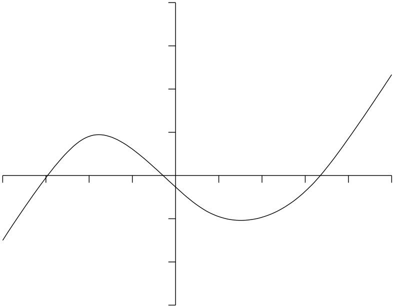
Graphs, Part I

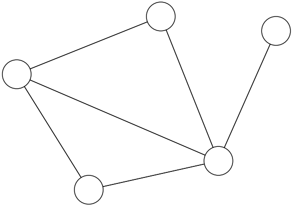
January 28, 2011

What Is a Graph?

NO



YES



What Is a Graph?

- Set of nodes (or vertices)
- Set of edges between pairs of nodes

What Are Graphs Good For?

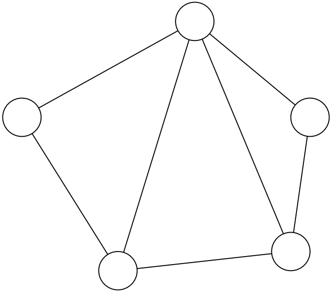
LOTS of situations can be viewed as graphs.
$\longrightarrow$ Any situation that involves relationships between pairs of things.

Examples:

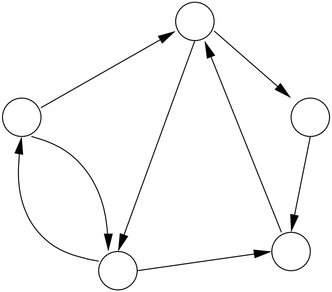
Problem	Nodes	Edges
Message routing in network	computers	wires
Transportation	cities	highways
Social relationships	people	friendships
Exam scheduling	exams	shared student
Planning courses to take	courses	prerequisites
Chess	board configurations	moves

Types of Graphs

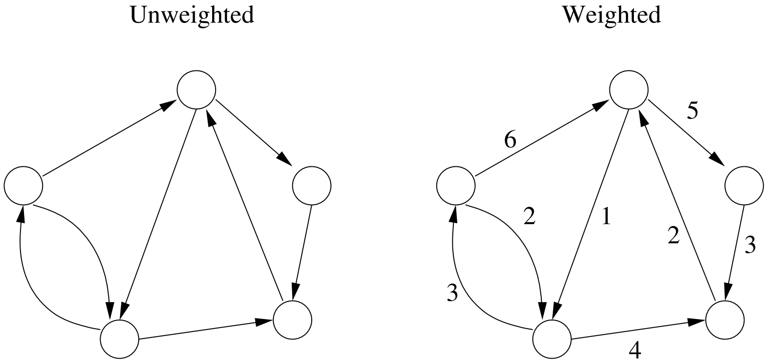
Undirected



Directed

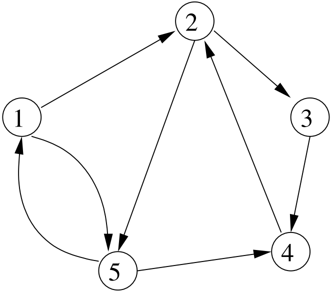


Types of Graphs



Adjacency Matrix Representation (Directed Graph)

A graph can be represented in a computer as an *adjacency matrix* of boolean values.



		TO				
		1	2	3	4	5
F	1	0	1	0	0	1
R	2	0	0	1	0	1
O	3	0	0	0	1	0
M	4	0	1	0	0	0
	5	1	0	0	1	0

(1 = true and 0 = false.)

Adjacency Matrix in Java

Create an $n \times n$ two-dimensional array of booleans:
`boolean A[][] = new boolean[n][n]`

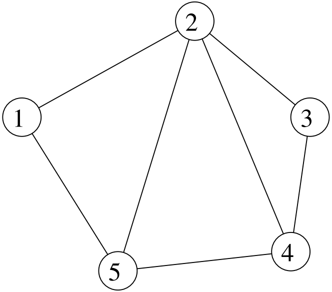
To access an entry representing the edge from node i to node j, use `A[i][j]`.

Remember that array indices will be $0..n-1$.

If you want to number the nodes from 1 to n, create an $(n+1) \times (n+1)$ array and just don't use index 0.

Adjacency Matrix Representation (Undirected Graph)

A graph can be represented in a computer as an *adjacency matrix* of boolean values.

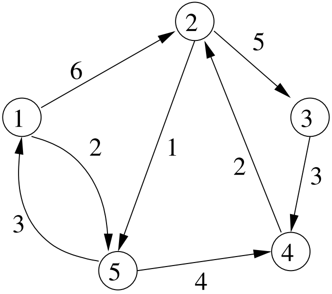


		TO				
		1	2	3	4	5
F	1	0	1	0	0	1
R	2	1	0	1	1	1
O	3	0	1	0	1	0
M	4	0	1	1	0	1
	5	1	1	0	1	0

(We treat each edge as if it goes in both directions.)

Adjacency Matrix Representation (Weighted Graph)

A *weighted* graph can be represented in a computer as an adjacency matrix of *weights*.



		TO				
		1	2	3	4	5
F	1	0	6	0	0	2
R	2	0	0	5	0	1
O	3	0	0	0	3	0
M	4	0	2	0	0	0
	5	3	0	0	4	0

When to Use Adjacency Matrix

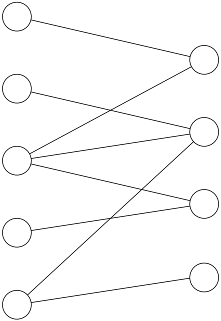
Most of the edges are present in the graph.

If graph has fewer edges, use adjacency lists (next week).

Note: If graph has 1000 nodes, adjacency matrix uses about 1 MB of memory.

Bipartite graphs

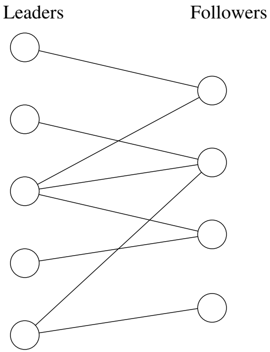
Another flavour of graph is the bipartite graph.



Useful for situations where there are two *types* of things.
Relationships are between things of opposite types.

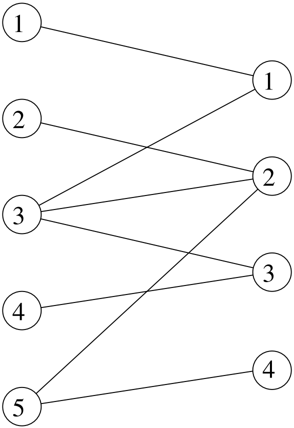
Bipartite Graph Example

When two people dance together, one leads and the other follows. At a dance, there are 5 leaders and 4 followers. The graph indicates which pairs of people are willing to dance with each other.



Bipartite Graph as Adjacency Matrix

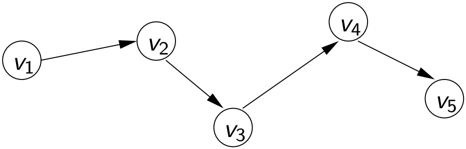
Bipartite graphs can also be represented as an adjacency matrix.



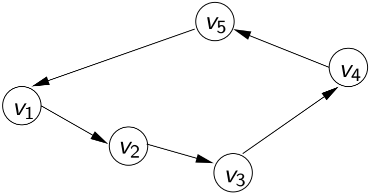
		RIGHT			
		1	2	3	4
L	1	1	0	0	0
E	2	0	1	0	0
F	3	1	1	1	0
T	4	0	0	1	0
	5	0	1	0	1

Some Basic Graph Terminology

- Path: sequence of nodes $v_1, v_2, \ldots, v_k$ where $v_i \to v_{i+1}$ is an edge for all i.

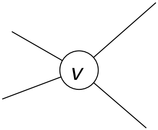


- Cycle: Path that starts and ends at the same node.



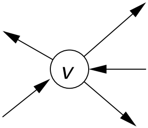
Some Basic Graph Terminology

- Degree of a node v: number of undirected edges that connect v to other nodes.



degree of v is 4

- In-degree of a node v: number of directed edges that point to v.
- Out-degree of a node v: number of directed edges that point out of v.



in-degree of v is 2
out-degree of v is 3