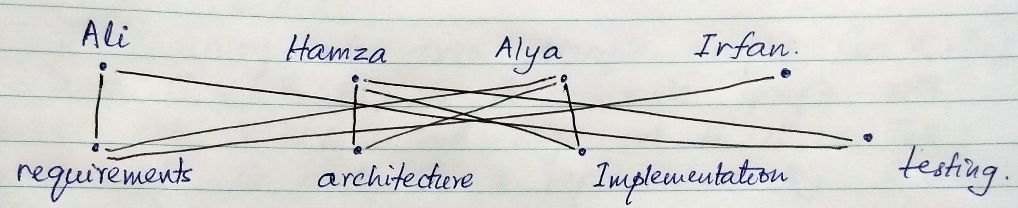
Bipartite graphs and matching.

Bipartite graphs are used to model applications that involve matching the elements of one set to the elements in another, for example:

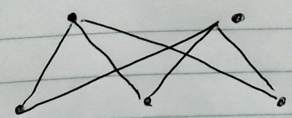
Example: (Job Assignments) Vertices represent the jobs and the employees, edges link employee with those jobs they have been ~~assigned~~ trained to do.
A common goal is to match jobs to employees so that the most jobs are done.

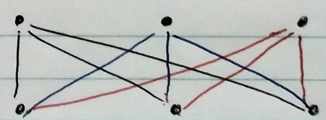


Complete bipartite graph

A complete bipartite graph $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets V_1 of size m and V_2 of size n such that there is an edge from every vertex in V_1 to every vertex in V_2.

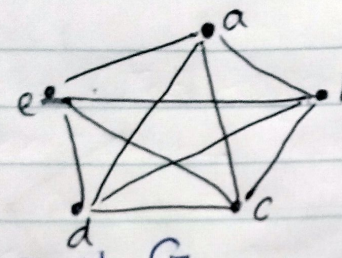
Examples:



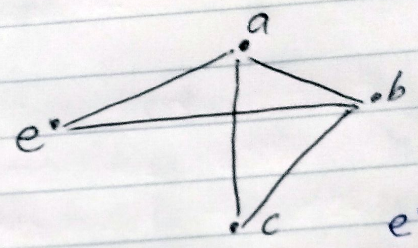


Subgraphs:

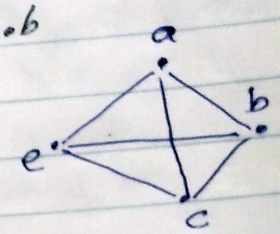
Definition: A subgraph of graph $G = (V, E)$ is a graph (W, F) where $W \subset V$ and $F \subset E$.
A subgraph H of G is proper subgraph if $H \neq G$.



graph G



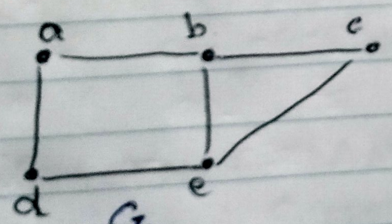
graph H



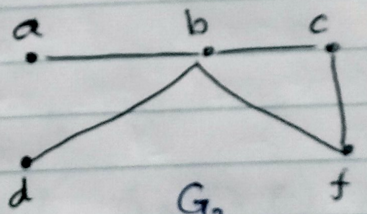
Union of Graphs:

The union of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. The union of G_1 and G_2 is represented by $G_1 \cup G_2$.

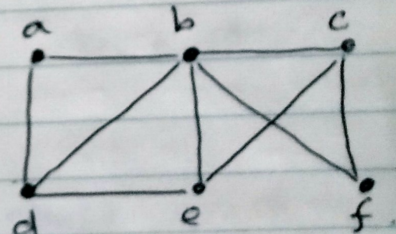
Example:



G_1



G_2

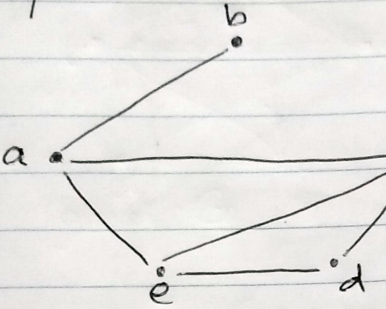


$G_1 \cup G_2$.

Representation of Graphs.

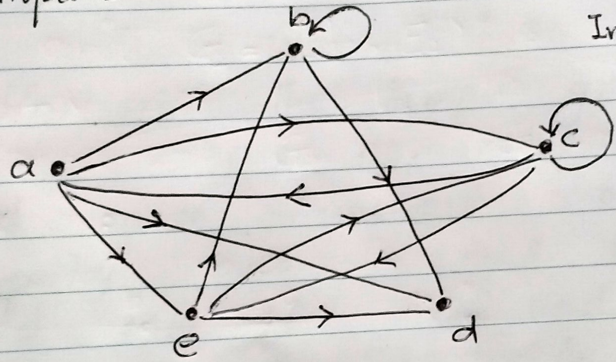
Definition: An adjacency list can be used to represent a graph with no multiple edges by specifying the vertices that are adjacent to each vertex of the graph.

Example 1



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

Example 2



Initial Vertex	Terminal vertex.
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d.

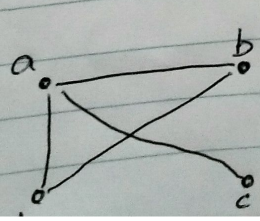
ADJACENCY MATRICES.

Definition: Suppose that $G = (V, E)$ is a simple graph wher $|V| = n$. arbitrary list of vertices of G as $v_1, v_2 \ldots v_n$.
The adjacency matrix A_G of G, with respect to the listing of vertices, is $n \times n$ zero-one matrix with 1 as its $(i,j)^{th}$ entry when v_i and v_j are adjacent, and 0 as its $(i,j)^{th}$ entry when they are not adjacent.

- In other words $A_G = [a_{ij}]$ where

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G. \\ 0 & \text{otherwise.} \end{cases}$$

Example:



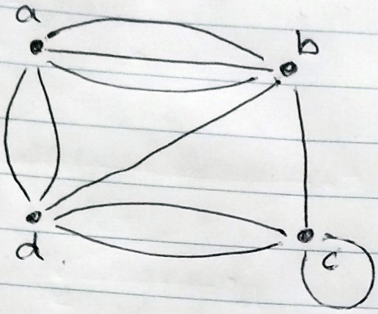
$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The ordering of vertices is a, b, c, d.

(Adjacency matrices ...contd)

* Adjacency matrices can also be used to represent graphs with loops and multiple edges.

Example:

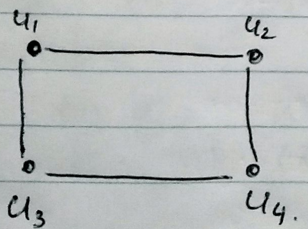


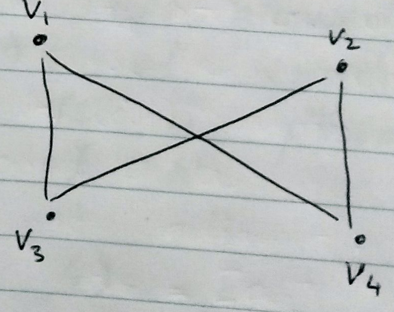
$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

Graph Isomorphism.

Definition: The simple graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are isomorphic if there is one-to-one and on-to function f from V_1 to V_2 with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1. Such a function f is called isomorphism. Two simple graphs that are not isomorphic are called nonisomorphic.

Example: Are the two graphs isomorphic





$u_1 \to V_1$.

$u_2 \to V_2$.

$u_3 \to V_3$.

$u_4 \to V_4$

Connectivity in the graphs, paths.

Informal Definition: A path is a sequence of edges that begins at a vertex of a graph and travels from vertex to vertex along edges of the graph. As the path travels along its edges, it visits the vertices along this path

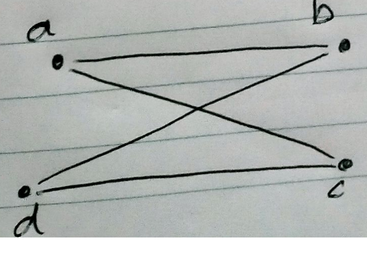
Applications: Numerous problems can be formed/ modeled with paths formed by traveling along edges of graph such as:

- determining whether a message can be sent between two computers.
- efficiently planning routes for mail/message delivery

We can use the adjacency matrix of a graph to find the number of the different paths between two vertices in the graph.

THEOREM: Let G be a graph with adjacency matrix A with respect to the ordering $v_1 \dots v_n$ of vertices (with directed or undirected edges, multiple edges and loops allowed). The number of different paths of length r from v_i to v_j, where $r>0$ is a positive integer, equals the $(i,j)^{th}$ entry of A^r.

Example: (paths of length 4).



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

ordering of vertices is a, b, c, d.