n-2)!}{(n-2)!\,2!} = \frac{n(n-1)}{2}$$

Assume that, the graph has a total of "m" edges. So, graph density would be:

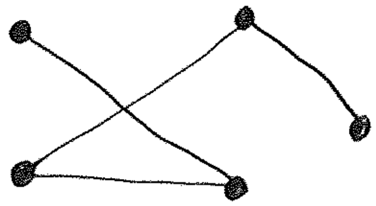
$$\frac{m}{\frac{n(n-1)}{2}} = \frac{2m}{n(n-1)}$$

For a directed graph: For a directed graph, total possible edges is $n(n-1)$. So, the density is

$$\frac{m}{n(n-1)}$$

Density of a simple graph, that is a graph without any loop or multiedges, has a maximum value 1. That is, a completely connected graph has density 1.

For instance, consider the simple graph as



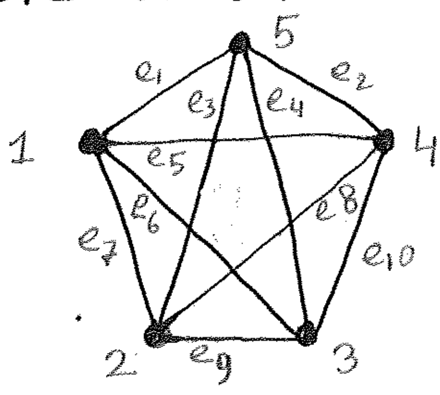
Here, edges $m = 4$

nodes $n = 5$

So, $\text{Density} = \dfrac{2m}{n(n-1)}$

$= \dfrac{4 \times 2}{5 \times 4} = \dfrac{1 \times 2}{5} = \dfrac{2}{5}$

Consider a complete and simple graph as shown



Here, $m = 10$

$n = 5$

So, $\text{density} = \dfrac{2m}{n(n-1)}$

$= \dfrac{2 \times 10}{5 \times 4} = 1$ (which is the maximum density for any simple graph.)

- Density of multigraphs and pseudographs can be higher than 1.

Bipartite Graphs :

Bipartite Graphs: A graph $G = (V, E)$ consists of non-empty set of vertices V and a set of edges.

Interestingly, in some graphs the set of vertices can be partitioned into two subsets of vertices. For instance,

consider a scenario of a customer service office, where each member of the service office and the customers are represented as vertices. Also,

receiving customer service by any customer is an edge.

Now, if we form a graph based on the provided customer service and received service, the vertices can be portioned into two subsets.

1. Vertices denoting service providers.
2. vertices denoting customers.

Such characteristics along with further tuning leads us to define the Bipartite graph

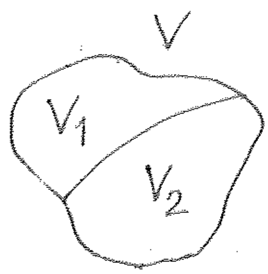
Definition :

A simple graph G is Bipartite if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that

every edge of the graph connects a vertex in V_1 and a vertex in V_2.

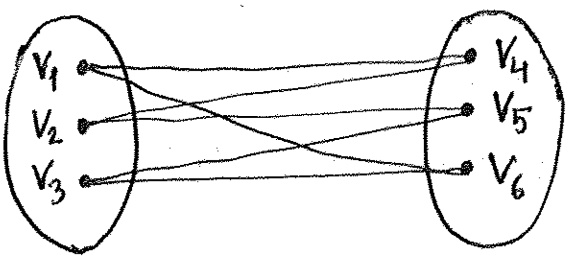
In otherwords,

no edge in graph G connects two vertices from same subsets.



Concept of Partition

Example :



$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$

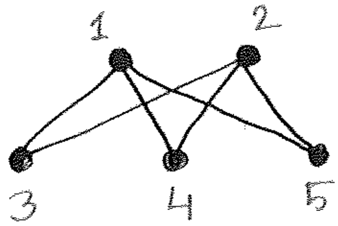
Here,

$V_1 = \{v_1, v_2, v_3\}$

$V_2 = \{v_4, v_5, v_6\}$

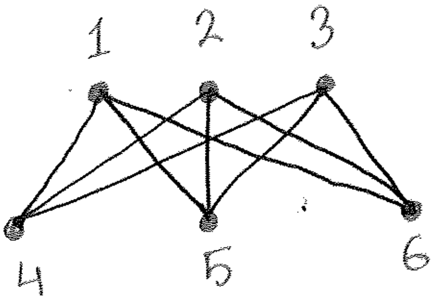
Also, no edge exists between any two vertices of the same subset.

A few examples of bipartite graph :



Complete graph, that is, exactly one edge between each pair of distinct vertices.

bipartite with $V_1 = \{1, 2\}, V_2 = \{3, 4, 5\}$



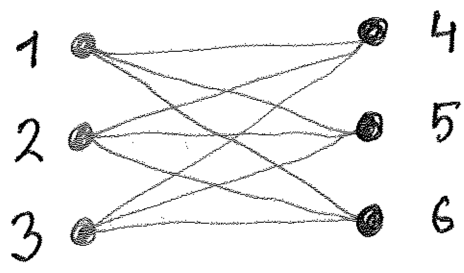
complete graph, bipartite

$V_1 = \{1, 2, 3\}, \quad V_2 = \{4, 5, 6\}$

四. How to determine whether a graph is bipartite ?

Theorem : A simple graph is bipartite if and only if it is possible to assign one of two different colors to each vertex so that two adjacent vertices do not have the same color.

$V = \{1, 2, 3, 4, 5, 6\}$



adjacency:

Also, known as neighbor nodes. If u and v nodes are endpoints for an edge, then they are adjacent.

Proof: Assume $G = (V, E)$ is a bipartite graph. So, $V = V_1 \cup V_2$, where $V_1 \cap V_2 = \phi$ and every edge in E connects a vertex from V_1 and a vertex from V_2.

Now, if we assign one color to each vertex in V_1 and the second color to each vertex in V_2, then it is evident that no two adjacent vertices are assigned the same color.

Again, suppose that it is possible to assign colors to vertices of the graph just by using two different colors. so that no two adjacent nodes have the same color.

Let V_1 be the set of vertices assigned for one color, and V_2 be the other colored-nodes. Then, V_1 and V_2 are disjoint, and $V = V_1 \cup V_2$.

Furthermore, every edge connects a vertex in V_1 and a vertex in V_2 and no two adjacent vertices are either both in V_1 or both in V_2.

That is, Graph G is bipartite

Connectivity

Consider that we want to transmit a message from computer A to computer D. Similar to such scenario, one might be interested to know if in a protein-protein interaction network protein A regulates the dynamics or production by interacting with a protein D.

In the former case, we need to traverse along the edges and see if we have the connectivity available from A to D

In the later case, the protein-protein interaction network must include an interaction network that ensures role of A in regulating the proteins that interact with D, or A directly interact with D.

Such requirement of connectivity can be solved using the concept of paths in graph.

Paths: A path, generally, is a sequence of edges that begins at a vertex and travels from ~~vet~~ vertex to vertex along the edges of the graph. A formal definition of a Path goes as follows:

Consider G is an undirected graph and n is a non-negative integer.

A path of length n is for the graph G is a sequence of edges $e_1, e_2 \dots e_n$ such that

$$
\begin{array}{llll}
e_1 & \text{is associated with} & & \{x_0, x_1\} \\
e_2 & " & " & \{x_1, x_2\} \\
\vdots & & & \\
e_n & " & " & \{x_{n-1}, x_n\}
\end{array}
$$

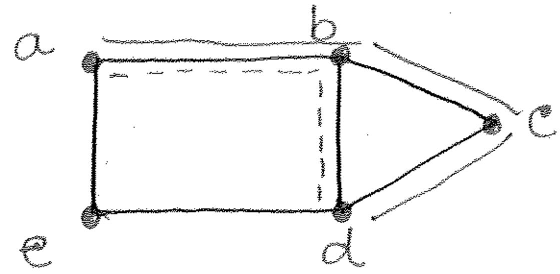
and $x_0 = u$, $x_n = v$, and the Path is from u to v of length n.

For a simple graph, the above path can be stated as a sequence of vertices

$$x_0, x_1, x_2 \dots \dots x_n$$

Circuits: A Path is a circuit if it begins and ends at the same vertex. That is, if $u = v$, the above path is a circuit.

Example :



Path from a to d
{a, b, d}
ab, bd

Path from a to d
{a, b, c, d}
ab, bc, cd

If we define a path as : {a, b, c, d, e, a}, then it is a circuit.

equivalently, {a, b, c, d, e}

other circuits possible are {a, b, d, e} and {b, c, d}

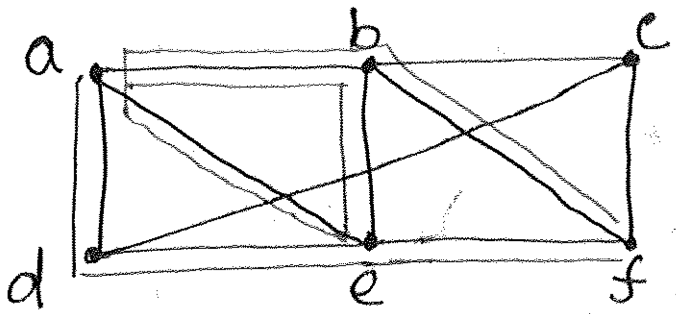
Alternatively, in many books, Walk is introduced to denote the travel from one node to the other along the edges. A path becomes a special case of Walk under such definition of traversal in the graph.

Walk : A sequence of edges that travel from node u to node v, where both vertices and node can revisited

whereas,

Path : vertices and nodes can not be repeated.

Consider a graph:



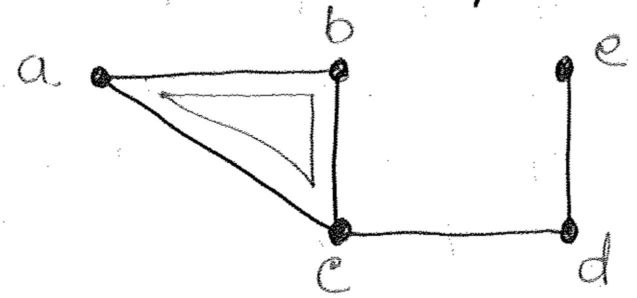
Say, we want to travel from node a to f.

Walk: The red line drawn is a walk

Path: Blue line forms a path.

Cycle: This is similar to circuit concept as defined previously. Precisely,

A path starting and ending at the same node in a graph is defined as the cycle. For instance



$\{a, b, c\}$ is a cycle

For directed graph: The definition remains very similar.

So, a more generic definition goes as

Given, n is non-negative
G is a directed graph

A path of length n from vertex u to vertex v in G is a sequence of of edges $e_1, e_2, \ldots e_n$ of the graph G such that

e_1 is associated with $\{x_0, x_1\}$

e_2 " " " $\{x_1, x_2\}$

$\vdots$

e_n " " " $\{x_{n-1}, x_n\}$

where $x_0 = u$, $x_n = v$.

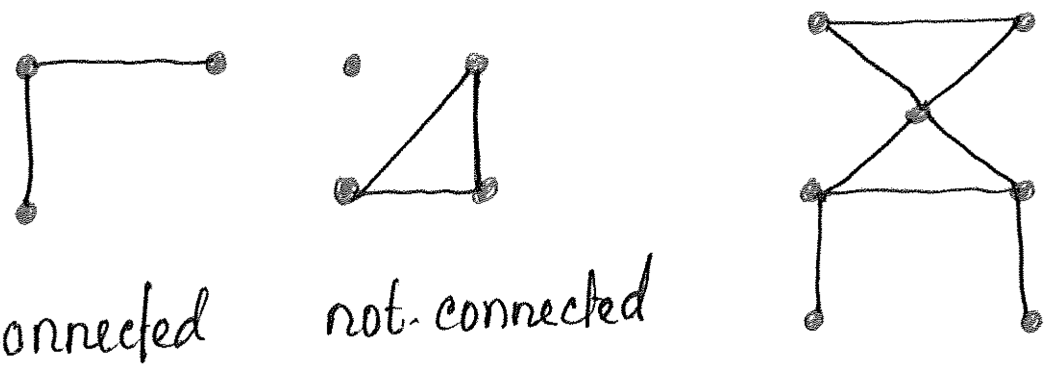
When no multiple edges exist, the path can be represented as

$$x_0, x_1, \ldots \ldots x_n$$

Also, A path of length greater than zero that begins and ends at the same node is a cycle/circuit.

Connectedness in undirected graph

An undirected graph is called connected if there is a path between every pair of distinct vertices. For instance,



Connected not-connected connected