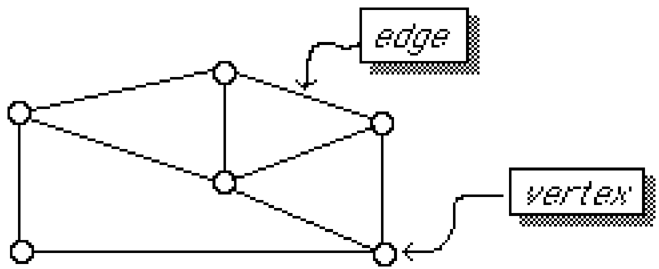
Graphs and Networks:
basic definitions & concepts

This Hypercard stack was prepared by:
Dennis Bricker,
Dept. of Industrial Engineering,
University of Iowa,
Iowa City, Iowa 52242
e-mail: dennis-bricker@uiowa.edu

author

A GRAPH consists of

- a collection of VERTICES or NODES
- a collection of LINKS or EDGES



Formally, a GRAPH is a pair of sets (V,A) where

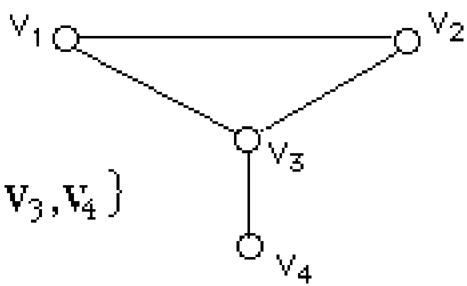
- V is non-empty
- A is an irreflexive, symmetric relation on V

irreflexive: $(v_i,v_i) \notin A$

symmetric: $(v_i,v_j) \in A \Leftrightarrow (v_j,v_i) \in A$

vertex set:

$$V = \{v_1, v_2, v_3, v_4\}$$

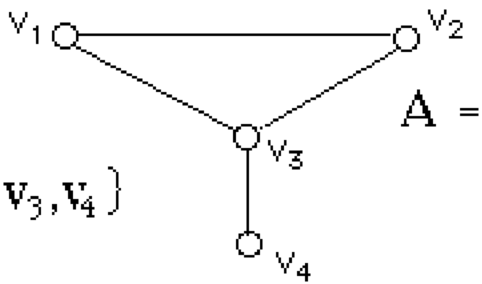


$$A = \{(v_1,v_2), (v_2,v_1), (v_2,v_3), (v_3,v_2), (v_3,v_4), (v_4,v_3), (v_1,v_3), (v_3,v_1)\}$$

a symmetric pair of elements (v_i,v_j), (v_j,v_i) *is called an EDGE*

The number of vertices is the **ORDER** of the graph

The number of edges is the **SIZE** of the graph



vertex set:

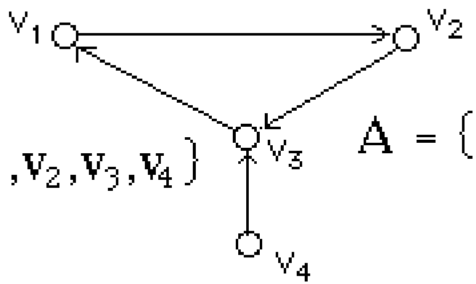
$$V = \{v_1, v_2, v_3, v_4\}$$

ORDER=4

$$A = \{(v_1,v_2), (v_2,v_1), (v_2,v_3), (v_3,v_2), (v_3,v_4), (v_4,v_3), (v_1,v_3), (v_3,v_1)\}$$

SIZE = 4

A **DIGRAPH** or DIRECTED GRAPH is a pair of sets (V,A) where A is not symmetric, that is, the links have directions

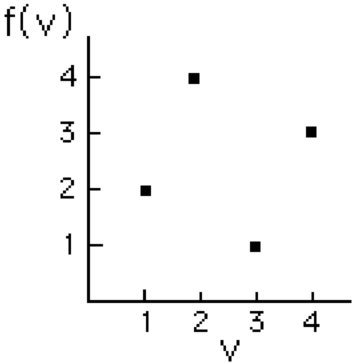


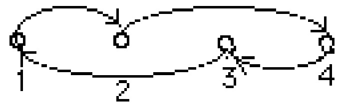
$$V = \{v_1, v_2, v_3, v_4\}$$

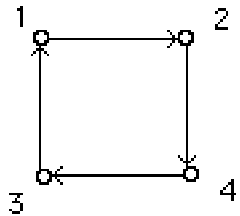
$$A = \{(v_1,v_2),(v_2,v_3),(v_4,v_3),(v_3,v_1)\}$$

Directed links are often called ARCS

Three representations of a digraph G=(V,A)
where V=(1,2,3,4) and A={(1,2),(2,4),(4,3),(3,1)}

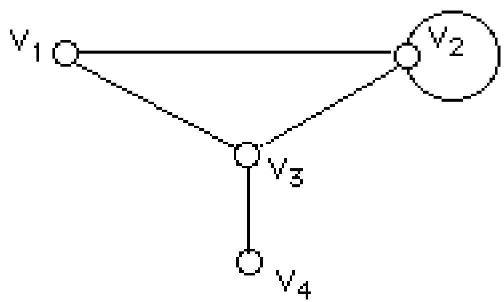




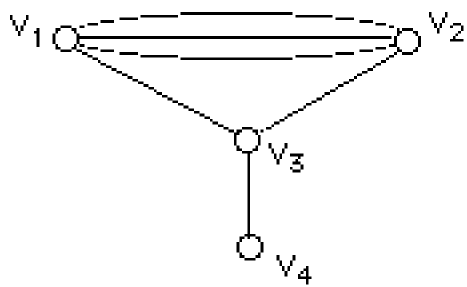


f(1)=2,
f(2)=4,
f(3)=1,
f(4)=3

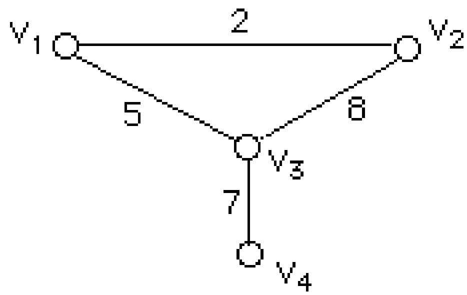
A "pure" graph has no loops, i.e., (v_i, v_i) is not a valid edge. If the edge set includes (v_i, v_i), the entity is called a LOOP-GRAPH



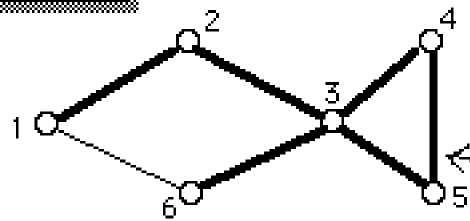
If multiple edges are allowed joining pairs of vertices, then the entity is called a MULTI-GRAPH



If each edge of a graph has an associated number, the entity is called a **NETWORK**

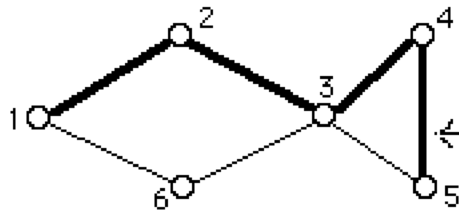


GRAPH



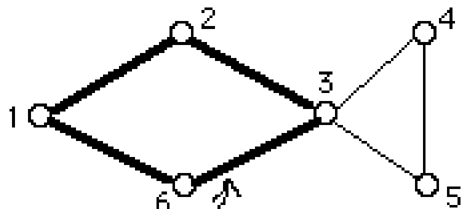
CHAIN : a sequence of vertices, $(x_0, x_1, \ldots x_i, x_{i+1}, \ldots x_s)$ where each pair (x_i, x_{i+1}) is an edge

(1,2,3,4,5,3,6)



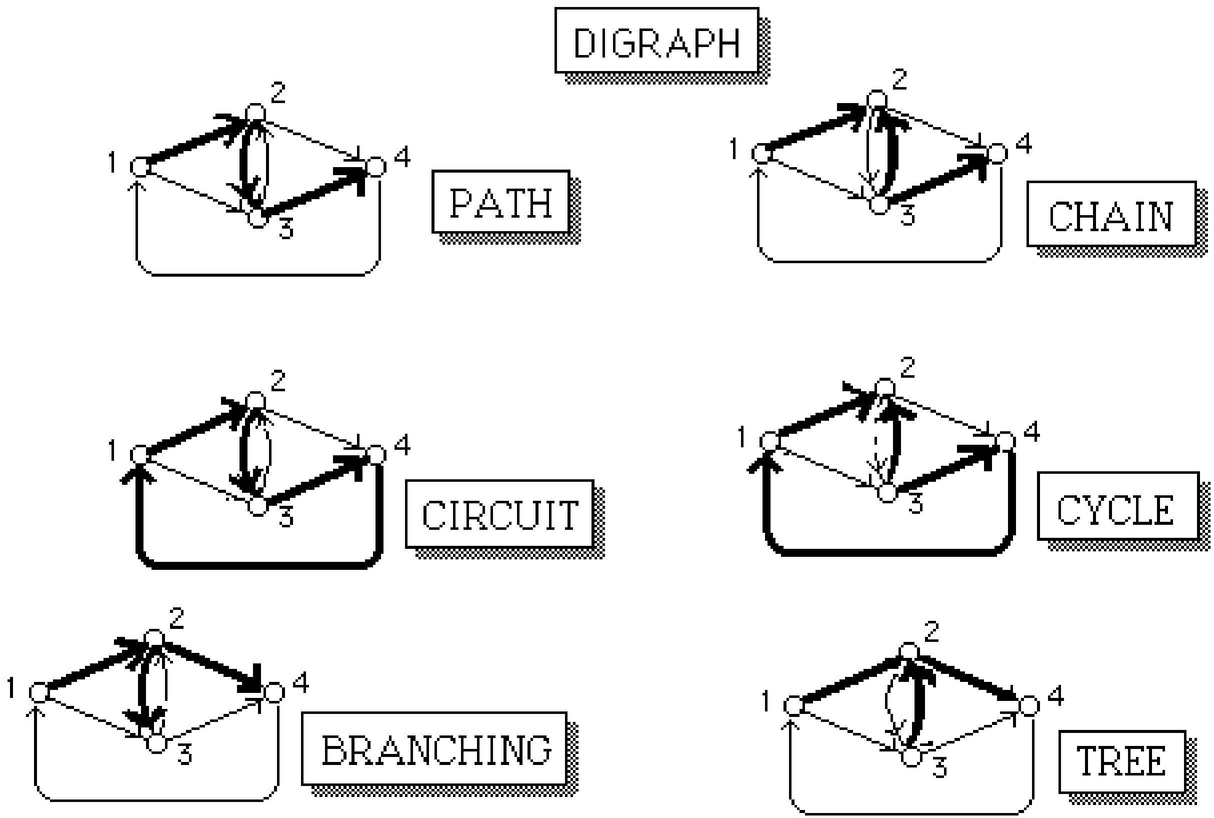
ELEMENTARY CHAIN (no vertices are repeated)

(1,2,3,4,5)

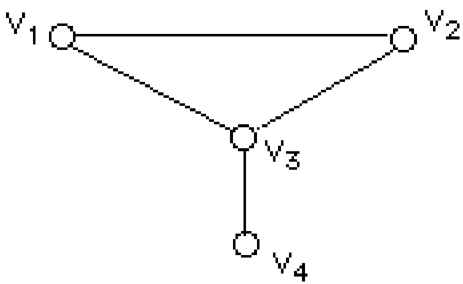


CYCLE (a closed chain, i.e., the first and last vertices of the chain are the same)

(1,2,3,6,1)


DIGRAPH
PATH
CHAIN
CIRCUIT
CYCLE
BRANCHING
TREE
1
2
3
4

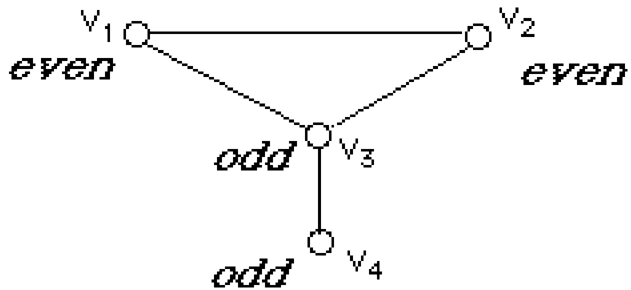
The **DEGREE** of a vertex is the number of edges incident with the vertex



vertex	*degree*
1	2
2	2
3	3
4	1

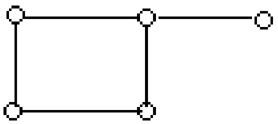
Theorem: The sum of the degrees of the vertices of a graph is twice the number of edges

A vertex of a graph is **EVEN** or **ODD** according to whether its degree is an even or odd integer, respectively.

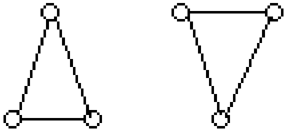


Theorem: Every graph contains an even number of odd vertices

A graph is **CONNECTED** if, for every pair of vertices, x & y, there is a chain of edges from vertex x to vertex y.



connected

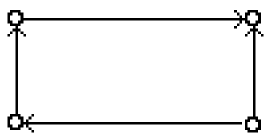


not connected

A directed graph is **CONNECTED**

if, for every pair of vertices, x & y, there is a chain of edges from vertex x to vertex y,

and **STRONGLY CONNECTED** if there is a path of edges from vertex x to vertex y.



connected, but not strongly connected

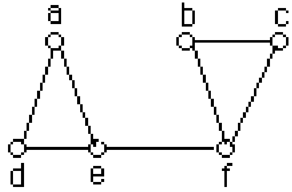
strongly connected

Suppose that we wish to assign directions to the edges of a connected graph so as to obtain a STRONGLY-CONNECTED digraph.

Under what conditions, if any, is this possible?

For example, can we make each street in a city one-way so that a vehicle at any intersection can reach any other intersection?

A **BRIDGE** of a connected graph is an edge which, if removed, destroys the graph's connectedness.



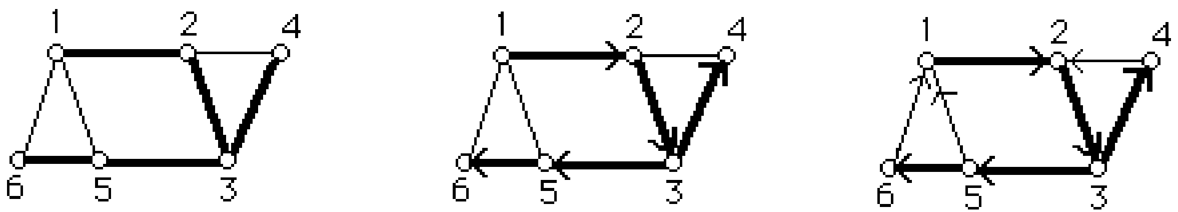
Edge (e,f) is a BRIDGE of the graph

Robbins' Theorem

A graph has a strongly-connected orientation
if and only if
the graph is connected and has no bridge.

Finding a Strongly-Connected Orientation

- First, find a DEPTH-FIRST-SEARCH SPANNING TREE
- Orient all edges ON the spanning tree from the vertex with smaller label to the vertex with the larger label
- Orient all edges NOT on the spanning tree from the vertex with larger label to the vertex with smaller label



DEPTH-FIRST-SEARCH SPANNING TREE

[0] Select any vertex, and label it "1". Let $i \leftarrow j \leftarrow 1$.

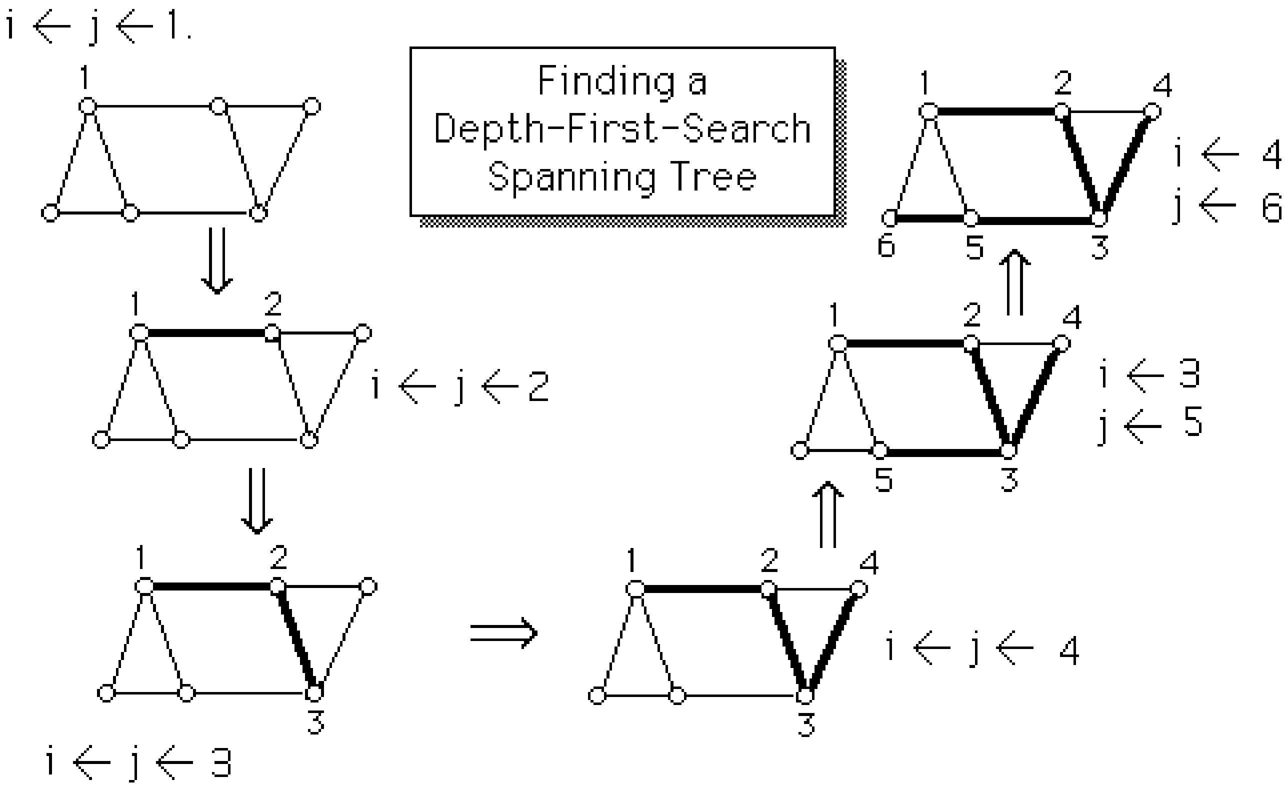
[1] Select any vertex which is connected by a single edge to the vertex labeled "i". If none, go to step [4]; otherwise, proceed to step [2]

[2] Label the selected vertex "j+1"

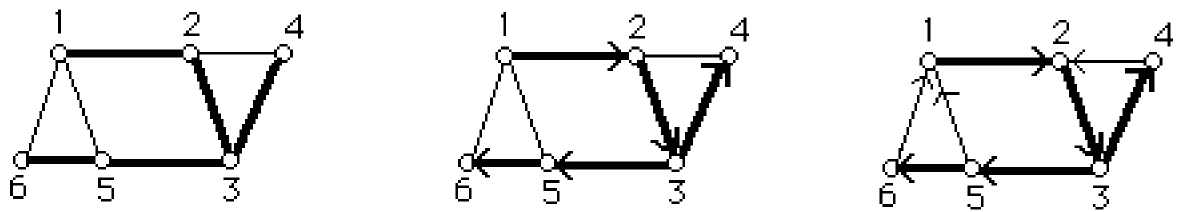
[3] Let $i \leftarrow j \leftarrow j+1$. Go to step [1].

[4] Let $i \leftarrow i-1$. If $i=0$, STOP; otherwise, go to step [1].

Finding a Depth-First-Search Spanning Tree



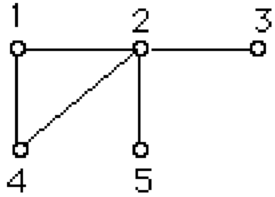
Example: Finding a strongly-connected orientation of a connected graph



depth-first-search spanning tree | *orient edges on the tree* | *orient edges not on the tree*

ADJACENCY MATRIX

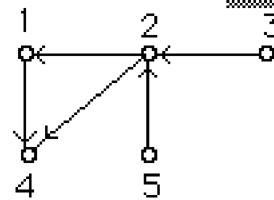
graph



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

$$A_i^j = \begin{cases} 1 & \text{if there is an edge } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

digraph

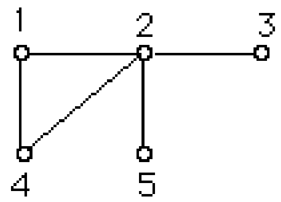


$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$A_i^j = \begin{cases} 1 & \text{if there is an arc } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

REACHABILITY MATRIX

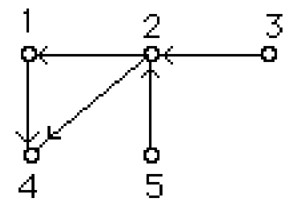
graph



$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$R_i^j = \begin{cases} 1 & \text{if there is a chain from vertex } i \text{ to vertex } j \\ 0 & \text{otherwise} \end{cases}$$

digraph

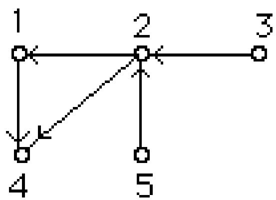


$$R = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$R_i^j = \begin{cases} 1 & \text{if there is a path from vertex } i \text{ to vertex } j \\ 0 & \text{otherwise} \end{cases}$$

Consider the generalized inner product $\vee.\wedge$ in APL notation:

(*or* → $\vee$, *and* → $\wedge$)



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

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 \end{bmatrix} \vee.\wedge \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \equiv (1\wedge 1)\vee(0\wedge 1)\vee(0\wedge 0)\vee(1\wedge 0)\vee(0\wedge 0)$$

$$\equiv 1 \vee 0 \vee 0 \vee 0 \vee 0$$

$$\equiv 1$$

row #2

column #4

indicates that there is an arc (2,1) ***and*** *an arc (1,4)*

indicates that there is a path of 2 arcs from 2 to 4

The value in row i & column j of the matrix

$$A \vee . \wedge A$$

is 1 if there is a path, consisting of 2 arcs, from vertex i to vertex j,
and 0 otherwise

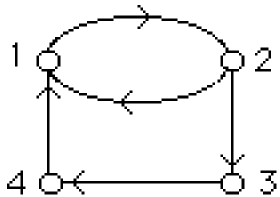
$(A \vee . \wedge A) \vee . \wedge A$ has a 1 in row i&column j if there is a path consisting of 3 arcs from i to j
etc.

How can the reachability matrix be computed?

An APL function to compute the reachability matrix:

```
    ∇R←A REACH N
[1]  →(N=0)/LAST
[2]  R ← A ∨.∧ A REACH N-1
[3]  →0
[4] LAST: R ← IDENTITY 1↑ρA
    ∇
```

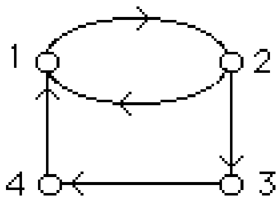
Powers of the Adjacency Matrix



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

$$A^2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} +.\times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

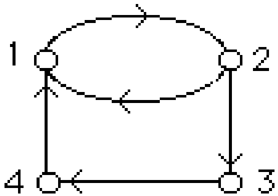
inner product (APL)



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad A^2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} +.\times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Theorem: If A is the adjacency matrix of a digraph, then the entry in row i & column j of A^k is the number of paths of length k edges from vertex i to vertex j



$$A^3 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \qquad A^4 = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$