Graph Theory and Representation

Some Applications of Graph Theory

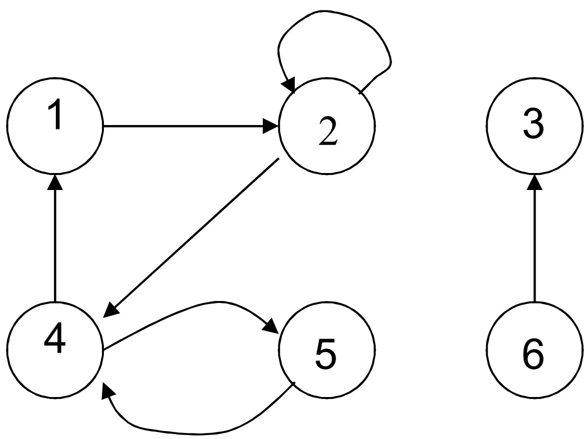
- Some high level problems in networks, eg
- 1 Topology planning
- 2 Dimensioning
- 3 Routing
- 4 Traffic engineering

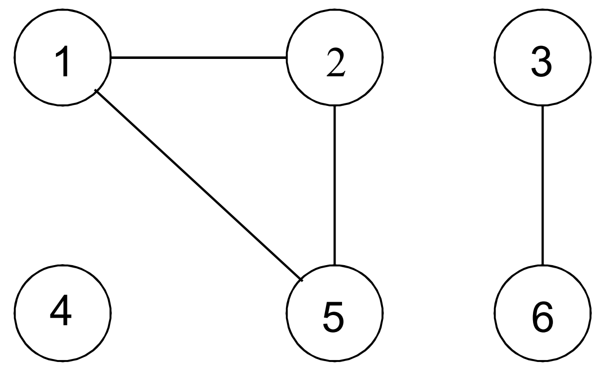
Graphs ↔ Networks

Graph (Network)	Vertexes (Nodes)	Edges (Arcs)	Flow
Communications	Telephones exchanges, computers, satellites	Cables, fiber optics, microwave relays	Voice, video, packets
Circuits	Gates, registers, processors	Wires	Current
Mechanical	Joints	Rods, beams, springs	Heat, energy
Hydraulic	Reservoirs, pumping stations, lakes	Pipelines	Fluid, oil
Financial	Stocks, currency	Transactions	Money
Transportation	Airports, rail yards, street intersections	Highways, railbeds, airway routes	Freight, vehicles, passengers

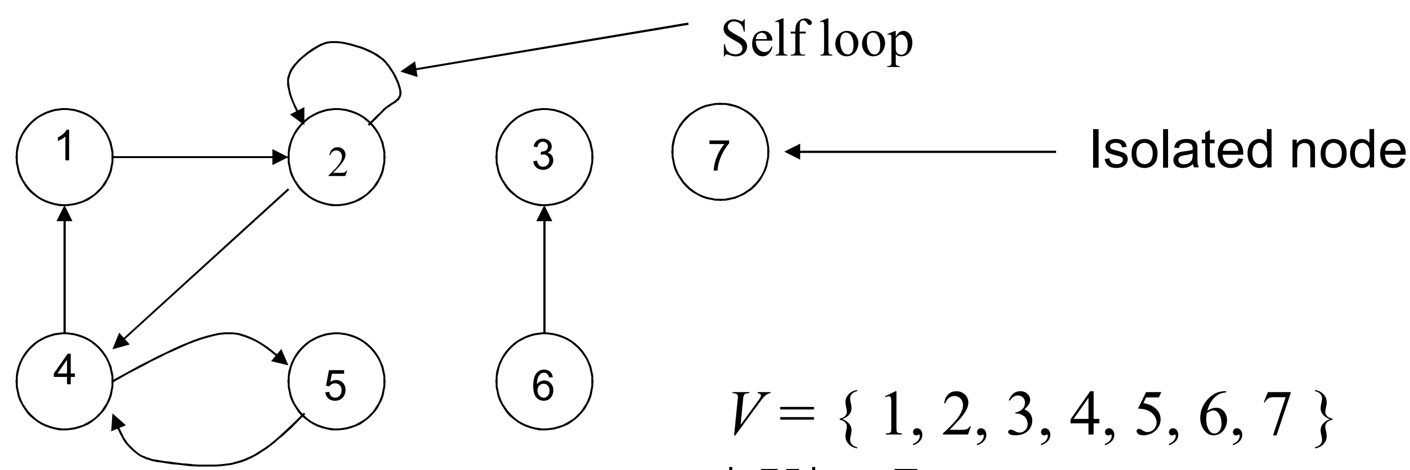
What is a Graph?

- Informally a *graph* is a set of nodes joined by a set of lines or arrows.





A ***directed graph****,* also called a ***digraph*** **G** is a pair (*V*, *E*), where the set *V* is a finite set and *E* is a binary relation on *V* . The set ***V*** is called the **vertex set** of *G* and the elements are called vertices. The set *E* is called the **edge set** of *G* and the elements are *edges* (also called *arcs*). A edge from node *a* to node *b* is denoted by the ordered pair (*a*, *b*).



$V = \{ 1, 2, 3, 4, 5, 6, 7 \}$

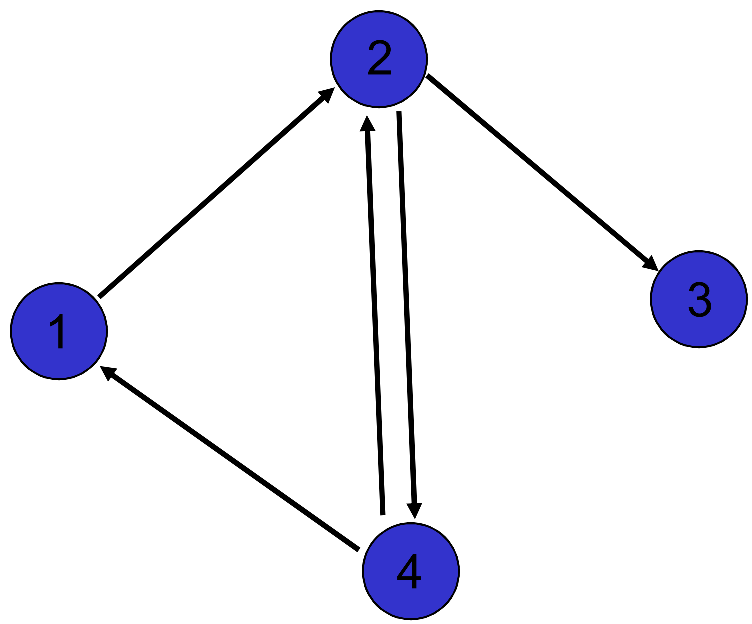
$| V | = 7$

$E = \{ (1,2), (2,2), (2,4), (4,5), (4,1), (5,4),(6,3) \}$

$| E | = 7$

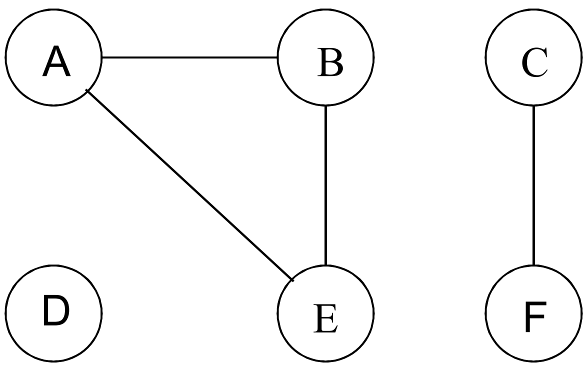
Directed Graph

An edge $e \in E$ of a directed graph is represented as an ordered pair (u,v), where $u, v \in V$. Here u is the initial vertex and v is the terminal vertex. Also assume here that $u \neq v$



$V = \{ 1, 2, 3, 4\}$, $| V | = 4$
$E = \{(1,2), (2,3), (2,4), (4,1), (4,2)\}$, $| E | = 5$

An ***undirected graph*** $G = (V, E)$, but unlike a digraph the edge set E consist of unordered pairs. We use the notation (a, b) to refer to a directed edge, and $\{a, b\}$ for an undirected edge.



$V = \{A, B, C, D, E, F\}$

$|V| = 6$

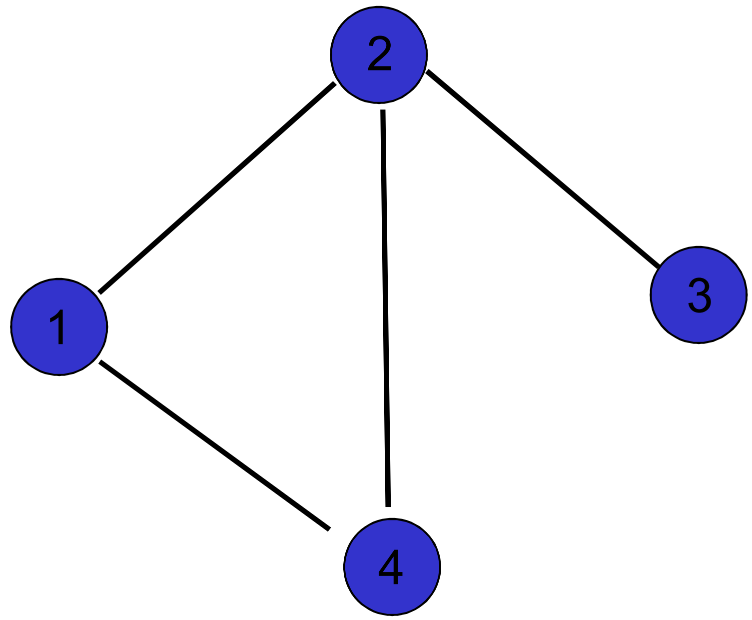
$E = \{\{A, B\}, \{A,E\}, \{B,E\}, \{C,F\}\}$

$|E| = 4$

Some texts use (a, b) also for undirected edges.
So (a, b) and (b, a) refers to the same edge.

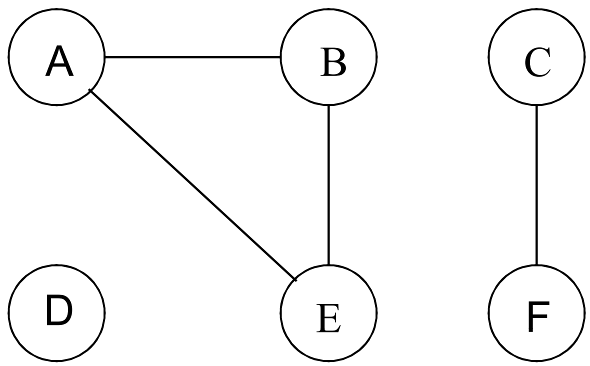
Undirected Graph

An edge $e \in E$ of an undirected graph is represented as an unordered pair $(u,v)=(v,u)$, where $u, v \in V$. Also assume that $u \neq v$



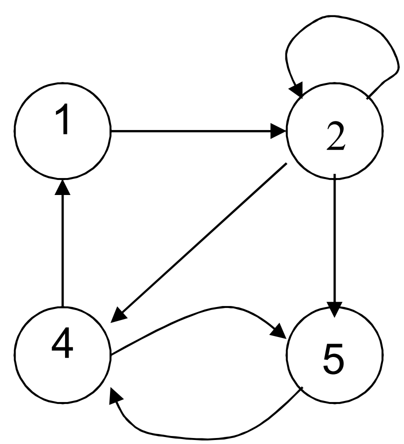
$V = \{ 1, 2, 3, 4\}$, $|V| = 4$
$E = \{(1,2), (2,3), (2,4), (4,1)\}$, $|E| = 4$

Degree of a Vertex in an undirected graph is the number of edges incident on it. In a directed graph , the ***out degree*** of a vertex is the number of edges leaving it and the ***in degree*** is the number of edges entering it.



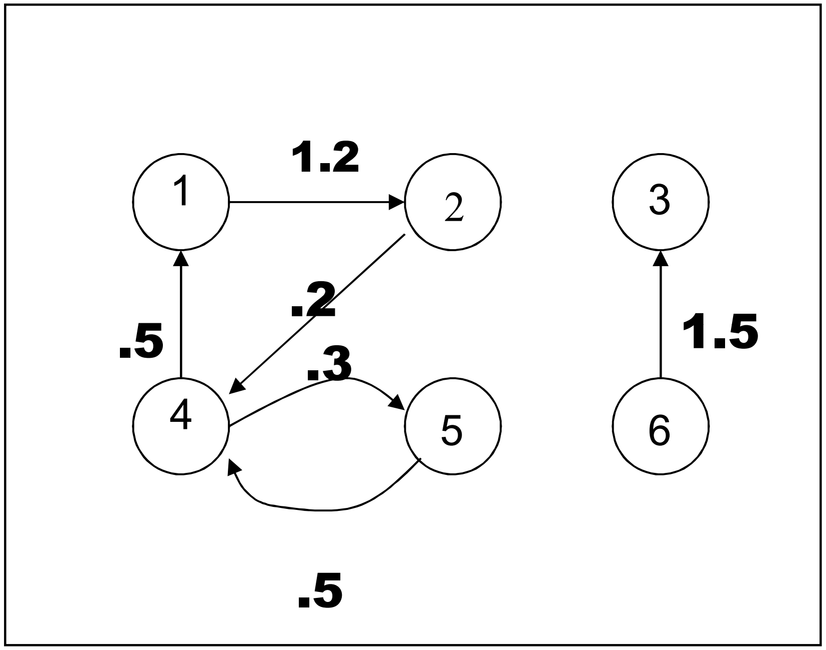
The degree of B is 2.

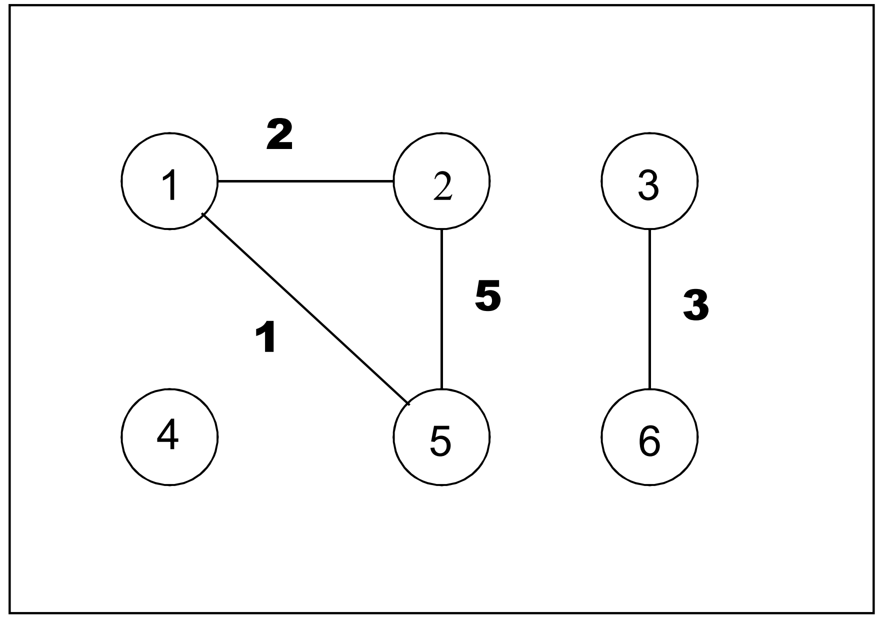
Self-loop



The in degree of 2 is 2 and the out degree of 2 is 3.

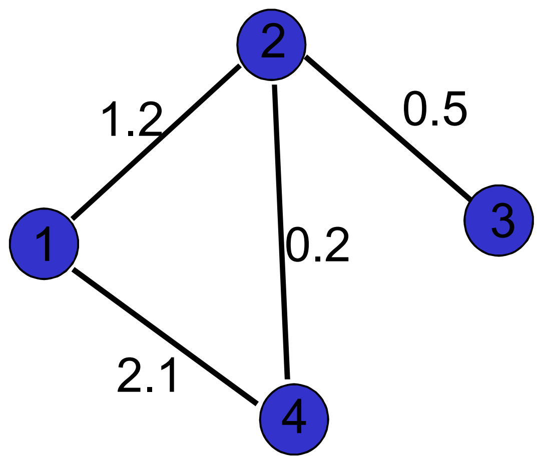
A ***weighted graph*** is a graph for which each edge has an associated ***weight***, usually given by a ***weight function*** $w: E \rightarrow \mathbf{R}$.

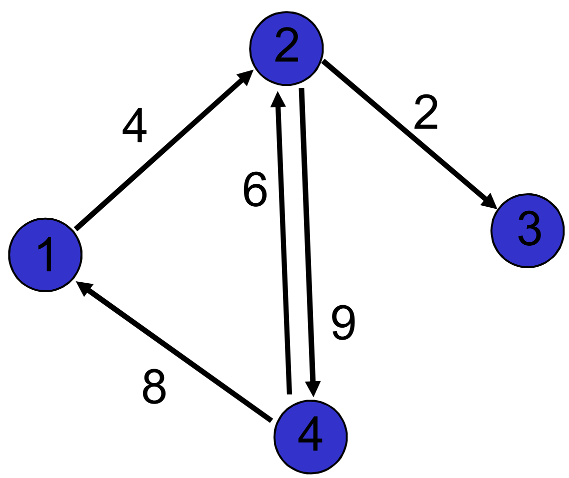




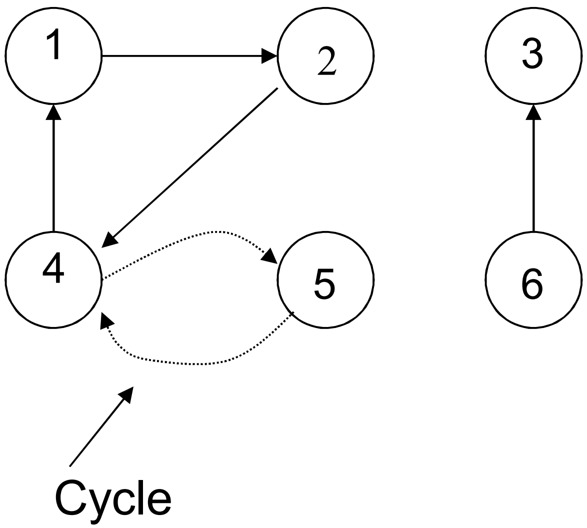
Weighted Graph

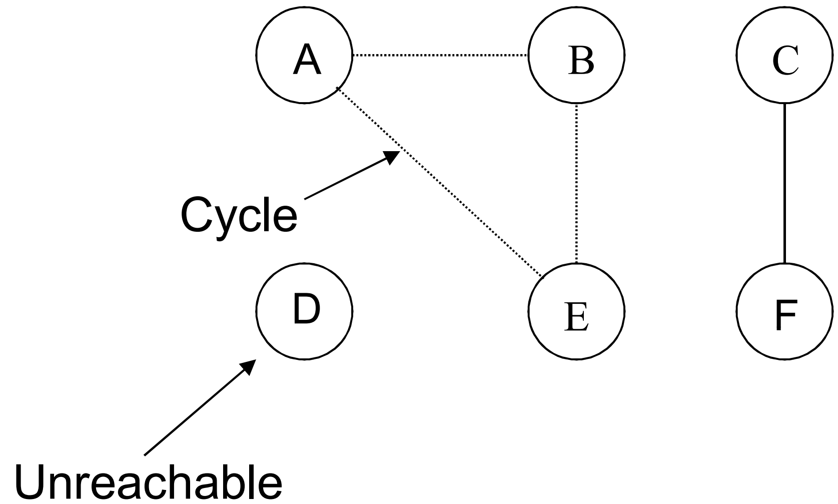
A *weighted graph* is a graph for which each edge has an associated *weight*, usually given by a *weight function* $w: E \rightarrow \mathrm{R}$





A ***path*** is a sequence of vertices such that there is an edge from each vertex to its successor. A path from a vertex to itself is called a ***cycle***. A graph is called ***cyclic*** if it contains a cycle; otherwise it is called ***acyclic*** A path is ***simple*** if each vertex is distinct.

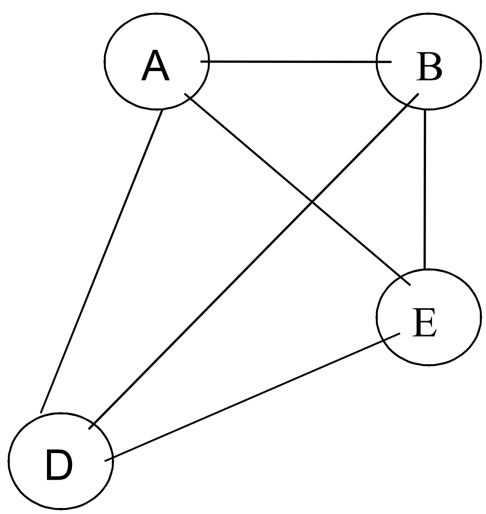




Simple path from 1 to 5
= (1, 2, 4, 5)
or as in our text
((1, 2), (2, 4), (4,5))

If there is path *p* from *u* to *v* then we say *v* is **reachable** from *u* via *p*.

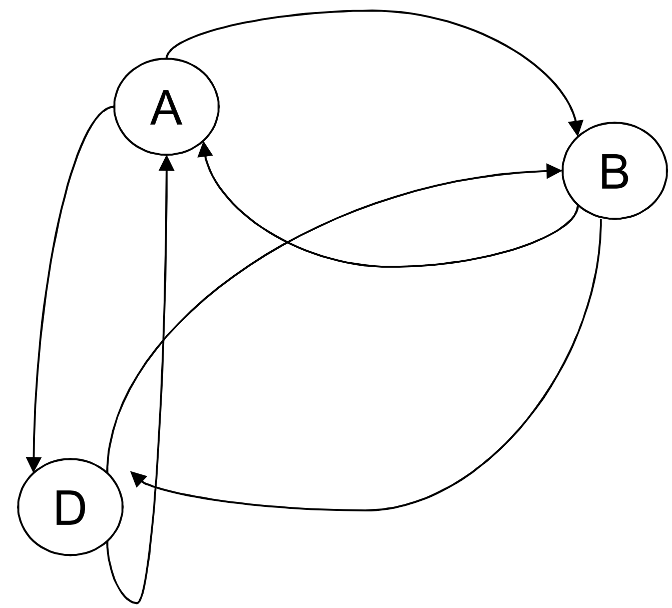
A ***Complete graph*** is an undirected/directed graph in which every pair of vertices is adjacent. If (u, v) is an edge in a graph G, we say that vertex v is ***adjacent*** to vertex u.



4 nodes and (4*3)/2 edges

V nodes and V*(V-1)/2 edges

Note: if self loops are allowed V(V-1)/2 +V edges



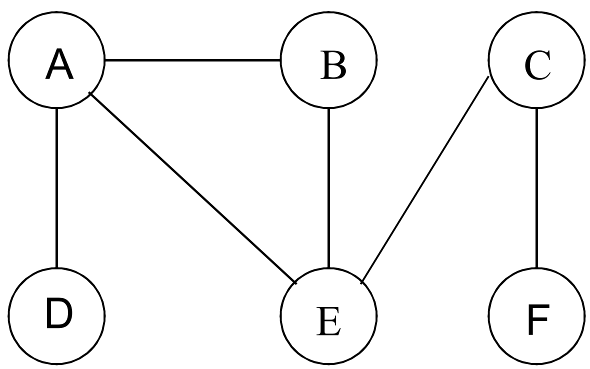
3 nodes and 3*2 edges

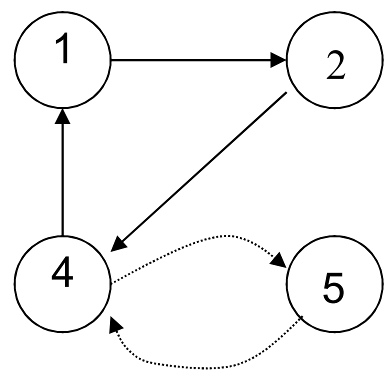
V nodes and V*(V-1) edges

Note: if self loops are allowed V^2 edges

An undirected graph is ***connected*** if you can get from any node to any other by following a sequence of edges OR any two nodes are connected by a path.

A directed graph is ***strongly connected*** if there is a directed path from any node to any other node.





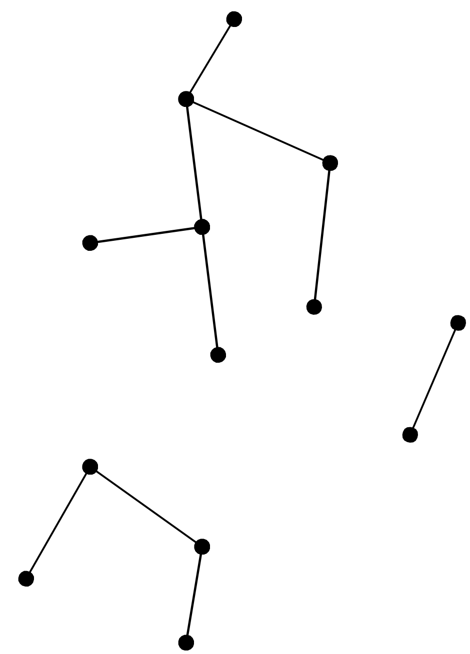
- A graph is ***sparse*** if $| E | \approx | V |$
- A graph is ***dense*** if $| E | \approx | V |^2$.

A ***free tree*** is an acyclic, connected, undirected graph. A ***forest*** is an acyclic undirected graph. A ***rooted tree*** is a tree with one distinguished node, ***root***.

Let $G = (V, E)$ be an undirected graph.

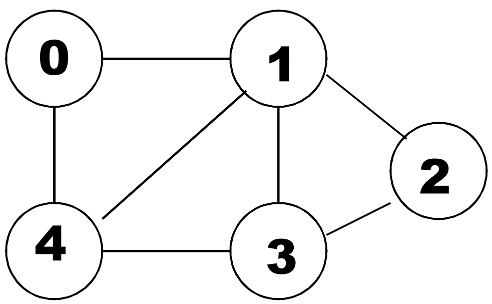
The following statements are equivalent.

1. G is a tree
2. Any two vertices in G are connected by unique simple path.
3. G is connected, but if any edge is removed from E, the resulting graph is disconnected.
4. G is connected, and $|E| = |V| - 1$
5. G is acyclic, and $|E| = |V| - 1$
6. G is acyclic, but if any edge is added to E, the resulting graph contains a cycle.



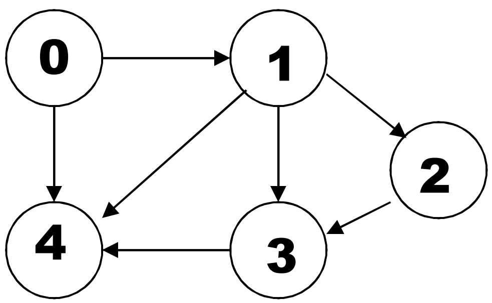
Adjacency-matrix-representation of a graph $G = (V, E)$ is a $|V| \times |V|$ matrix $A = (a_{ij})$ such that

$$a_{ij} = \begin{cases} 1 \text{ (or some Object)} & \text{if } (i, j) \in E \text{ and} \\ 0 \text{ (or null)} & \text{otherwise.} \end{cases}$$



$$\begin{array}{c|ccccc|} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 2 & 0 & 1 & 0 & 1 & 0 \\ 3 & 0 & 1 & 1 & 0 & 1 \\ 4 & 1 & 1 & 0 & 1 & 0 \\ \hline \end{array}$$

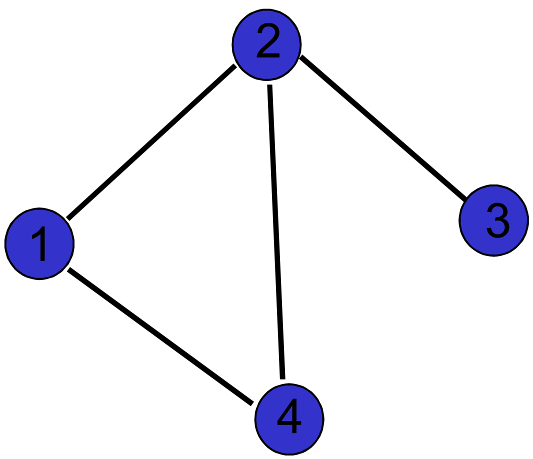
Adjacency Matrix Representation for a Directed Graph



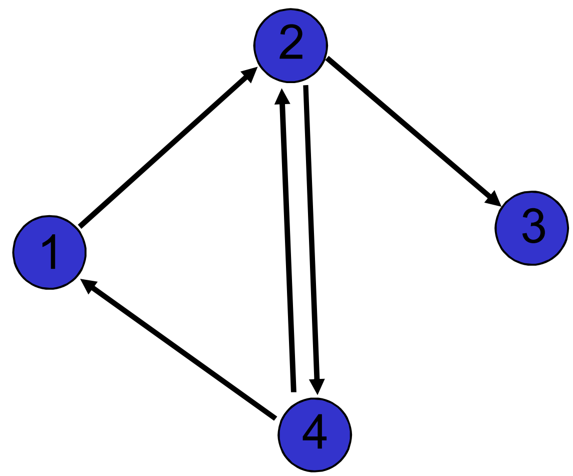
$$
\begin{array}{c|ccccc|}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 \\
2 & 0 & 0 & 0 & 1 & 0 \\
3 & 0 & 0 & 0 & 0 & 1 \\
4 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
$$

Degree of a Vertex

Degree of a vertex in an undirected graph is the number of edges incident on it. In a directed graph, the *out degree* of a vertex is the number of edges leaving it and the *in degree* is the number of edges entering it



The *degree* of vertex 2 is 3



The *in degree* of vertex 2 is 2 and the *in degree* of vertex 4 is 1