Chapter 10: Graphs

Basic Concepts

Lydia Sinapova, Simpson College

Basic Graph Definitions

A graph is a mathematical object that is used to model different situations – objects and processes:

- Linked list
- Tree (partial instance of graph)
- Flowchart of a program
- City map
- Electric circuits
- Course curriculum

Vertices and Edges

Definition: A graph is a collection (nonempty set) of vertices and edges

Vertices (Nodes): can have names and properties
Edges (Connection):

connect two vertices,

can be labeled,

can be directed

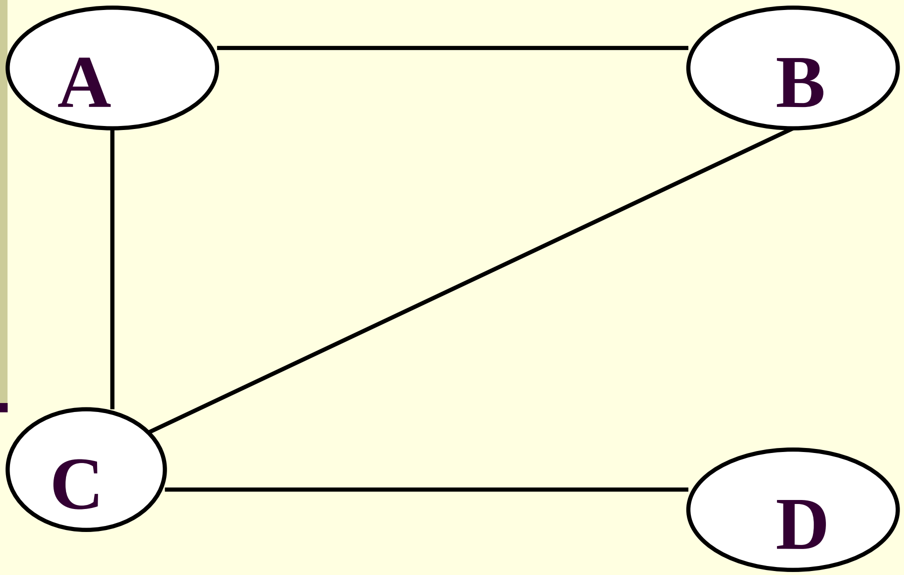
Adjacent vertices: there is an edge between them

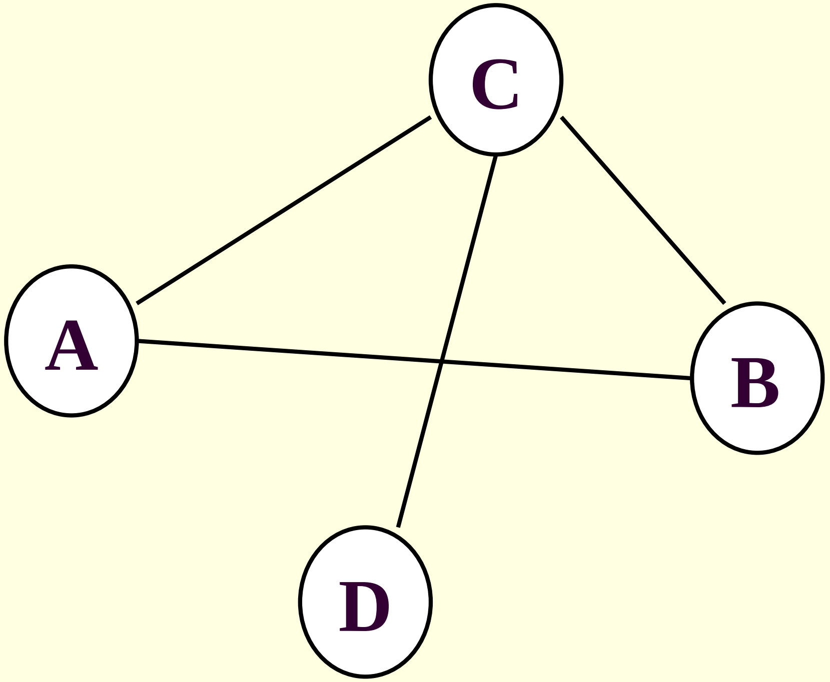
Example

Graph1

Vertices: A,B,C,D

Edges: AB, AC, BC, CD

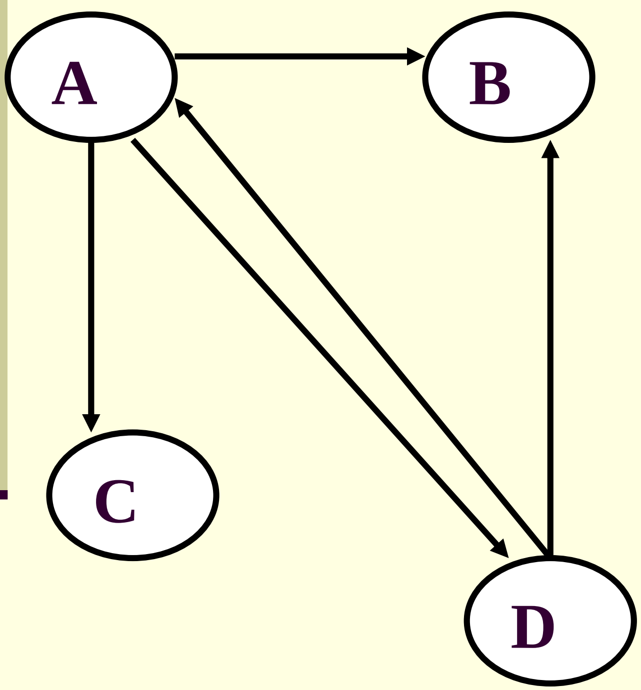




Two ways to draw the same graph

Degree of Directed and undirected graphs

Graph2



In-Degree (Indeg): No. of edges Entering Node

Out-Degree(Outdeg): No. of edges Exiting Node

Examples: In Graph2

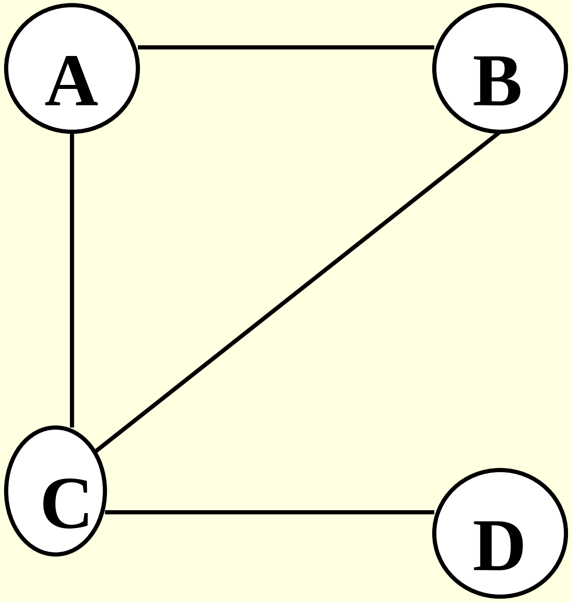
Indeg(A)=1	Outdeg(A)=3	
Indeg(B)=2	Outdeg(B)=0	CALLED ***Sink***
Indeg(C)=1	Outdeg(A)=0	CALLED ***Sink***
Indeg(D)=1	Outdeg(A)=2	

IF Indeg(N)=0 & Outdeg(N)>0, Then N is ***Source***

IF Outdeg(N)=0 & Indeg(N)>0, Then N is ***Sink***

Degree of Directed and <u>undirected</u> graphs

Graph3



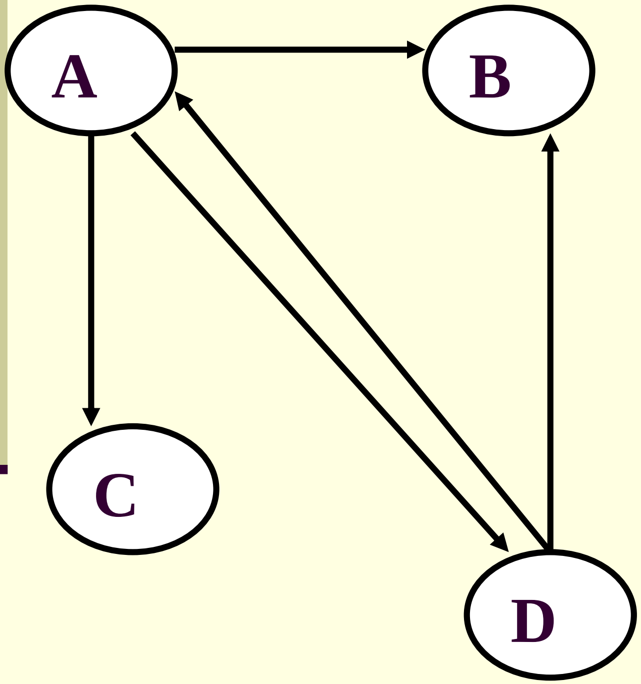
Degree (deg): No. of edges Connected to a Node

Examples: In Graph3

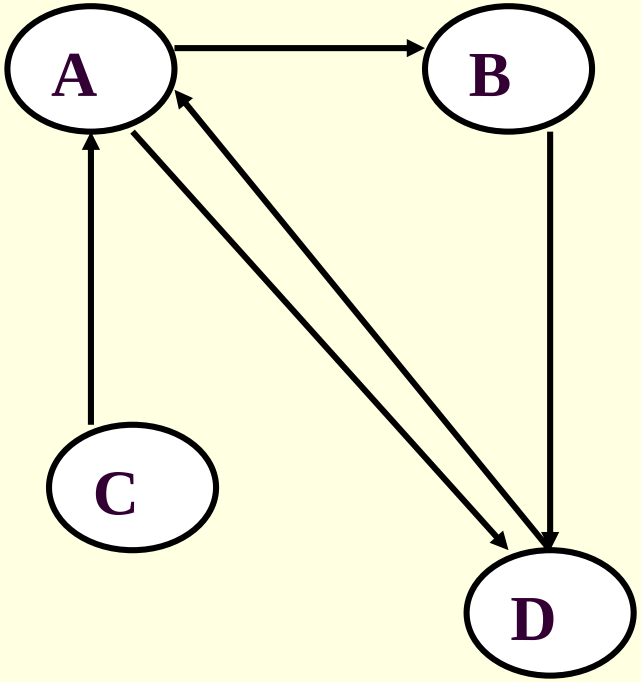
Deg(A)=2
Deg(B)=2
Deg(C)=3
Deg(D)=1

Directed and undirected graphs

Graph2



Graph3



These are two different graphs

More definitions : Path

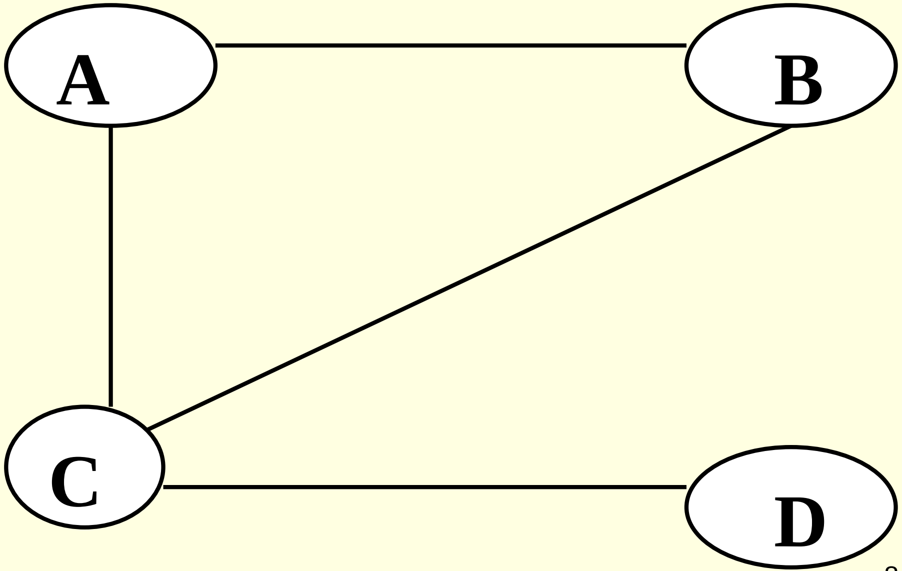
A list of vertices in which successive vertices are connected by edges

A B C

B A C D

A B C A B C A B C D

B A B A C



More definitions : Simple Path

No vertex is repeated.

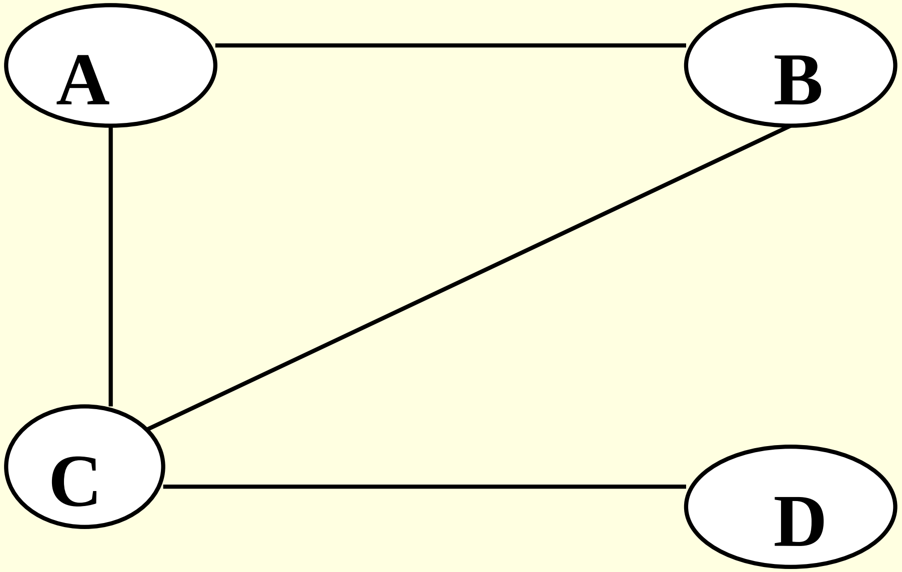
A B C D

D C A

D C B

A B

A B C



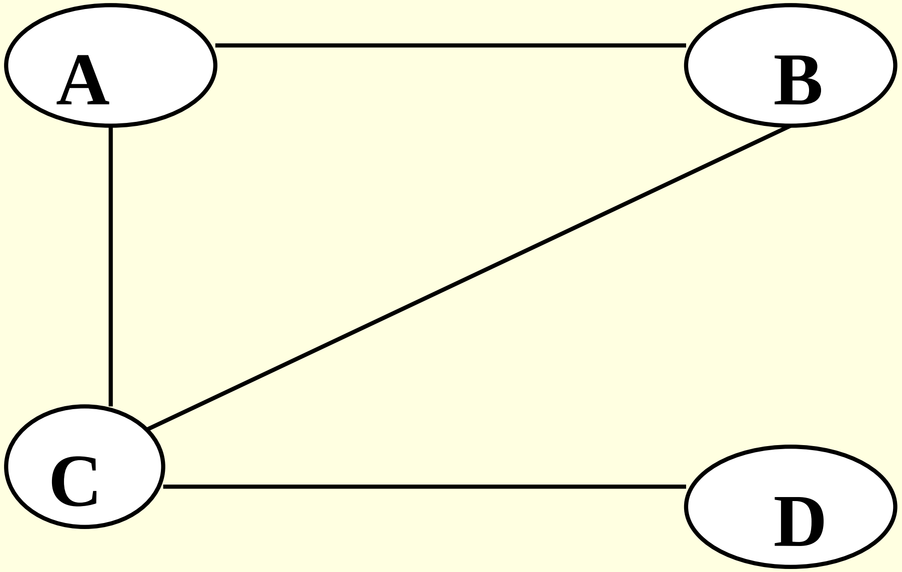
More definitions : Cycle

Simple path with distinct edges, except that the first vertex is equal to the last

A B C A

B A C B

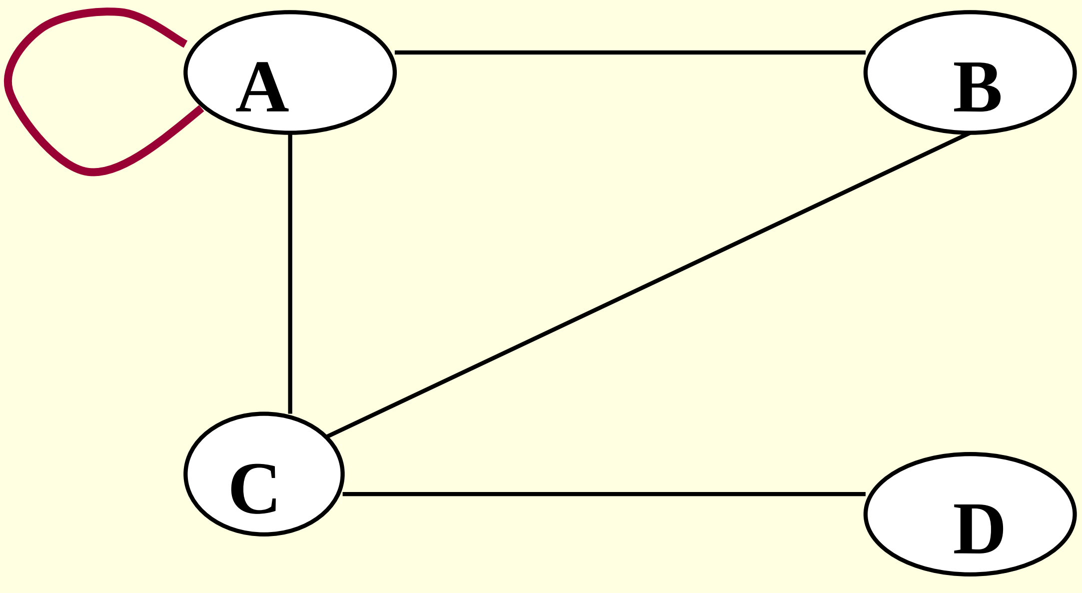
C B A C



A graph without cycles is called **acyclic graph**.

More definitions : Loop

An edge that connects the vertex with itself

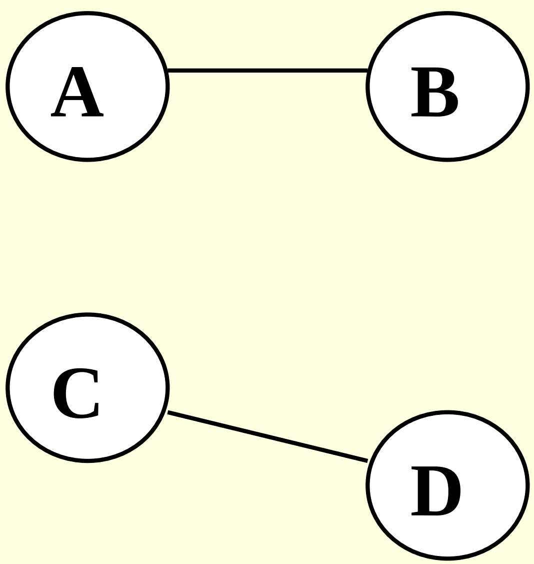


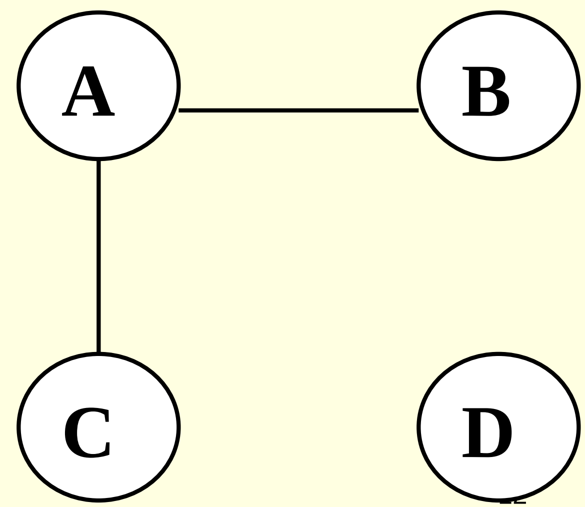
Connected and Disconnected graphs

Connected graph: There is a path between each two vertices

Disconnected graph : There are at least two vertices not connected by a path.

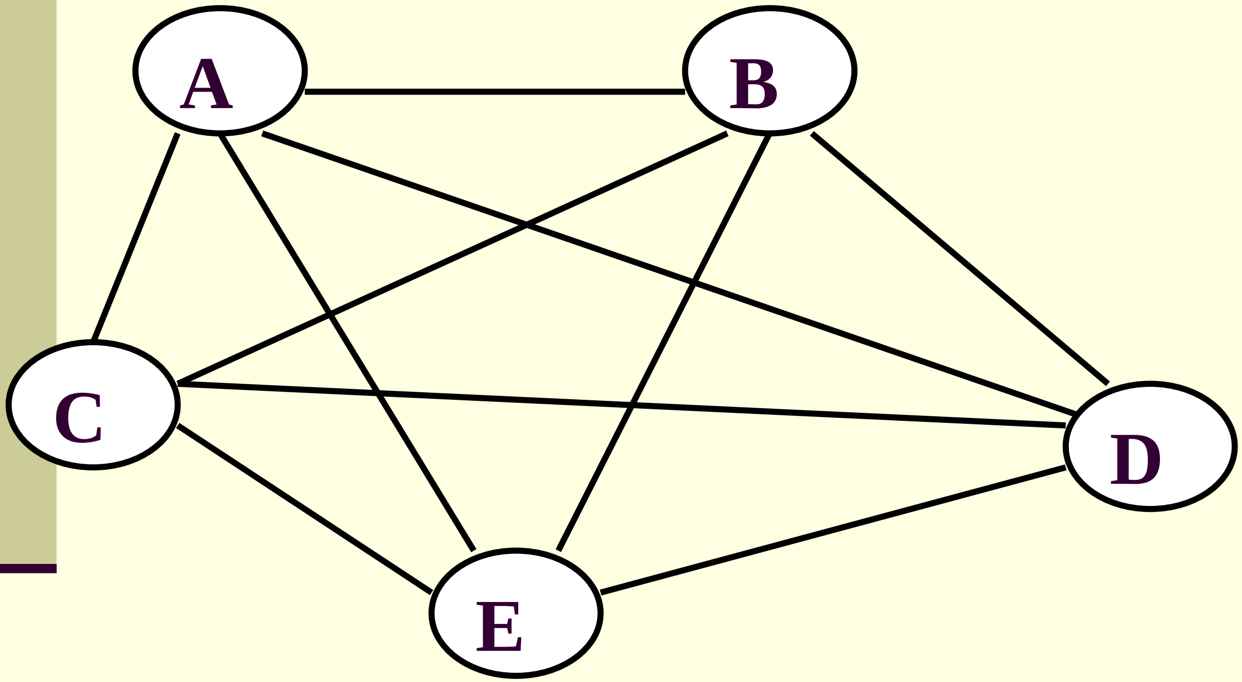
Examples of disconnected graphs:





Complete graphs

Graphs with all edges present – each vertex is connected to all other vertices



A complete graph

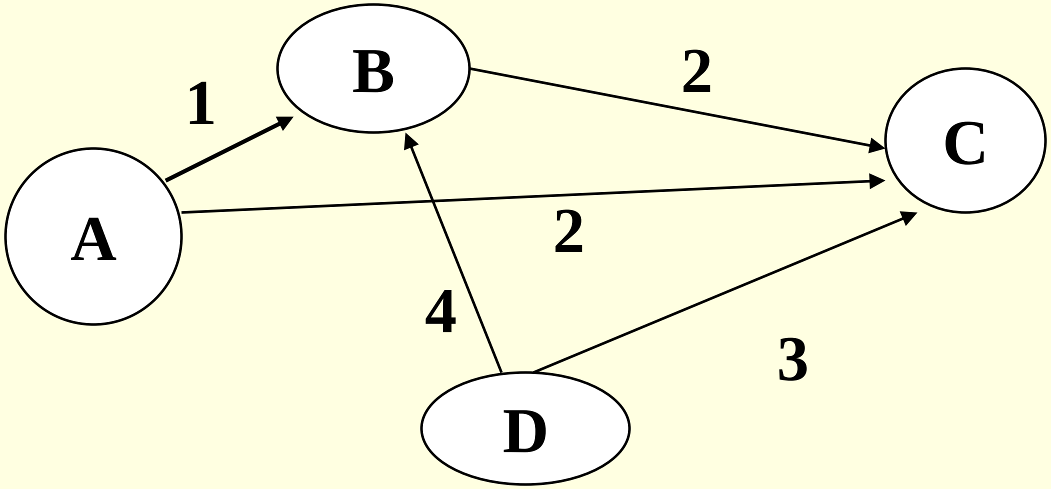
Dense graphs: relatively few of the possible edges are missing

Sparse graphs: relatively few of the possible edges are present

Weighted graphs and Networks

Weighted graphs – weights are assigned to each edge (e.g. road map)

Networks: directed weighted graphs (some theories allow networks to be undirected)



Graph Representation

- Adjacency matrix
- Adjacency lists

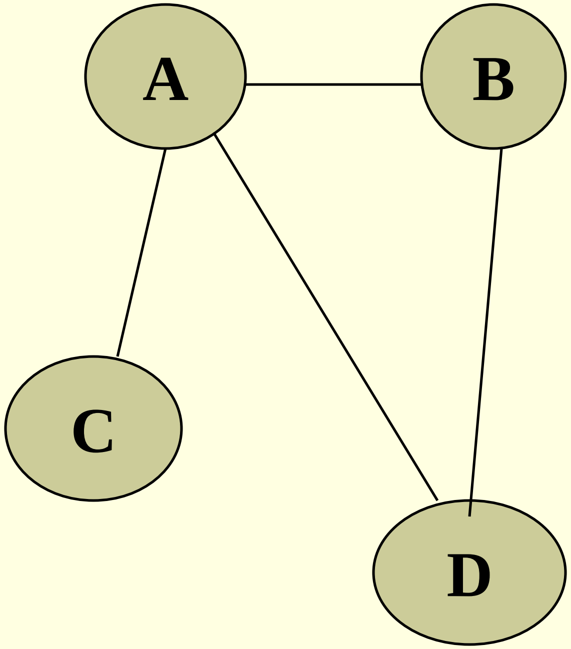
Adjacency matrix – undirected graphs

Vertices: A,B,C,D

Edges: AC, AB, AD, BD

The matrix is symmetrical

	A	B	C	D
A	0	1	1	1
B	1	0	0	1
C	1	0	0	1
D	1	1	0	0

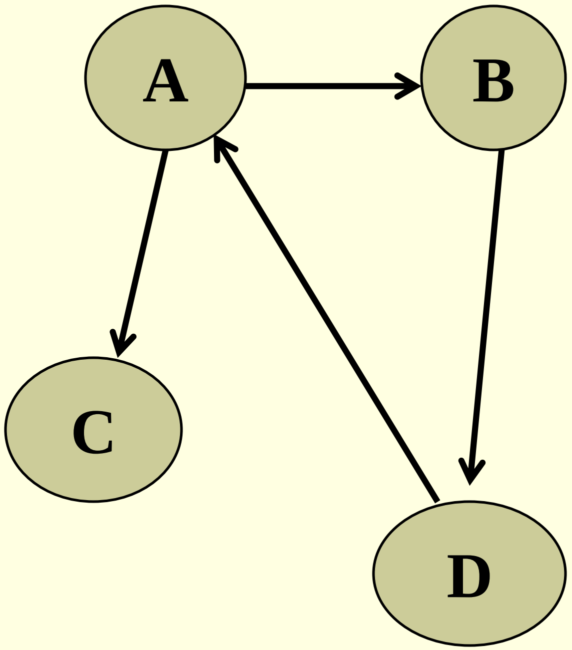


Adjacency matrix – directed graphs

Vertices: A,B,C,D

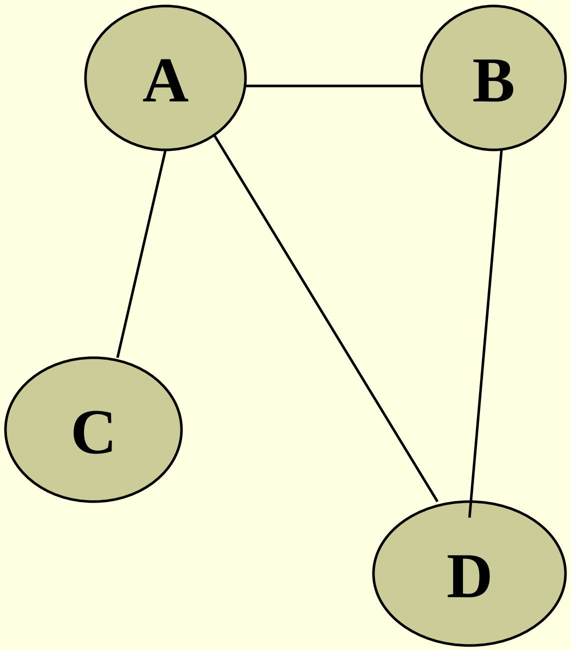
Edges: AC, AB, BD, DA

	A	B	C	D
A	0	1	1	0
B	0	0	0	1
C	0	0	0	0
D	1	0	0	0



Adjacency lists – undirected graphs

Vertices:	**A,B,C,D**
Edges:	**AC, AB, AD, BD**
Heads	**lists**
A	**B C D**
B	**A D**
C	**A**
D	**A B**

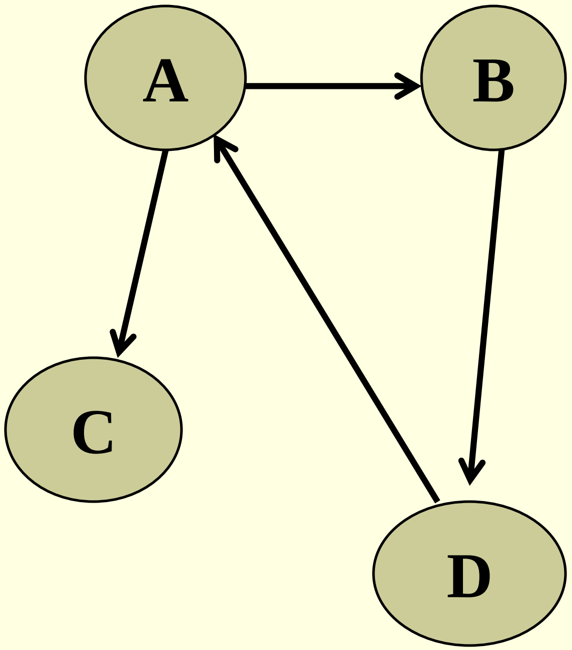


Adjacency lists – directed graphs

Vertices: A,B,C,D

Edges: AC, AB, BD, DA

Heads	lists
A	B C
B	D
C	=
D	A



Graph Traversal

- Breadth First Search BFS (uses Queue)
- Depth First Search DFS (uses Stack)

BFS – Basic Idea

Given a graph with N vertices and a selected vertex A:

for (i = 1;

there are unvisited vertices ; i++)

Visit all unvisited vertices at distance **i**

(**i** is the length of the shortest path between **A** and currently processed vertices)

Queue-based implementation

BFS – Algorithm

BFS algorithm

1. Store source vertex **S** in a **queue** and mark as processed

2. **while** queue is not empty

Read vertex **v** from the queue

for all neighbors **w**:

If **w** is not processed

Mark as processed

Append in the queue

Record the parent of **w** to be **v** (necessary only if we need the shortest path tree)

Example

Breadth-first traversal: 1, 2, 3, 4, 6, 5

1: starting node

2, 3, 4 : adjacent to 1

(at distance 1 from node 1)

6 : unvisited adjacent to node 2.

5 : unvisited, adjacent to node 3

Adjacency lists

1: 2, 3, 4

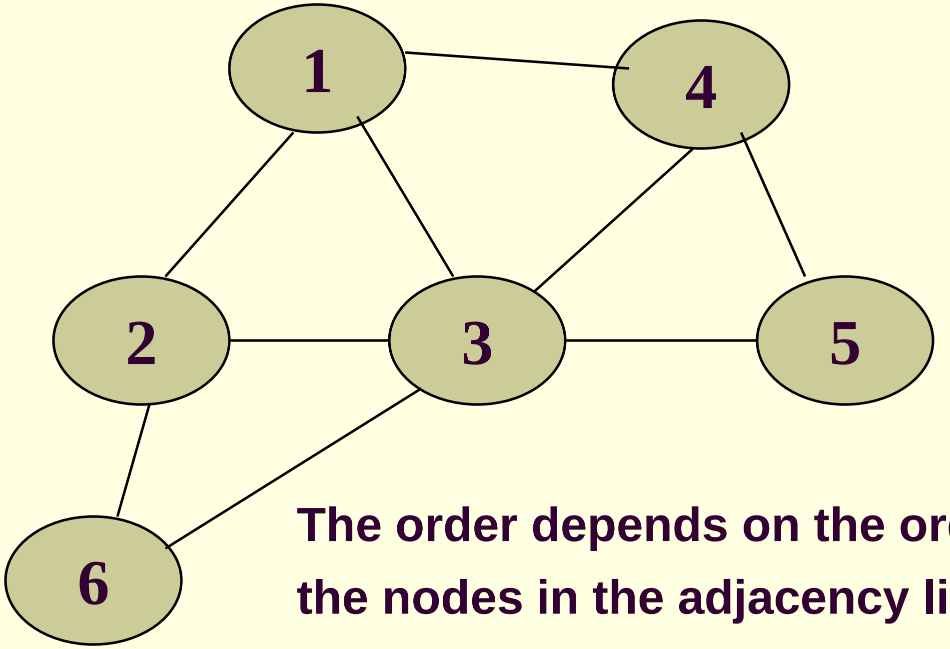
2: 1, 3, 6

3: 1, 2, 4, 5, 6

4: 1, 3, 5

5: 3, 4

6: 2, 3



The order depends on the order of the nodes in the adjacency lists