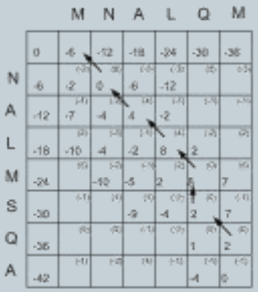
Introduction to Graph Theory

HANDBOOK OF GRAPH THEORY FOR FRESHER'S



Prem Sankar C

M Tech Technology Management

Dept of Futures Studies ,Kerala University

Outline

1. History of Graph Theory
2. Basic Concepts of Graph Theory
3. Graph Representations
4. Graph Terminologies
5. Different Type of Graphs

Why Graph Theory ?

- Graphs used to model pair wise relations between objects
- Generally a network can be represented by a graph
- Many practical problems can be easily represented in terms of graph theory

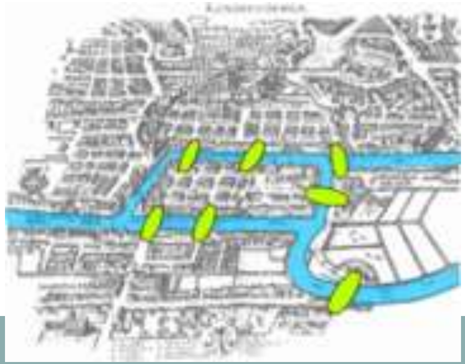
Graph Theory - History

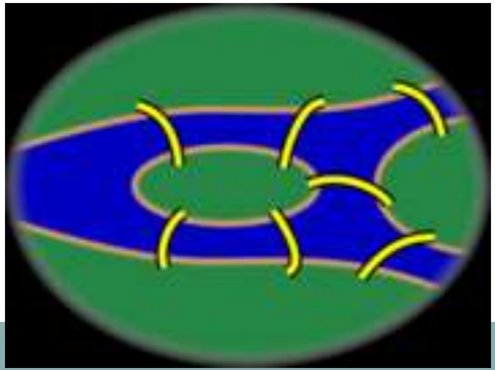
The origin of graph theory can be traced back to Euler's work on the Konigsberg bridges problem (1735), which led to the concept of an Eulerian graph. The study of cycles on polyhedra by the Thomas P. Kirkman (1806 - 95) and William R. Hamilton (1805-65) led to the concept of a Hamiltonian graph.

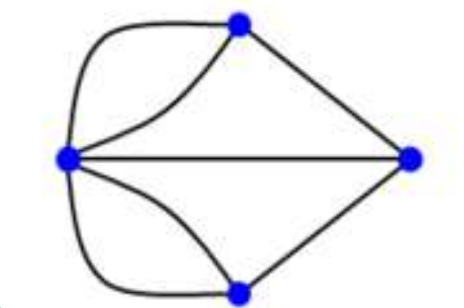
Graph Theory - History

- Begun in 1735
- Mentioned in Leonhard Euler's paper on "*Seven Bridges of* Konigsberg ".

Problem : Walk all 7 bridges without crossing a bridge twice





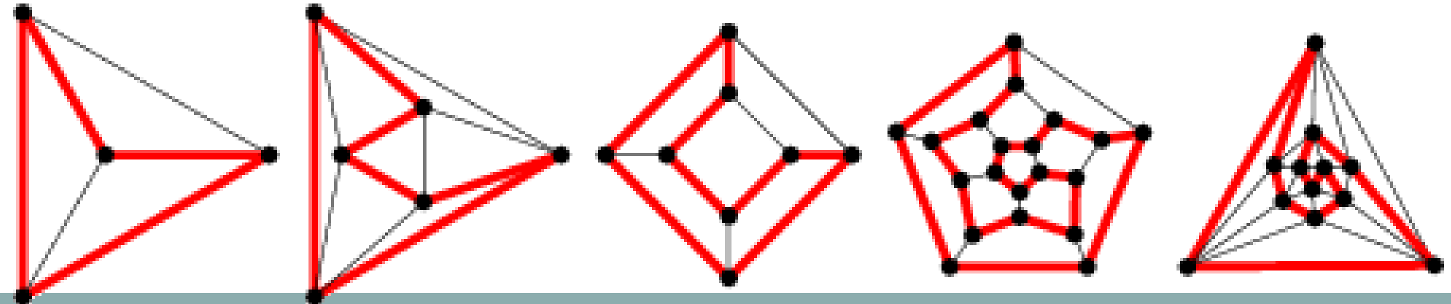


Graph Theory – History…….

Cycles in Polyhedra - polyhedron with no Hamiltonian cycle

Thomas P. Kirkman

William R. Hamilton

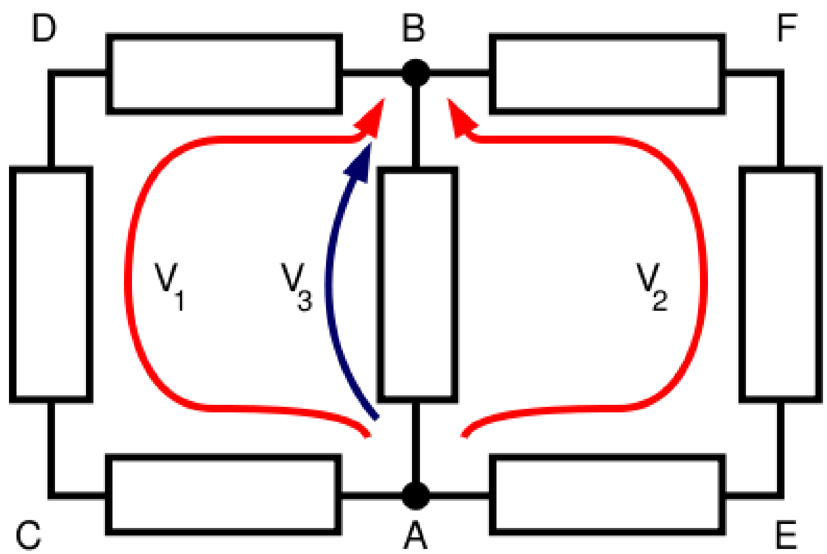


Hamiltonian cycles in Platonic graphs

Graph Theory – History…..

Trees in Electric Circuits

Gustav Kirchhoff



$$V_1 = V_2 = V_3$$

Basic Concepts of Graph Theory

Definition: Graph

- A graph is a collection of nodes and edges
- Denoted by $G = (V, E)$.

  V = **nodes** (vertices, points).

  E = **edges** (links, arcs) between pairs of nodes.

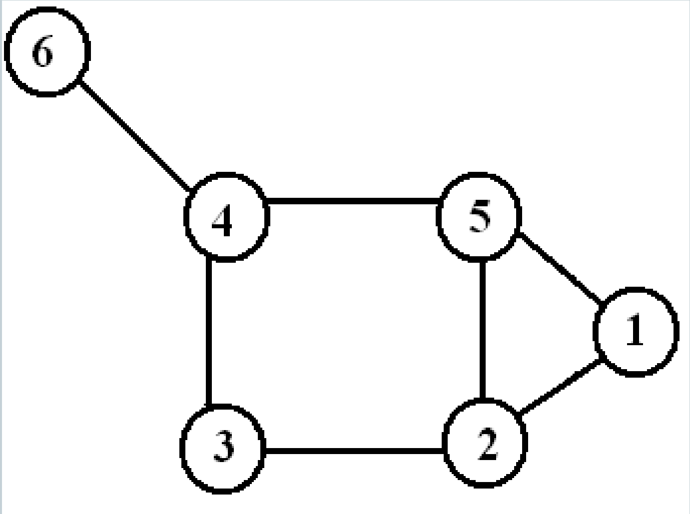
  Graph size parameters: $n = |V|$, $m = |E|$.

-

Vertex & Edge

- Vertex /Node
  - Basic Element
  - Drawn as a node or a dot.
  - V**ertex set** of G is usually denoted by V(G), or V or V_G
- Edge /Arcs
  - A set of two elements
  - Drawn as a line connecting two vertices, called end vertices, or endpoints.
  - The edge set of G is usually denoted by E(G), or E or E_G
- Neighborhood
  - For any node v, the set of nodes it is connected to via an edge is called its neighborhood and is represented as N(v)

Graph :Example



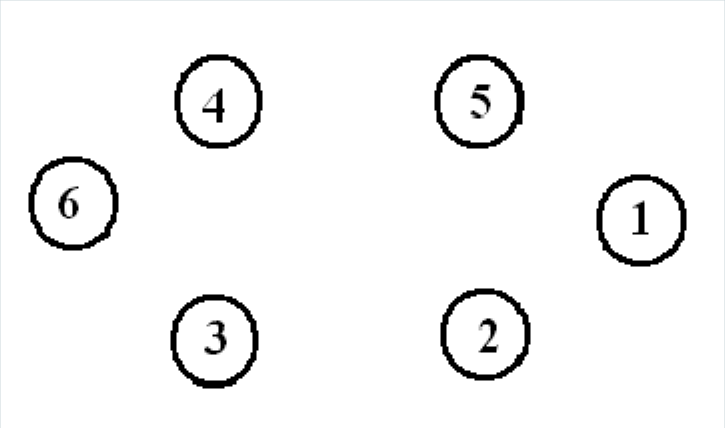
- n:= 6 , m:=7
- Vertices (V) :={1,2,3,4,5,6}
- Edge (E) := {1,2},{1,5},{2,3},{2,5},{3,4},{4,5},{4,6}}
- *N(4) := Neighborhood (4) ={6,5,3}*

Edge types:

- **Undirected**;
  - E.g., distance between two cities, friendships…
- **Directed**; ordered pairs of nodes.
  - E.g ,…
  - Directed edges have a **source** (head, origin) and **target** (tail, destination) vertices
- **Weighted ;** usually weight is associated .

Empty Graph / Edgeless graph

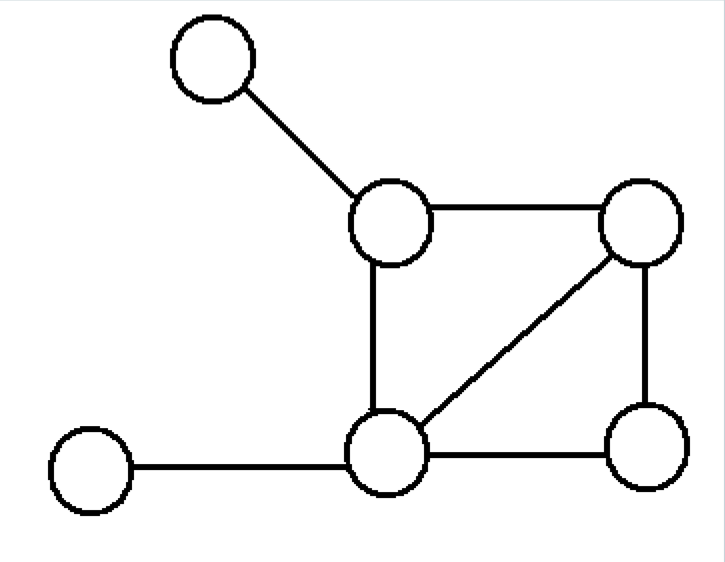
- No edge



- Null graph
  - No nodes
  - Obviously no edge

Simple Graph (Undirected)

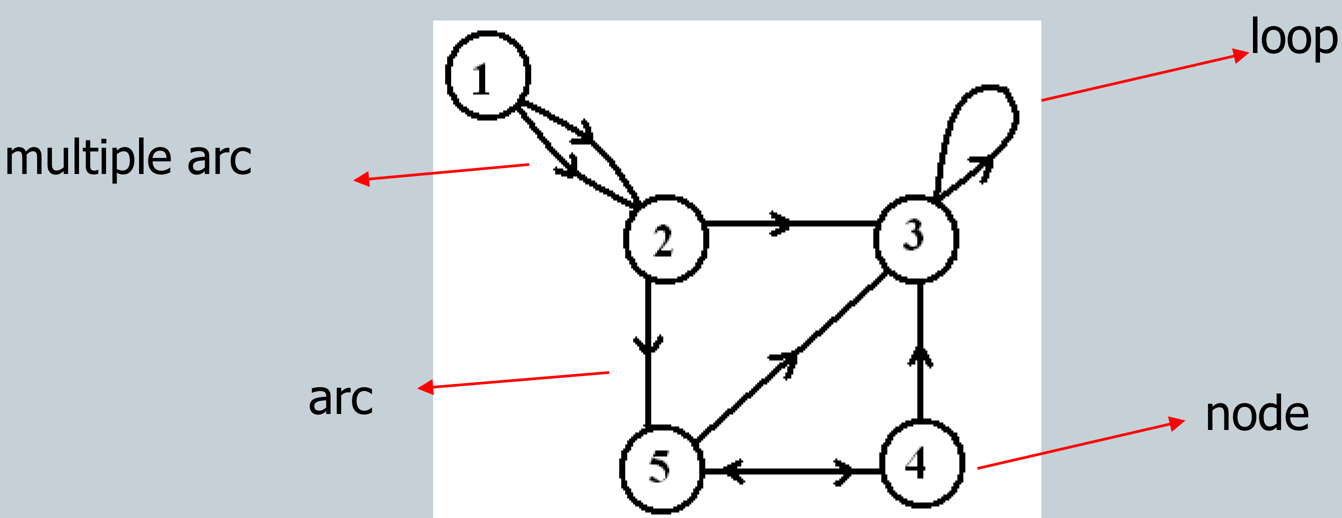
- Simple Graph are undirected graphs without loop or multiple edges
- A = AT



For simple graphs, $\sum_{v_i \in V} \deg(v_i) = 2|E|$

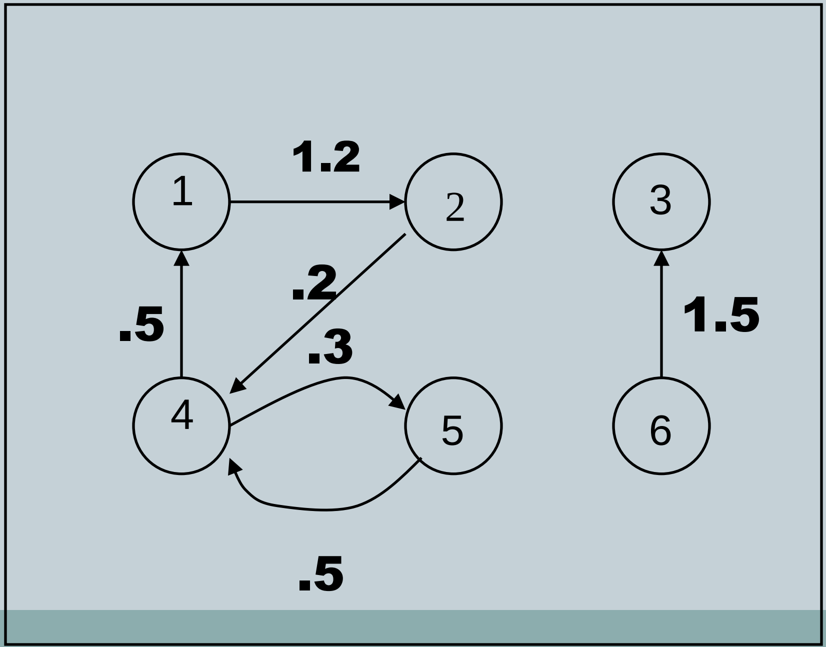
Directed graph : (digraph)

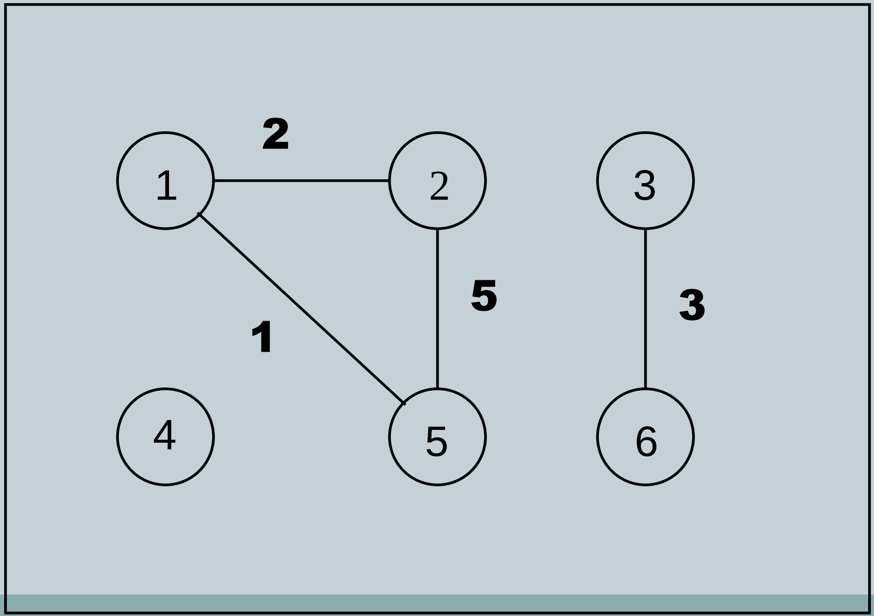
- Edges have directions
- A !=AT



Weighted graph

- is a graph for which each edge has an associated ***weight***



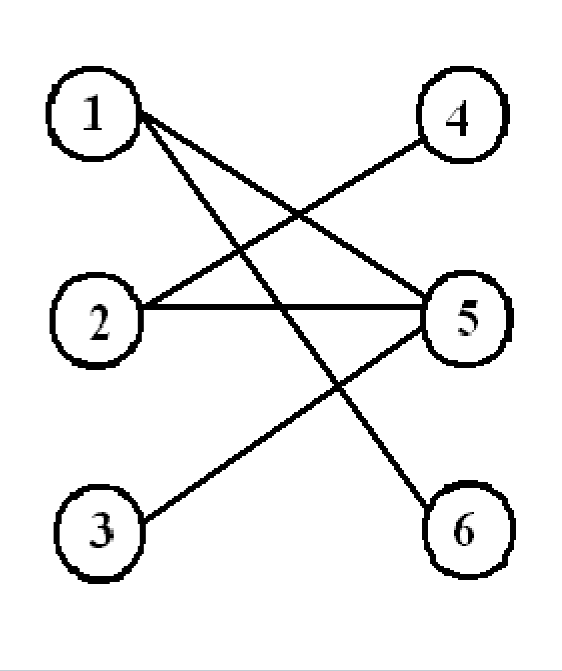


Bipartite Graph

V can be partitioned into 2 sets V_1 and V_2 such that $(u,v)\in E$ implies

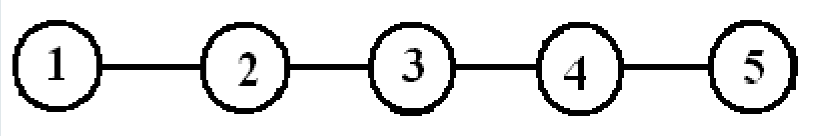
either $u \in V_1$ and $v \in V_2$

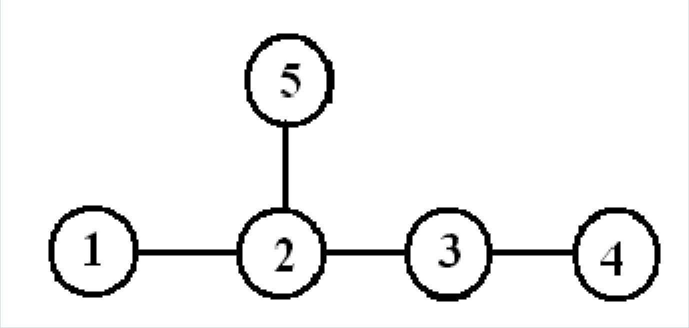
OR $v \in V_1$ and $u \in V_2$.

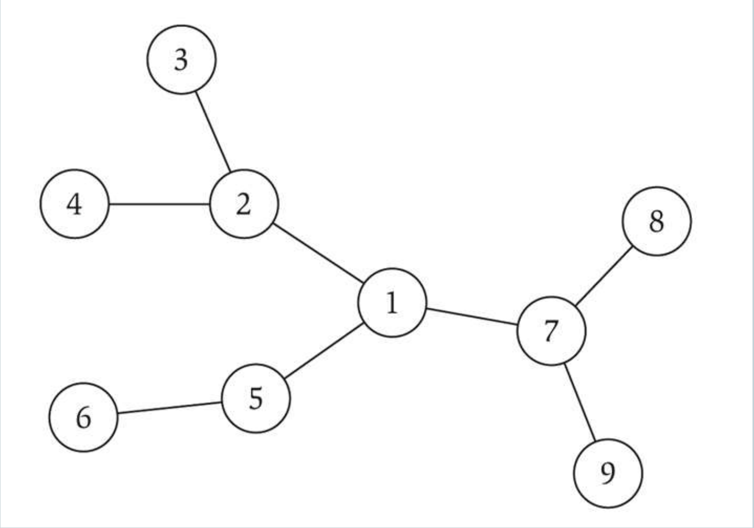


Trees

- An undirected graph is a **tree** if it is connected and does not contain a cycle (Connected Acyclic Graph)
- Two nodes have *exactly* one path between them

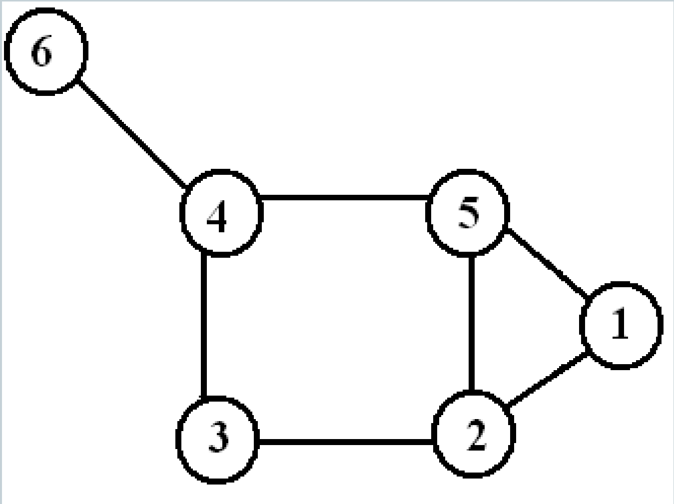


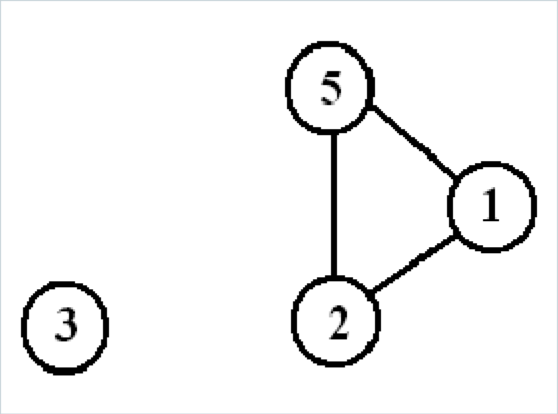




Subgraph

- Vertex and edge sets are subsets of those of G
  - a *supergraph* of a graph G is a graph that contains G as a subgraph.

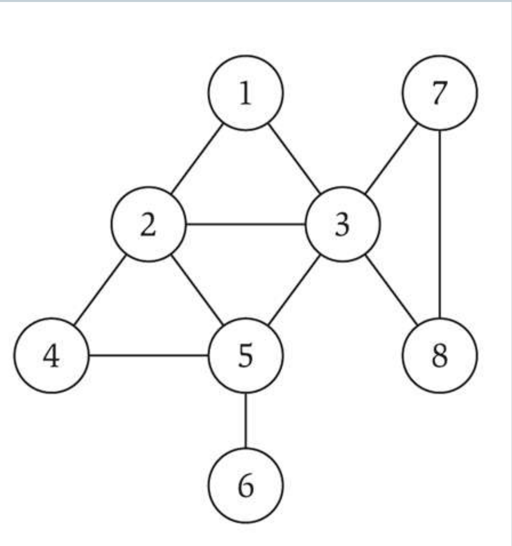




Graph Representations

1. Adjacency Matrix

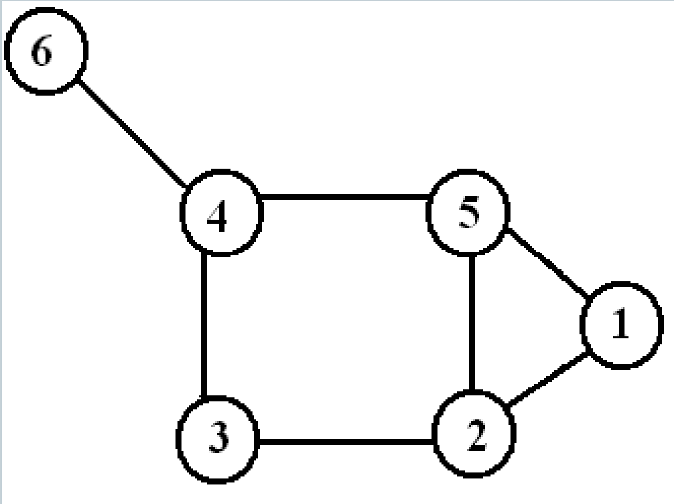
- n-by-n matrix with $A_{uv} = 1$ if (u, v) is an edge.
  - Diagonal Entries are self-links or loops
  - Symmetric matrix for undirected graphs



	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	1	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

2. Incidence Matrix

- V x E
- [vertex, edges] contains the edge's data



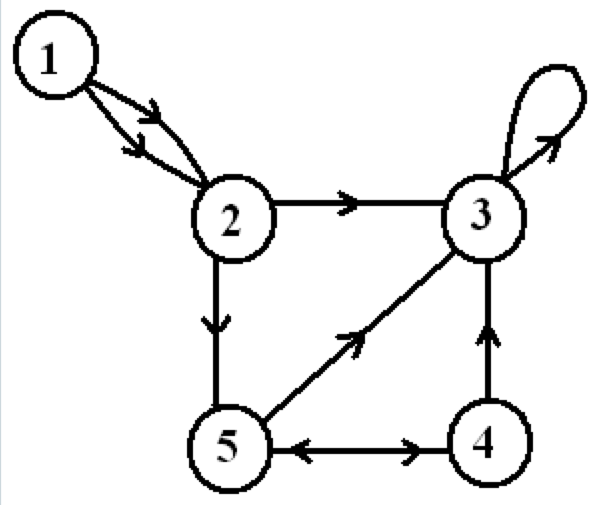
	1,2	1,5	2,3	2,5	3,4	4,5	4,6
1	1	1	0	0	0	0	0
2	1	0	1	1	0	0	0
3	0	0	1	0	1	0	0
4	0	0	0	0	1	1	1
5	0	1	0	1	0	1	0
6	0	0	0	0	0	0	1

3. Adjacency List

- Edge List

Edge List

1 2
1 2
2 3
2 5
3 3
4 3
4 5
5 3
5 4

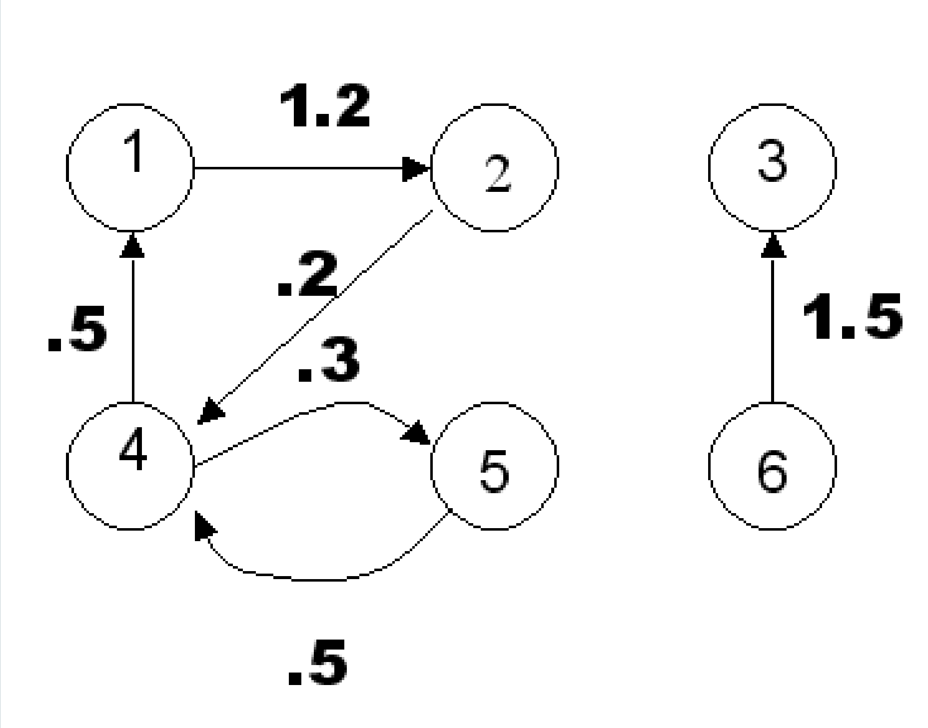


- Adjacency List (node list)

Node List

1 2 2
2 3 5
3 3
4 3 5
5 3 4

Edge Lists for Weighted Graphs



Edge List

1 2 1.2
2 4 0.2
4 5 0.3
4 1 0.5
5 4 0.5
6 3 1.5

Graph Terminologies

Classification of Graph Terms

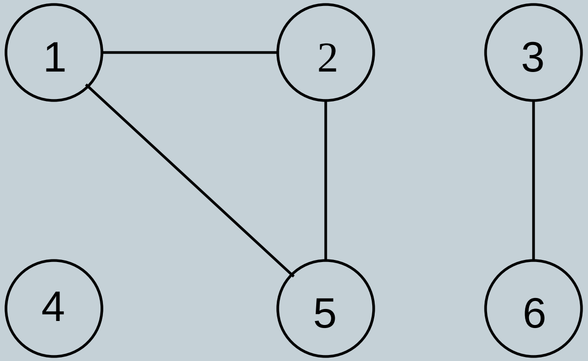
- Global terms refer to a whole graph
- Local terms refer to a single node in a graph

Connected and Isolated vertex

- Two vertices are **connected** if there is a path between them

- Isolated vertex – not connected

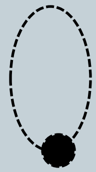
isolated vertex



Adjacent nodes

- **Adjacent nodes -**Two nodes are adjacent if they are connected via an edge.
  - If edge ***e={u,v} ∈ E(G),*** we say that *u* and *v* are **adjacent** or **neigbors**

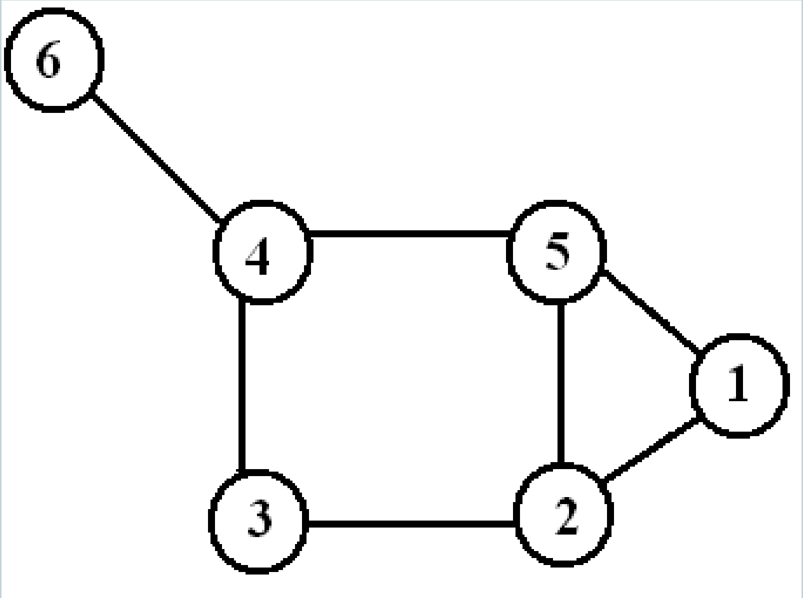
- **An edge where the two end vertices are the same is called a loop, or a self-loop**



Degree (Un Directed Graphs)

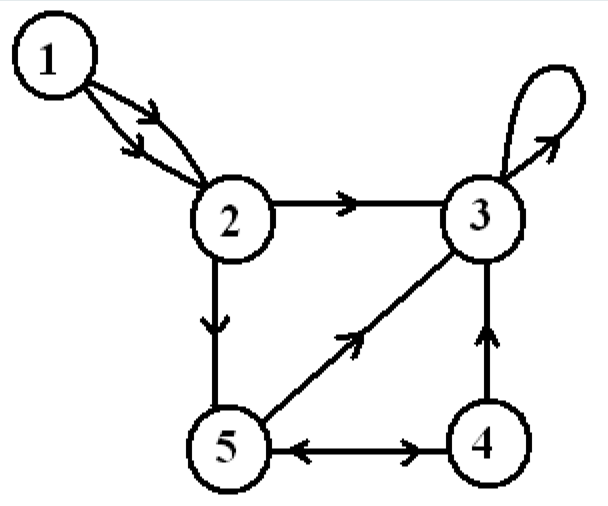
- Number of edges incident on a node

The degree of 5 is 3



Degree (Directed Graphs)

- In-degree: Number of edges entering
- Out-degree: Number of edges leaving
- Degree = indeg + outdeg



outdeg(1)=2
indeg(1)=0

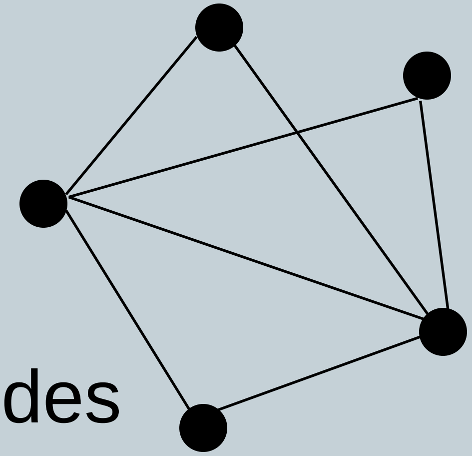
outdeg(2)=2
indeg(2)=2

outdeg(3)=1
indeg(3)=4

Walk

- **trail**: no edge can be repeat

  a-b-c-d-e-b-d



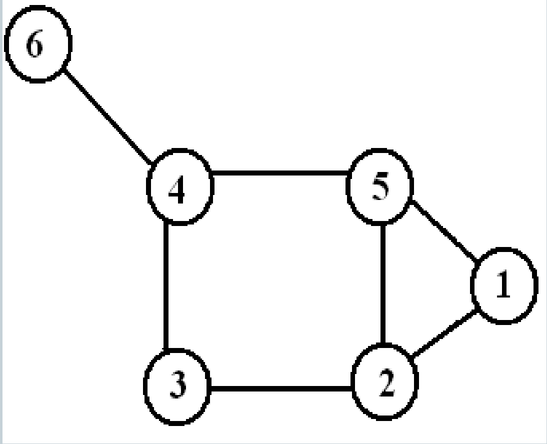
- **walk**: a path in which edges/nodes can be repeated.

  a-b-d-a-b-c

- A walk is ***closed*** is a=c

Paths

- **Path**: is a sequence P of nodes $v_1, v_2, \ldots, v_{k-1}, v_k$
- No vertex can be repeated
- A closed path is called a cycle
- The length of a path or cycle is the number of edges visited in the path or cycle

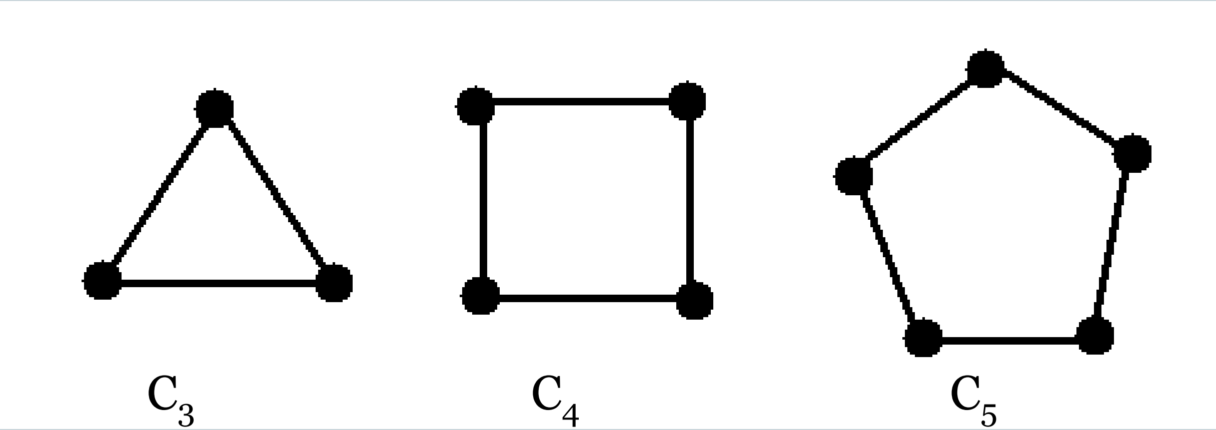


Walks and Paths

1,2,5,2,3,4	1,2,5,2,3,2,1	1,2,3,4,6
walk of length 5	CW of length 6	path of length 4

Cycle

- Cycle - closed path: **cycle (a-b-c-d-a)** , closed if $x=y$
- Cycles denoted by C_k, where k is the number of nodes in the cycle



Shortest Path

- **Shortest Path is the path between two nodes that has the shortest length**
- **Length** – number of edges.
- **Distance** between u and v is the length of a shortest path between them
- The diameter of a graph is the length of the longest shortest path between any pairs of nodes in the graph

THANK YOU

Prem Sankar C

M Tech Technology Management

Dept of Futures Studies

Kerala University