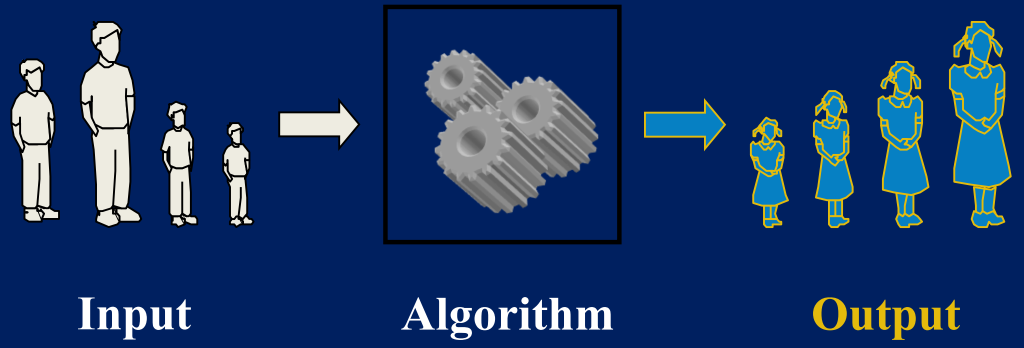
GLA UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)
Accredited with A Grade by NAAC

DESIGN & ANALYSIS OF ALGORITHM (BCSC0012)

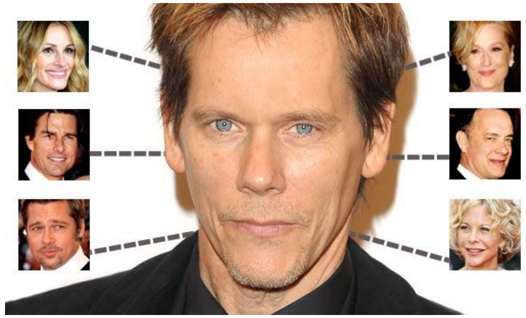
Chapter 11: Graph

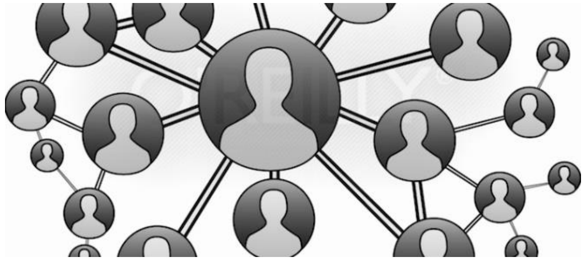


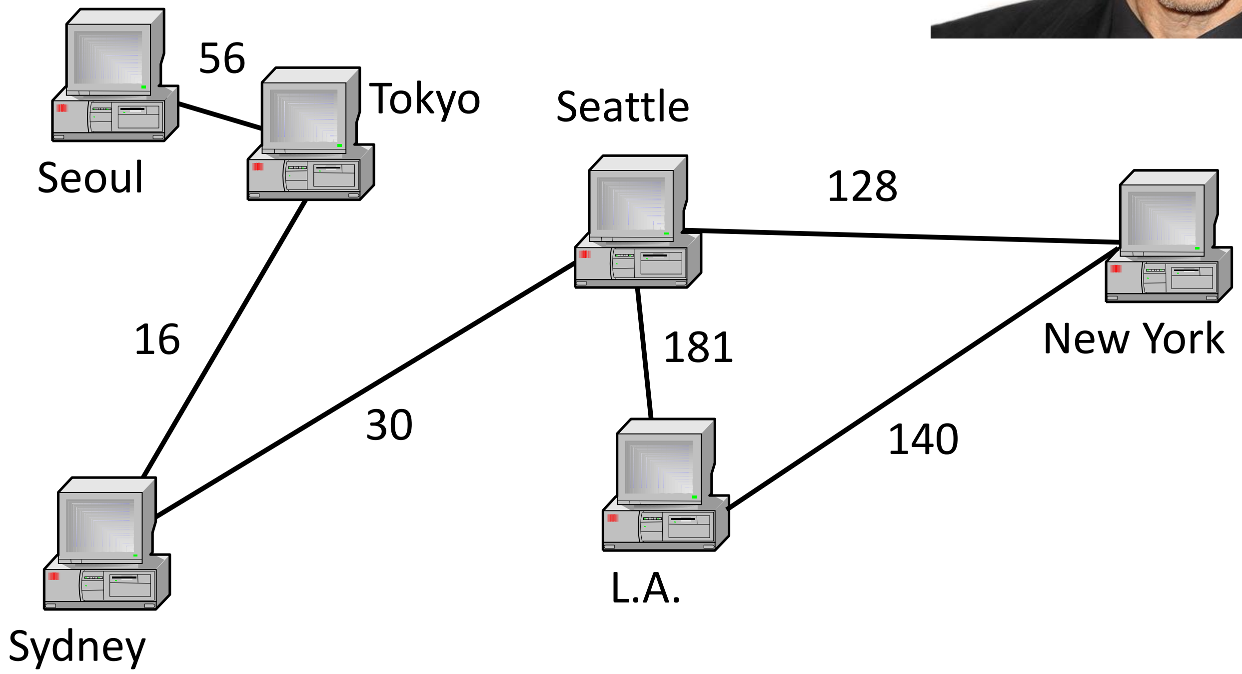
Prof. Anand Singh Jalal

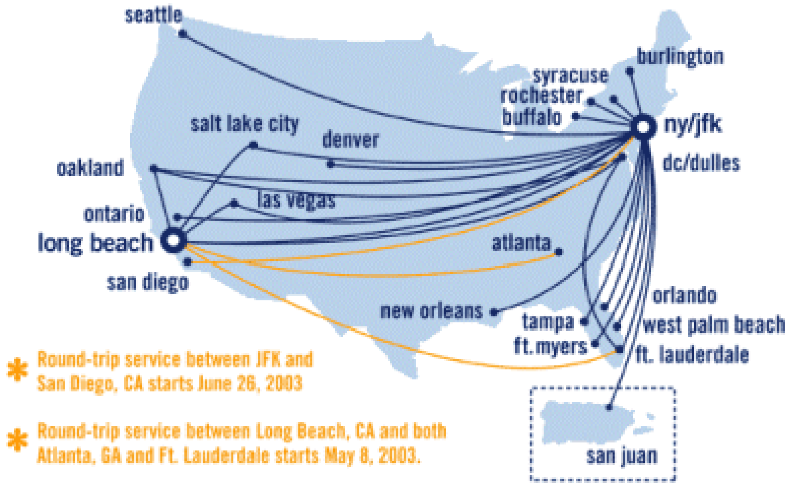
Department of Computer Engineering & Applications

What is a Graph?



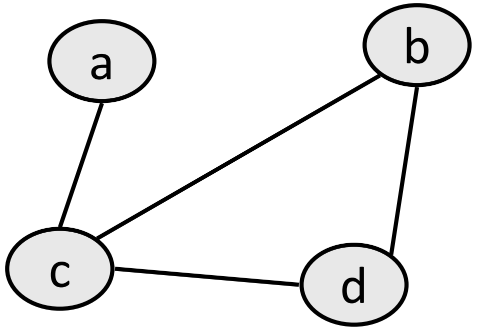






What is a Graph?

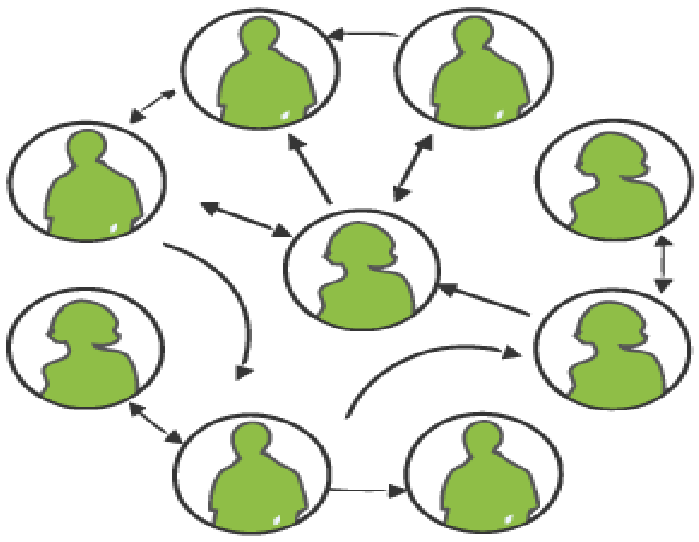
- **Graph**: A data structure containing:
  - a set of **vertices** *V*, *(sometimes called nodes)*
  - a set of **edges** *E*, where an edge represents a connection between 2 vertices.
    - Graph $G = (V, E)$
    - an edge is a pair (v, w) where v, w are in V
- the graph at right:
  - $V = \{a, b, c, d\}$
  - $E = \{(a, c), (b, c), (b, d), (c, d)\}$



- **Degree**: number of edges touching a given vertex.
  - at right: a=1, b=2, c=3, d=2

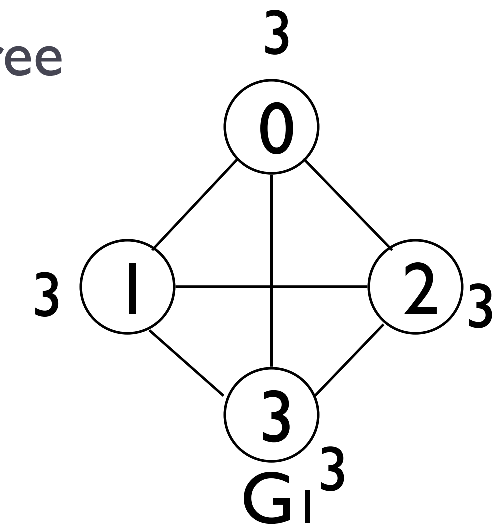
Graph Examples

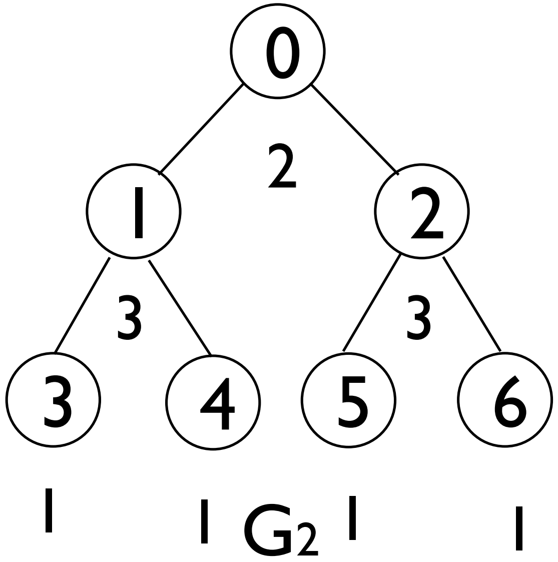
- For each, what are the vertices and what are the edges?
  - Web pages with links
  - Methods in a program that call each other
  - Road maps (e.g., Google maps)
  - Airline routes
  - Facebook friends
  - Course pre-requisites
  - Family trees
  - Paths through a maze



Degree of Graph

Degree

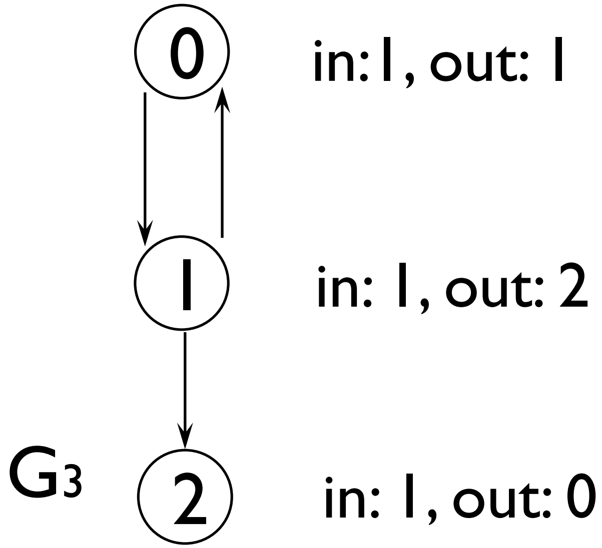




directed graph

in-degree

out-degree



Degree of a Vertex:
How many edges are passing from a vertex

In an undirected graph, in total degree every edge counted twice.

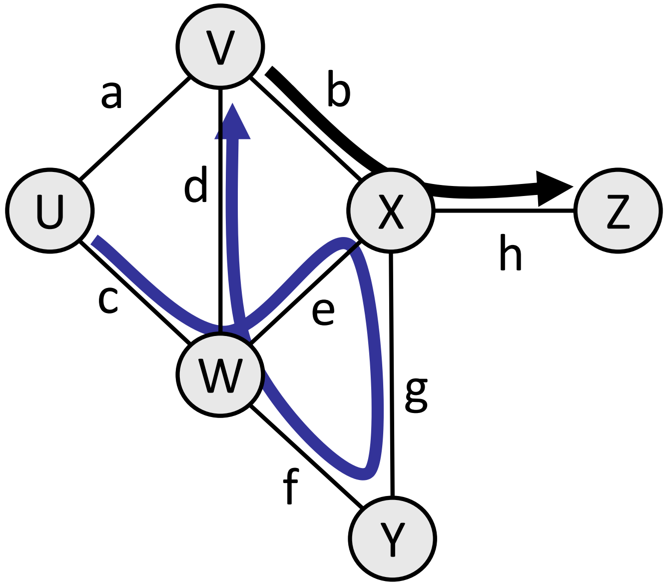
$$E_{\max} = \frac{n(n-1)}{2}$$

Path

- **Path**: A path from vertex *a* to *b* is a sequence of edges that can be followed starting from *a* to reach *b*.
  - can be represented as vertices visited, or edges taken
  - example, one path from *V* to *Z*: {b, h} or {V, X, Z}
  - What are two paths from U to Y?

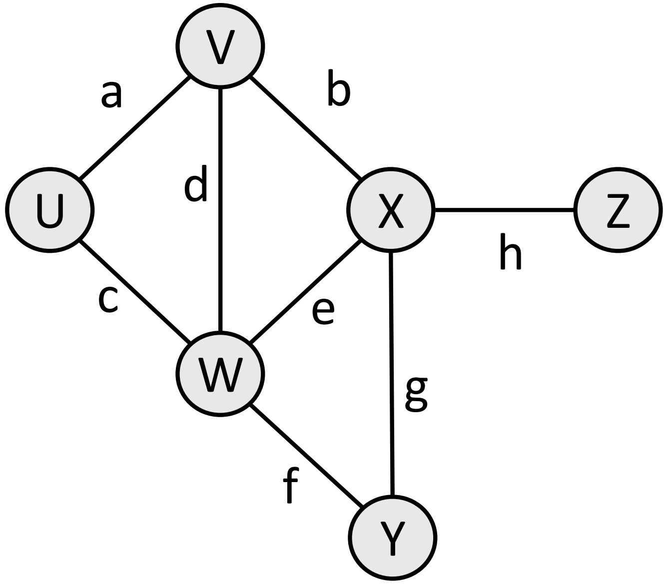
- **Path length**: Number of edges contained in the path.

- **Neighbor** or **Adjacent**: Two vertices connected directly by an edge.
  - example: V and X



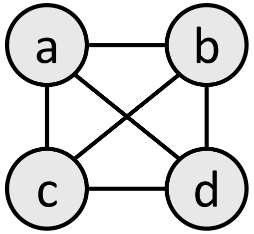
Reachability, Connectedness

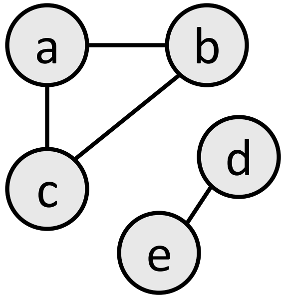
- **Reachable**: Vertex *a* is *reachable* from *b* if a path exists from *a* to *b*.



- **Connected**: A graph is *connected* if every vertex is reachable from any other.
  - Is the graph at top right connected?

- **Strongly connected**: When every vertex has an edge to every other vertex.





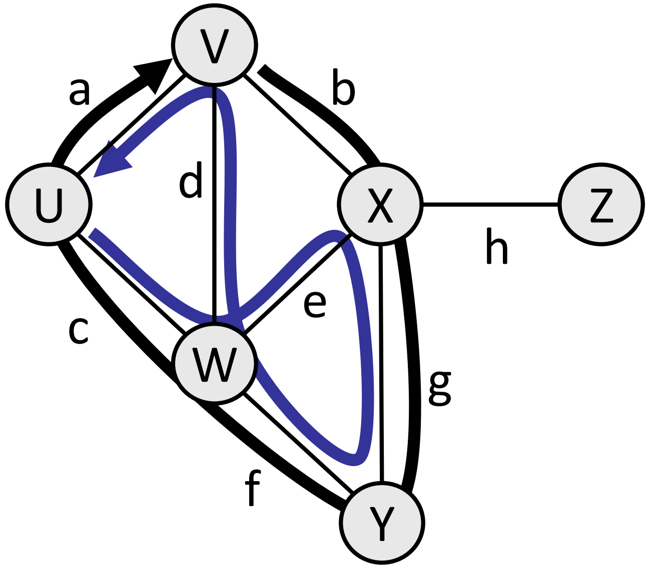
A complete graph G of n nodes has **n*(n-1)/2** edges.

Loops and Cycles

- **Cycle**: A path that begins and ends at the same node.
  - example: {b, g, f, c, a} or {V, X, Y, W, U, V}.
  - example: {c, d, a} or {U, W, V, U}.

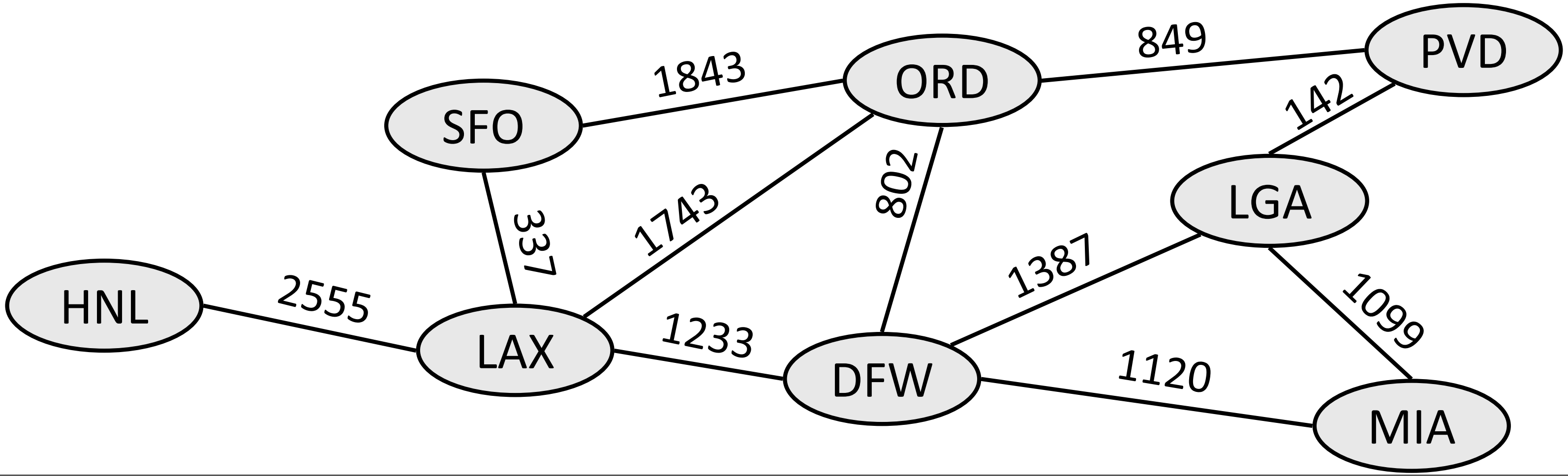
- **Acyclic graph**: One that does not contain any cycles.

- **Loop**: An edge directly from a node to itself.
  - Many graphs don't allow loops.



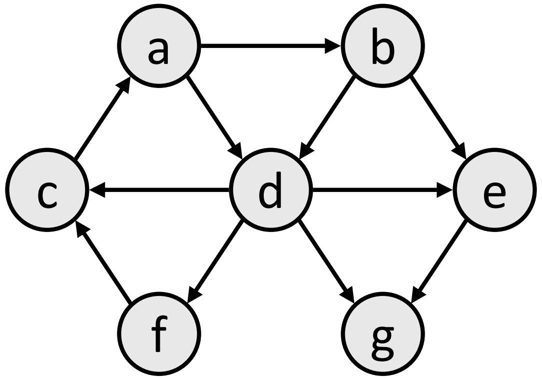
Weighted Graphs

- **Weight**: Cost associated with a given edge.
  - Some graphs have weighted edges, and some are unweighted.
  - Edges in an unweighted graph can be thought of as having equal weight (e.g. all 0, or all 1, etc.)
  - Most graphs do not allow negative weights.
- *example*: graph of airline flights, weighted by miles between cities:



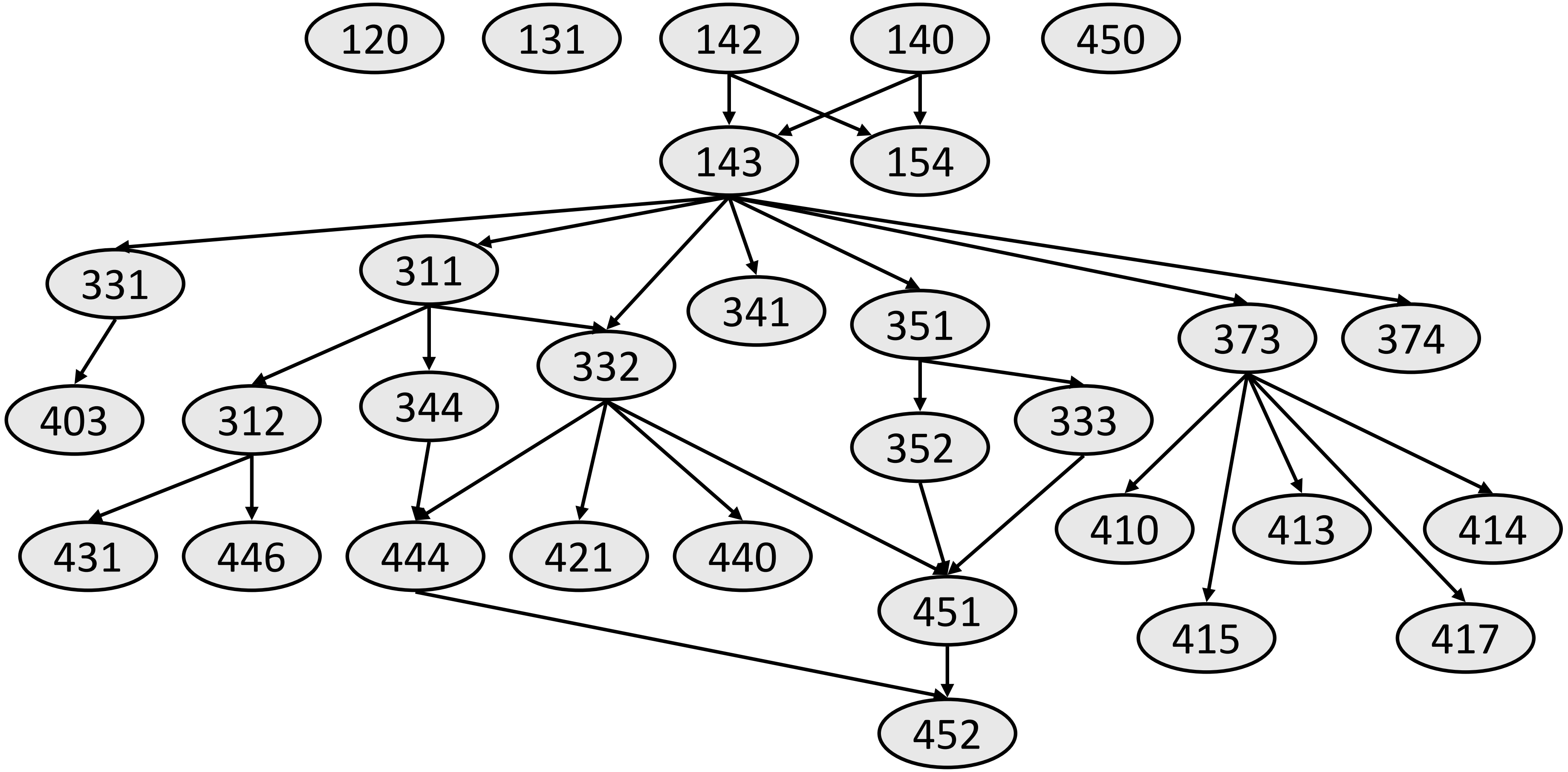
Directed Graphs

- **Directed graph** ("digraph"): One where edges are *one-way* connections between vertices.
  - If graph is directed, a vertex has a separate in/out degree.
  - A digraph can be weighted or unweighted.



Directed Graphs: Example

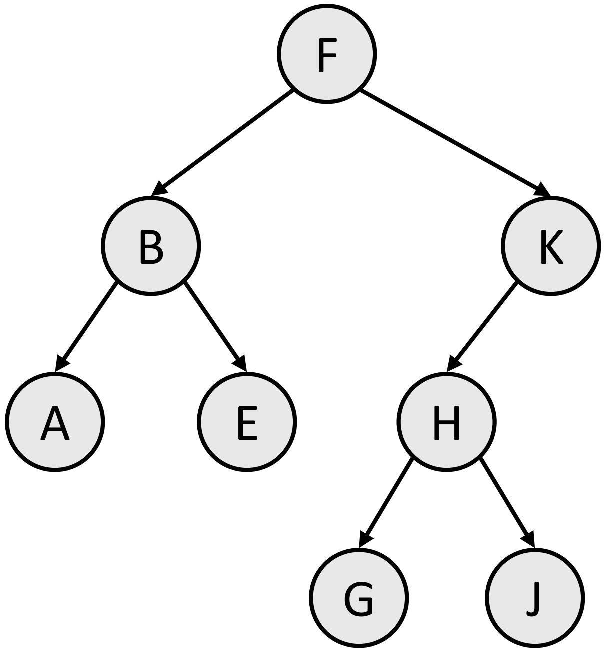
- Vertices = UW CSE courses (incomplete list)
- Edge (a, b) = *a* is a prerequisite for *b*

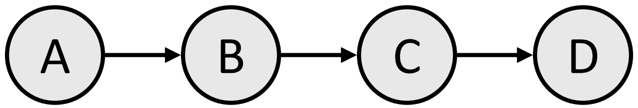


Linked Lists, Trees, Graphs

- A ***binary tree*** is a graph with some restrictions:
  - The tree is an unweighted, directed, acyclic graph (**DAG**).
  - Each node's in-degree is at most 1, and out-degree is at most 2.
  - There is exactly one path from the root to every node.

- A ***linked list*** is also a graph:
  - Unweighted DAG.
  - In/out degree of at most 1 for all nodes.





Graph Representation of Graphs

There are two ways to represent graphs:

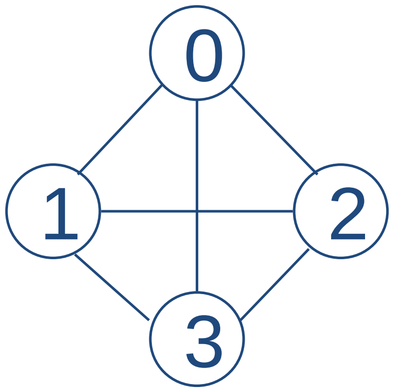
- Sequential Representation
  - Adjacency matrix
  - Path Matrix

- Linked List Representation

Graph Representation of Graphs ...

- Sequential Representation
  - **Adjacency matrix:** Adjacency matrix representation of a graph G consists of V x V matrix A=(a_{ij}) such that

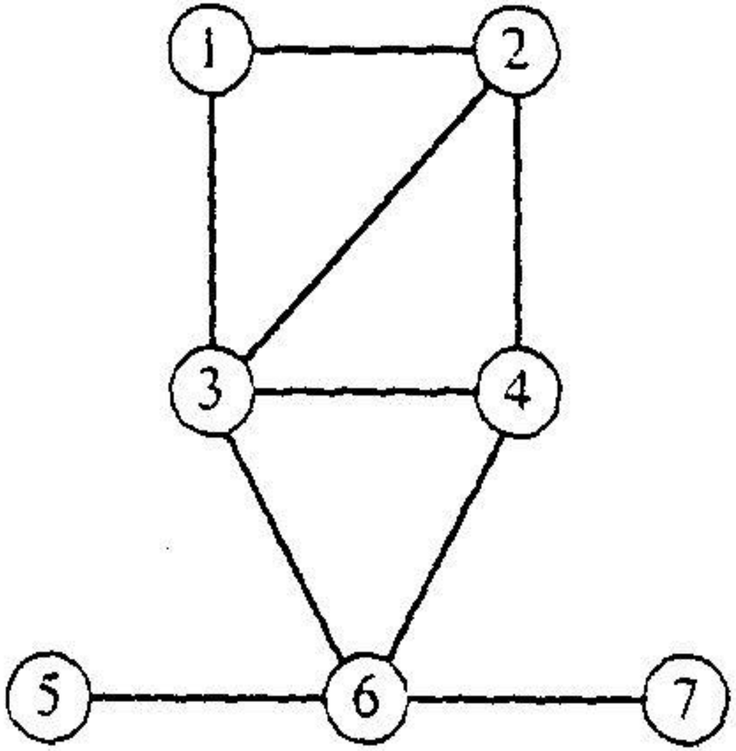
$$aij = \begin{cases} 1 & if(i,\ j) \in E \\ 0 & otherwise \end{cases}$$



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

Adjacency matrix:

Graph Representation of Graphs ...



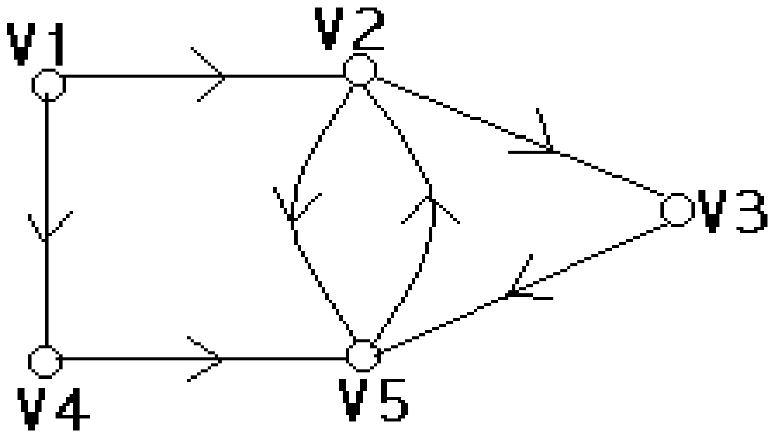
(a) An undirected graph

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

(b) Its adjacency matrix

Graph Representation of Graphs ...

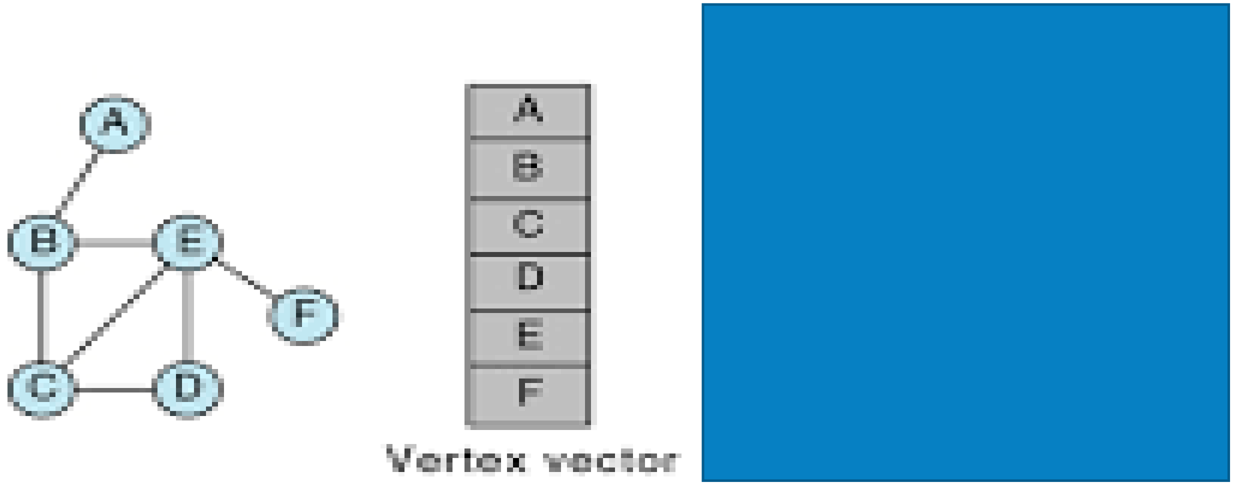
Example: Given a directed graph G as:



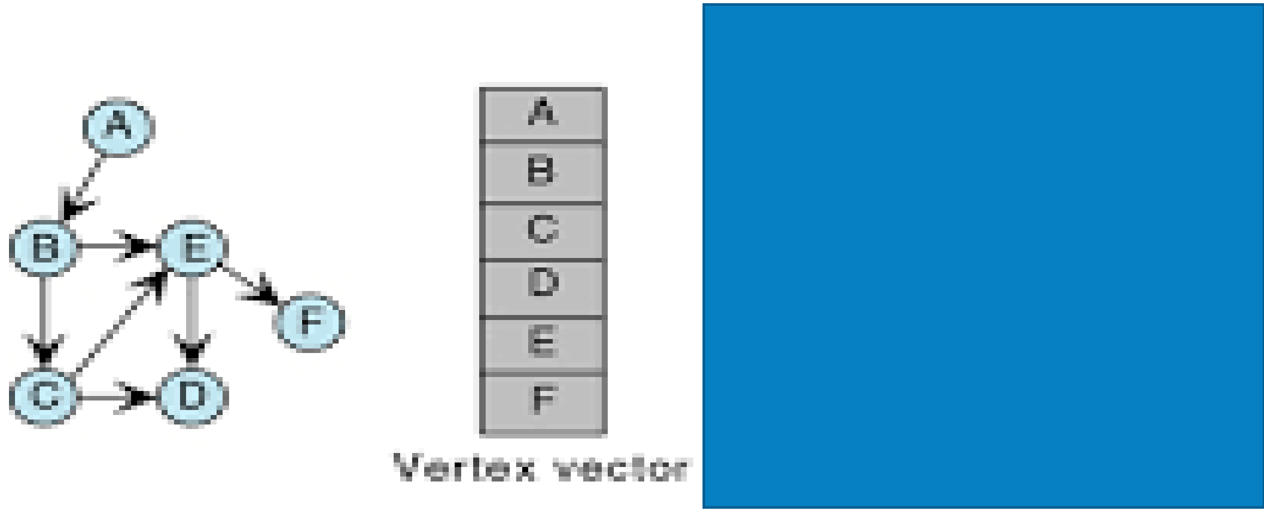
	V1	V2	V3	V4	V5
V1	0	1	0	1	0
V2	0	0	1	0	1
V3	0	0	0	0	1
V4	0	0	0	0	1
V5	0	1	0	0	0

Adjacency matrix

Graph Representation of Graphs …



(a) Adjacency matrix for non-directed graph



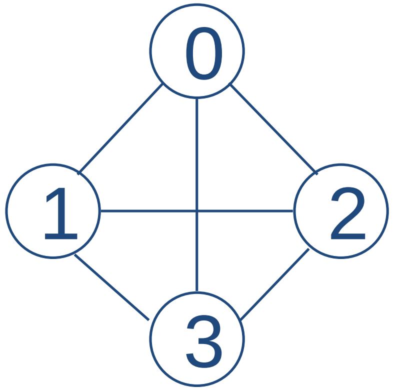
(a) Adjacency matrix for directed graph

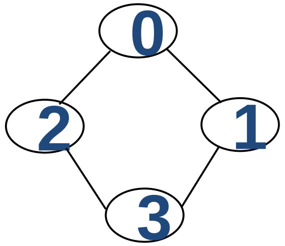
Graph Representation of Graphs ...

Sequential Representation

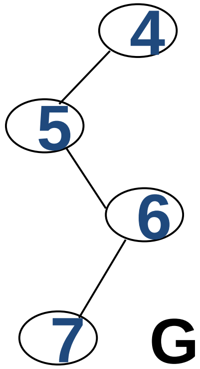
- Linked List Representation

G1

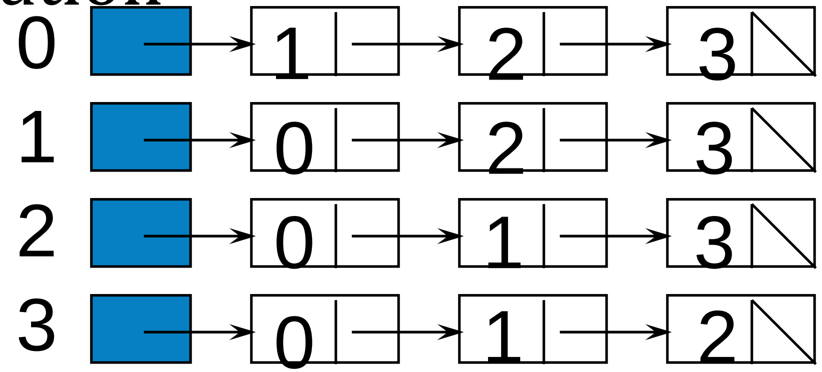




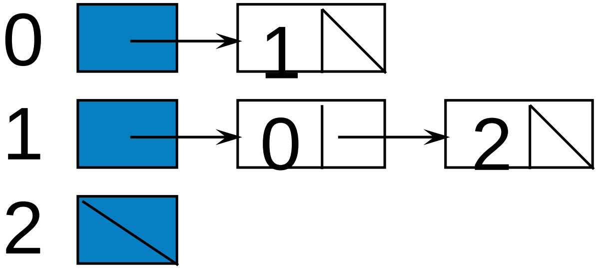
G2



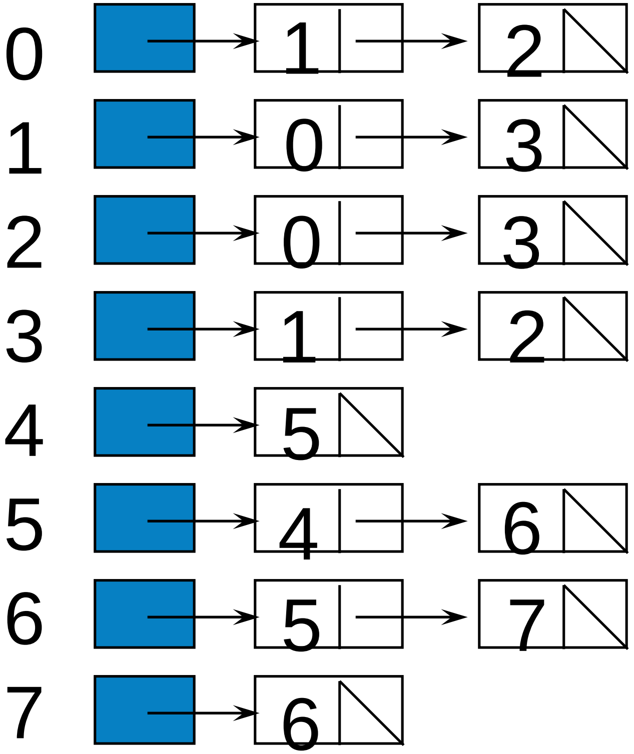
G3



G_1



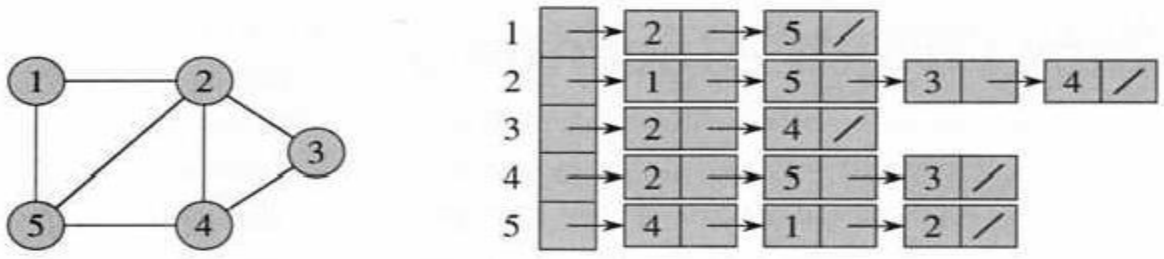
G_2



G_4

Graph Representation of Graphs ...

Example 1

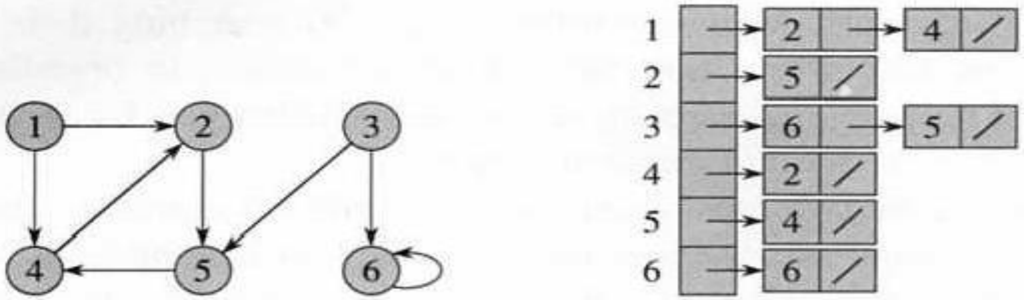


(a) (b)

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

(c)

Example 2



(a) (b)

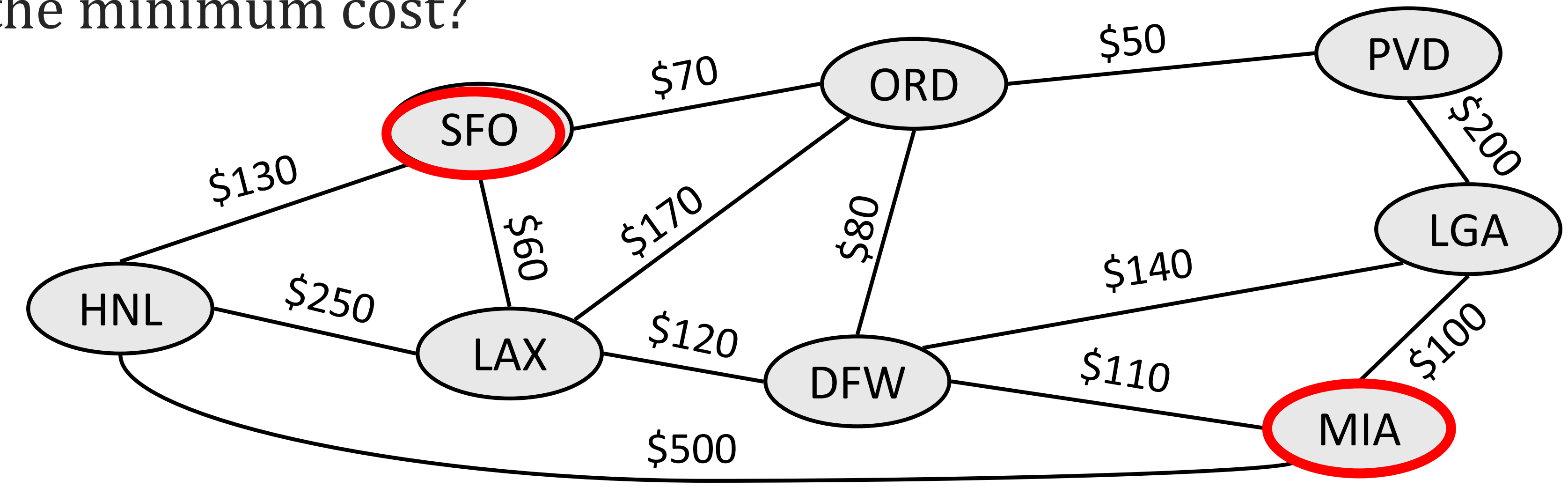
	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

(c)

Searching for Paths

- Searching for a path from one vertex to another:
  - Sometimes, we just want *any* path (or want to know there *is* a path).
  - Sometimes, we want to minimize path *length* (# of edges).
  - Sometimes, we want to minimize path *cost* (sum of edge weights).

- What is the shortest path from MIA to SFO?
  Which path has the minimum cost?



Any Questions ?

Dr. Anand Singh Jalal
Professor
Email: asjalal@gla.ac.in