CSC-391: Data Structures

Lecture: 14

Graph-1: Basic Theory

Lecture-14:

Graph-1: Basic Theory

Objectives of this Lecture:

- Definition of graphs and related concepts
  - Vertices/nodes, edges, adjacency, incidence
  - Degree, in-degree, out-degree
  - Subgraphs, unions, isomorphism
  - Adjacency matrices

- Types of Graphs
  - Trees
  - Undirected graphs
  - Simple graphs, Multigraphs, Pseudographs
  - Digraphs, Directed multigraph
  - Bipartite
  - Complete graphs, cycles, wheels, cubes, complete bipartite

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Graph Basic:

- Some problems that can be represented by a graph
    - computer networks
    - airline flights
    - road map
    - course prerequisite structure
    - tasks for completing a job
    - flow of control through a program
    - many more

What is a Graph?

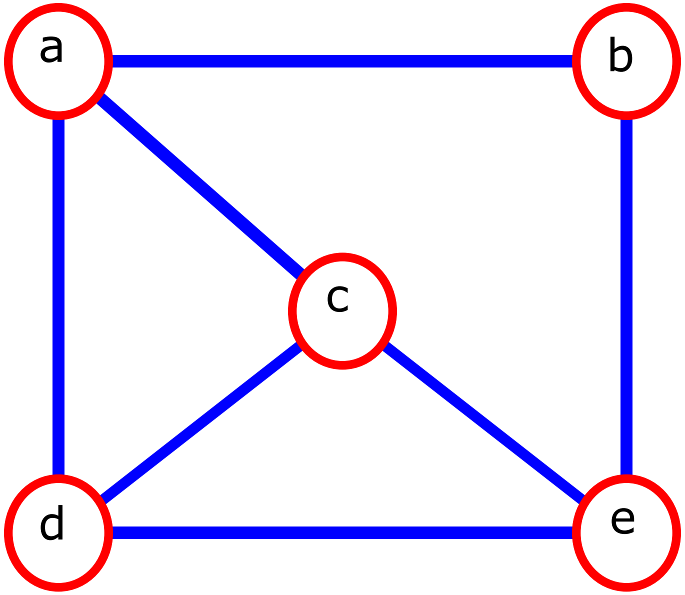
- In mathematics, a graph is an abstract representation of a set of objects where some pairs of the objects are connected by links.
- The interconnected objects are represented by mathematical abstractions called vertices (or nodes), and the links that connect some pairs of vertices are called edges.
- Typically, a graph is depicted in diagrammatic form as a set of dots for the vertices, joined by lines or curves for the edges. Therefore, a graph is a finite set of nodes with edges between nodes.
- Graphs are one of the objects of study in discrete mathematics.

What is a Graph?

- A graph, indicated by G = (V,E), is composed of:

  V: set of vertices

  E: set of edges connecting the vertices in V
- An edge e = (u,v) is a pair of vertices
- Example of a graph:



V= {a,b,c,d,e}

E= {(a,b),(a,c),(a,d),

(b,e),(c,d),(c,e),(d,e)}

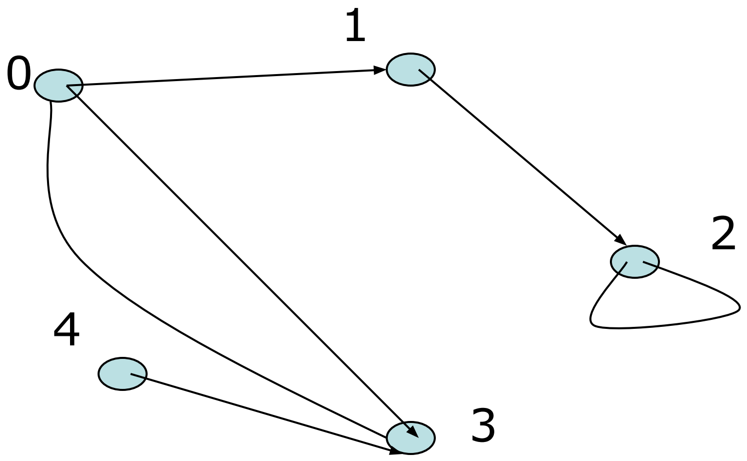
Vertices are also called *nodes* or *points*, and edges are also called *lines* or *arcs*.

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

What is a Graph?

Examples of Graphs:

- V={0,1,2,3,4}
- E={(0,1), (1,2), (0,3), (3,0), (2,2), (4,3)}



□ When (x,y) is an edge, we say that x is *adjacent to* y, and y is *adjacent from* x.

❑ 0 is adjacent to 1.
❑ 1 is not adjacent to 0.
❑ 2 is adjacent from 1.

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

What is a Graph?

- The edges in a graph may be directed (asymmetric) or undirected (symmetric).
- For example, if the vertices represent people at a party, and there is an edge between two people if they shake hands, then this is an undirected graph, because if person A shook hands with person B, then person B also shook hands with person A.
- On the other hand, if the vertices represent people at a party, and there is an edge from person A to person B when person A knows of person B, then this graph is directed, because, one person knowing another person does not necessarily imply the reverse; for example, many fans may know of a celebrity, but the celebrity is unlikely to know of all their fans). This latter type of graph is called a *directed* graph and the edges are called *directed edges* or *arcs*.

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Graph Terminology:

- The vertices belonging to an edge are called the **ends**, **endpoints**, or **end vertices** of the edge.
- A vertex may exist in a graph and not belong to an edge.
- The **order** of a graph is the number of vertices. It is denoted as **|V|**.
- The **size** of a graph is the number of edges. It is denoted as |E|.
- The **degree** of a node or vertex is the number of edges that connect to it, that is, degree of a vertex is the number of edges incident to that vertex. If **u** is a node, then its degree is denoted as **deg(u).**
- For directed graph,
  - the in-degree of a vertex *v* is the number of edges that have *v* as the head. That is, ***in-degree*** of a vertex (deg$^-$) counts the number of edges that stick *in* to the vertex.
  - the out-degree of a vertex *v* is the number of edges that have *v* as the tail. That is, ***out-degree*** (deg$^+$) counts the number of edges sticking *out*.
  - if d_i is the degree of a vertex *i* in a graph *G* with *n* vertices and *e* edges, the number of edges is

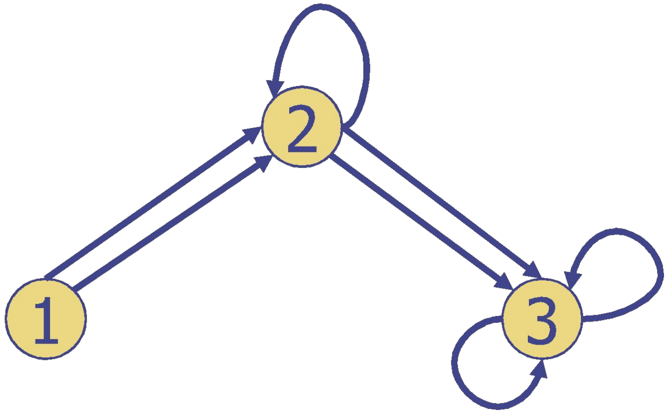
$$e = \left(\sum_{0}^{n-1} d_i\right) / 2$$

Graph Terminology:

Examples: in-degree and out-degree

Q: What are in-degrees and out-degrees of all the vertices?

Ans: $\deg^{-}(1) = 0$
$\deg^{-}(2) = 3$
$\deg^{-}(3) = 4$
$\deg^{+}(1) = 2$
$\deg^{+}(2) = 3$
$\deg^{+}(3) = 2$



Types of Graph:

Different purposes require different types of graphs.

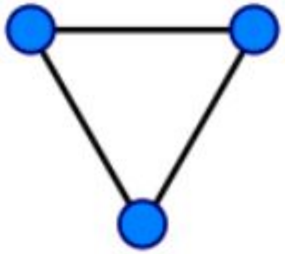
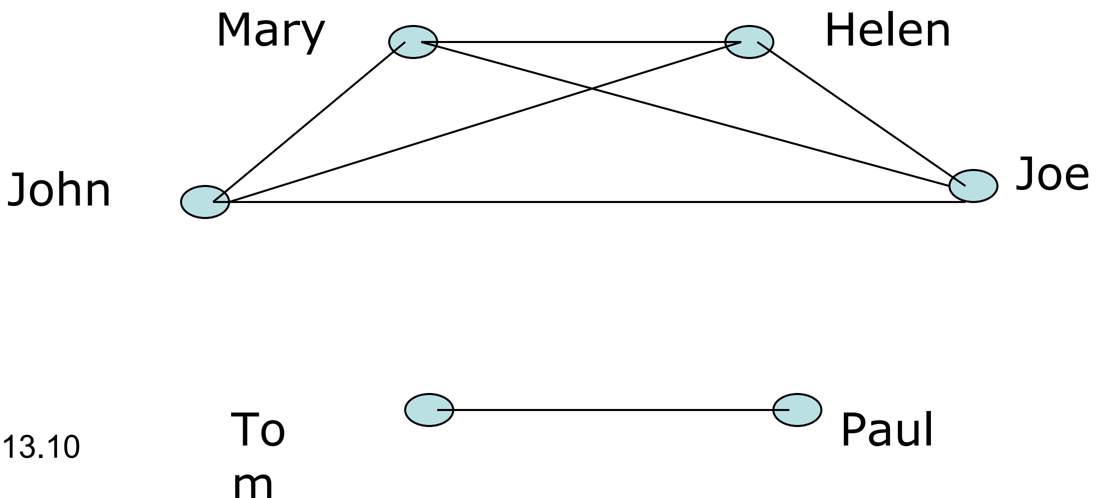


Figure: An undirected graph

Undirected graph:

- An undirected graph is one in which edges have no orientation. The edge (a, b) is identical to the edge (b, a), i.e., they are not ordered pairs, i.e. (v0, v1) = (v1,v0).
- Figure shows a simple undirected graph with three vertices and three edges. Each vertex has degree two, so this is also a regular graph.

- V=set of 6 people: John, Mary, Joe, Helen, Tom, and Paul, where the first 4 are siblings, and the last two are siblings
- E ={(x,y) | x and y are siblings}



if (x,y) is an edge: we say that x is *adjacent to* y, & y adjacent to x. We also say that x and y are *neighbors*

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

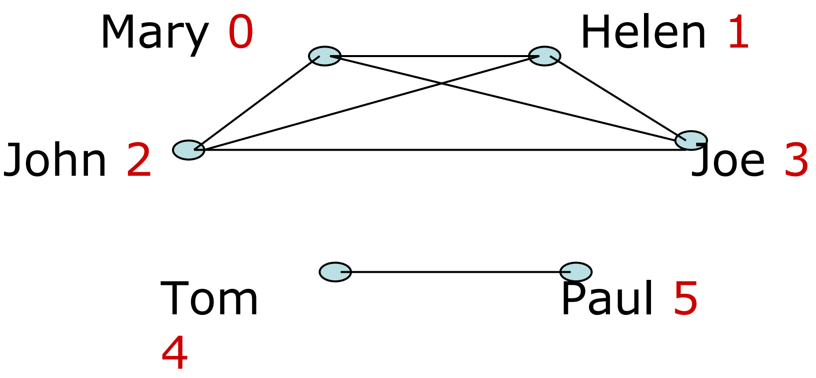
Representations of Undirected Graphs

- The same two representations for directed graphs can be used for undirected graphs
- **Adjacency matrix A:**
  - A[i][j]=1 if (i,j) is an edge; 0 otherwise
- **Adjacency Lists:**
  - L[i] is the linked list containing all the neighbors of i

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

Example of Representations:



Linked Lists representation:

L[0]: 1, 2, 3

L[1]: 0, 2, 3

L[2]: 0, 1, 3

L[3]: 0, 1, 2

L[4]: 5

L[5]: 4

Adjacency Matrix representation:

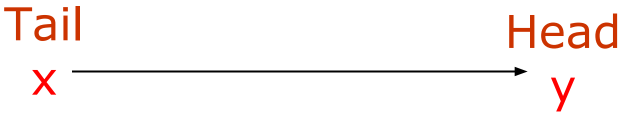
$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

Directed graph:

- A **directed graph** or **digraph** is one in which each edge is assigned a direction, or in other words, each edge is identified with an ordered pair of vertices such that <v0, v1> != <v1,v0>. The set of ordered pairs of vertices are called **arcs**, **directed edges**, or **arrows**.
- An arc *a* = (*x*, *y*) is considered to be directed **from** *x* **to** *y*; *y* is called the **head** (or target) and *x* is called the **tail** (or source) of the arc; *y* is said to be a **direct successor** of *x*, and *x* is said to be a **direct predecessor** of *y*.
- Figure shows a directed graph with three vertices and three edges.



Tail
x

Head
y

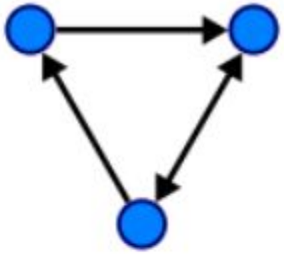


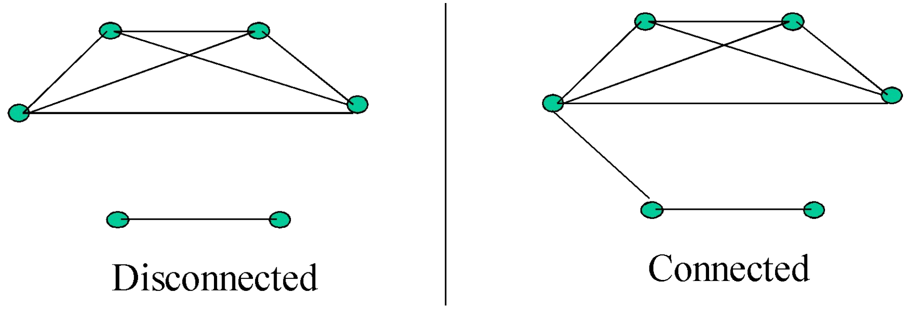
Figure: A directed graph

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

Connected graph:

- An undirected graph is said to be connected if there is a path between every pair of nodes. Otherwise, the graph is disconnected.
- A graph is called connected if every pair of distinct vertices in the graph is connected; otherwise, it is called disconnected.
- A directed graph is called weakly connected if replacing all of its directed edges with undirected edges produces a connected (undirected) graph. It is strongly connected or strong if it contains a directed path from u to v and a directed path from v to u for every pair of vertices u, v.



Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

Mixed graph:

- A mixed graph *G* is a graph in which some edges may be directed and some may be undirected. It is written as an ordered triple $G = (V, E, A)$ with V, E, and A defined as above.
- Figure shows a mixed graph with three vertices and three edges.

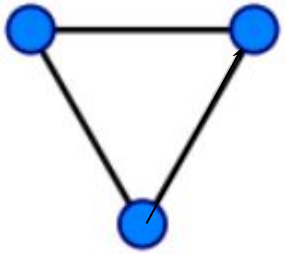


Figure: A mixed graph

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

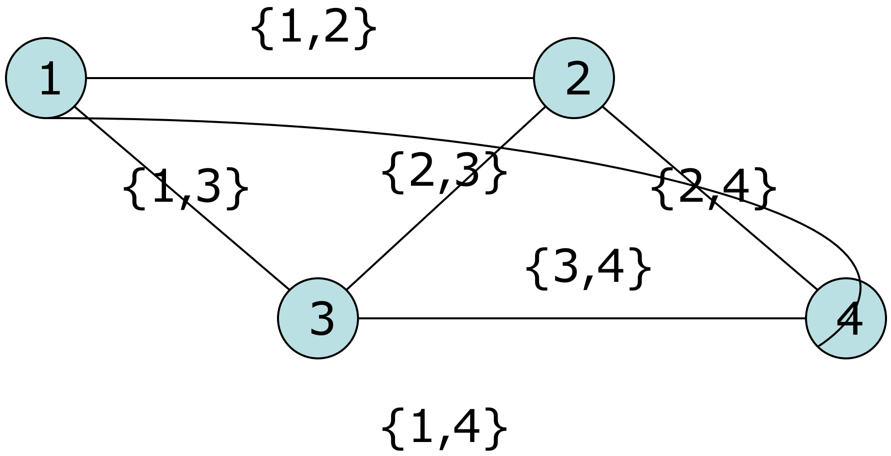
Simple graph:

- A simple graph, as opposed to a multigraph, is an undirected graph that has no loops and no more than one edge between any two different vertices.
- In a simple graph the edges of the graph form a set (rather than a multiset) and each edge is a distinct pair of vertices. In a simple graph with n vertices every vertex has a degree that is less than n (the converse, however, is not true — there exist non-simple graphs with n vertices in which every vertex has a degree smaller than n).

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

- Figure shows a simple graph with four vertices and six edges.



EG: Suppose a local computer network

- Is bidirectional (undirected)
- Has no loops (no "self-communication")
- Has unique connections between computers

Sensible to represent as follows:

- Vertices are labeled to associate with particular computers
- Each edge can be viewed as the set of its two endpoints

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

Multi graph:

- A **loop** is an edge (directed or undirected) which starts and ends on the same vertex; these may be permitted or not permitted according to the application. In this context, an edge with two different ends is called a **link**.
- The term "multigraph" is generally understood to mean that multiple edges (and sometimes loops) are allowed. Where graphs are defined so as to allow loops and multiple edges, a multigraph is often defined to mean a graph without loops, however, where graphs are defined so as to disallow loops and multiple edges, the term is often defined to mean a "graph" which can have both multiple edges and loops, although many use the term "pseudograph" for this meaning.

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

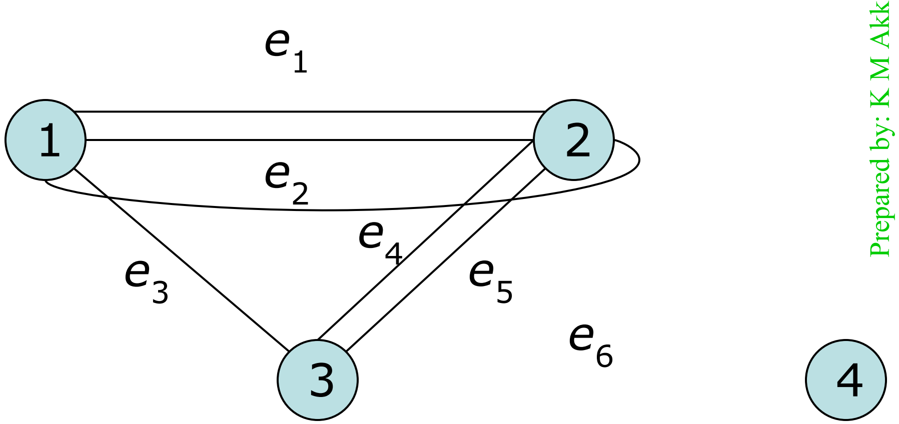
Consider the case in a local computer network again.

If computers are connected via internet instead of directly, there may be several routes to choose from for each connection. Depending on traffic, one route could be better than another.

Makes sense to allow multiple edges, but still no self-loops:

Edge-labels distinguish between edges sharing same endpoints. Labeling can be thought of as function:

e_1 □ {1,2}, e_2 □ {1,2}, e_3 □ {1,3}, e_4 □ {2,3}, e_5 □ {2,3}, e_6 □ {1,2}



Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

Weighted graph:

- A graph is a weighted graph if a number (weight) is assigned to each edge. Such weights might represent, for example, costs, weight, lengths or capacities, etc. depending on the problem at hand. Some authors call such a graph a network.

Regular graph:

- A regular graph is a graph where each vertex has the same number of neighbors, i.e., every vertex has the same degree or valency. A regular graph with vertices of degree k is called a k-regular graph or regular graph of degree k.

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

Complete graph:

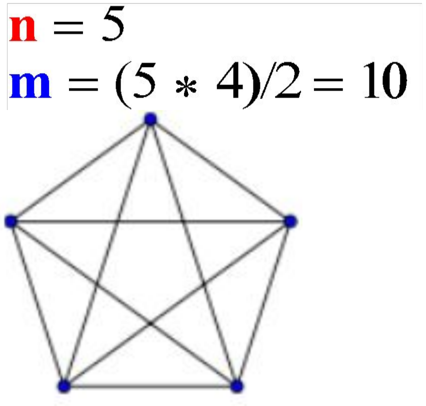
- Complete graphs have the feature that each pair of vertices is joined by an edge. That is, the graph contains all possible edges. Figure below shows three complete graphs. Each vertex has an edge to every other vertex.

Let n = no. of vertices, and **m** = no. of edges.

How many total edges in a complete graph?

Each of the *n* vertices is incident to **n**-1 edges, however, we would have counted each edge twice! Therefore, intuitively, m = **n**(**n** -1)/2.

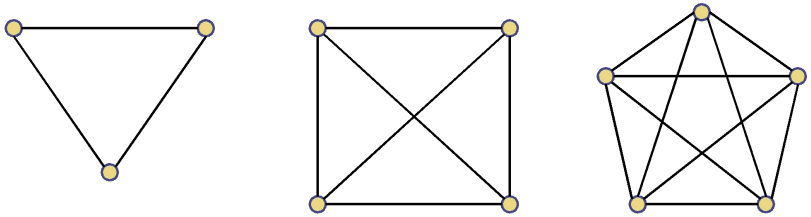
Therefore, if a graph is not complete, m < **n**(**n** -1)/2



$\mathbf{n} = 5$

$\mathbf{m} = (5 * 4)/2 = 10$

Figure: A complete graph



Types of Graph:

Trivial graph:

- The graph with only one vertex and no edges is called the **trivial graph**.

Edgeless graph:

- A graph with only vertices and no edges is known as an **edgeless graph**.

Null or Empty graph:

- The graph with no vertices and no edges is sometimes called the **null graph** or **empty graph**.

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Types of Graph:

Bipartite graph (Bigraph):

- A bipartite graph is a graph whose vertices can be divided into two disjoint sets , U and V, so that no two vertices in U are adjacent and no two vertices in V are adjacent. That is, every edge connects a vertex in U to one in V.

The two sets U and V may be thought of as a coloring of the graph with two colors:
- if one colors all nodes in U blue, and all nodes in V green, each edge has endpoints of differing colors, as is required in the graph coloring problem. In contrast, such a coloring is impossible in the case of a non-bipartite graph, such as a triangle: after one node is colored blue and another green, the third vertex of the triangle is connected to vertices of both colors, preventing it from being assigned either color.

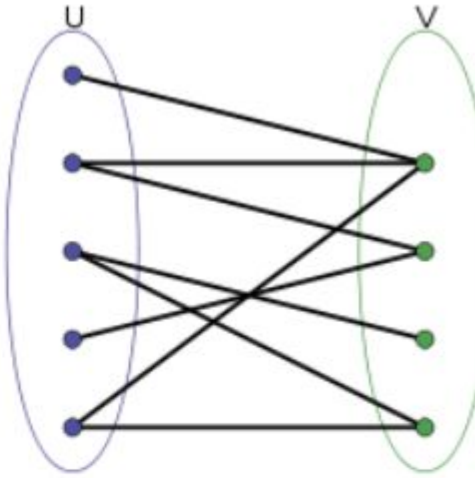


Figure: A bipartite graph

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Properties of a Graph:

Adjacent edges:

- Two edges of a graph are called **adjacent** (sometimes **coincident**) if they share a common vertex. Two arrows of a directed graph are called **consecutive** if the head of the first one is at the nock (notch end) of the second one.

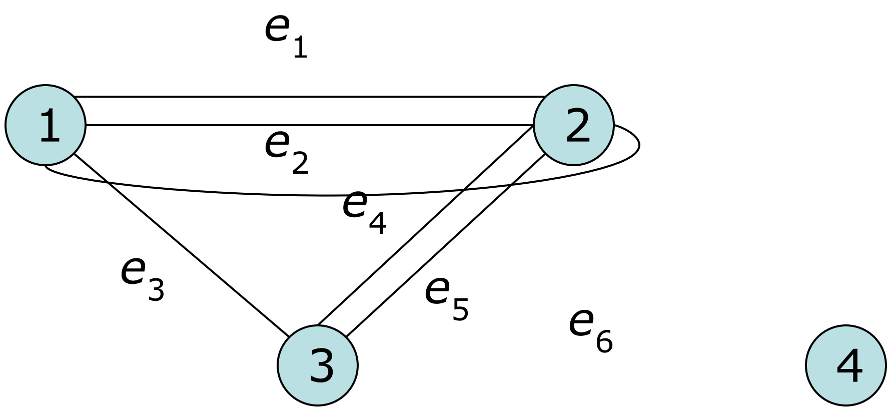
Adjacent nodes or neighbor or vertices:

- Two vertices are called **adjacent** if they share a common edge . In this case, the common edge is said to **join** the two vertices.
- A vertex is **incident** with an edge (and the edge is incident with the vertex) if it is the endpoint of the edge. Two vertices are called **consecutive** if they are at the notch and at the head of an arrow.

Prepared by: K M Akkas Ali, Associate Professor, IIT, JU

Properties of a Graph:

Q: In the figure below, which vertices are adjacent to 1? How about adjacent to 2, 3, and 4?



Ans: 1 is adjacent to 2 and 3

2 is adjacent to 1 and 3

3 is adjacent to 1 and 2

4 is not adjacent to any vertex

Q: Which edges are incident to 1? How about incident to 2, 3, and 4?

Ans:

e_1, e_2, e_3, e_6 are incident with 2

2 is incident with e_1, e_2, e_4, e_5, e_6

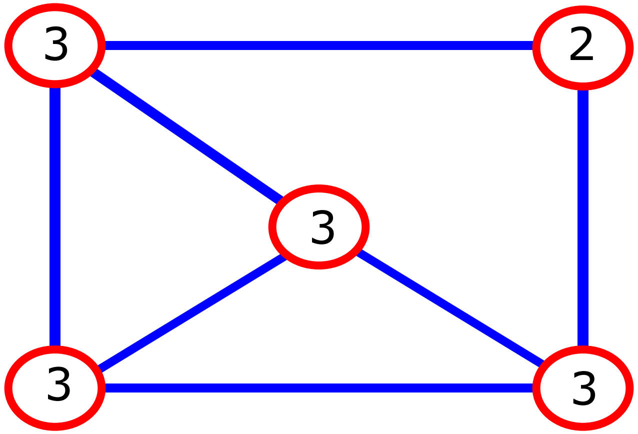
3 is incident with e_3, e_4, e_5

4 is not incident with any edge

Properties of a Graph:

Path:

- It is the sequence of vertices $v_1, v_2, . . . v_k$ such that consecutive vertices v_i and v_{i+1} are adjacent.

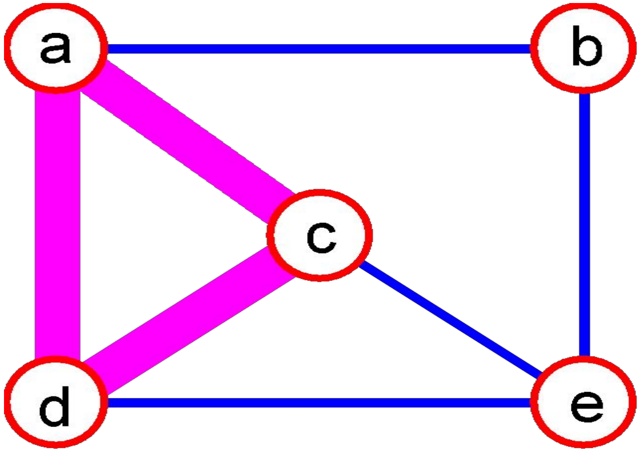


Simple path:

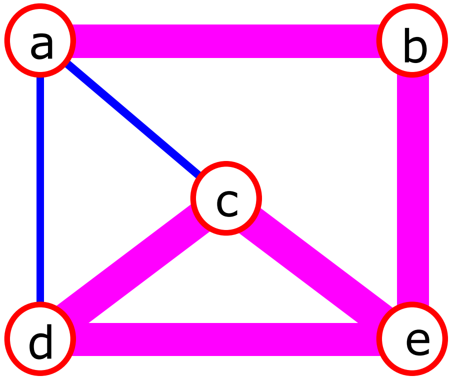
- It is a path with no repeated vertices.

Cycle:

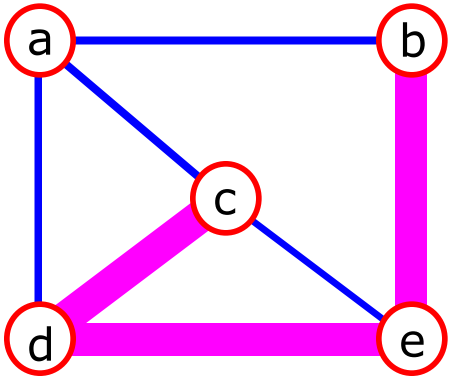
- It is a simple path, except that the last vertex is the same as the first vertex



Cycle: acda



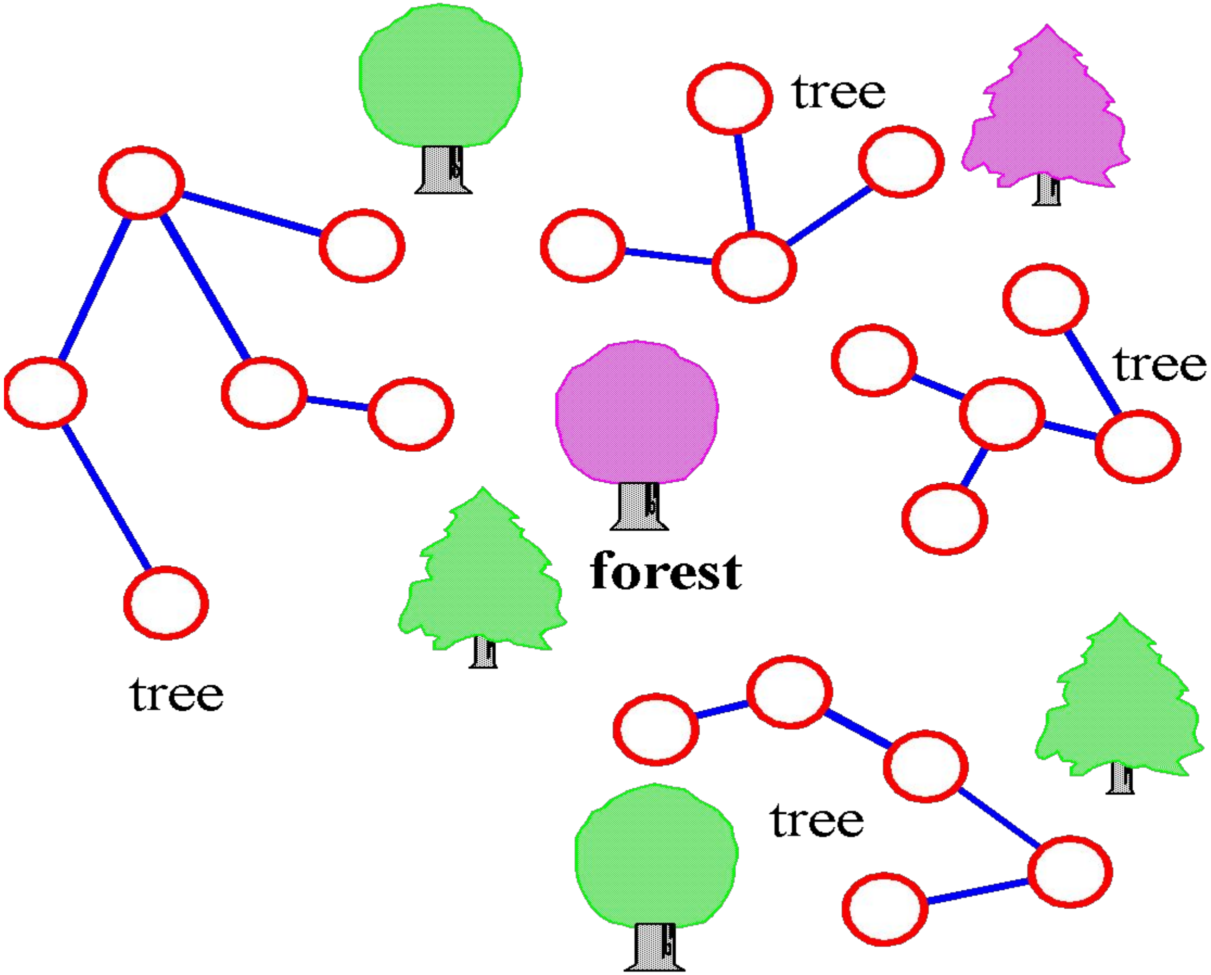
a b e d c



Simple path: b e d c

Properties of a Graph:

- Tree - connected graph without cycles
- Forest - collection of trees


tree
tree
forest
tree
tree