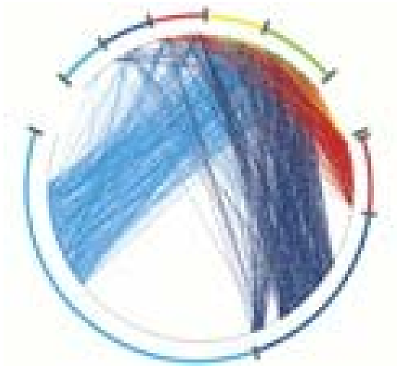
Lecture 2: Network Metrics



Understanding large graphs

- What are the statistics of real life networks?
- In which terms we can describe the networks?
- How we can measure a large network?
- Can we explain how the networks were generated?
- Can we make models for network construction?
- To how much extent do the artificially constructed networks describe real networks?

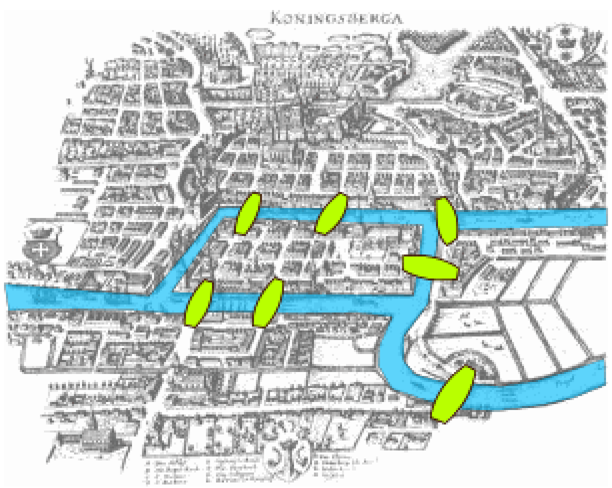
First step: Introducing network metrics

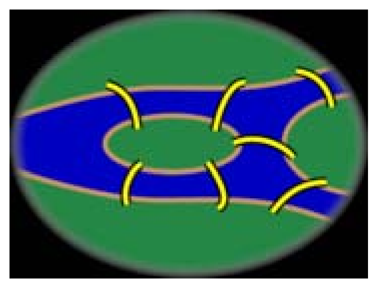
Networks became hot topic !

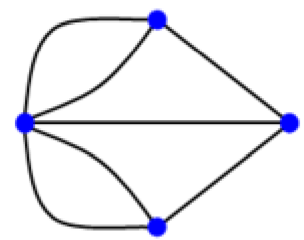
- Around 1999
  - Watts and Strogatz, Collective dynamics of small-world networks
  - Faloutsos^3, On power-law relationships of the Internet Topology
  - Kleinberg et al., The Web as a graph
  - Barabasi and Albert, The emergence of scaling in real networks

History: graph theory

- Euler's **Seven Bridges of Königsberg** – one of the first problems in graph theory
- Is there a route that crosses each bridge only once and returns to the starting point?

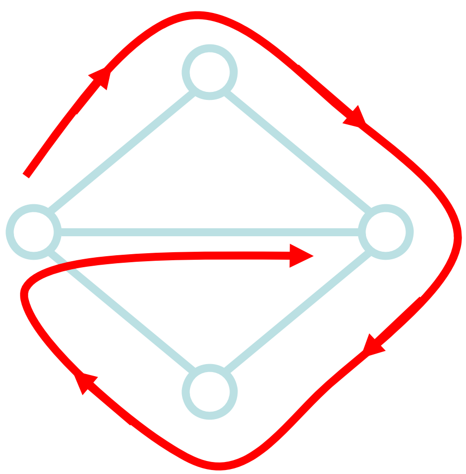




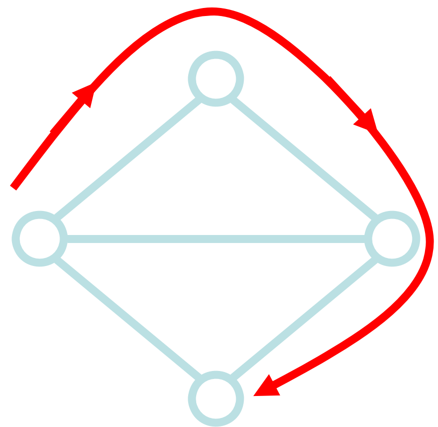


Eulerian paths

- If starting point and end point are the same:
  - only possible if no nodes have an odd degree
- If don't need to return to starting point
  - can have 0 or 2 nodes with an odd degree



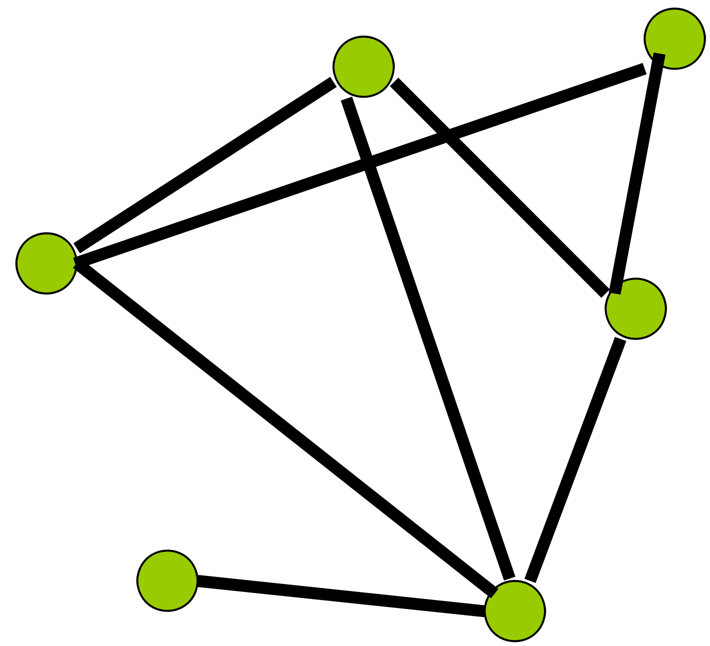
Eulerian path: traverse each edge exactly once



Hamiltonian path: visit each vertex exactly once

Graph Theory

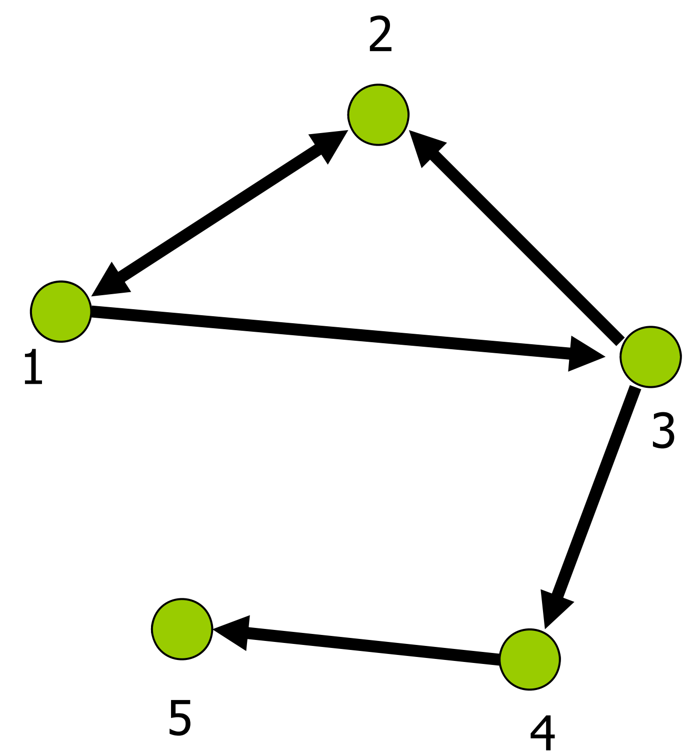
- Graph G=(V,E)
  - V = set of vertices
  - E = set of edges



undirected graph
E={(1,2),(1,3),(2,3),(3,4),(4,5)}

Graph Theory

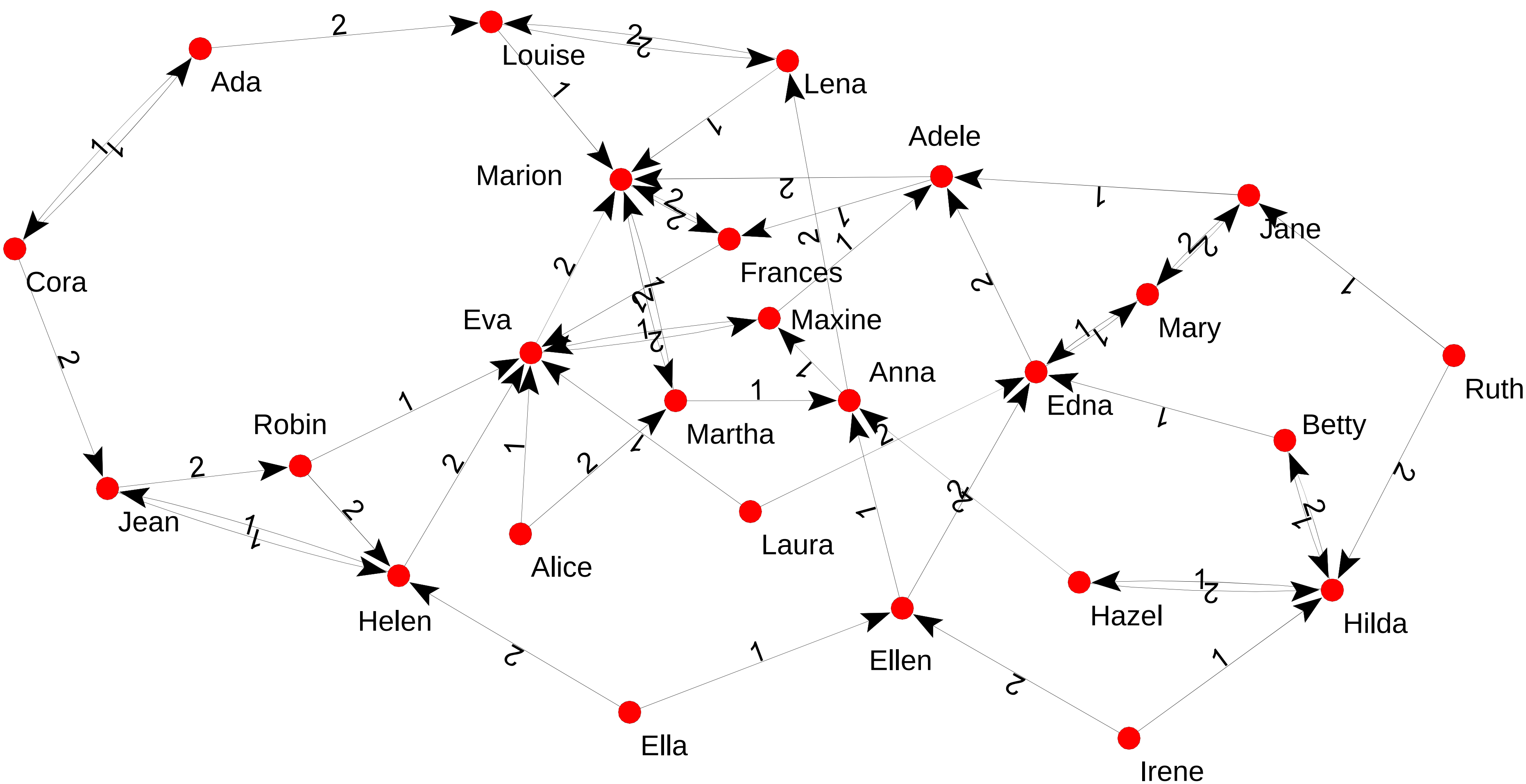
- Graph G=(V,E)
  - V = set of vertices
  - E = set of edges



directed graph
E={‹1,2›, ‹2,1› ‹1,3›, ‹3,2›, ‹3,4›, ‹4,5›}

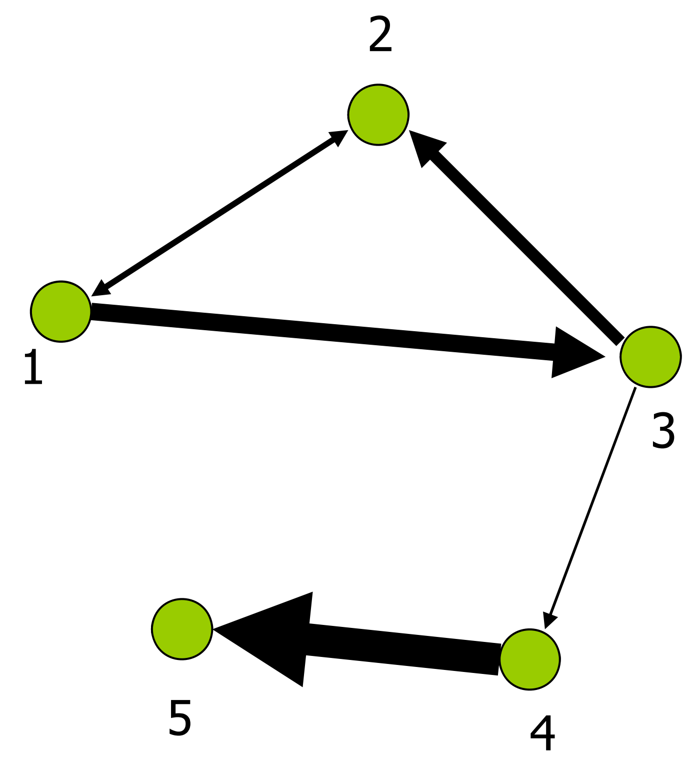
Graph Theory

- girls' school dormitory dining-table partners (Moreno, *The sociometry reader*, 1960)
- first and second choices shown



Graph Theory

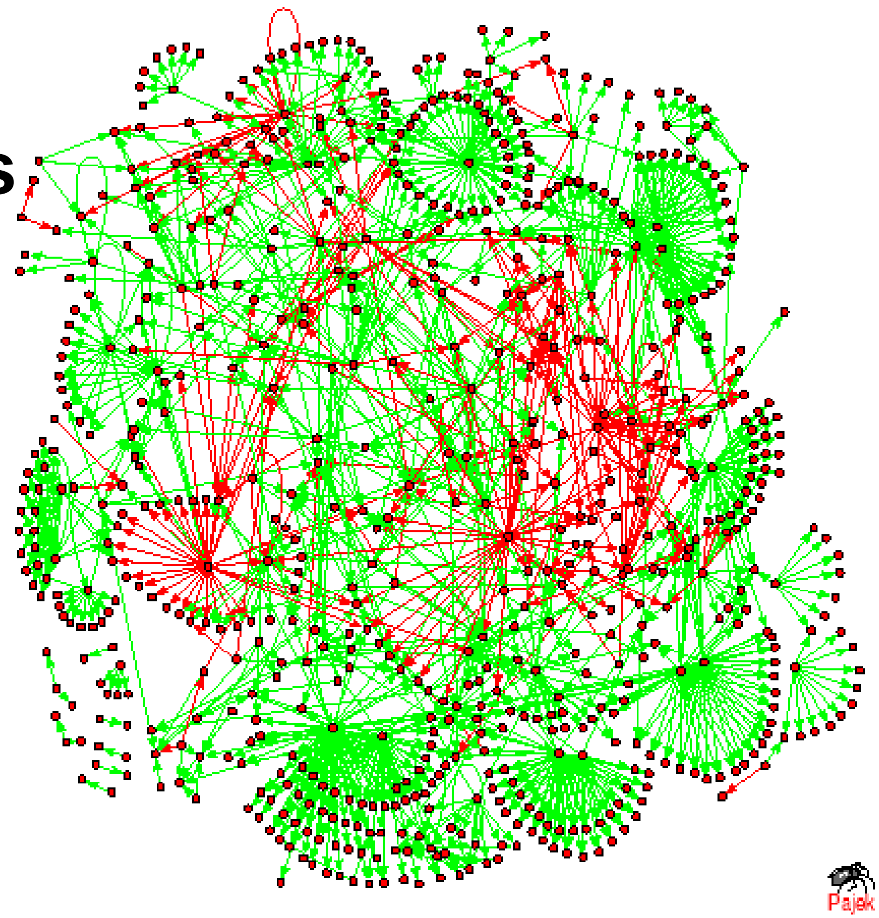
- Graphs might be weighted and/or directed
- Width of each edge (link) proportional to its weight



Graph Theory

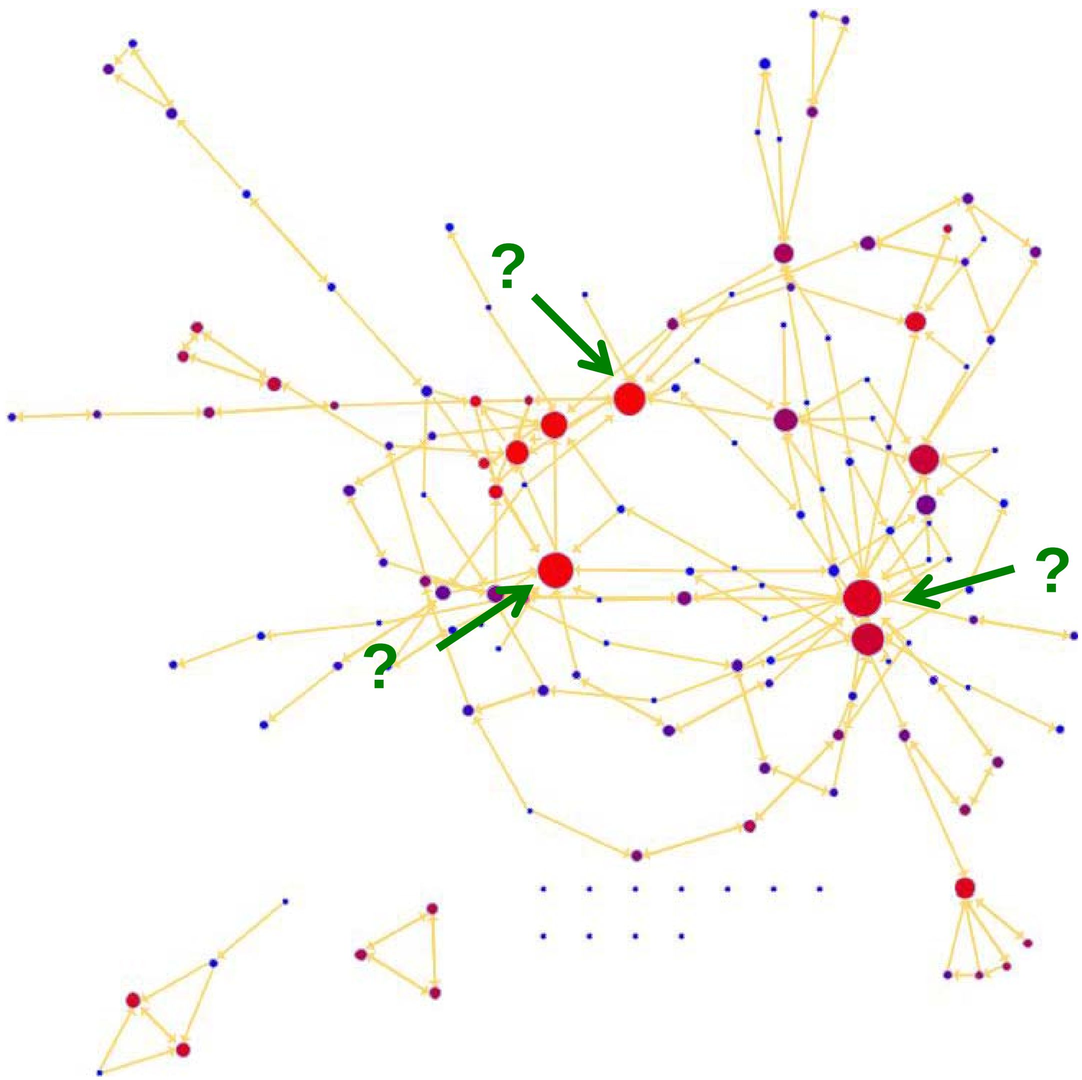
Edge weights can have positive or negative values

- One gene activates/inhibits another
- One person trusting/distrusting another
  - Research challenge: How does one 'propagate' negative feelings in a social network? Is my enemy's enemy my friend?



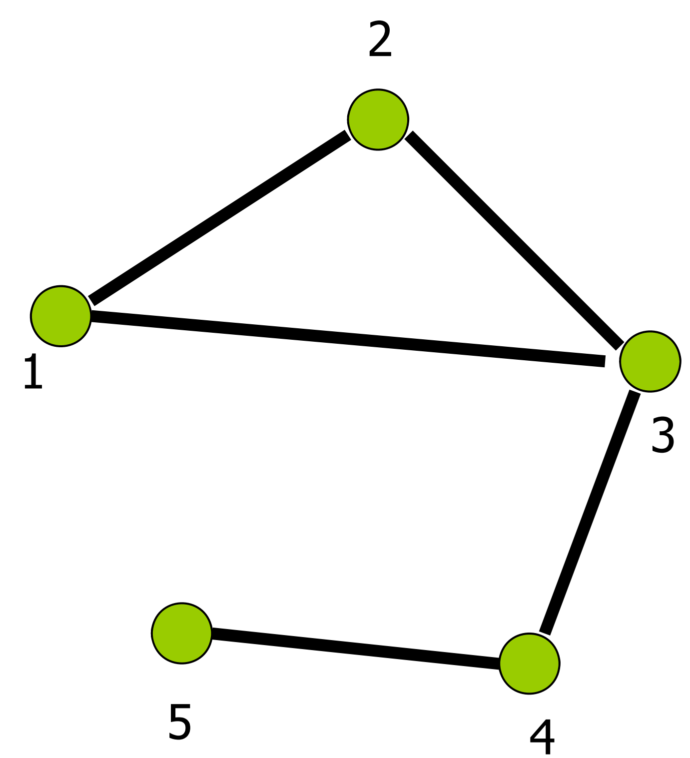
Transcription regulatory network in baker's yeast

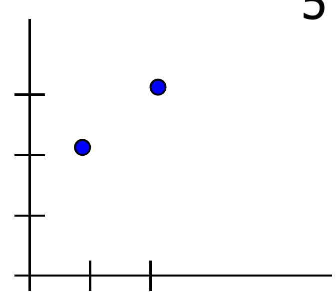
Who is most central?



Undirected graph

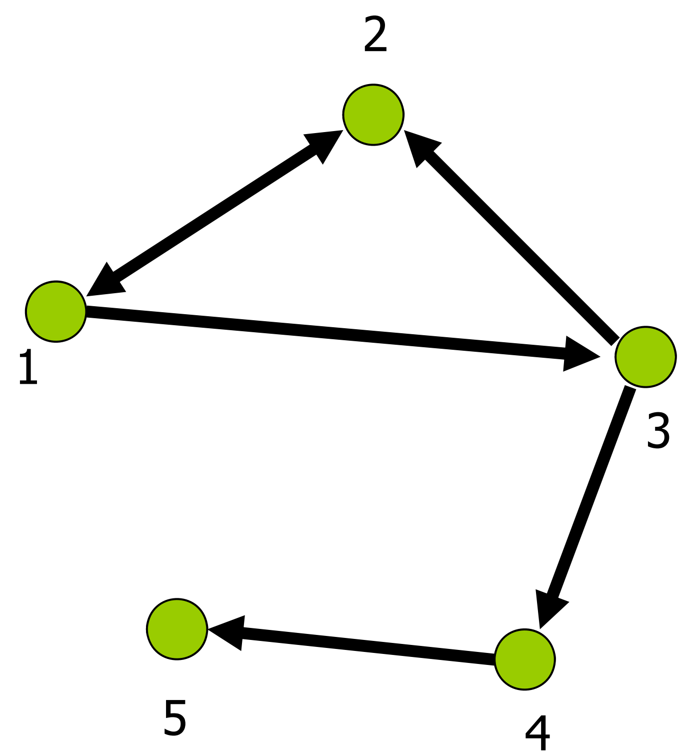
- degree k(i) of node i
  - number of edges incident on node i
- degree sequence
  - [k(1),k(2),k(3),k(4),k(5)]
  - [2,2,2,1,1]
- degree distribution
  - [(1,2),(2,3)]





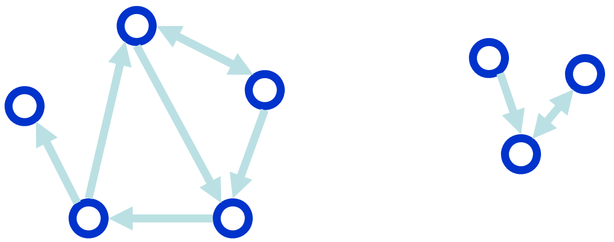
Directed Graph

- in-degree $k_{in}(i)$ of node i
  - number of edges pointing to node i
- out-degree $k_{out}(i)$ of node i
  - number of edges leaving node i
- in-degree sequence
  - [1,2,1,1,1]
- out-degree sequence
  - [2,1,2,1,0]

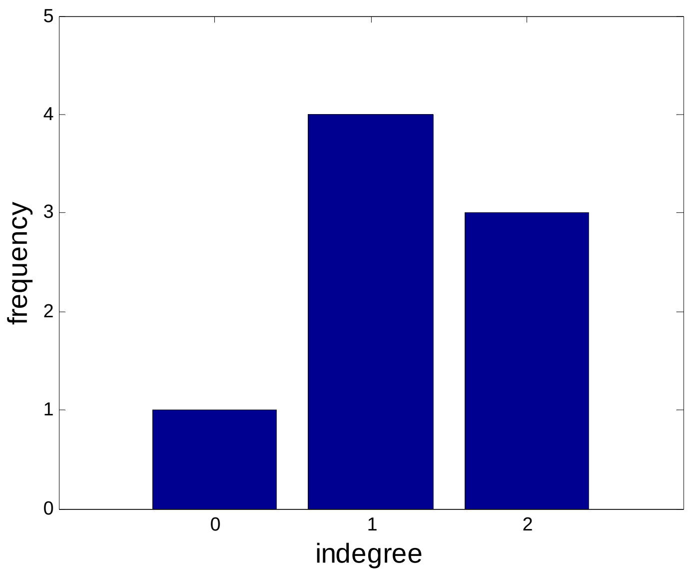


Another example for degree

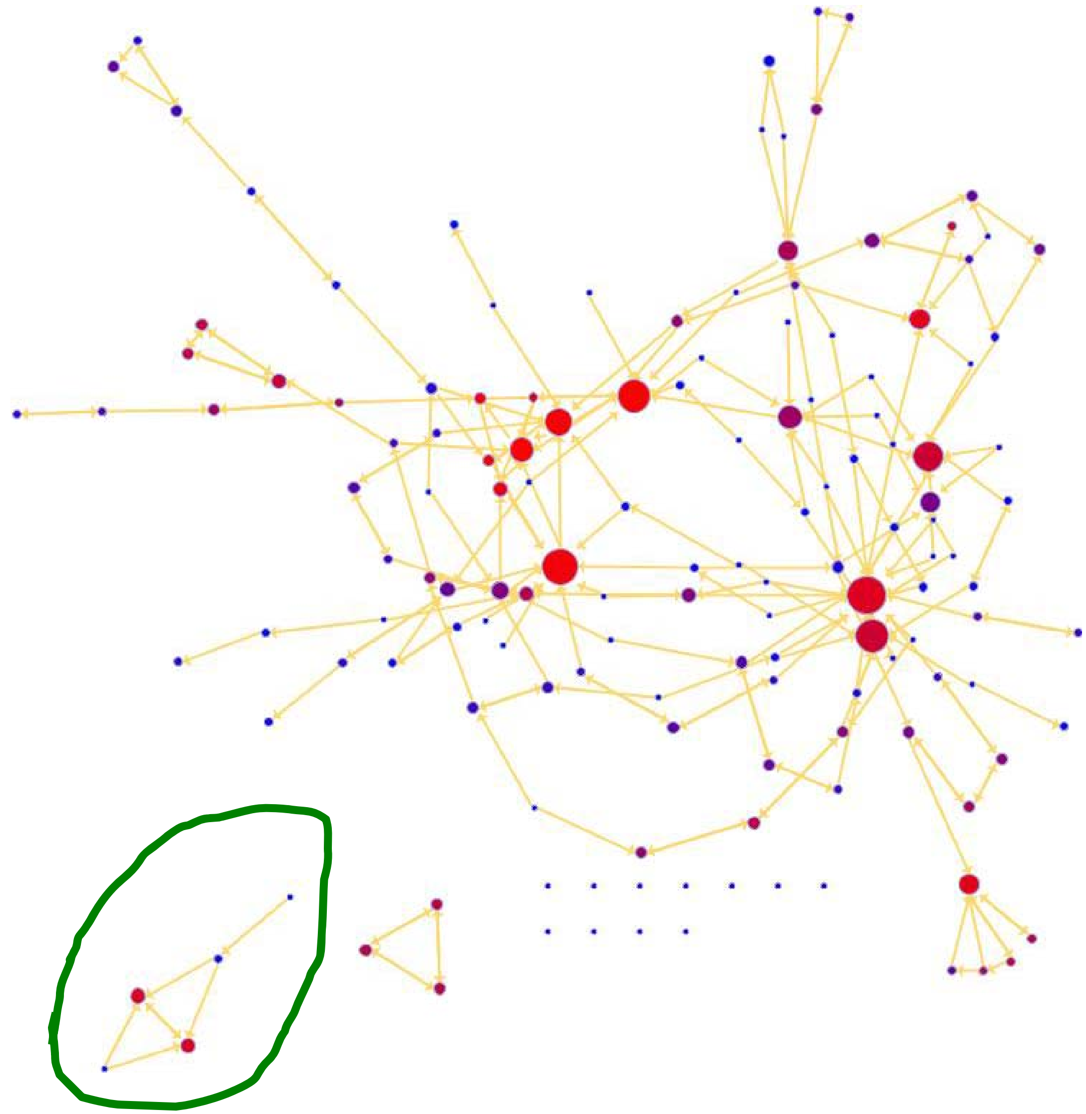
- Degree sequence: An ordered list of the (in,out) degree of each node
  - In-degree sequence:
    - [2, 2, 2, 1, 1, 1, 1, 0]
  - Out-degree sequence:
    - [2, 2, 2, 2, 1, 1, 1, 0]
  - (undirected) degree sequence:
    - [3, 3, 3, 2, 2, 1, 1, 1]



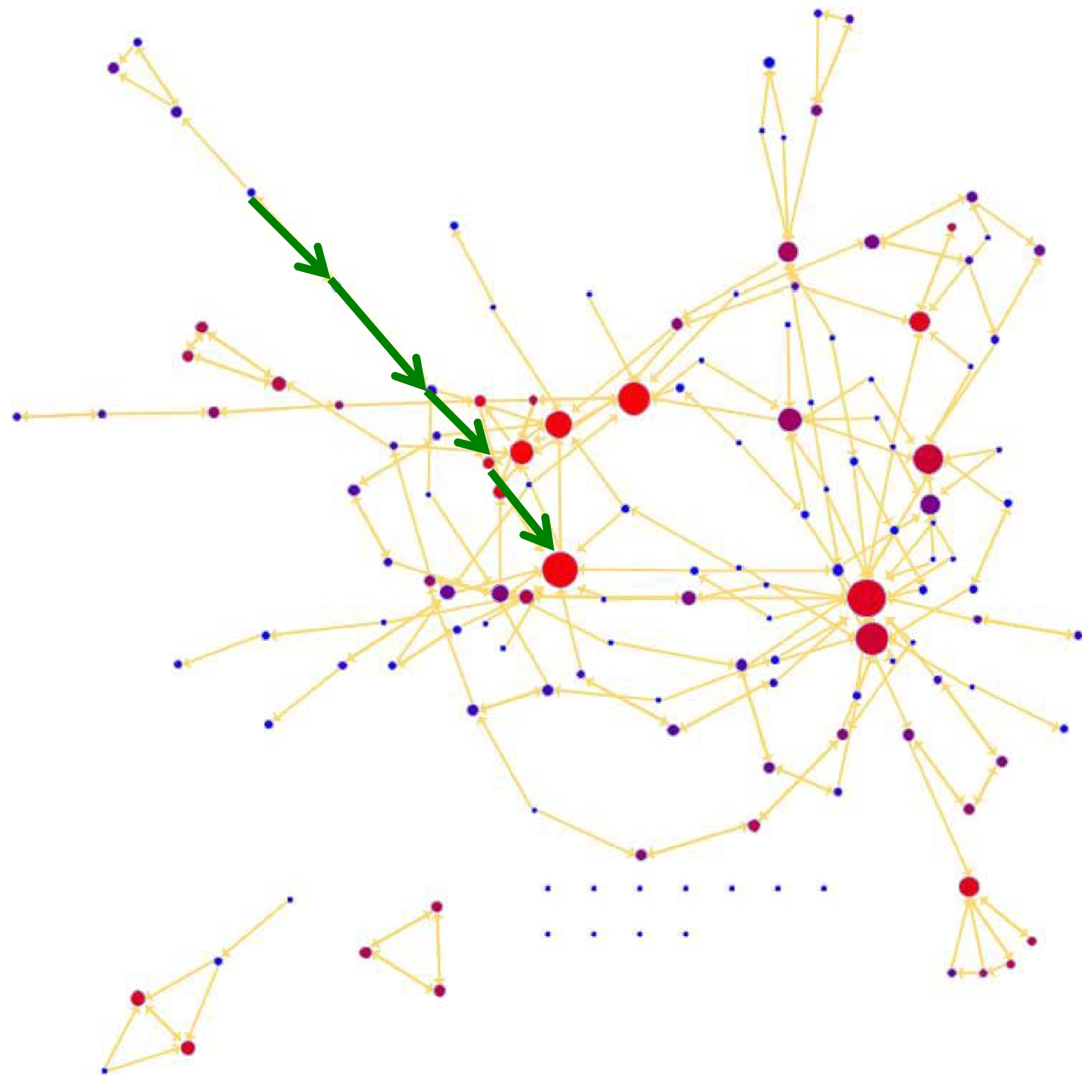
- Degree distribution: A frequency count of the occurrence of each degree
  - In-degree distribution:
    - [(2,3) (1,4) (0,1)]
  - Out-degree distribution:
    - [(2,4) (1,3) (0,1)]
  - (undirected) distribution:
    - [(3,3) (2,2) (1,3)]



Is everyone connected?

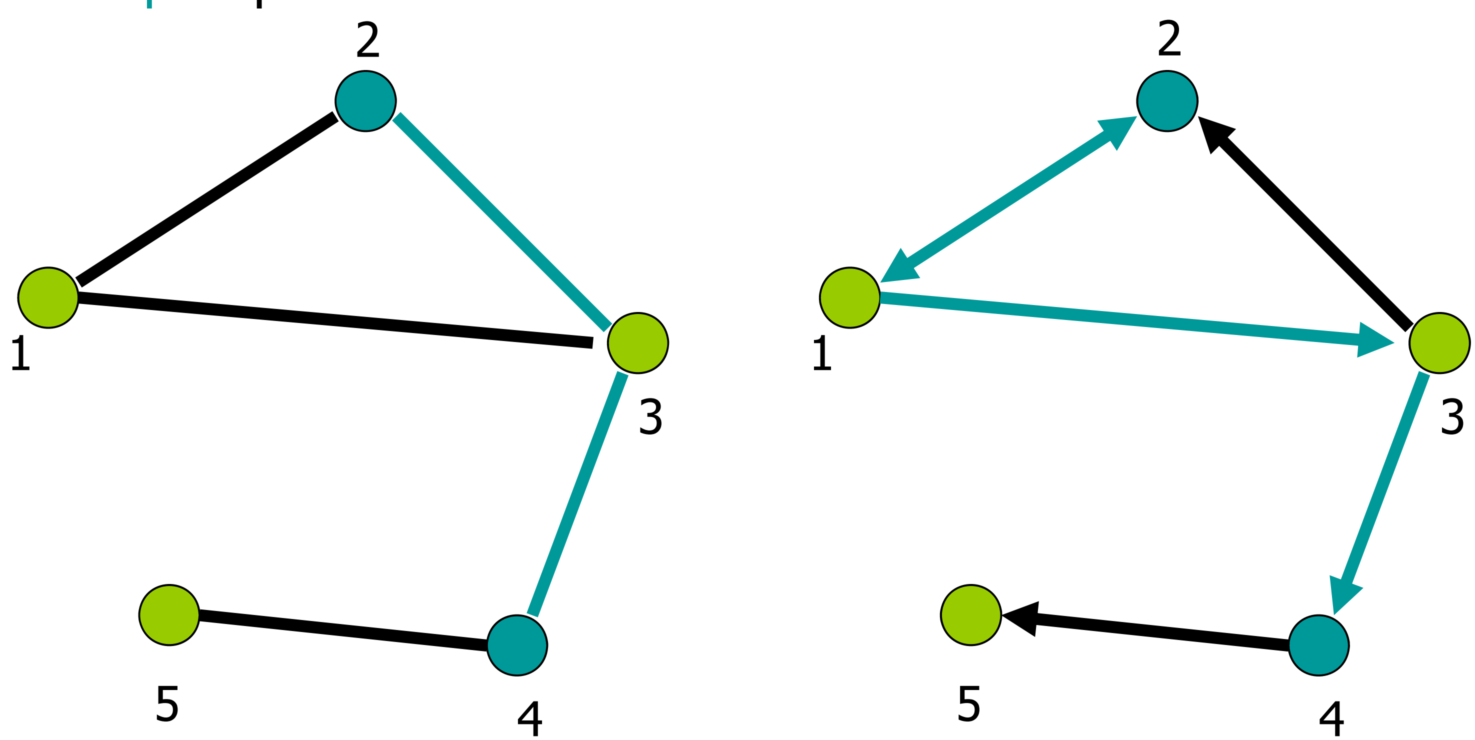


How far apart are nodes?



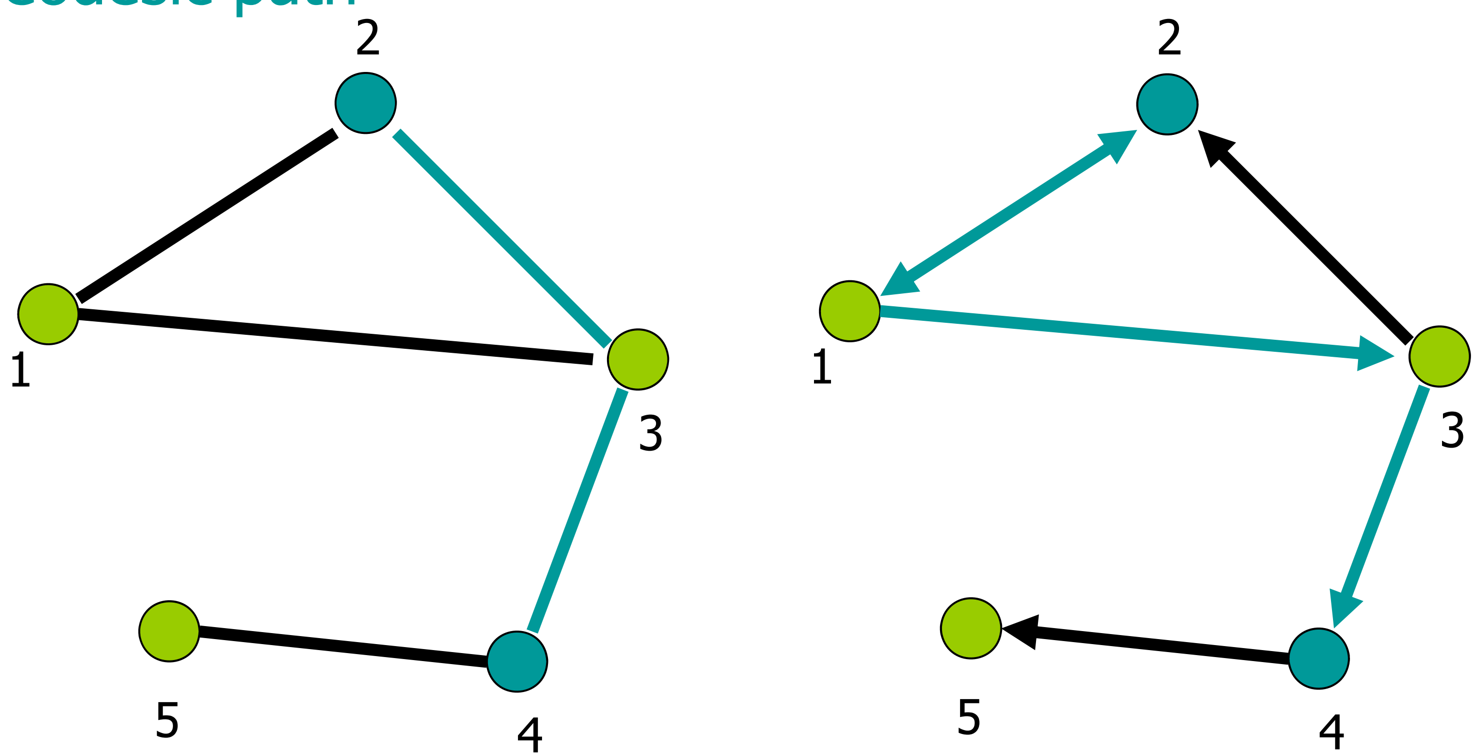
Paths

- Path from node i to node j: a sequence of edges (directed or undirected from node i to node j)
  - path length: number of edges on the path (unweighted networks)
  - nodes i and j are connected
  - Cycle (loop): a path that starts and ends at the same node
  - Self-loop: a path from a node to itself



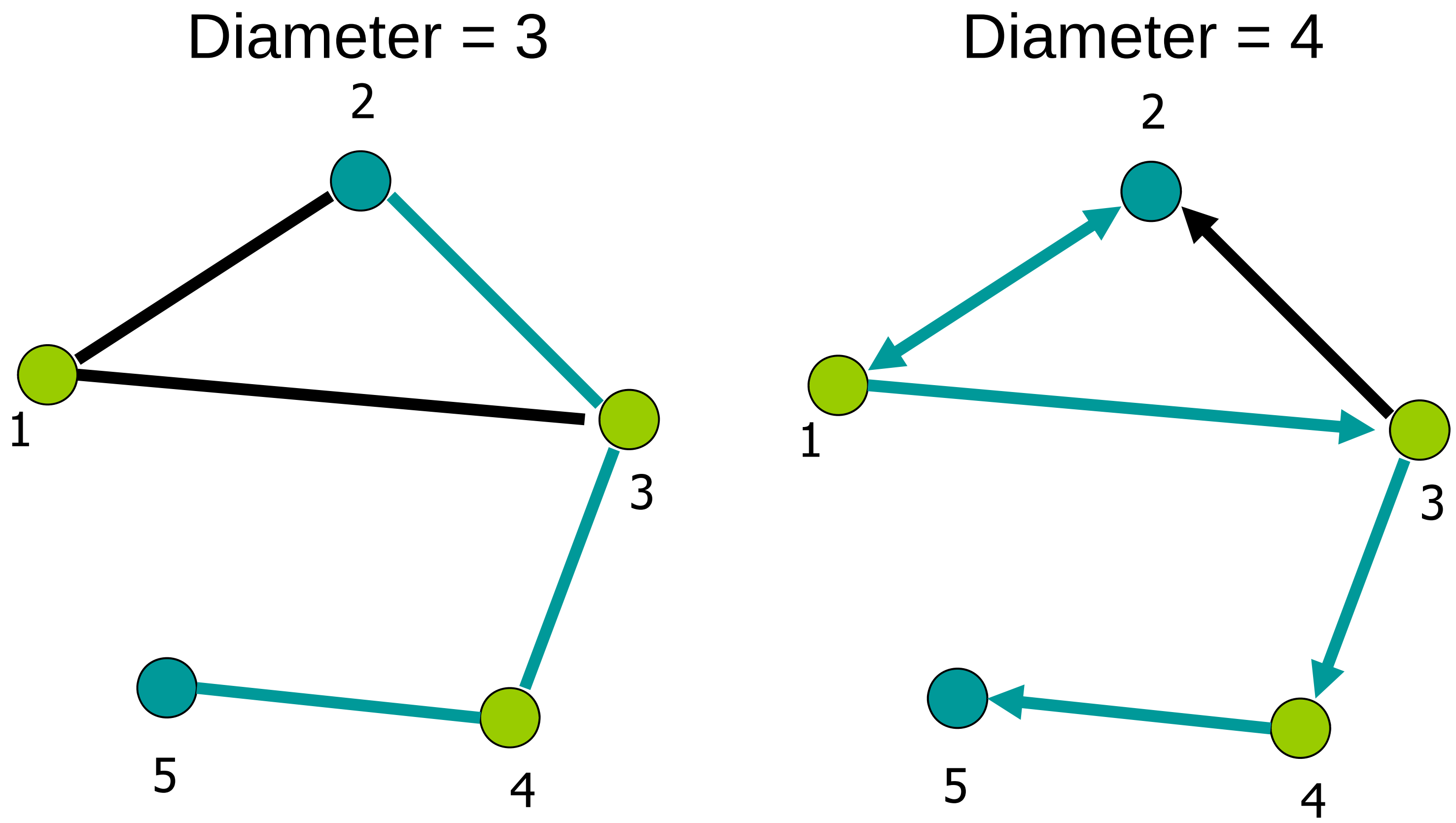
Shortest Paths

- Shortest Path from node i to node j (i and j are connected)
  - also known as BFS path, Characteristic path or geodesic path



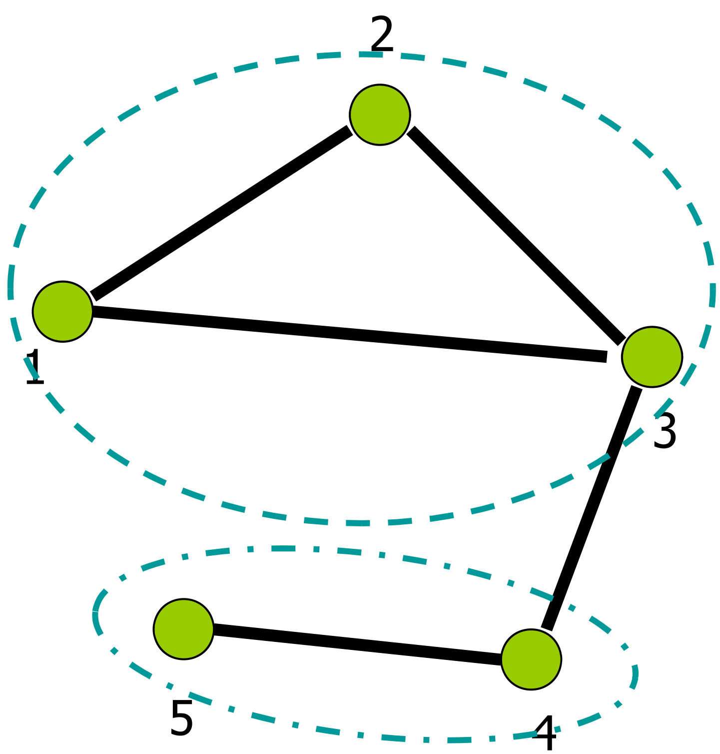
Diameter

- The longest shortest path in the graph



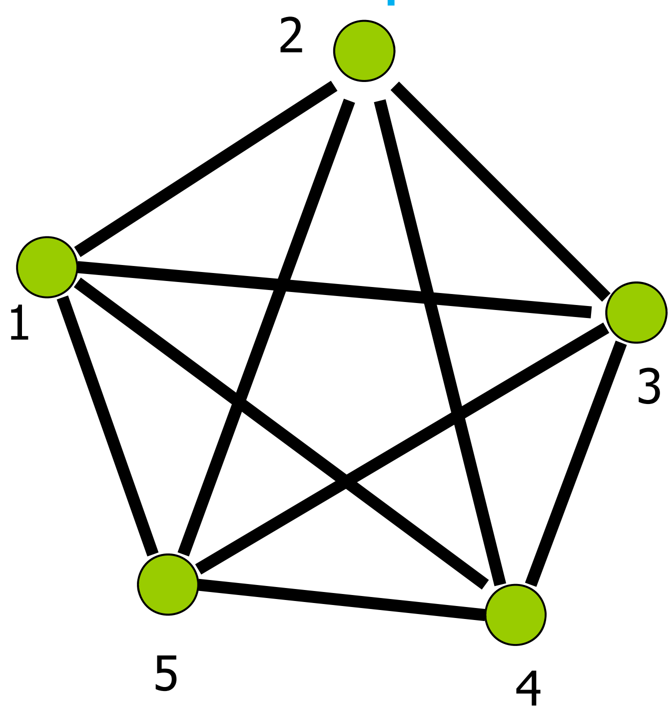
Undirected graph

- Connected graph: a graph where every pair of nodes is connected
- Disconnected graph: a graph that is not connected
- Connected Components: subsets of vertices that are connected
- Largest Connected Component: the connected component with the largest number of nodes



Fully Connected Graph

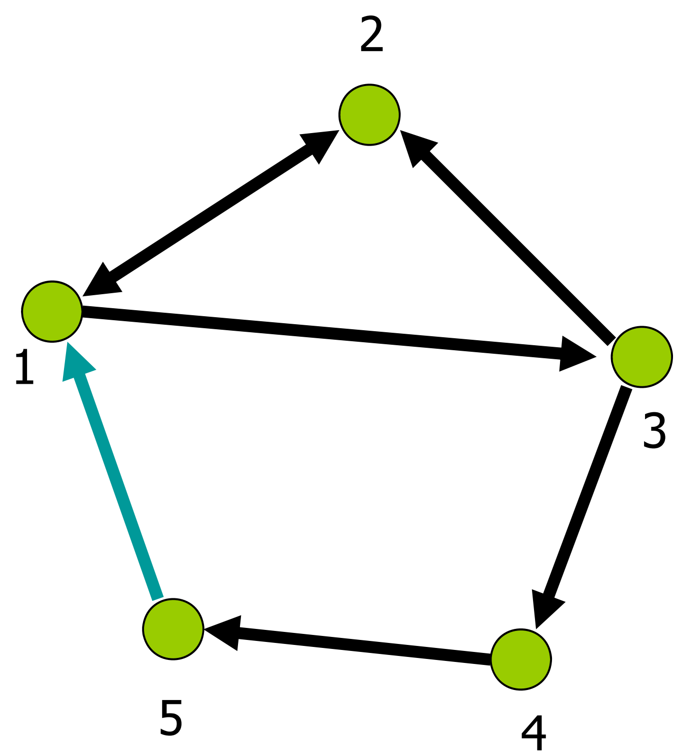
- Clique K_n
- A graph that has all possible n(n-1)/2 edges (n is the number of nodes)
- Sometimes called n-clique



A 5-clique graph

Directed Graph

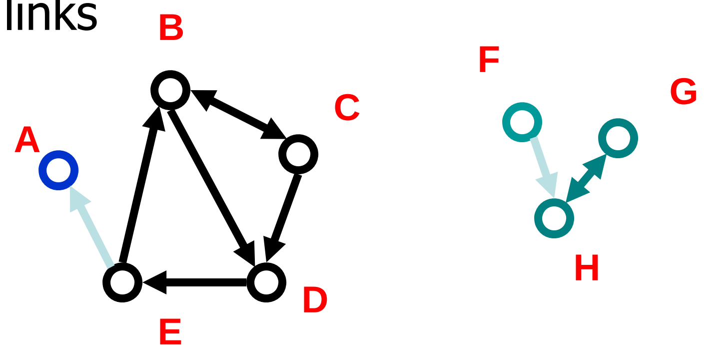
- Strongly connected graph: there exists a path from every i to every j
- Weakly connected graph: If edges are made to be undirected the graph is connected



- ❖ A graph is sparse if $| E | \approx | V |$
- ❖ A graph is dense if $| E | \approx | V |^2$

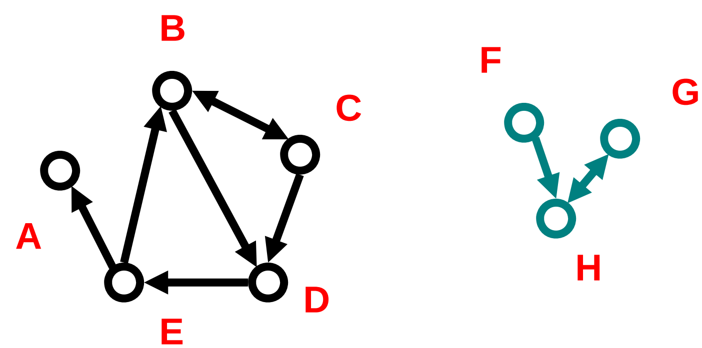
Connected Component

- Strongly connected components
  - Each node within the component can be reached from every other node in the component by following directed links



- Strongly connected components
  - B C D E
  - A
  - G H
  - F

- Weakly connected components: every node can be reached from every other node by following links in either direction

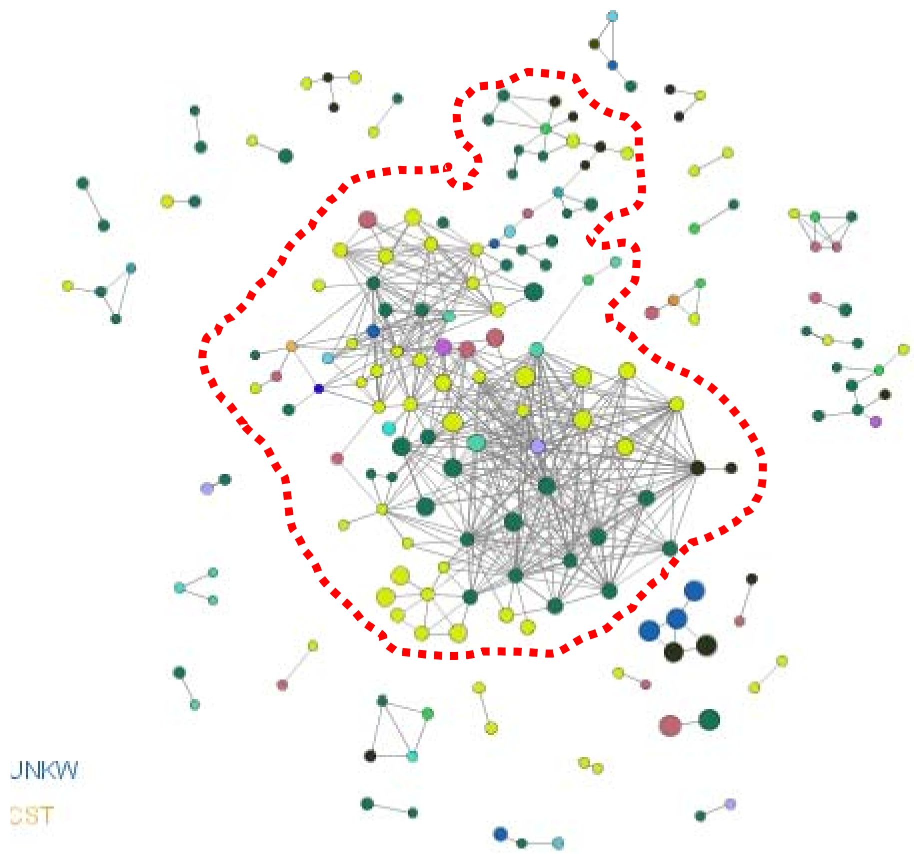


- Weakly connected components
  - A B C D E
  - G H F

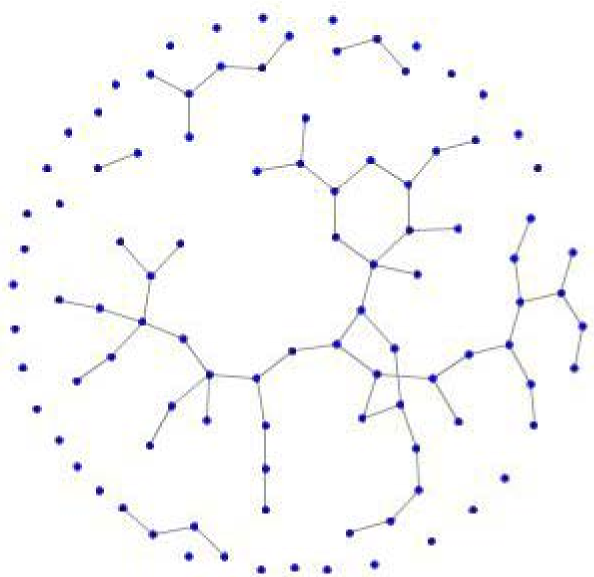
- In undirected networks one talks simply about 'connected components'

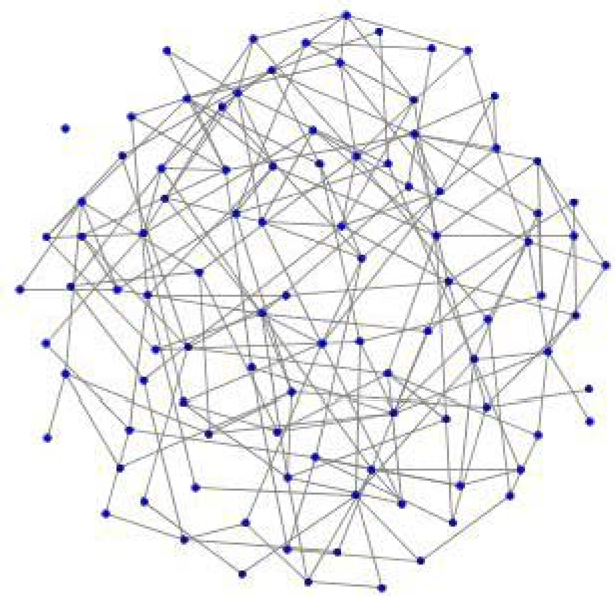
Largest Connected Component

if the largest component encompasses a significant fraction of the graph, it is called the **giant component** or **largest connected component**



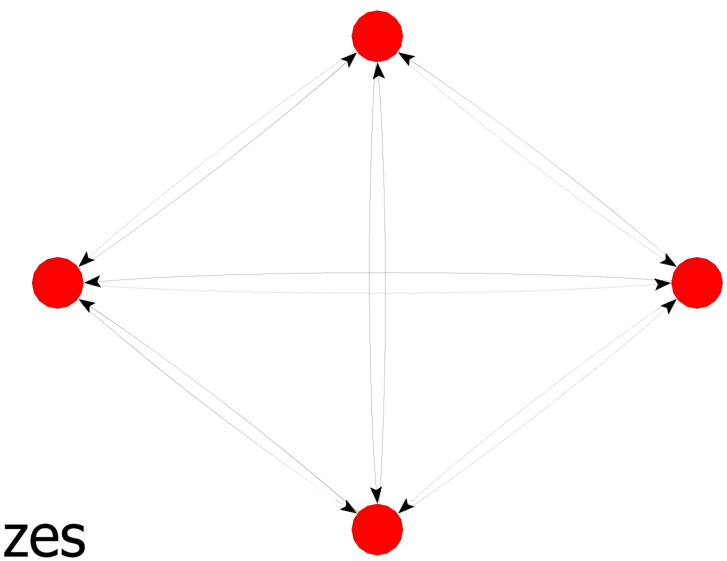
How dense the networks are?





Graph density

- Number of the connections that may exist between n nodes
  - directed graph
    $e_{max} = n*(n-1)$
    each of the n nodes can connect to (n-1) other nodes
  - undirected graph
    $e_{max} = n*(n-1)/2$
    since edges are undirected, count each one only once
- What fraction are present?
  - density = e / e_{max}
  - For example, out of 12 possible connections, this graph has 7, giving it a density of 7/12 = 0.583

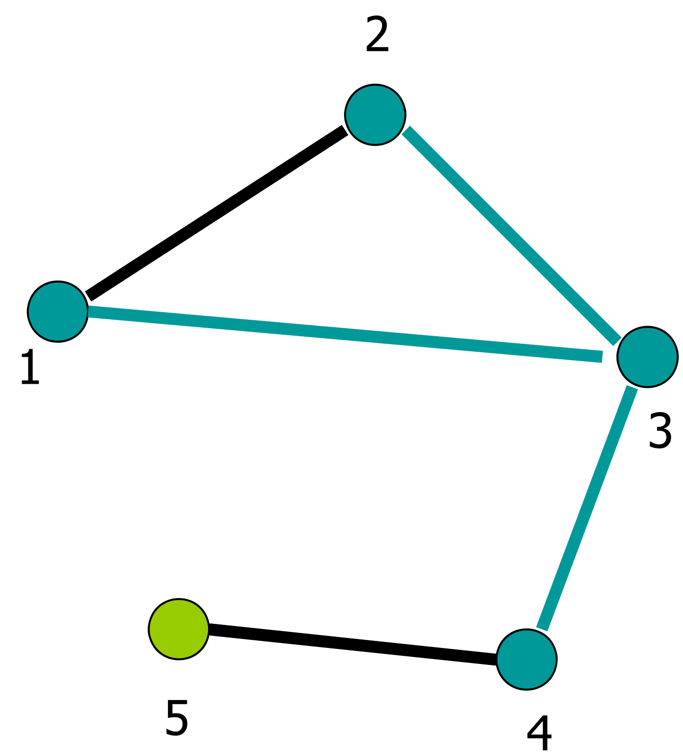


- Would this measure be useful for comparing networks of different sizes (different numbers of nodes)?

Subgraphs

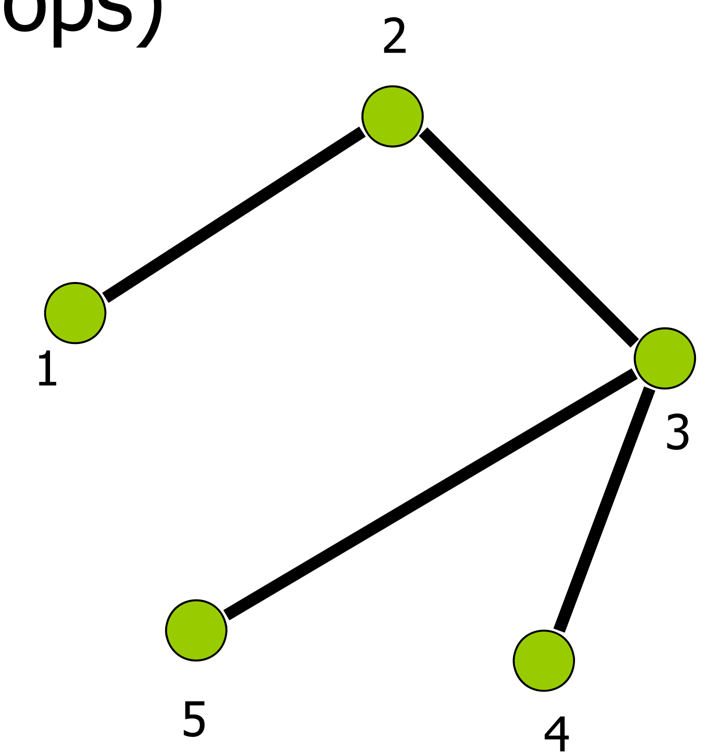
- Subgraph: Given V′ C V, and E′ C E, the graph G′=(V′,E′) is a subgraph of G.

- Induced subgraph: Given V′ C V, let E′ C E is the set of all edges between the nodes in V′. The graph G′=(V′,E′), is an induced subgraph of G



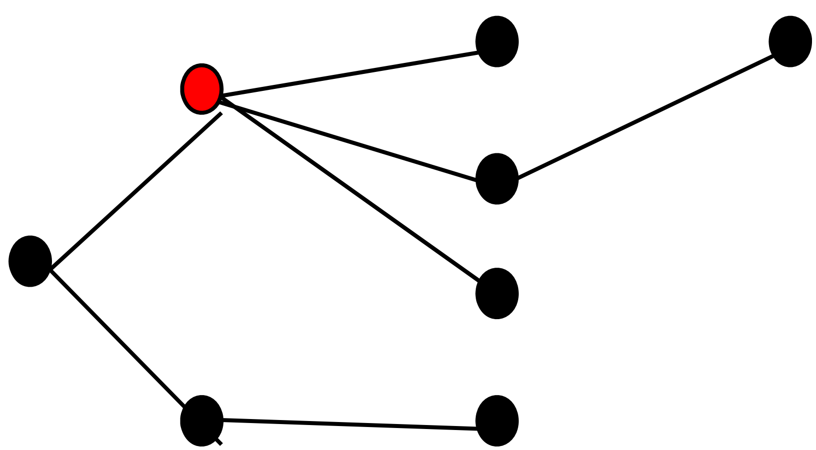
Trees

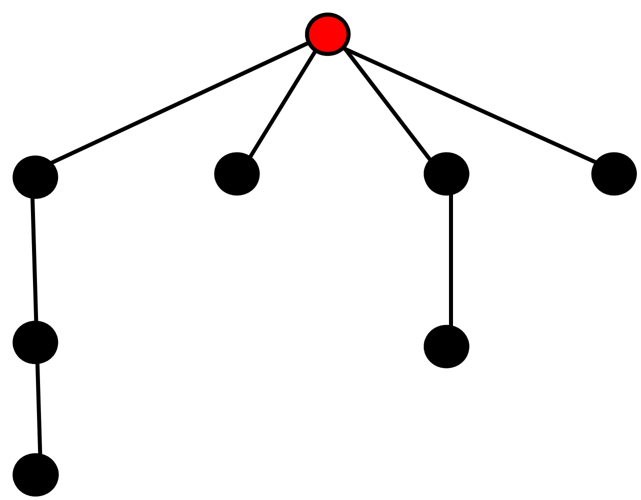
- Connected Undirected graphs without cycles (loops)



Rooted trees

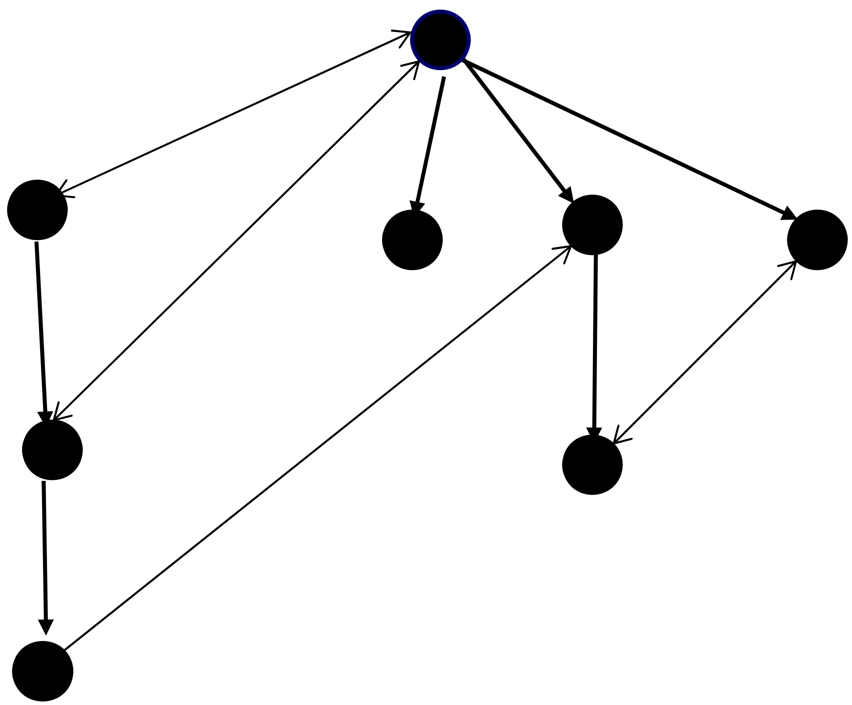
- Sometimes it is useful to distinguish one vertex of a tree and call it the *root* of the tree.
- For instance we might, for whatever reasons, take the tree below and declare the red vertex to be its root. In that case we often redraw the tree to let it all "hang down" from the root (or invert this picture so that it all "grows up" from the root, which suits the metaphor better)





Rooted directed trees

- It is sometimes useful to turn a rooted tree into a rooted directed tree T′ by directing every edge away from the root.



Rooted directed trees

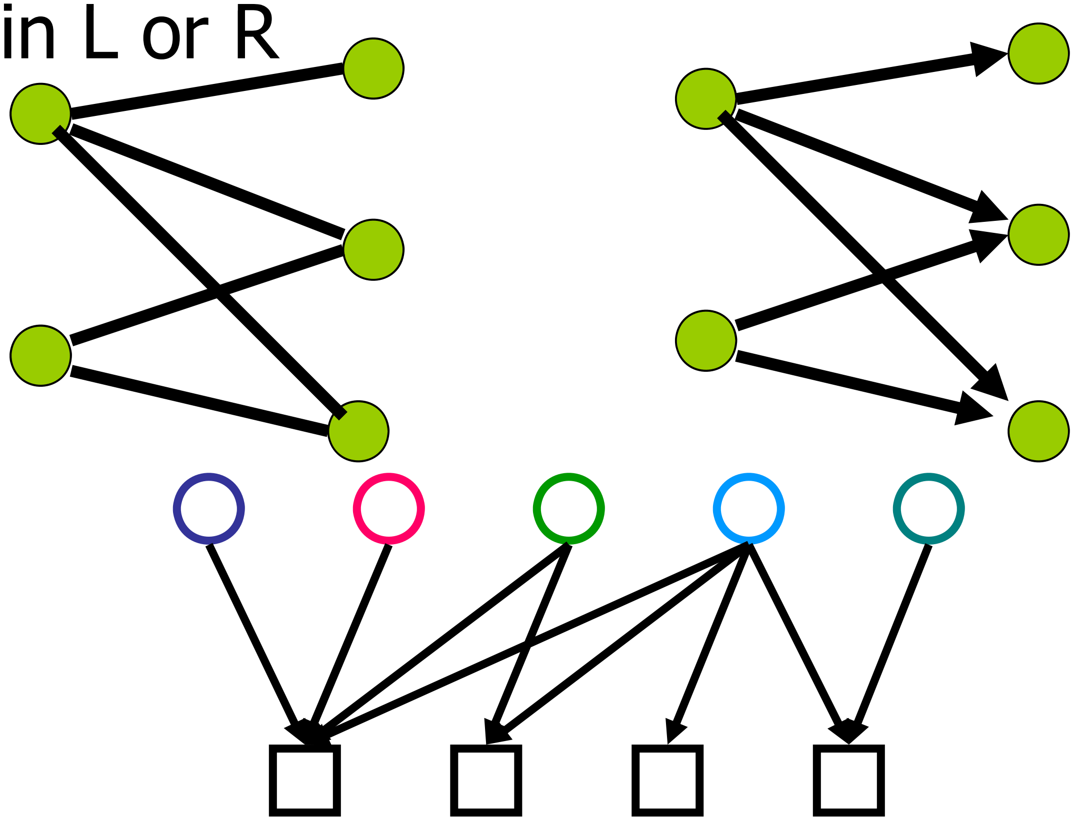
- Rooted trees and their derived rooted directed trees have some useful terminology, much of which is suggested by family trees. The *level* of a vertex is the length of the path from it to the root. The *height* of the tree is the length of the longest path from a leaf to the root. If there is a directed edge in T′ from a to b, then a is the *parent* of b and b is a *child* of a. If there are directed edges in T′ from a to b and c, then b and c are *siblings*. If there is a directed path from a to b, then a is an *ancestor* of b and b is a *descendant* of a.

Spanning tree of a graph

- If G(V,E) is a graph and T(V,F) is a subgraph of G and is a tree, then T is a *spanning tree* of G. That is, T is a tree that includes every vertex of G and has only edges to be found in G. Using a procedure (remove edges from cycles until only a tree remains), we can easily prove that every connected graph has a spanning tree.

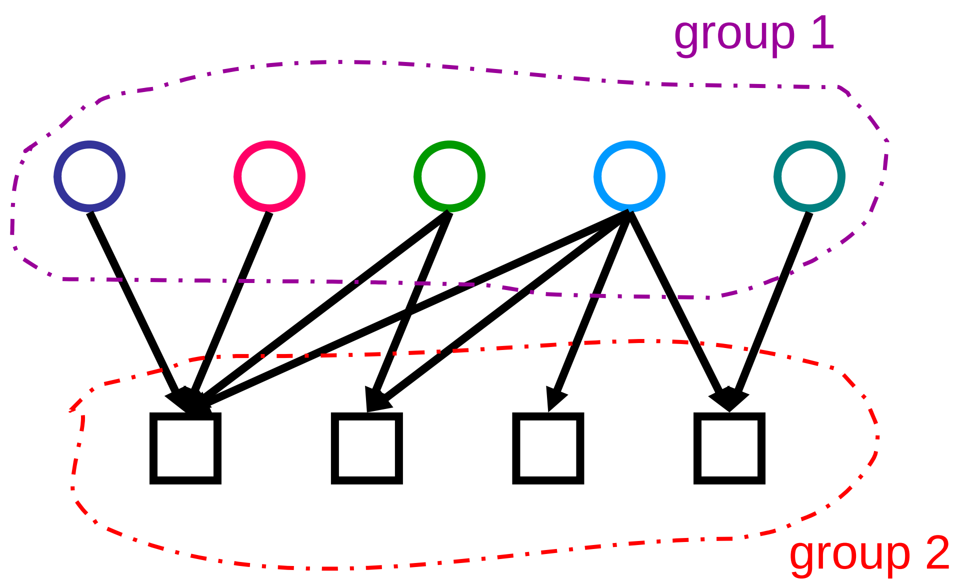
Bipartite Graphs

- Graphs where the set V can be partitioned into two sets L and R, such that all edges are between nodes in L and R, and there is no edge within L or R

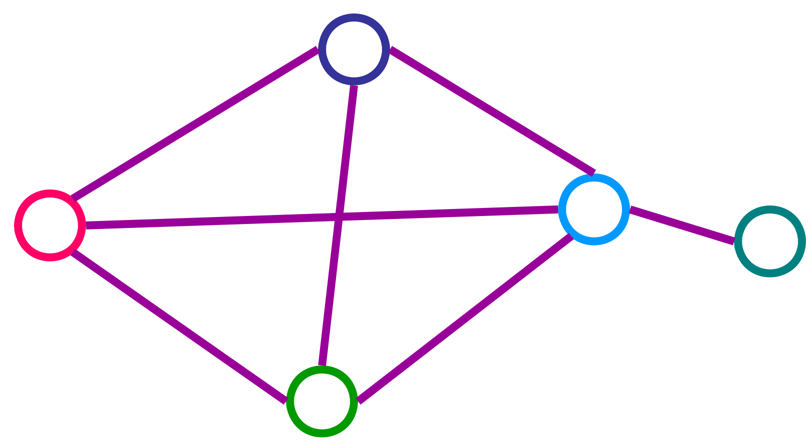


Going from Bipartite to one-mode

- Two-mode (bipartite) network



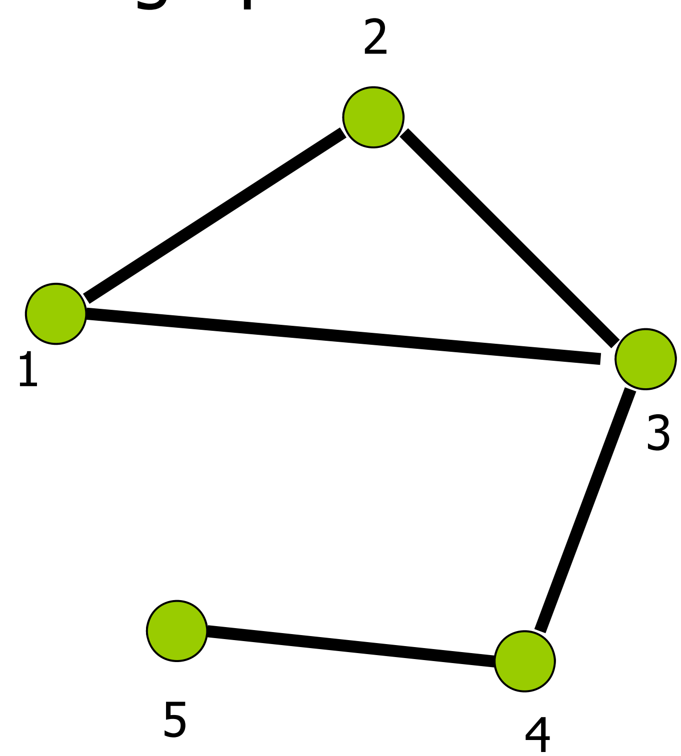
- One-mode projection
  - two nodes from the first group are connected if they link to the same node in the second group
  - some loss of information
  - naturally high occurrence of cliques



Adjacency Matrix

- Representing edges (who is adjacent to whom) as a matrix
- Symmetric matrix for undirected graphs

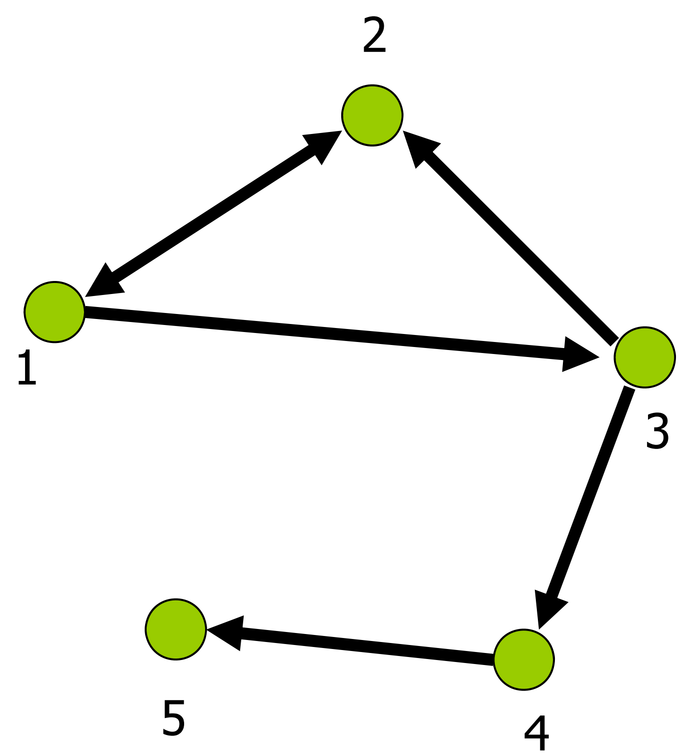
$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$



Adjacency Matrix

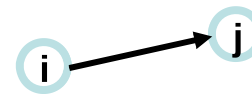
- Representing edges (who is adjacent to whom) as a matrix
- Non-symmetric matrix for directed graphs

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

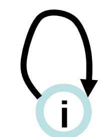


Adjacency Matrix

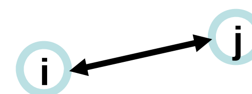
- A_{ij} = 1 if node i has an edge to node j
  = 0 if node i does not have an edge to j



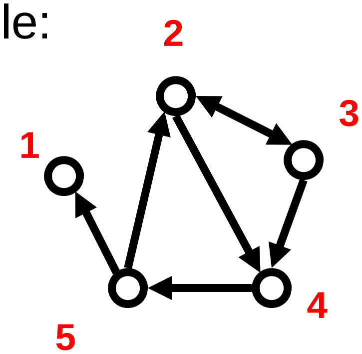
- A_{ii} = 0 unless the network has self-loops



- $A_{ij} = A_{ji}$ if the network is undirected, or if i and j share a reciprocated edge



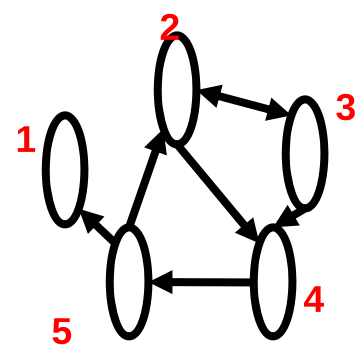
Example:



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

Adjacency Lists

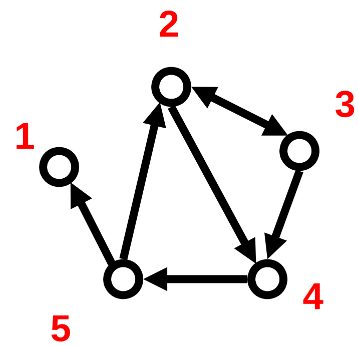
- Edge list
  - 2 3
  - 2 4
  - 3 2
  - 3 4
  - 4 5
  - 5 2
  - 5 1



- Adjacency list
  - is easier to work with if network is
    - large
    - sparse
  - quickly retrieve all neighbors for a node
    - 1:
    - 2: 3 4
    - 3: 2 4
    - 4: 5
    - 5: 1 2

Node degree from matrix values

- Out-degree = $\sum_{j=1}^{n} A_{ij}$



$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ \boxed{0} & \boxed{1} & \boxed{0} & \boxed{1} & \boxed{0} \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

example: out-degree for node 3 is 2, which we obtain by summing the number of non-zero entries in the 3rd *row*

$$\sum_{j=1}^{n} A_{3j}$$

- In-degree = $\sum_{i=1}^{n} A_{ji}$

$$A = \begin{pmatrix} 0 & 0 & \boxed{0} & 0 & 0 \\ 0 & 0 & \boxed{1} & 1 & 0 \\ 0 & 1 & \boxed{0} & 1 & 0 \\ 0 & 0 & \boxed{0} & 0 & 1 \\ 1 & 1 & \boxed{0} & 0 & 0 \end{pmatrix}$$

example: the in-degree for node 3 is 1, which we obtain by summing the number of non-zero entries in the 3rd *column*

$$\sum_{i=1}^{n} A_{i3}$$

Weighted Graph

- In weighted networks instead of degree, strength of nodes are defined
- If the weighted adjacency matrix is $W=(w_{ij})$, the strength of node i is defined as

$$s_i = \sum_{j=1}^{n} w_{ij}$$

- For weighted directed network the in strength and out-strength are defined
- The strength distribution of the graph is also correspondingly defined

Eigenvalues and Eigenvectors

- The value λ is an eigenvalue of matrix A if there exists a non-zero vector x, such that $Ax=\lambda x$. Vector x is an eigenvector of matrix A
  - The largest eigenvalue is called the principal eigenvalue
  - The corresponding eigenvector is the principal eigenvector
  - Corresponds to the direction of maximum change

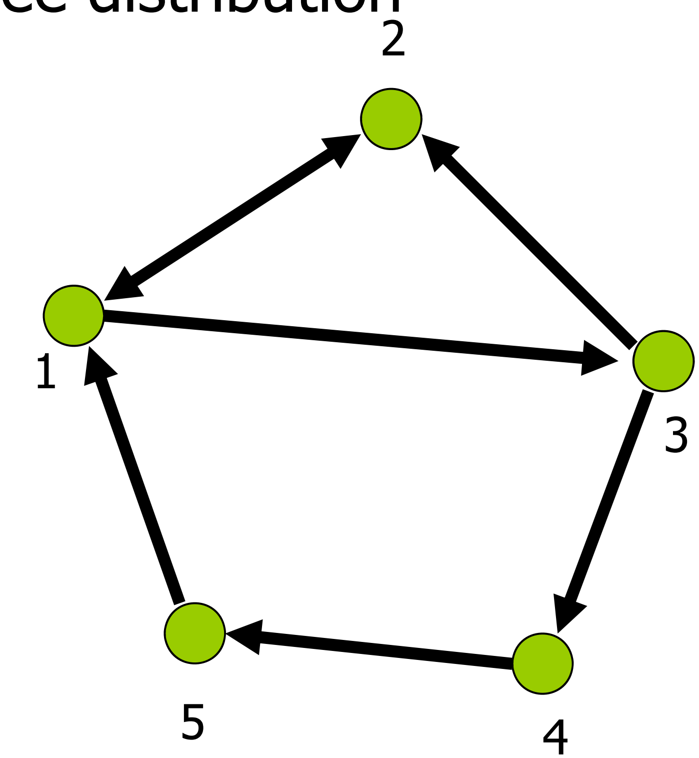
Random Walks

- Start from a node, and follow links uniformly at random.
- Stationary distribution: The fraction of times that you visit node i, as the number of steps of the random walk approaches infinity
  - if the graph is strongly connected, the stationary distribution converges to a unique vector.

Random Walks

- stationary distribution: principal left eigenvector of the normalized adjacency matrix
  - x = xP
  - for undirected graphs, the degree distribution

$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Readings

- L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas. Characterization of complex networks: A survey of measurements. Advances in Physics, 56(1):167 – 242, 2007.