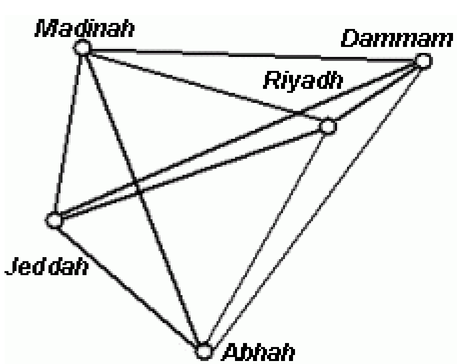
Introduction to Graphs

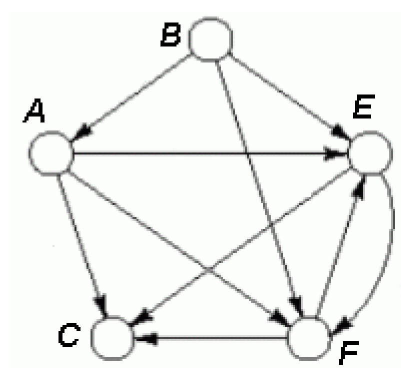
What is a Graph?

- Graphs are Generalization of Trees.
- A simple graph G = (V, E) consists of a non-empty set V, whose members are called the vertices of G, and a set E of pairs of distinct vertices from V, called the edges of G.

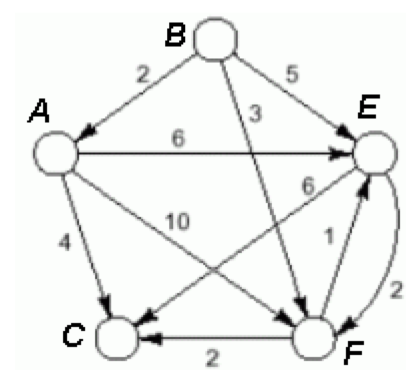
Undirected



Directed (Digraph)



Weighted



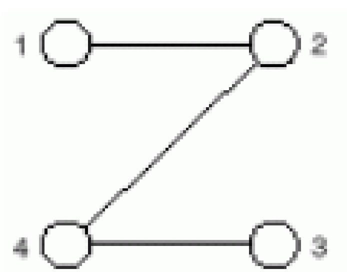
Some Example Applications of Graph

- Finding the least congested route between two phones, given connections between switching stations.
- Determining if there is a way to get from one page to another, just by following links.
- Finding the shortest path from one city to another.
- As a traveling sales-person, finding the cheapest path that passes through all the cities that the sales person must visit.
- Determining an ordering of courses so that prerequisite courses are always taken first.

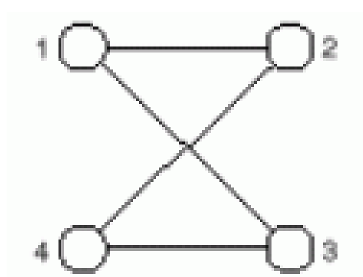
Graphs Terminologies

- Adjacent Vertices: there is a connecting edge.
- A Path: A sequence of adjacent vertices.
- A Cycle: A path in which the last and first vertices are adjacent.
- Connected graph: There is a path from any vertex to every other vertex.

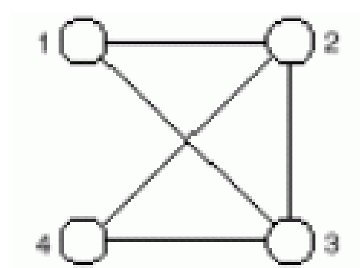
Path



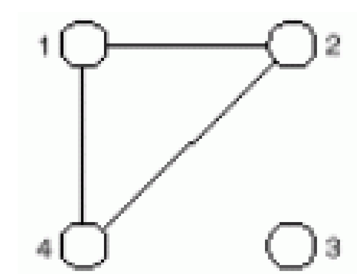
Cycle



Connected



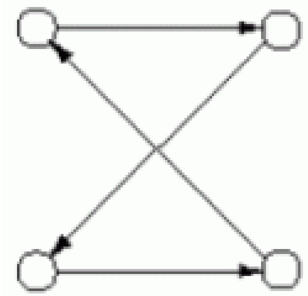
Disconnected



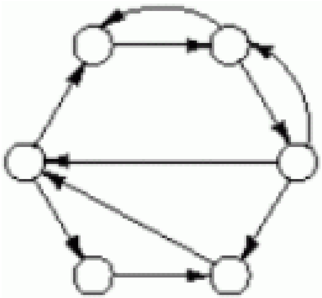
More Graph Terminologies

- Path and cycles in a digraph: must move in the direction specified by the arrow.
- Connectedness in a digraph: strong and weak.
- Strongly Connected: If connected as a digraph - following the arrows.
- Weakly connected: If the underlying undirected graph is connected (i.e. ignoring the arrows).

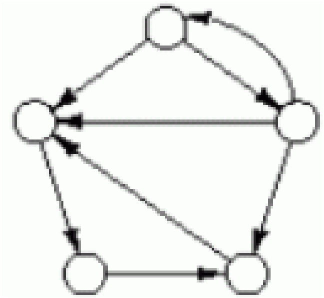
Directed Cycle



Strongly Connected



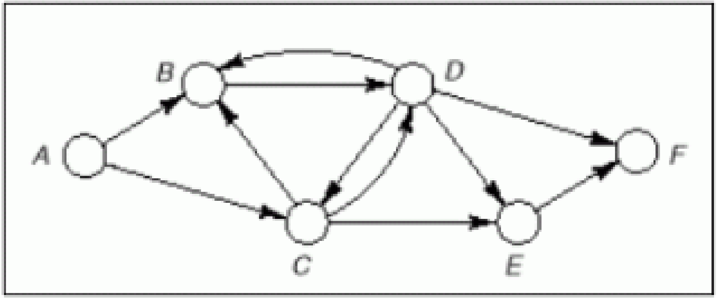
Weakly Connected



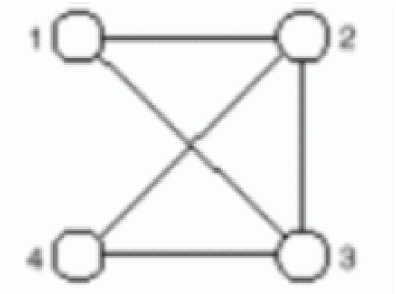
Further Graph Terminologies

- Emanate: an edge $e = (v, w)$ is said to emanate from v.
  - $A(v)$ denotes the set of all edges emanating from v.
- Incident: an edge $e = (v, w)$ is said to be incident to w.
  - $I(w)$ denote the set of all edges incident to w.
- Out-degree: number of edges emanating from v -- $|A(v)|$
- In-degree: number of edges incident to w -- $|I(w)|$.

Directed Graph



Undirected Graph



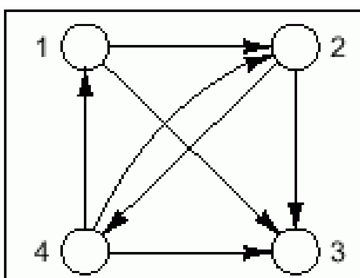
Graph Representations

- For vertices:
  - an array or a linked list can be used

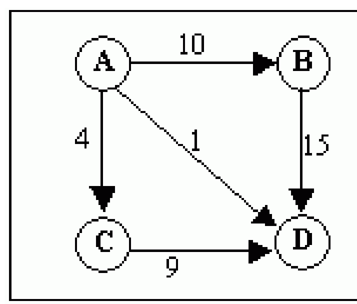
- For edges:
  - Adjacency Matrix (Two-dimensional array)
  - Adjacency List (One-dimensional array of linked lists)
  - Linked List (one list only)

Adjacency Matrix Representation

- Adjacency Matrix uses a 2-D array of dimension |V|x|V| for edges. (For vertices, a 1-D array is used)
- The presence or absence of an edge, (v, w) is indicated by the entry in row v, column w of the matrix.
- For an unweighted graph, boolean values could be used.
- For a weighted graph, the actual weights are used.



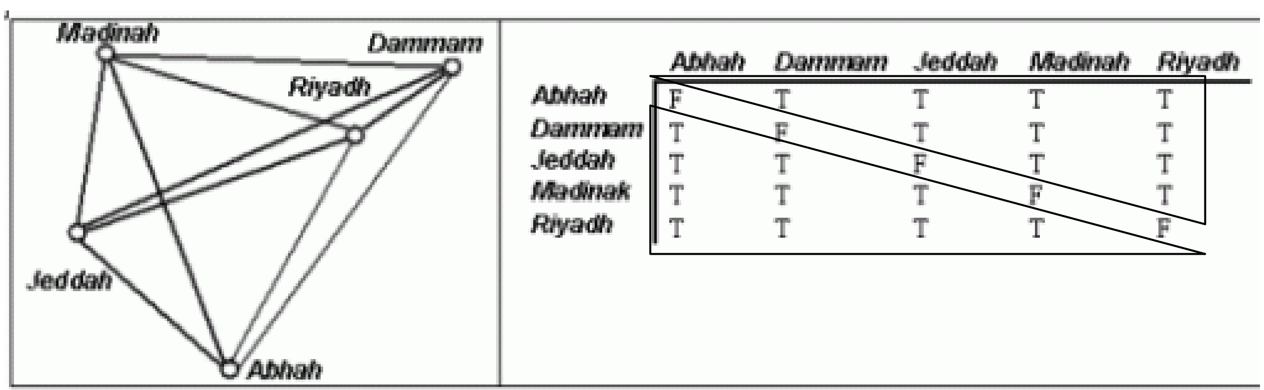
	1	2	3	4
1	F	T	T	F
2	F	F	T	T
3	F	F	F	F
4	T	T	T	F



	A	B	C	D
A	∞	10	4	1
B	∞	∞	∞	15
C	∞	∞	∞	9
D	∞	∞	∞	∞

Notes on Adjacency Matrix

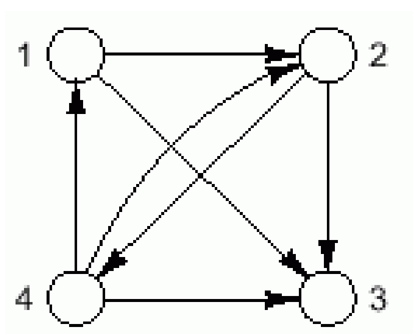
- For undirected graph, the adjacency matrix is always symmetric.
- In a Simple Graph, all diagonal elements are zero (i.e. no edge from a vertex to itself).
- The space requirement of adjacency matrix is $O(n^2)$ - most of it wasted for a graph with few edges.
- However, entries in the matrix can be accessed directly.

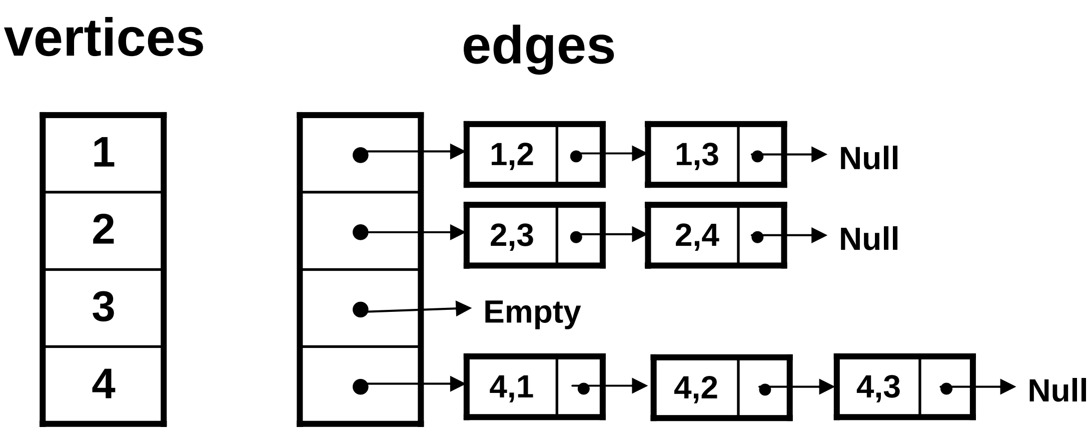


Adjacency List Representation

- This involves representing the set of vertices adjacent to each vertex as a list. Thus, generating a set of lists.

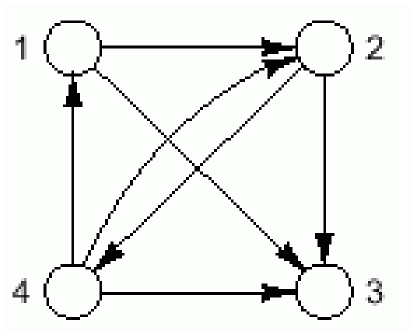
- This can be implemented in different ways.
- Our representation:
  - Vertices as a one dimensional array
  - Edges as an array of linked list (the emanating edges of vertex 1 will be in the list of the first element, and so on, …





Simple List Representation

- Vertices are represented as a 1-D array or a linked list
- Edges are represented as one linked list
  - Each edge contains the information about its two vertices



vertices

1
2
3
4

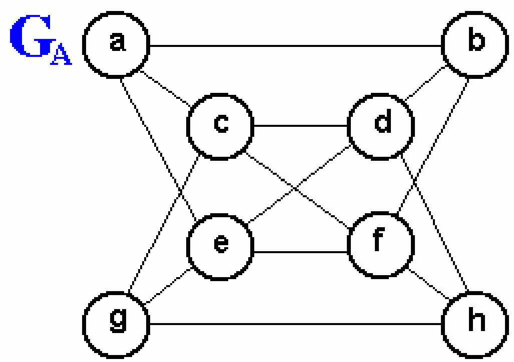
.
.
.

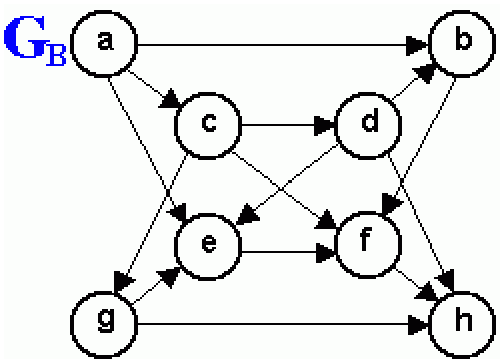
edges

edge(1,2)
edge(1,3)
edge(2,3)
edge(2,4)
edge(4,1)
edge(4,2)
edge(4,3)

.
.
.

Review Questions





1. Consider the undirected graph GA shown above. List the elements of V and E. Then, for each vertex v in V, do the following:
   1. Compute the in-degree of v
   2. Compute the out-degree of v
   3. List the elements of A(v)
   4. List the elements of I(v).
2. Consider the undirected graph GA shown above.
   1. Show how the graph is represented using adjacency matrix.
   2. Show how the graph is represented using adjacency lists.
3. Repeat Exercises 1 and 2 for the directed graph GB shown above.