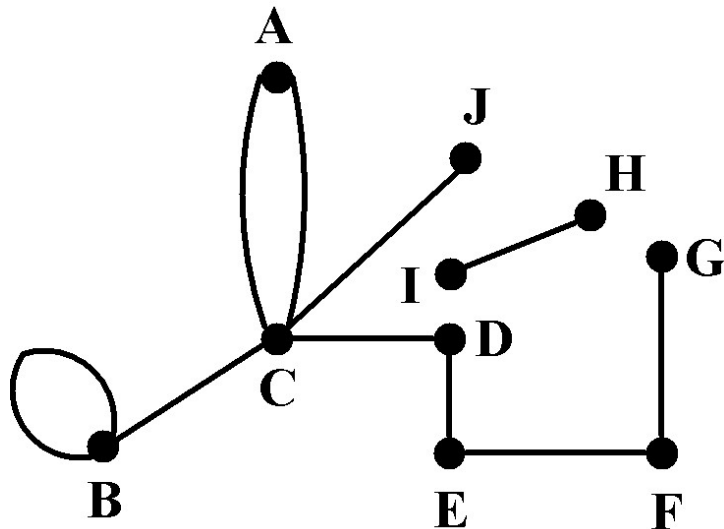


Graph concepts

Graphs are made up by **vertices (nodes)** and **edges (links)**.
An edge connects two vertices, or a vertex with itself – **loop**.



AC, AC - multiple edges
BB – loop

The shape of the graph does not matter, only the way the nodes are connected to each other.

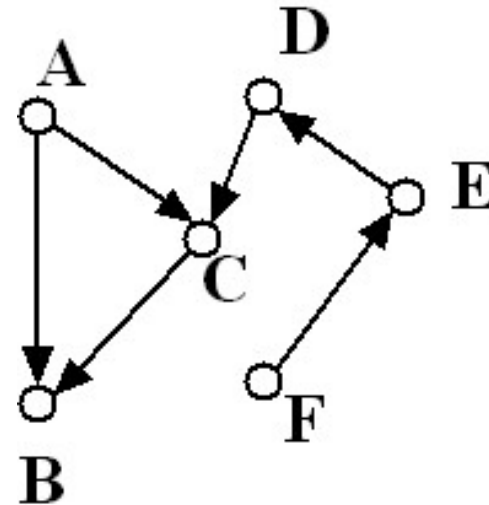
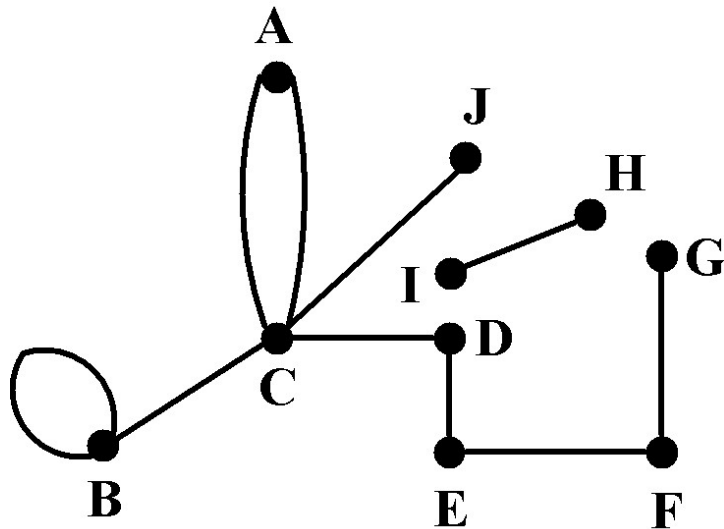
Simple graph - does not have loops (self-edges) and multiple identical edges.

Further reading:

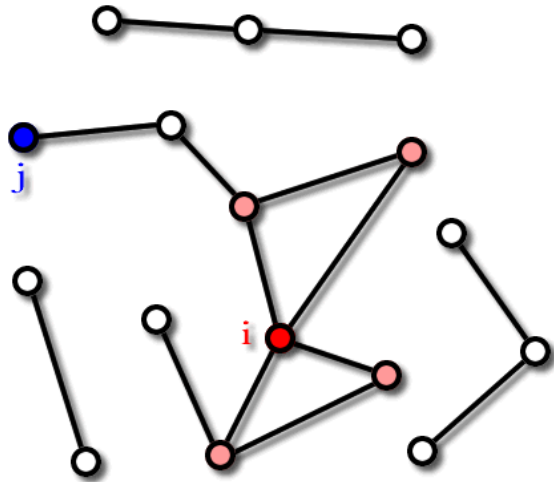
<http://www.utm.edu/departments/math/graph/glossary.html>

Symmetrical and directed graphs

Two distinct types of edges: symmetrical and directed (also called arcs).
Two different graph frameworks: graph, digraph.
In the digraph framework a symmetrical edge means the superposition of two opposite directed edges.

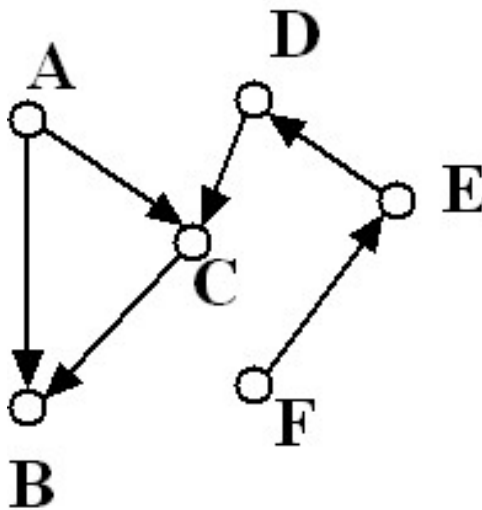


Node degrees



Node degree: the number of edges connected to the node. $k_i = 4$

In directed networks we can define an **in-degree** and **out-degree**. The (total) degree is the sum of in- and out-degree. $k_C^{in} = 2$ $k_C^{out} = 1$ $k_C = 3$

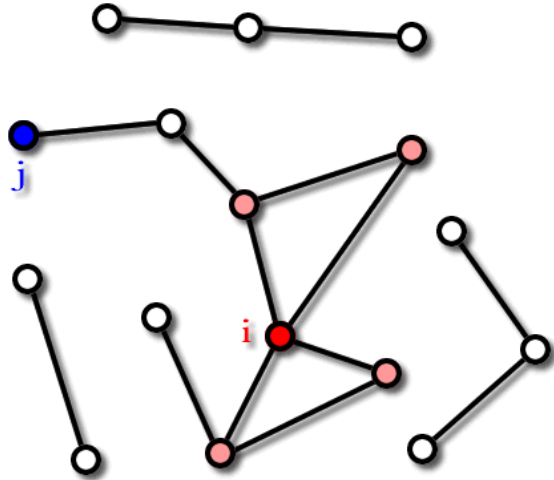


Source: a node with in-degree = 0.

Sink: a node with out-degree = 0.

E.g. A, F are sources, B is a sink.

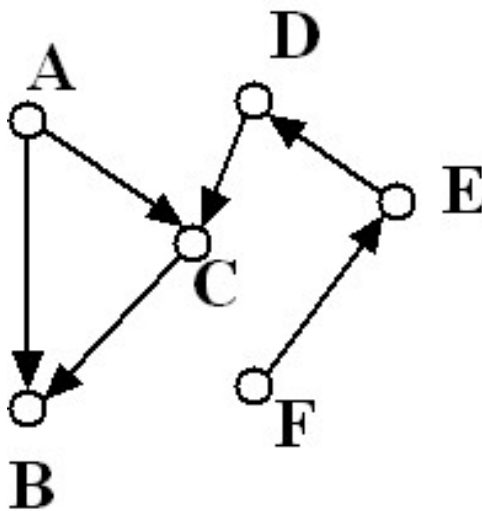
Average degree



$$\langle k \rangle \equiv \frac{1}{N} \sum_{i=1}^N k_i$$

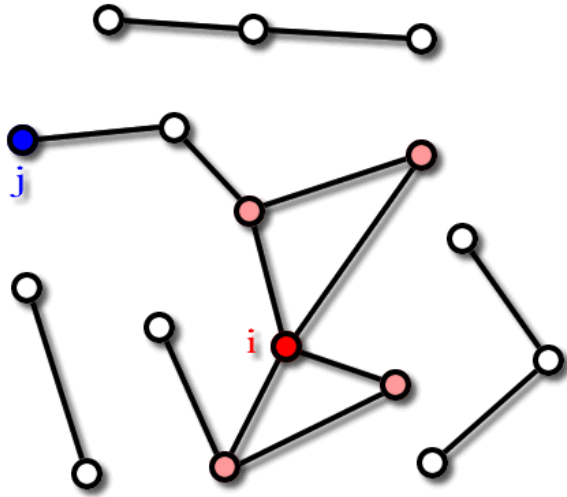
N – the number of nodes

$$\langle k^{in} \rangle \equiv \frac{1}{N} \sum_{i=1}^N k_i^{in}, \quad \langle k^{out} \rangle \equiv \frac{1}{N} \sum_{i=1}^N k_i^{out}, \quad \langle k^{in} \rangle = \langle k^{out} \rangle$$



Q: What is the relation between the number of edges in a (non-directed) graph and the sum of node degrees? How about in a directed graph?

Statistics of node degrees



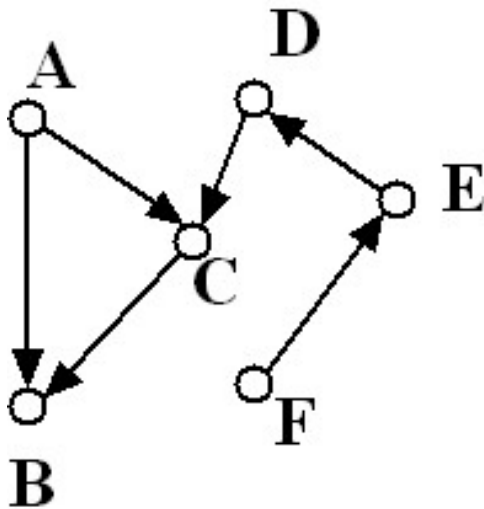
Average degree

$$\langle k \rangle \equiv \frac{1}{N} \sum_{i=1}^N k_i = \frac{2E}{N}$$

$$\langle k^{in} \rangle = \langle k^{out} \rangle = \frac{E}{N}$$

The degree distribution $P(k)$ gives the fraction of nodes that have k edges.

Similarly $P(k^{in}) / P(k^{out})$ gives the fraction of nodes that have in-degree k^{in} / out-degree k^{out} .



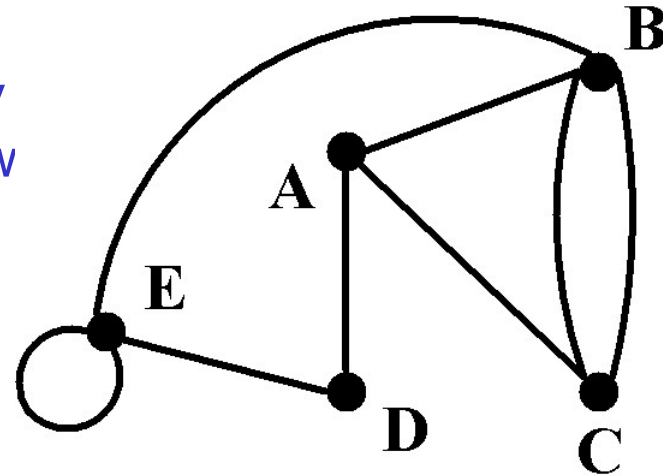
Paths and circuits

Adjacent nodes (vertices) – there is an edge joining them.

In the digraph framework the adjacency only defined in the direction of the arrow

Path: a sequence of nodes in which each node is adjacent to the next one.

Edges can be part of a path only once.



A path in a directed network needs to follow the direction of the edges, thus a symmetrical edge in the digraph framework can be used once in one direction and once in the opposite.

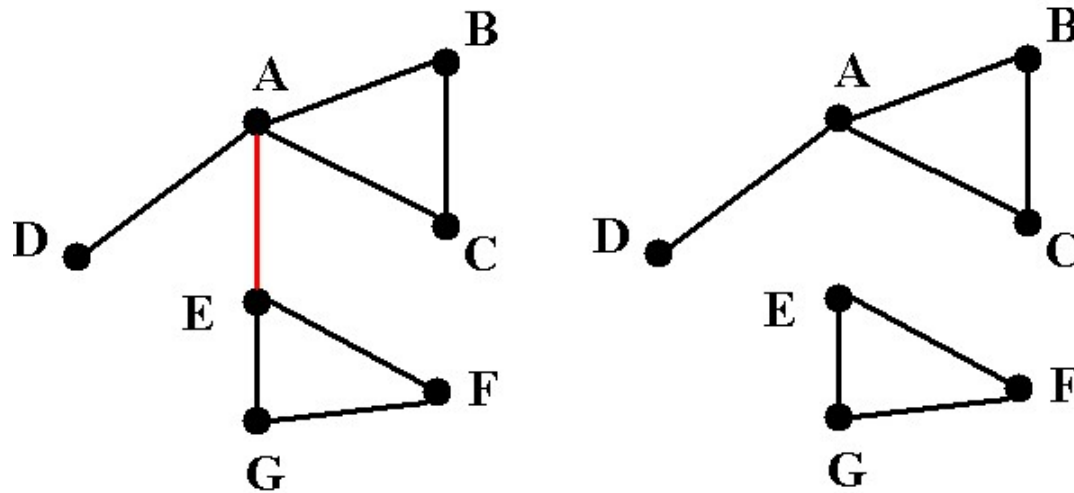
Circuit:

a path that starts and ends at the same vertex .

Connected and disconnected graphs

Connected graph: any two vertices can be joined by a path.

A disconnected graph is made up by connected components.



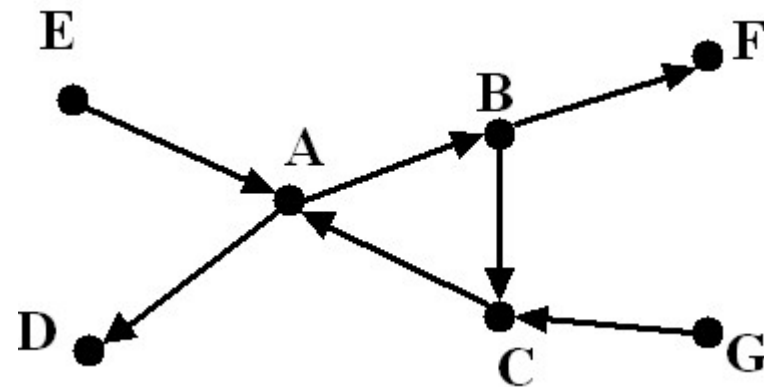
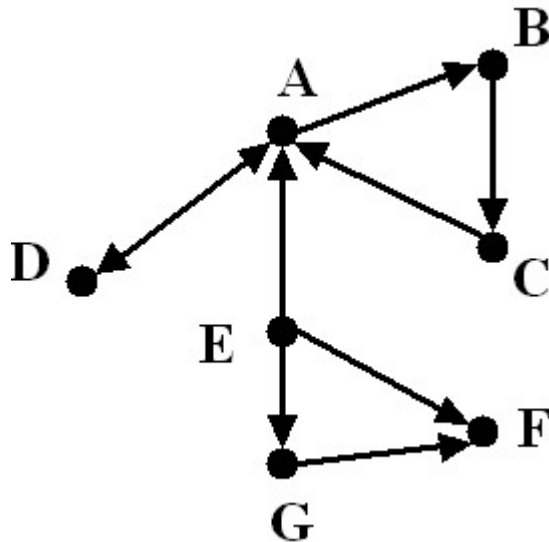
Bridge: if we erase it, the graph becomes disconnected.

Connectivity of directed graphs

Strongly connected directed graph: has a path from each node to every other node **and vice versa** (e.g. AB path and BA path).

Weakly connected directed graph: it is connected if we disregard the edge directions.

Strongly connected components can be identified, but not every node is part of a nontrivial strongly connected component.



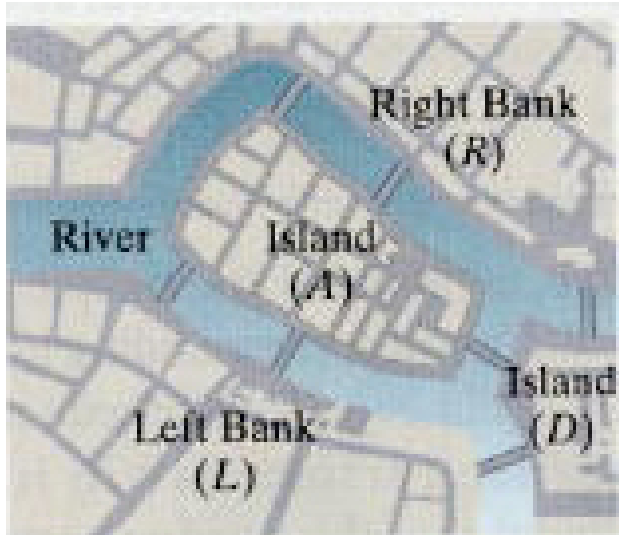
In-component: nodes that can reach the scc,

out-component: nodes that can be reached from the scc.

Exercises

1. Draw a graph or digraph with 4 nodes such that each node has degree 1 / 2 / 3. Try to use a variety of edges: symmetrical, directed, multiple edges, loops.
2. You have N nodes and need to build a connected graph from them. Each time you add an edge you must pay \$1. What is the minimum amount of money needed to build the graph?
3. You are constructing a disconnected graph from N nodes. For each edge you add you receive \$1. You are not allowed to use directed edges, loops or multiple edges, and you must stop before the graph becomes connected. What is the most money you can make?

Exercise: the bridges of Königsberg



Walking problem –
traverse each bridge
do not recross any bridge
return to the starting point

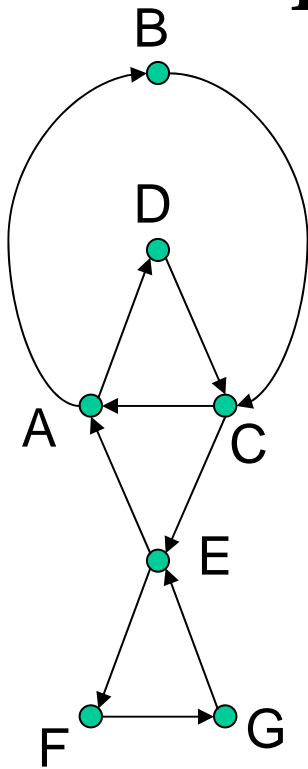
Euler circuit: return to the starting
point by traveling each edge of the
graph once and only once.

Is there a solution to the Königsberg bridge walking problem?

Euler's theorem:

- (a) If a graph has any vertices of odd degree, it cannot have an Euler circuit.
- (b) If a graph is connected and every vertex has an even degree, it has at least one Euler circuit.

Euler circuits in directed graphs

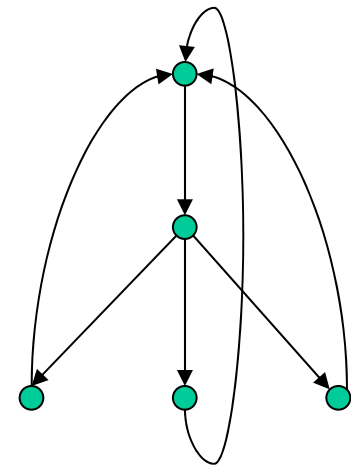


If a digraph is strongly connected and the in-degree of each node is equal to its out-degree, then there is an Euler circuit

Q: Give one possible Euler circuit

Otherwise there is no Euler circuit.

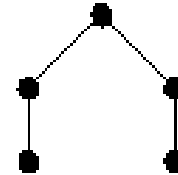
This is because in a circuit we need to enter each node as many times as we leave it.



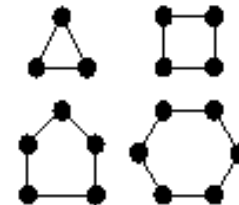
Common subgraphs

Subgraph: a subset of nodes of the original graph and of edges connecting them. Does not have to contain all the edges of a node included in the subgraph.

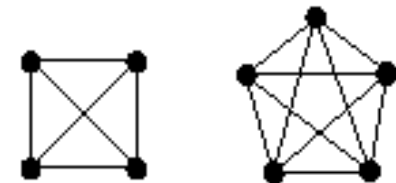
Trees: contain no circuits; N nodes and $N-1$ edges.



Cycles: circuits where nodes are not revisited;
 N nodes and N edges

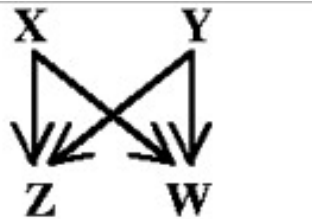


Cliques: completely connected subgraphs; N nodes
and $N(N-1)/2$ edges

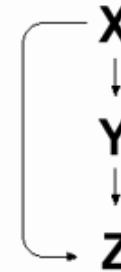


Special directed subgraphs

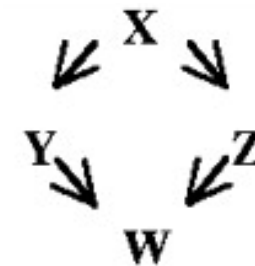
Bi-fan



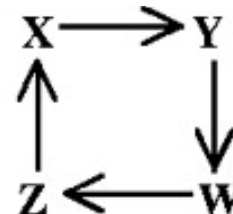
Feed-forward loop: two nonintersecting directed paths between a start and endpoint



Bi-parallel: two nonintersecting paths of identical length between a start and endpoint



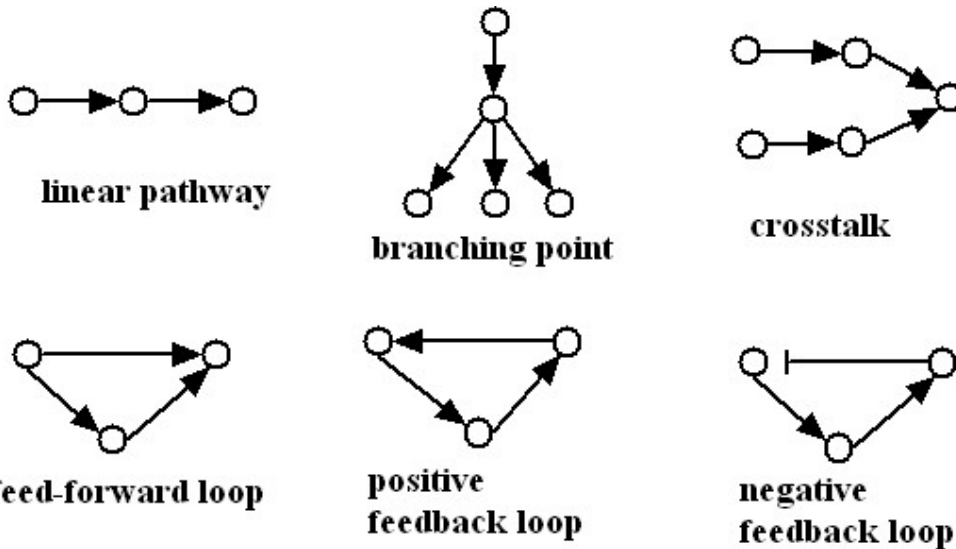
Feed-back loop: a directed cycle



S. S. Shen-Orr, R. Milo, S. Mangan and U. Alon, Nature Genetics (2002)

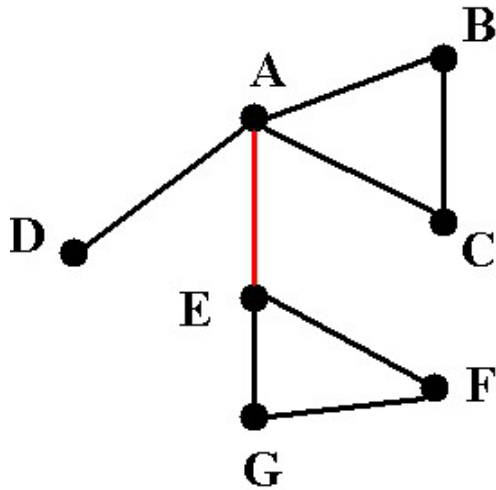
Network topology and dynamics

Network motifs can illustrate regulatory relationships



Further, dynamic details are needed to describe how multiple inputs on a node are integrated - additive action (e.g. same product for two chemical reactions)
- synergy (e.g. transcriptional regulation)

Distances between nodes

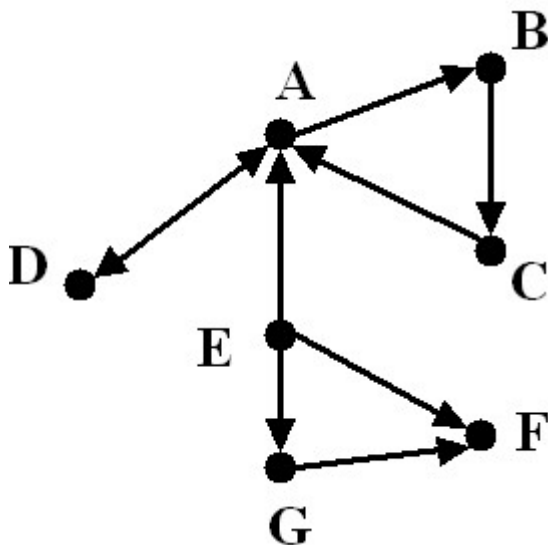


The distance between two nodes is defined as the number of edges along the shortest path connecting them.

If the two nodes are disconnected, the distance is infinity.

In **digraphs** each path needs to follow the direction of the arrows.

Thus in a digraph the distance from node A to B (on an AB path) is generally different from the distance from node B to A (on a BA path).



Average distance (or path length)

We can define an average path length or distance for a **connected graph** (component) or a **strongly connected** (component of a) **digraph**.

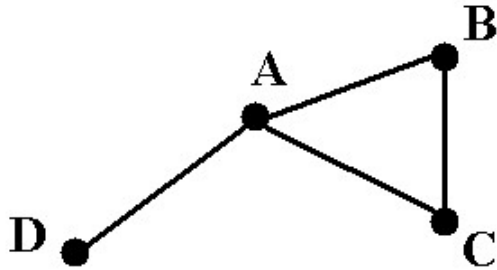
In general $\langle l \rangle \equiv \frac{1}{2N_{pairs}} \sum_{i,j \neq i} l_{ij}$, where l_{ij} is the distance from node i to node j .

N_{pairs} is the number of node pairs $N_{pairs} = \binom{N}{2} = \frac{N(N-1)}{2}$

Since in a (symmetrical) graph $l_{ij} = l_{ji}$, we only need to count them

once $\langle l \rangle \equiv \frac{1}{N_{pairs}} \sum_{i,j > i} l_{ij}$

Graph efficiency



To avoid infinities, one can define a graph efficiency (= average inverse distance)

$$\eta \equiv \frac{1}{2N_{pairs}} \sum_{i,j \neq i} \frac{1}{l_{ij}}$$

N_{pairs} is the number of node pairs

Q: What is the average distance and efficiency of the graph on the left?

Tip: fill out the matrix

	A	B	C	D
A	--	l_{AB}	l_{AC}	l_{AD}
B	l_{BA}	--	l_{BC}	l_{BD}
C	l_{CA}	l_{CB}	--	l_{CD}
D	l_{DA}	l_{DB}	l_{DC}	--

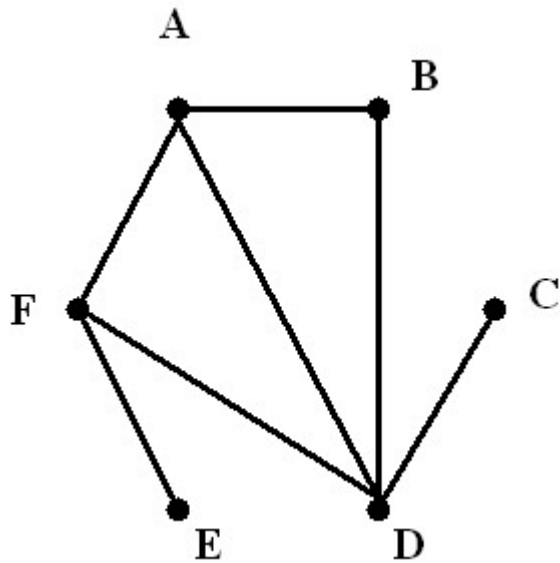
Betweenness centrality (load)



- Find all the shortest paths between nodes i and j - $C(i,j)$
- Determine how many of these pass through node k - $C_k(i,j)$
- The betweenness centrality of node k is

$$g_k = \sum_{i \neq j} \frac{C_k(i,j)}{C(i,j)}$$

L. C. Freeman, *Sociometry* 40, 35 (1977)



$$g_k = \sum_{i \neq j} \frac{C_k(i, j)}{C(i, j)}$$

$C(i, j)$ – nr. of shortest paths btw. i, j

$C_k(i, j)$ – nr. of these paths that contain k

Ex: Calculate the betweenness centrality of the nodes in this graph.

Do not count being the starting or ending point of a path ($k \neq i, k \neq j$).

Weighted networks

In some applications it is necessary quantify edges with weights, corresponding, e.g., to a traversal cost or a geographical distance.

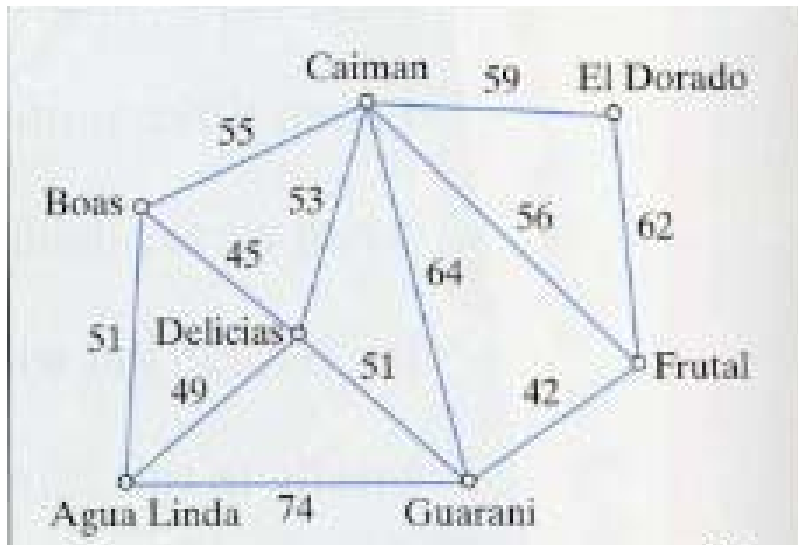
Then the shortest path between two nodes is redefined in terms of weight, e.g. “the path with lowest cost”, or “most efficient path”.

To find the distance between two nodes in a weighted network, calculate the sum of edge weights on each path (this is the path weight), then select the path with lowest weight

$$l_{ij} = w_{ik} + w_{kl} + \dots + w_{mj}$$

where $i k l..m j$ is the path with minimum weight
 w_{ij} is the weight of edge ij

Kruskal's algorithm to find the **minimum spanning tree** of a graph



- Find the cheapest edge in the graph and mark it.
- Continue selecting the cheapest remaining edge at each step, but **do not select edges that create circuits**.
- When the number of edges is one less than the number of vertices, **STOP**.

Ex: Delete the three highest-weight edges from the graph. Find the weighted distances between the nodes in the modified graph.

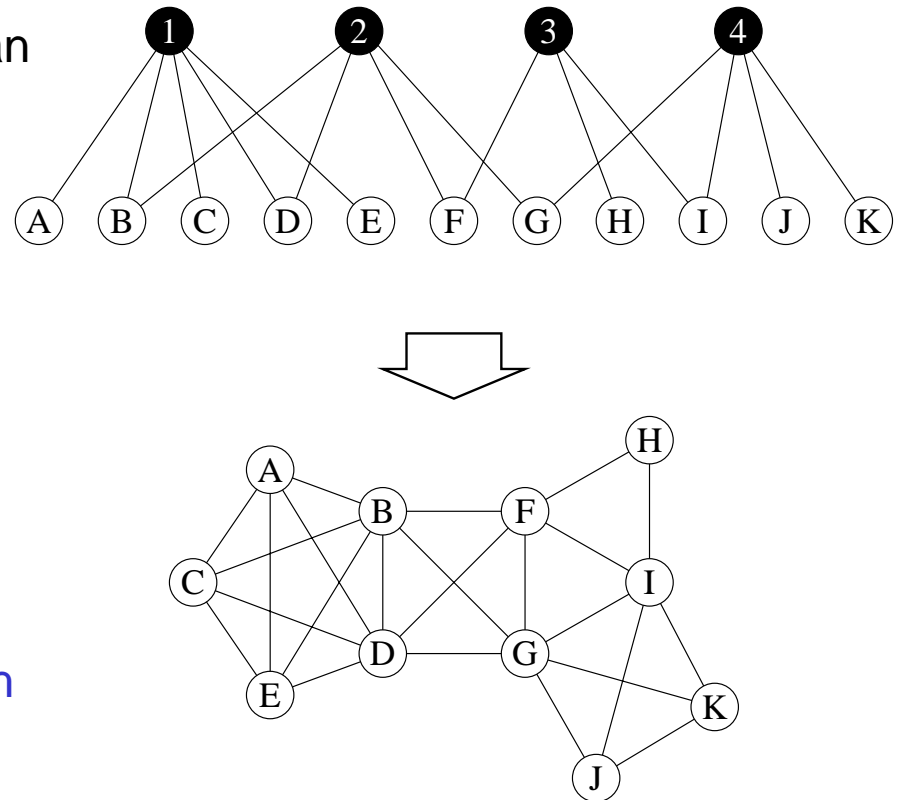
Bipartite graphs

Group structure (e.g. in social networks) can be incorporated into a bipartite graph.

A bipartite graph has two types of nodes:
group nodes (black, numbered)
member nodes (white, lettered)

Edges are possible only between different types of nodes: membership in group.

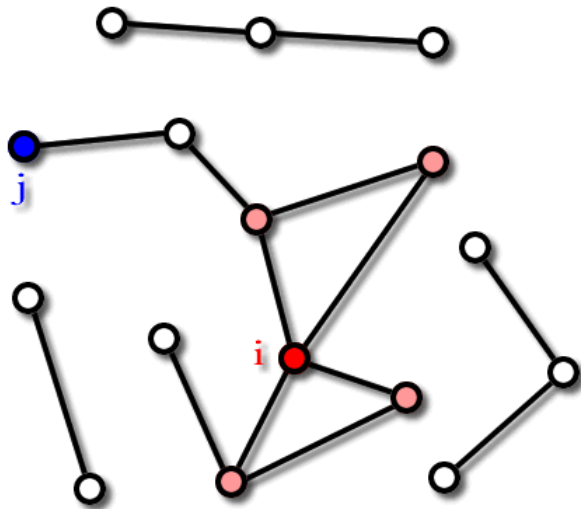
An alternative representation connects all members in a given group - **each group becomes a completely connected subgraph**



D. Watts, P. S. Dodds, M. E. J. Newman, Science 296 (2002)

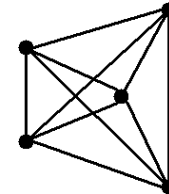
M. E. J. Newman, S. Strogatz, D. Watts, Phys. Rev. E 64, 026118 (2001)

Local order and clustering



Cliques (completely connected subgraphs)

$$k = N - 1, n = \frac{N(N - 1)}{2}$$



How close the neighborhood of a node is to a clique?

Edges among first neighbors of node i

Clustering coefficient

$$C_i \equiv \frac{n_i}{k_i(k_i - 1) / 2}, \quad k_i \neq 0, 1 \quad \text{or}$$

$$C_i \equiv \frac{\text{nr. of triangles connected to } i}{\text{nr. of triples centered on } i}$$

Average clustering coefficient

$$C \equiv \frac{1}{N} \sum_{i=1}^N C_i$$

Ex. 1

N nodes are connected by N edges such that they form a cycle.
How does the maximum distance between nodes (the diameter) depend on N ? How about the average distance?

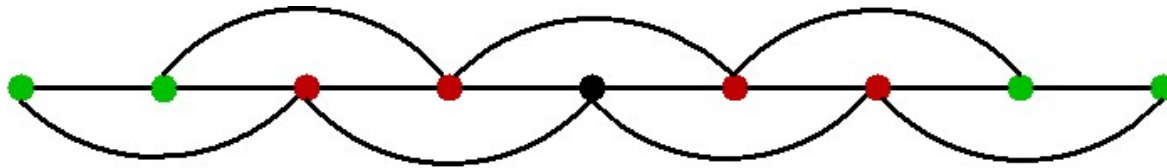
Ex. 2

On the ring lattice from above every second neighbor is connected by an edge. What is the clustering coefficient of the nodes?

Ex. 3

Construct a square lattice (grid) L edges long. How does the maximum distance between nodes depend on L ?

Regular lattices, ex. 1, 1D lattice (ring)



$k = 4$ for inside nodes

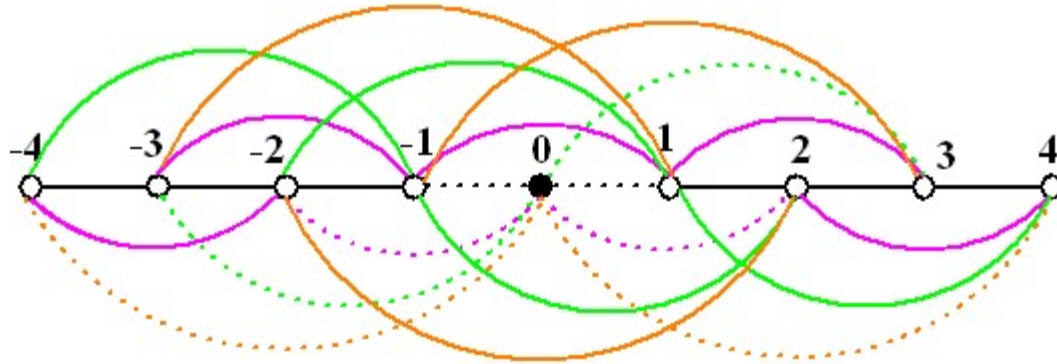
$C = \frac{1}{2}$ for inside nodes

$$1 + \sum_{l=1}^{l_{max}} 4 \approx N \Rightarrow l_{max} \approx \frac{N}{4} \quad \langle l \rangle = \frac{4 \sum_{l=1}^{l_{max}} l}{N} \Rightarrow \langle l \rangle \approx \frac{N}{8}$$

The average path-length varies as $\langle l \rangle \approx N$

Constant degree, constant clustering coefficient.

Clustering coefficient of 1D lattice



The origin (black) node is connected to k ($=4$) nodes on each side.

Total possible edges among neighbors: $2k(2k-1)/2$

Enumerate edges that are actually there:

$$n = [2k - 2] + [2k - 1 - 2] + [2k - 2 - 2] + \dots + [2k - (i - 1) - 2] + \dots + [k - 1]$$

First neighbors
on lattice

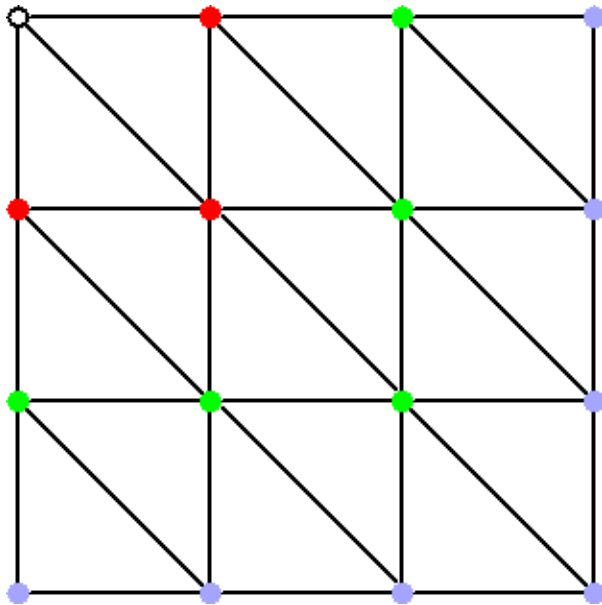
second neighbors

edge "length"

k th neighbors

$$C = \frac{2 \sum_{i=1}^k (2k - 1 - i)}{2k(2k - 1)} = \frac{3}{2} \frac{k - 1}{2k - 1}$$

Regular lattices, ex. 2, 2D lattice



$k = 6$ for inside nodes

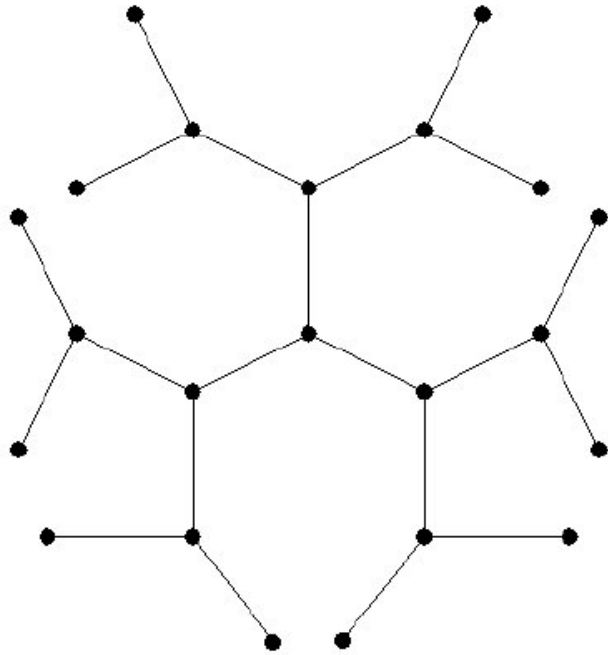
$C = \frac{6}{15}$ for inside nodes

$$1 + \sum_{l=1}^{l_{max}} 6l \approx N \Rightarrow l_{max} \propto N^{0.5}$$

$$\langle l \rangle \approx L \approx N^{1/2}$$

In general, the average distance varies as $\langle l \rangle \approx N^{1/D}$ where D is the dimensionality of the lattice. Constant degree (coordination number), constant clustering coefficient.

Regular lattices, ex. 3, the Cayley tree



$k = 3$ for inside nodes
 $k = 1$ for surface nodes

$$\langle k \rangle = 2$$

$$C = 0$$

$$1 + 3 \sum_{l=1}^{l_{max}} 2^{l-1} \approx N \Rightarrow l_{max} \propto \frac{\log N}{\log \langle k \rangle}$$

$$\langle l \rangle \approx \frac{\log N}{\log \langle k \rangle}$$

Distances vary logarithmically with N . Constant degree, no clustering.