Network Basics 1

Advanced Social Computing

Department of Computer Science
University of Massachusetts, Lowell
Fall 2020

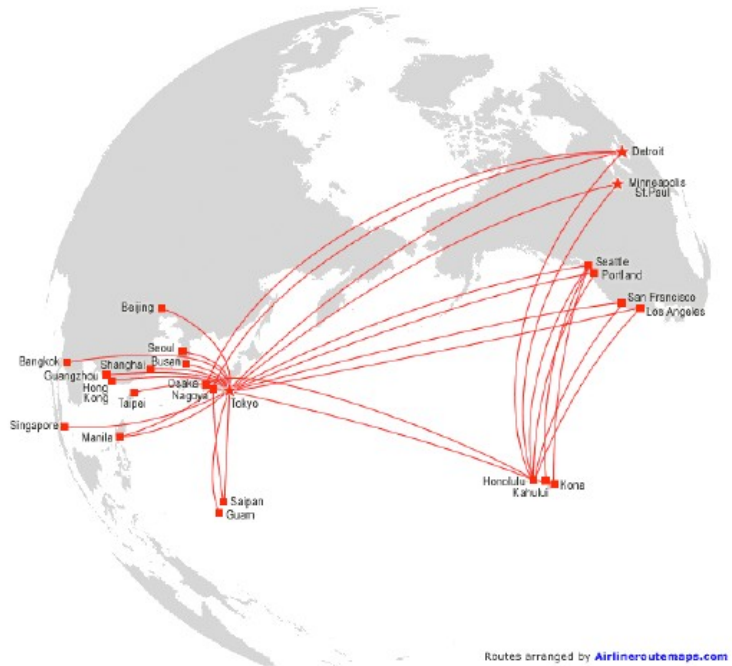
Hadi Amiri
hadi@cs.uml.edu

Lecture Topics

- Graph Theory
  - Node degree
  - Graph density
  - Complete Graph
  - Distance and Diameter
  - Adjacency matrix
  - Graph Connectivity
  - Reachability
  - Sub-graphs
  - Graph Types

Terminology

- Graph terminology is often derived from transportation metaphors
  - E.g. “shortest path”, “flow”, “diameter”



(a) *Airline routes*

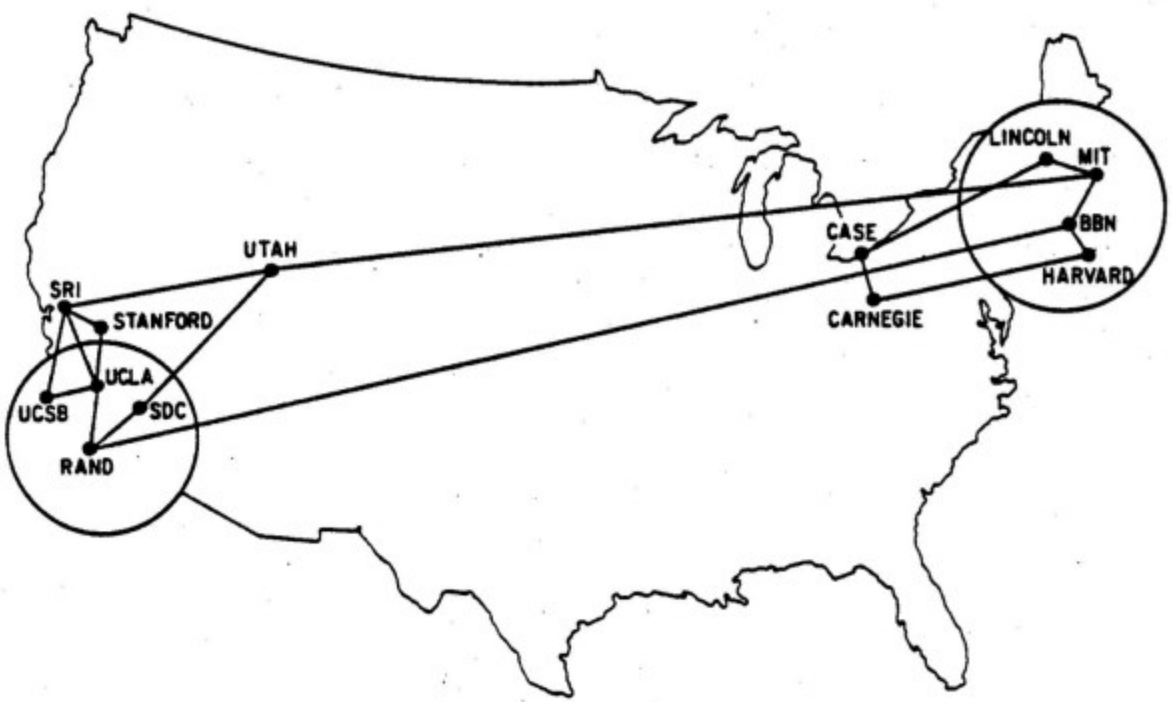
(b) *Subway map*

Graphs as Models of Networks

- Abstract graph theory is interesting in itself
- But in network science, items typically represent real-world entities
  - Examples
    - Communication networks
      - Companies, telephone wires
    - Social networks
      - People, friendship/contacts
    - Information networks
      - Web sites, hyperlinks

Graphs as Models of Networks – Cnt.

- ARPANET: Early Internet precursor
- December 1970 with 13 nodes



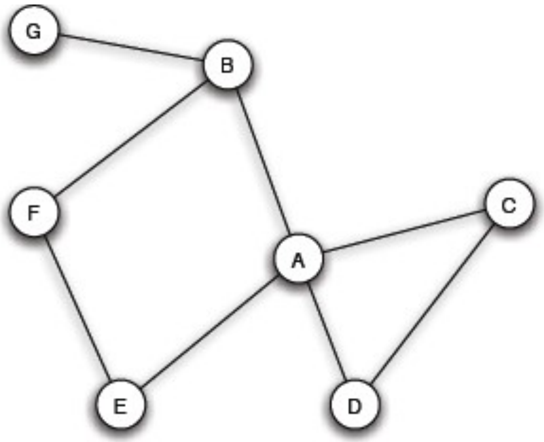
Basic Graph Concepts

Graph Theory

- A graph consists of
  - **N**: a set of nodes (items, entities, people, etc), and
  - **E**: a set of links or edges between nodes

- Graph is a way to **specify relationships** / links amongst a set of nodes.

- We define
  - N=|**N**| → size of **N**
  - E=|**E**| → size of **E**



Graph Theory. Cnt.

- Nodes *i* and *j* are *adjacent* or *neighbors* if:
  - There is an edge btw them!
    - $i \in \mathbf{N}$
    - $j \in \mathbf{N}$
    - $(i, j) \in \mathbf{E}$

Sample Graphs 1.

- “Lives Near” Graph

nodes

	Actor	Lives near:
n_1	Allison	Ross, Sarah
n_2	Drew	Eliot
n_3	Eliot	Drew
n_4	Keith	Ross, Sarah
n_5	Ross	Allison, Keith, Sarah
n_6	Sarah	Allison, Keith, Ross

Links or edges

$l_1 = (n_1, n_5)$

$l_2 = (n_1, n_6)$

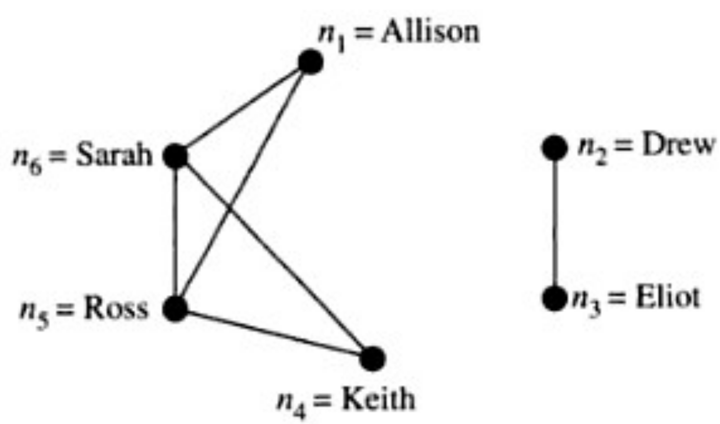
$l_3 = (n_2, n_3)$

$l_4 = (n_4, n_5)$

$l_5 = (n_4, n_6)$

$l_6 = (n_5, n_6)$

Graph

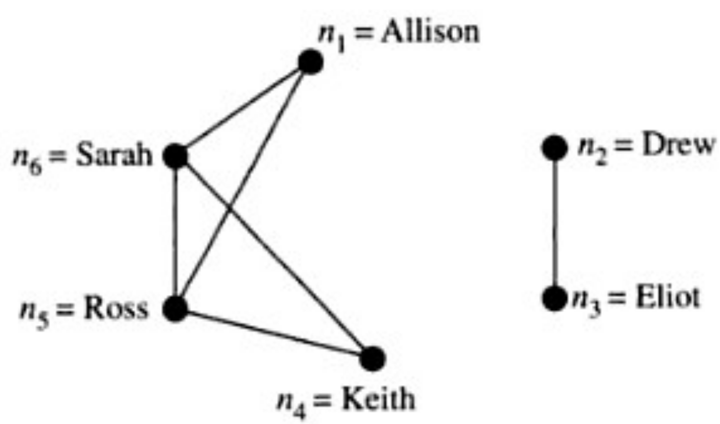


Node Degree $d(i)$

- Given Node i, its degree $d(i)$ is:
  - the number nodes adjacent to it.

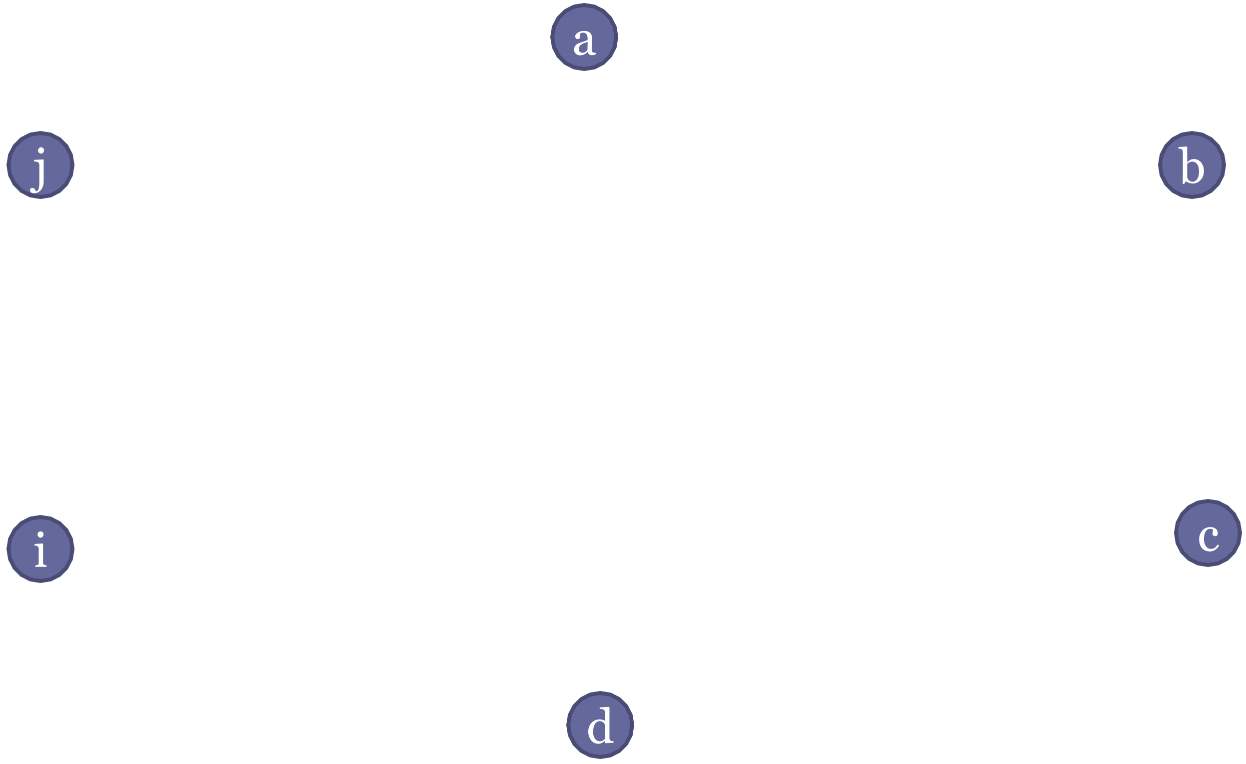
	Actor	Lives near:	Degree
n_1	Allison	Ross, Sarah	2
n_2	Drew	Eliot	1
n_3	Eliot	Drew	1
n_4	Keith	Ross, Sarah	2
n_5	Ross	Allison, Keith, Sarah	3
n_6	Sarah	Allison, Keith, Ross	3

$$l_1 = (n_1, n_5)$$
$$l_2 = (n_1, n_6)$$
$$l_3 = (n_2, n_3)$$
$$l_4 = (n_4, n_5)$$
$$l_5 = (n_4, n_6)$$
$$l_6 = (n_5, n_6)$$



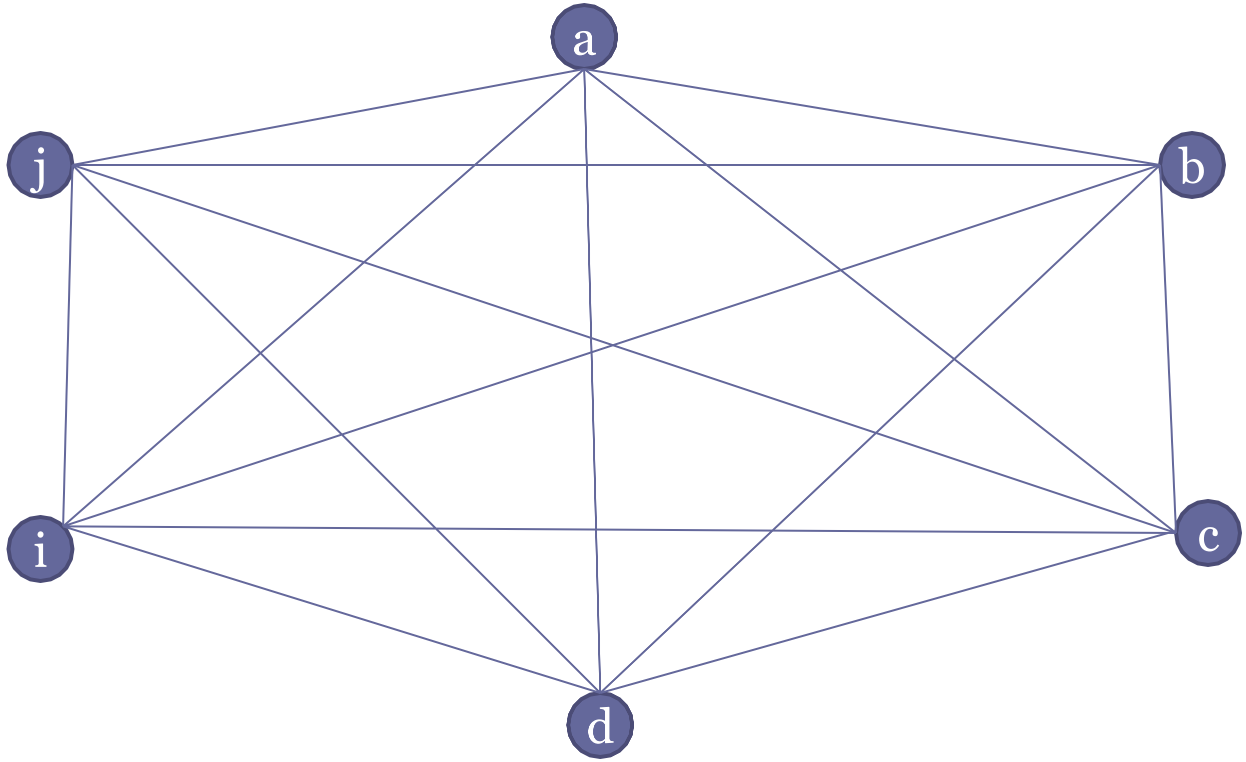
Graph Density

- How many edges are possible?



Graph Density- Cnt.

- (N-1) + (N-2) + (N-3) + … + 1 = N * (N-1) / 2



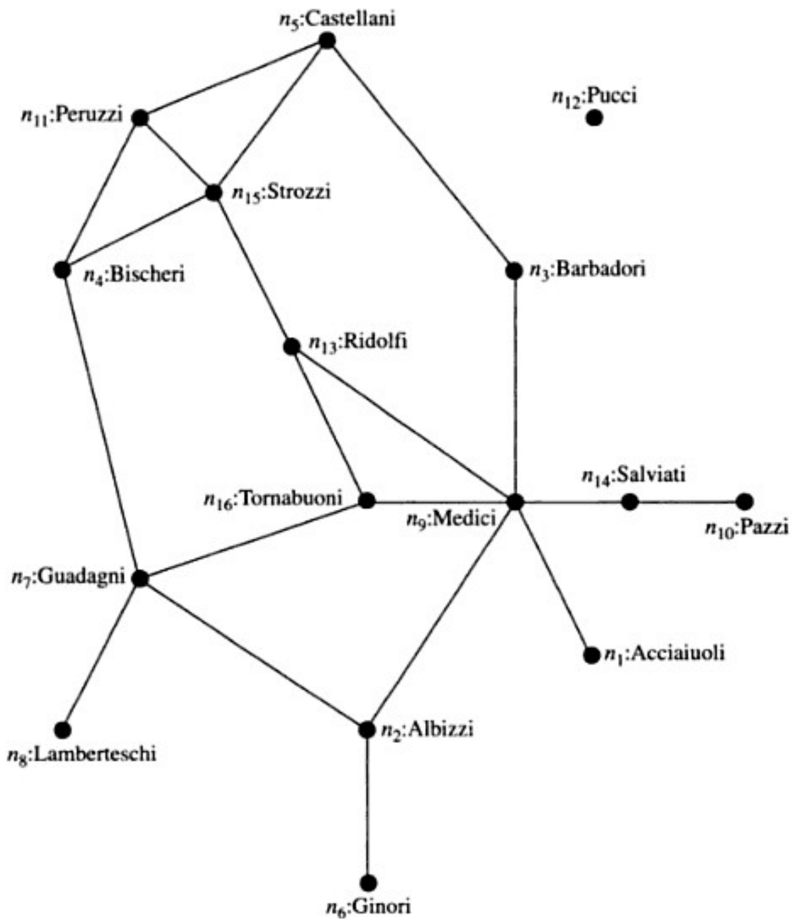
Graph Density- Cnt.

- Graph Density of a given graph G is determined by:
  - the proportion of all possible edges that are present in the graph.
  - with N nodes and E edges, graph density is:

$$Density = 2 * E / N * (N-1)$$

Graph Density- Cnt.

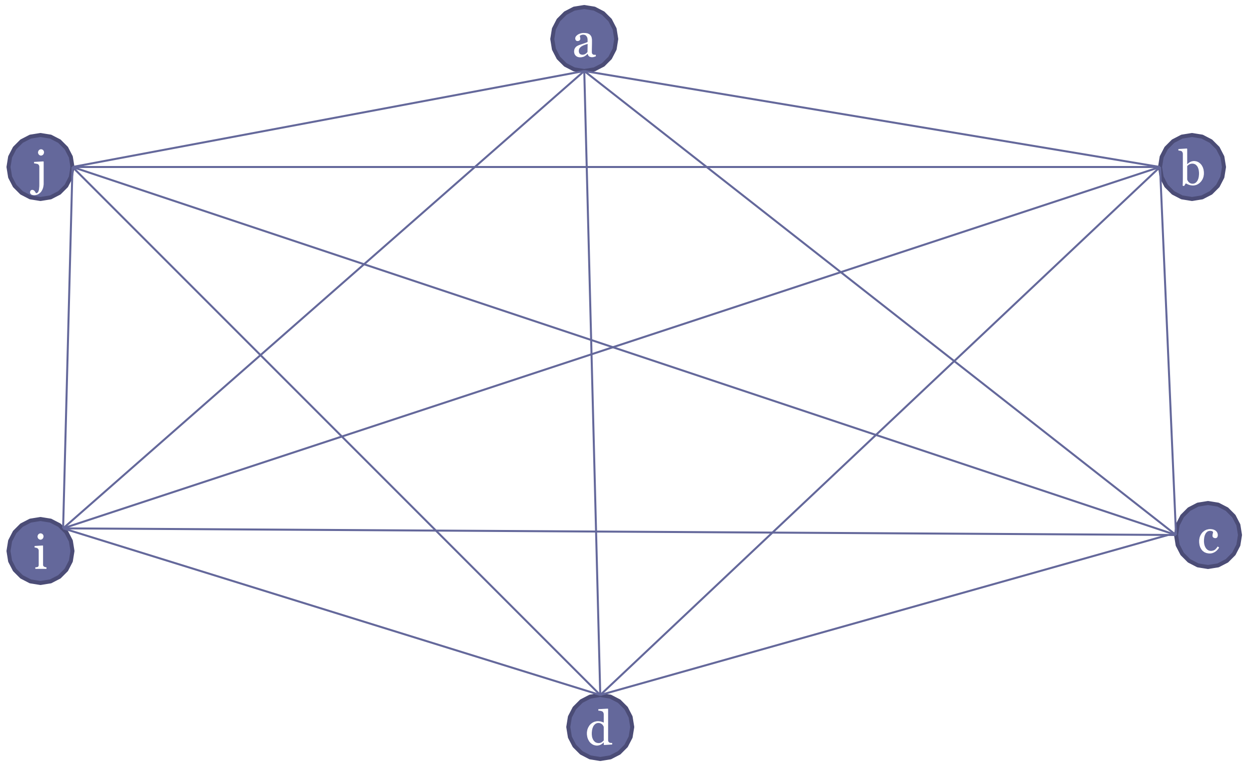
- What is the density of this graph?
  - N= 16
  - E= 20



Source: Social network analysis: Methods and applications. Wasserman, Stanley. Cambridge university press, 1994.

Complete Graph

- If all edges are present, then all nodes are adjacent (neighbors), and the graph is a *Complete Graph*.



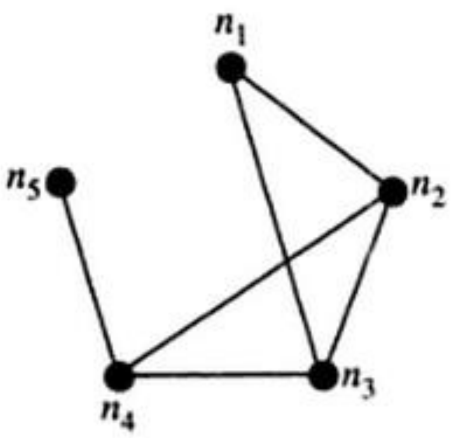
What is the density of a complete graph?

Distance and Diameter

- Distance btw node *i* and *j*: $d(i,j)$
  - length of the ***shortest path*** between *i* and *j*
- Diameter of a graph
  - the maximum value of $d(i,j)$ for all *i* and *j*

The path with min number of edges.

Distance and Diameter- Cnt.



distance

$d(1,2) = 1$

$d(1,3) = 1$

$d(1,4) = 2$

$d(1,5) = 3$

$d(2,3) = 1$

$d(2,4) = 1$

$d(2,5) = 2$

$d(3,4) = 1$

$d(3,5) = 2$

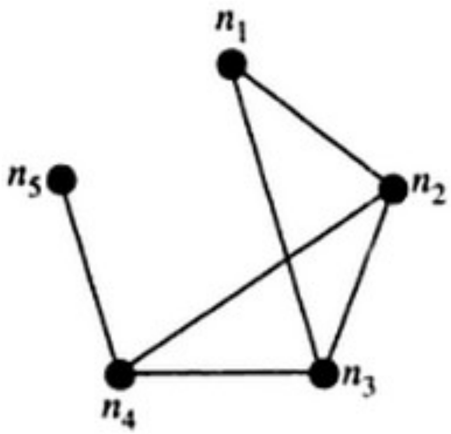
$d(4,5) = 1$

Diameter of graph $= \max d(i,j) = d(1,5) = 3$

What is the distance and diameter of a complete graph?

Source: Social network analysis: Methods and applications. Wasserman, Stanley. Cambridge university press, 1994.

Adjacency Matrix



A =

	n_1	n_2	n_3	n_4	n_5
n_1	0	1	1	0	0
n_2	1	0	1	1	0
n_3	1	1	0	1	0
n_4	0	1	1	0	1
n_5	0	0	0	1	0

- Each row or column represents a node!

$A = A^T$

Properties of adjacency matrix → next session

Graph Connectivity

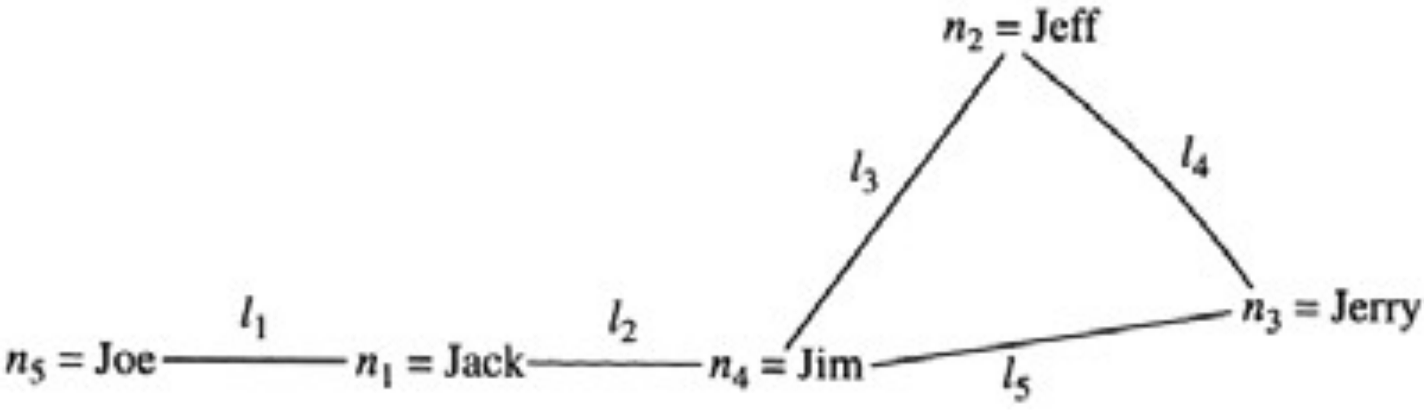
- Indirect connections between nodes:
  - Walks
  - Trails
  - Paths

Graph Connectivity- Cnt.

- Walk
  - A sequence of nodes and edges that starts and ends with nodes where each node is incident to the edges following and preceding it.
- Trail
  - A trail is a walk with distinct edges
- Path
  - A path is a walk with distinct nodes & edges.
- The length of a walk, trail, or path is the number of edges in it.

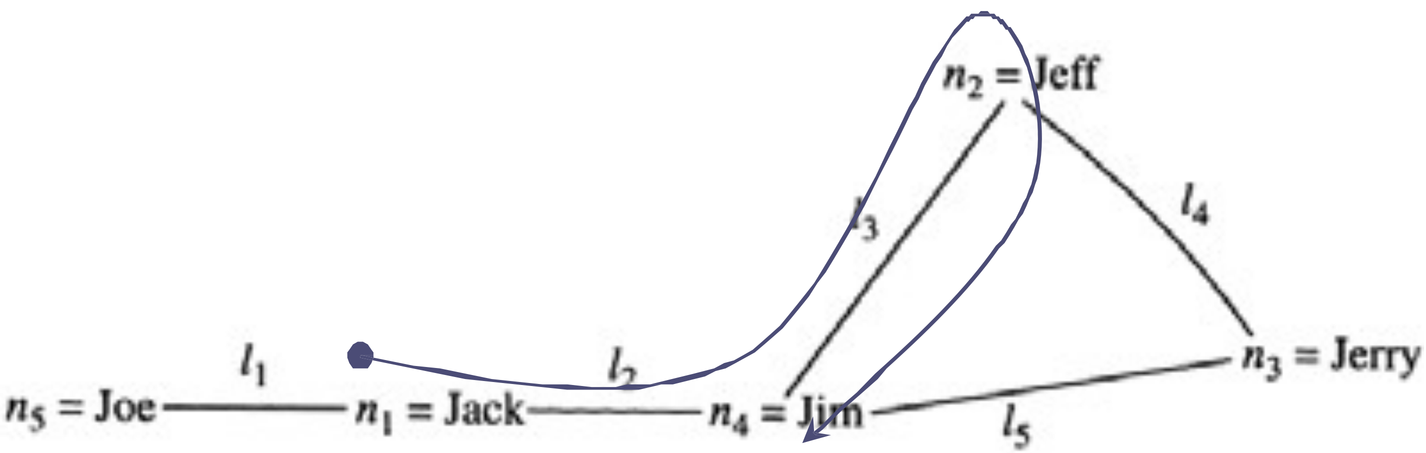
Graph Connectivity- Cnt.

- Walk
  - A sequence of nodes and edges that starts and ends with nodes where each node is incident to the edges following and preceding it.



Graph Connectivity- Cnt.

- Walk
  - A sequence of nodes and edges that starts and ends with nodes where each node is incident to the edges following and preceding it.

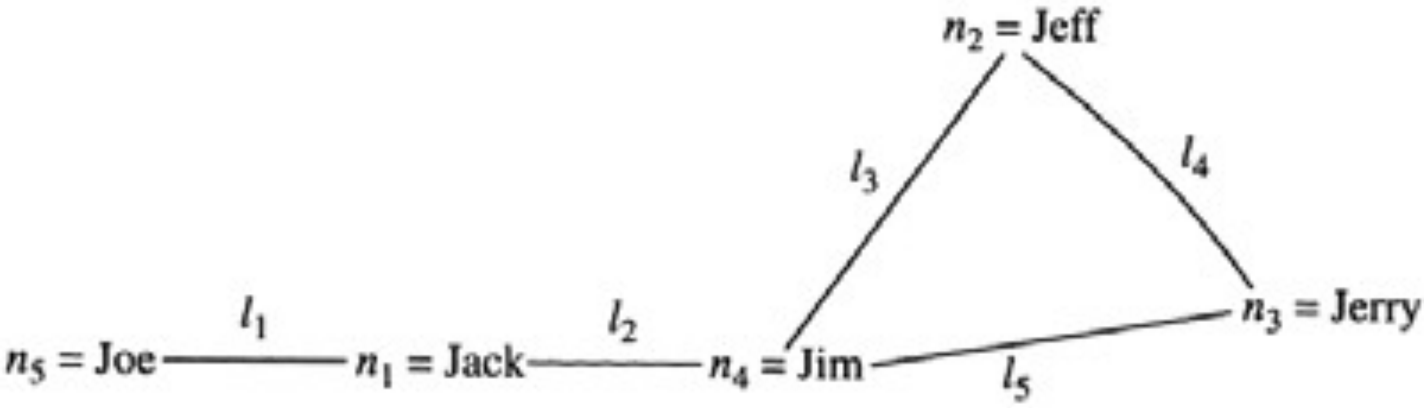


Sample Walk:

$W = n_1 l_2 n_4 l_3 n_2 l_3 \ n_4$

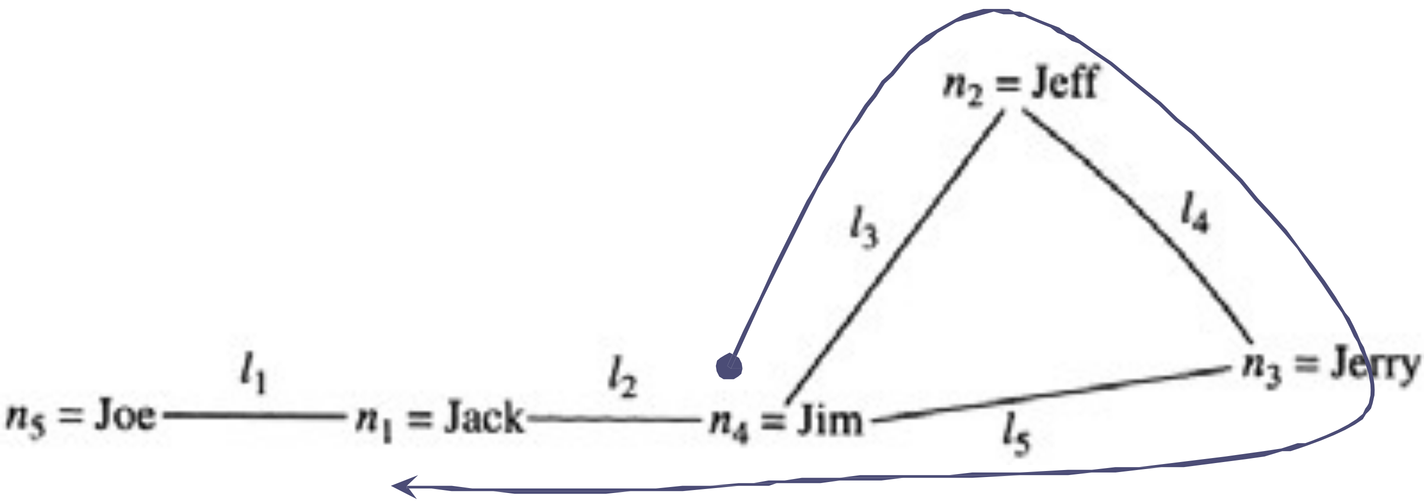
Graph Connectivity- Cnt.

- Trail
  - A trail is a walk in which all edges are distinct, although some node(s) may be included more than once.



Graph Connectivity- Cnt.

- Trail
  - A trail is a walk in which all edges are distinct, although some node(s) may be included more than once.

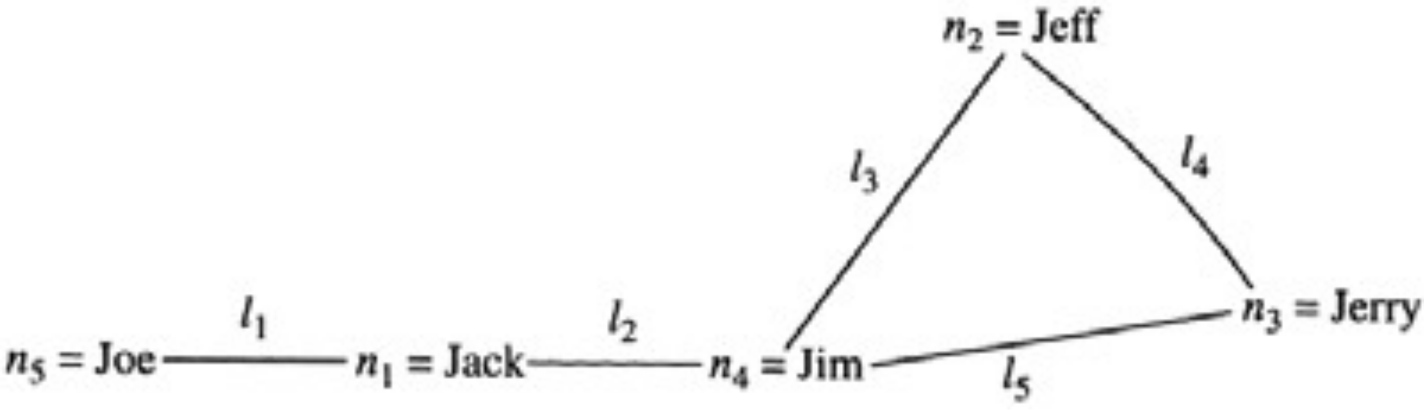


Sample Trail:

$T = n_4\, l_3\, n_2\, l_4\, n_3\, l_5\, n_4\, l_2\ \ n_1$

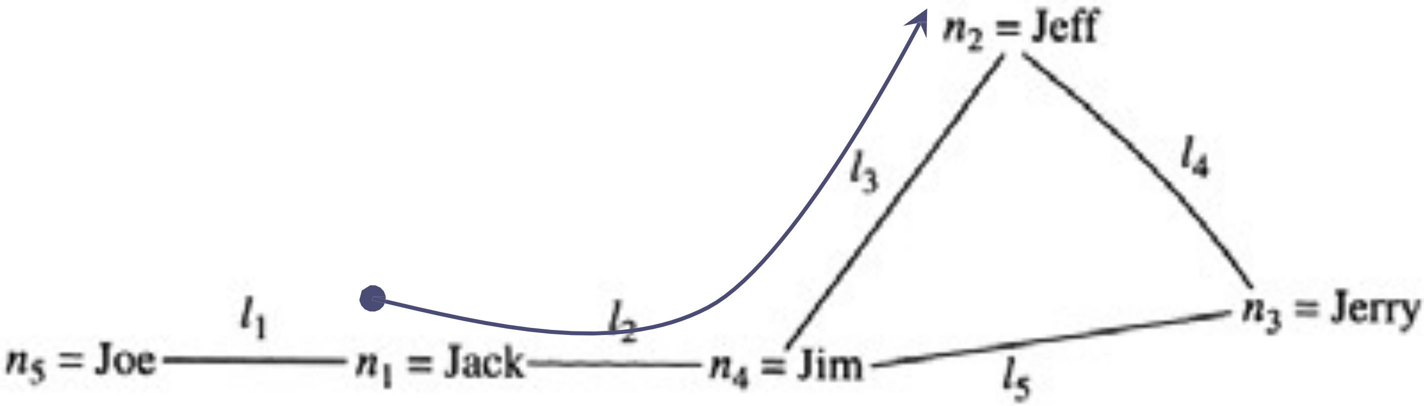
Graph Connectivity- Cnt.

- Path
  - A path is a walk in which all nodes and all edges are distinct.



Graph Connectivity- Cnt.

- Path
  - A path is a walk in which all nodes and all edges are distinct.

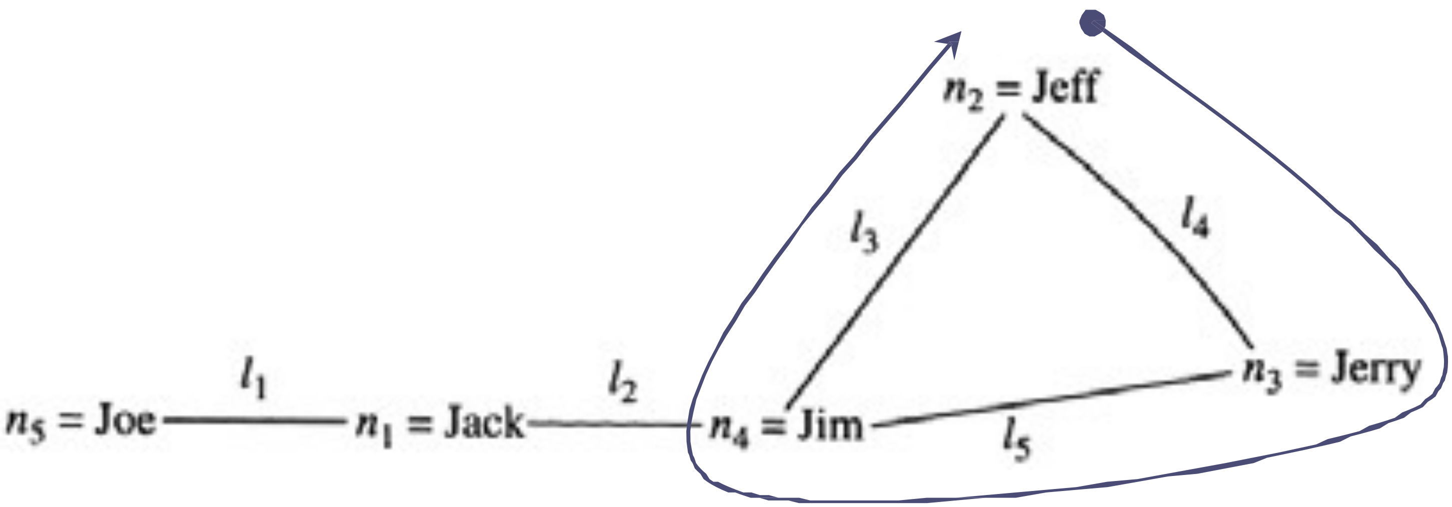


Sample Path:

$P = n_1 l_2 n_4 l_3 \; n_2$

Graph Connectivity- Cnt.

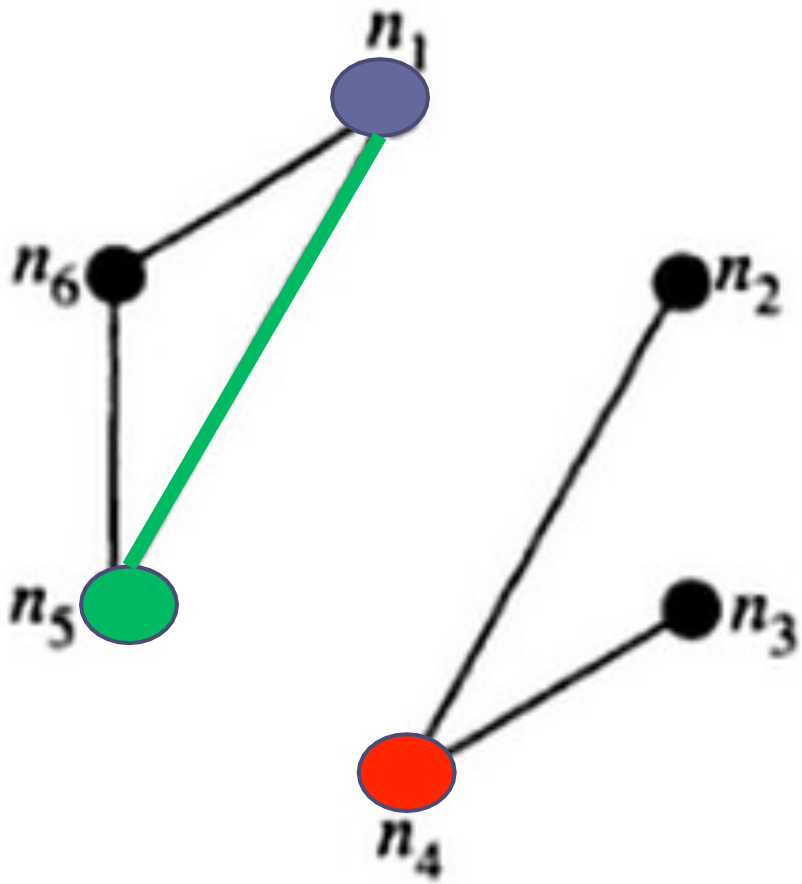
- Is this a Walk? Trail? Path?
  - We call a *closed walk* with distinct nodes & edges Cycle!



$n_2\, l_4\, n_3\, l_5\, n_4\, l_3\;\; n_2$

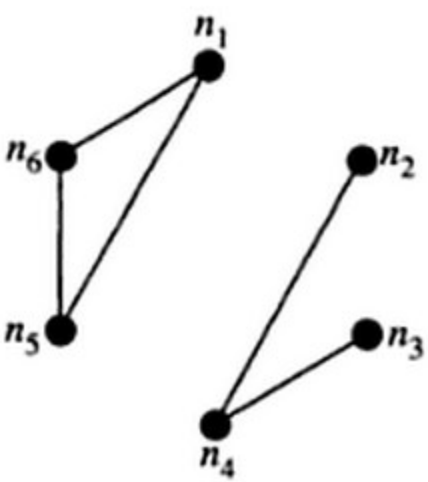
Reachability

- If there is a **path between nodes** *i* and *j*, then *i* and *j* are reachable from each other.



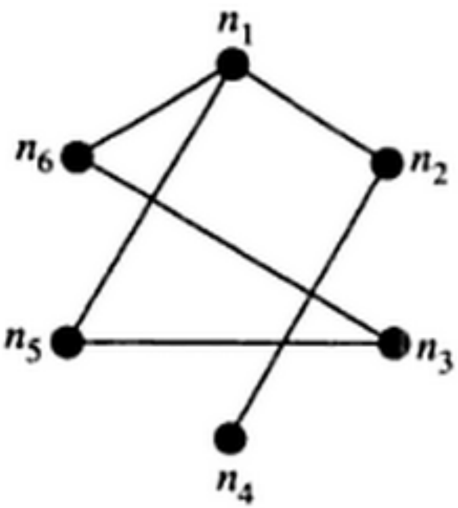
Connected Graph

- A graph is connected if ***every pair of its nodes*** are reachable from each other
  - i.e. there is a path between them.



Disconnected Graph

How can we make this graph connected?



Connected Graph

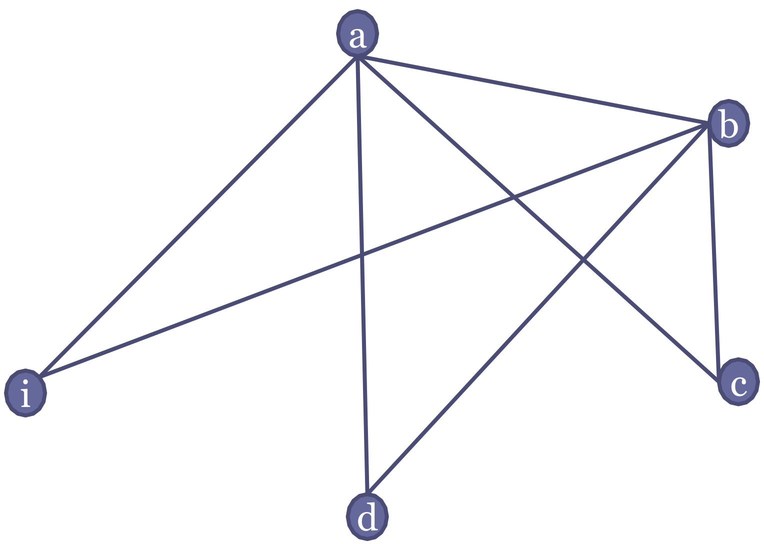
and this graph disconnected?

Sub-graphs

- Graph G_s is a sub-graph of G if its nodes and edges are a subset of G's nodes and edges respectively.

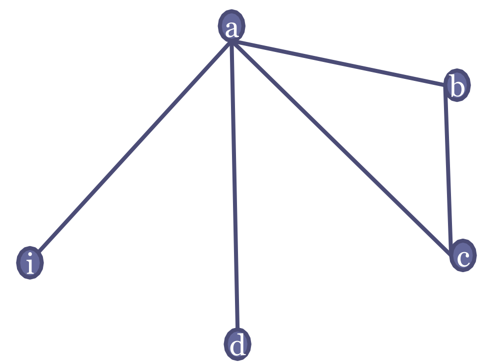
Sub-graphs- Cnt.

- Graph G_s is a sub-graph of G if its nodes and edges are a subset nodes and edges of G respectively.

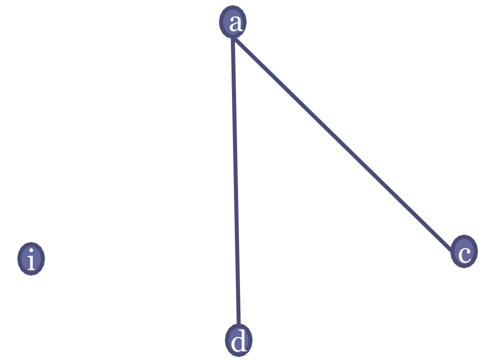


G

G_{s1}



G_{s2}

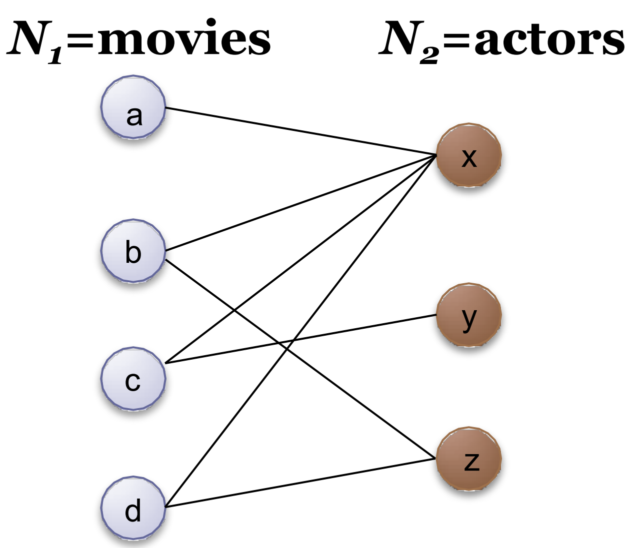


Graph Types

- Several types of graphs:
  - Bipartite graphs
  - Digraphs
  - Multigraphs
  - Hypergraphs
  - Weighted/Signed

Graph Types- Bipartite Graphs

- A bipartite graph is an undirected graph in which
  - nodes can be partitioned into two (disjoint) sets N_1 and N_2 such that:
    - $(u, v) \in E$ implies either $u \in N_1$ and $v \in N_2$ or vice versa
  - So, all edges go between the two sets N_1 and N_2 but not within N_1 or N_2.



N_1={a,b,c,d}

N_2={x,y,z}

Graph Types- Digraphs

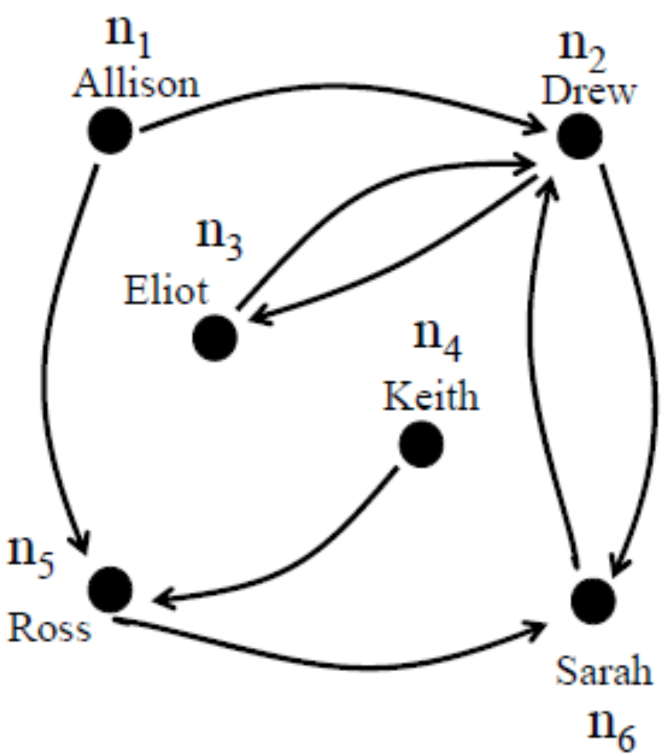
- Digraphs or Directed Graphs
  - Edges are directed
- Adjacency:
  - There is a direct edge btw nodes!
    - $i \in N$
    - $j \in N$
    - $(i, j) \in E$

Graph Types- Digraphs- Cnt.

- Node Indegree and Outdegree
  - Indegree
    - The indegree of a node, $d_I(i)$, is the number of nodes that link to *i*,
  - Outdegree
    - The outdegree of a node, $d_O(i)$, is the number of nodes that are linked by *i*,

- Indegree: number of edges terminating at *i*.
- Outdegree: number of edges originating at *i*.

Graph Types- Digraphs- Cnt.



$$d_O(n_i) = \sum_{j=1}^{n} A_{ij}$$

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

2
2
1
1
1
1

$$d_I(n_j) = \sum_{i=1}^{n} A_{ij}$$

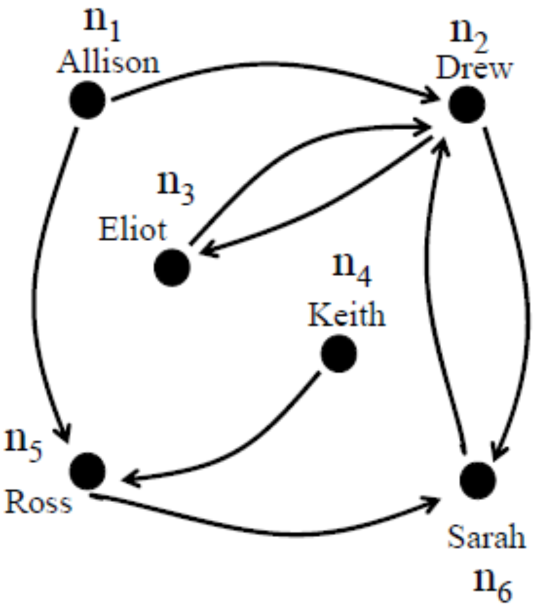
0	3	1	0	2	2

$A \mathrel{!=} A^T$

Graph Types- Digraphs- Cnt.

- Density of Digraph:
  - Number of all possible edges in Digraph?
    - N * (N-1)

$$\frac{E}{N * (N-1)}$$



Graph Types- Digraphs- Cnt.

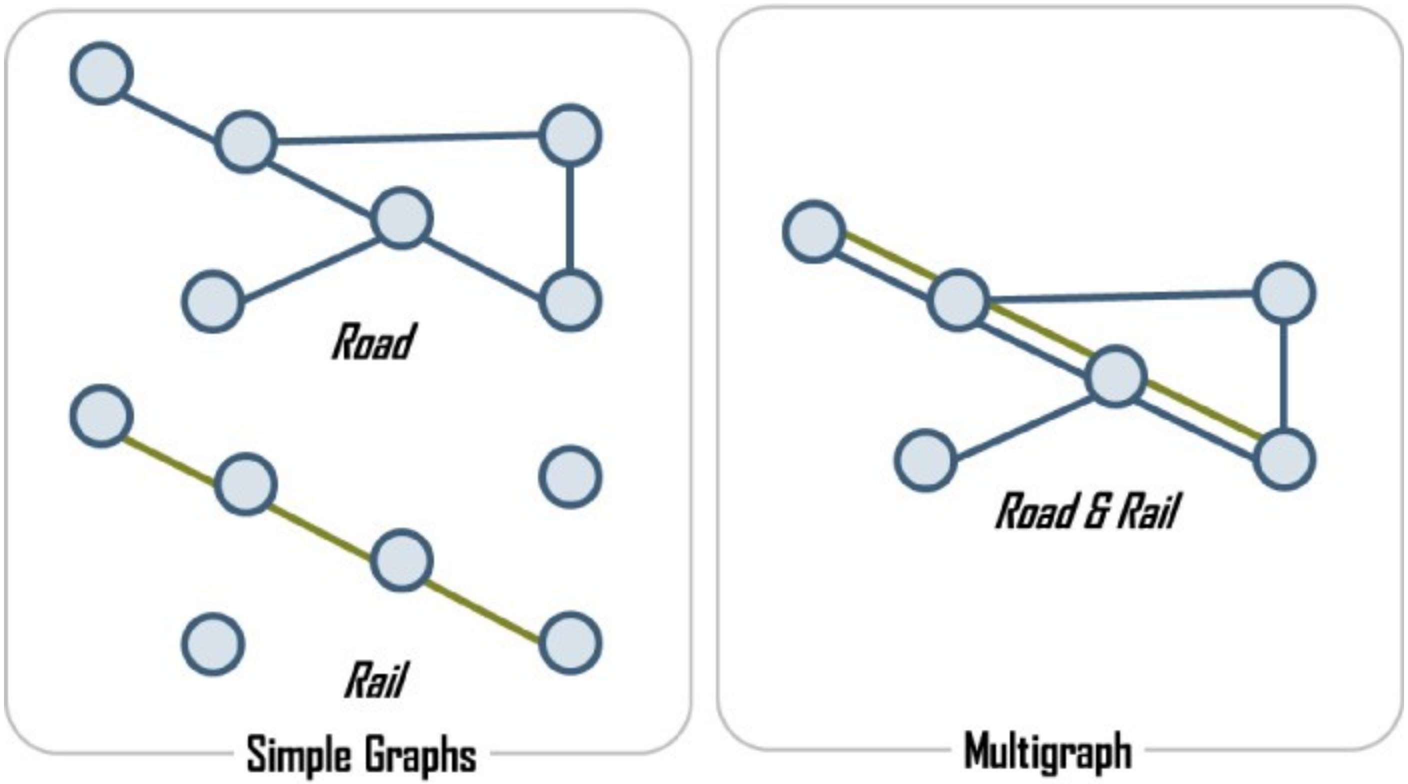
- Connectivity
  - Walks
  - Trails
  - Paths

- The same as before just links are directed!

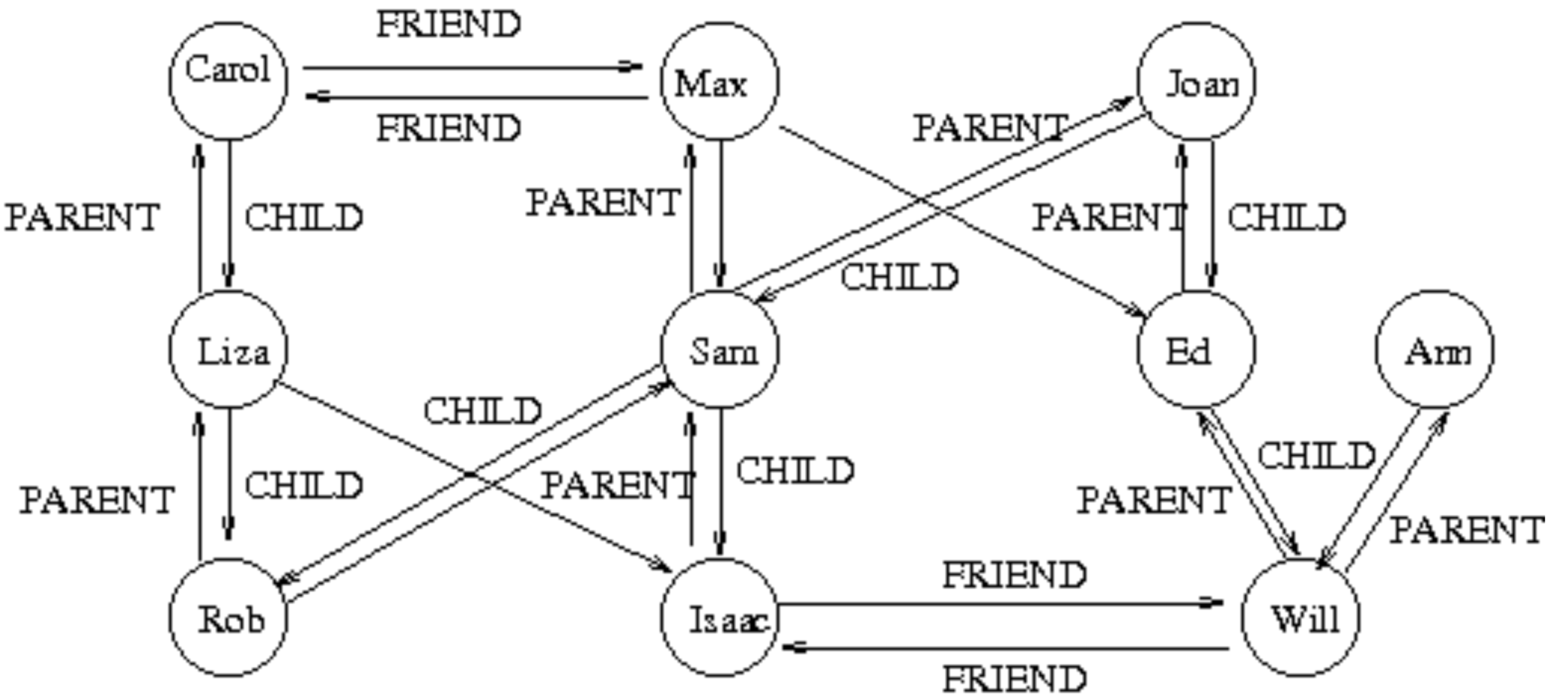
Graph Types- Multigraphs

- A Multigraph (or multivariate graph) *G* consists of:
  - a set of nodes, *and*
  - two or more sets of edges, $E^+ = \{\boldsymbol{E_1}, \boldsymbol{E_2}, \ldots, \boldsymbol{E_r}\}$, r is the number of edge sets.

Multigraph 1.



Multigraph 2.



Graph Types- Multigraphs- Cnt.

- Each E_i indicated one type of relationship, e.g.:
  - E_1: lives near relationship
  - E_2: friends at the beginning of the year
  - E_3: friends at the end of the year

Graph Types- Multigraphs- Cnt.

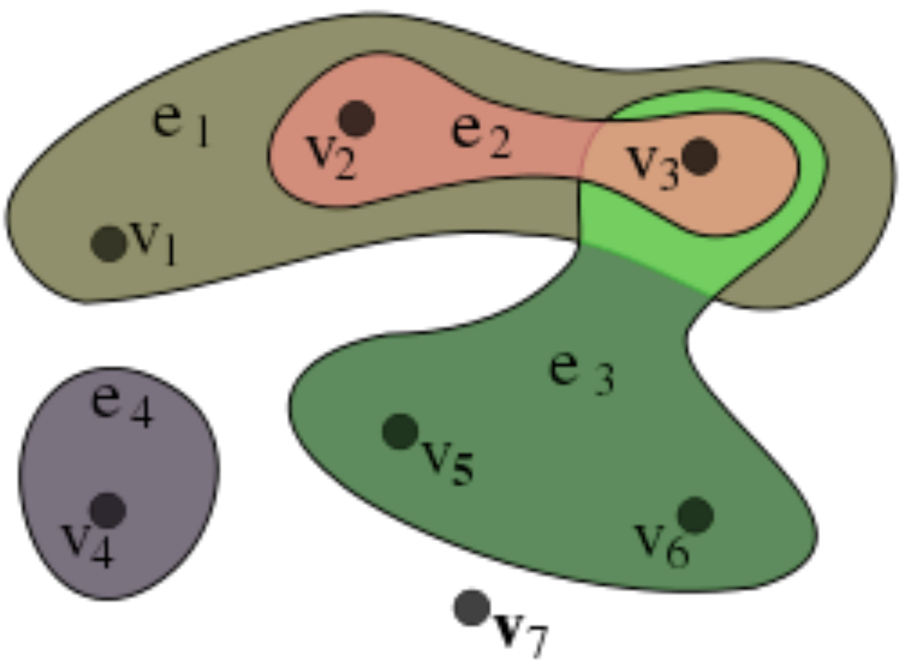
- Number of edges btw any two nodes in a multigraph?
  - $E^+ = \{\boldsymbol{E_1}, \boldsymbol{E_2}, \ldots, \boldsymbol{E_r}\}$, r is the number of sets of edges
    - Undirected multigraph
      - [0, r]
    - Directed multigraph
      - [0, 2*r]

Graph Types- Hypergraphs

- A hypergraph is a graph in which an edge can connect any number of nodes.
- In a hypergraph, ***E*** is a set of non-empty subsets of ***N*** called *hyperedges*.

Graph Types- Hypergraphs- Cnt.

- A hypergraph is a graph in which an edge can connect any number of nodes.
- In a hypergraph, ***E*** is a set of non-empty subsets of ***N*** called *hyperedges*.



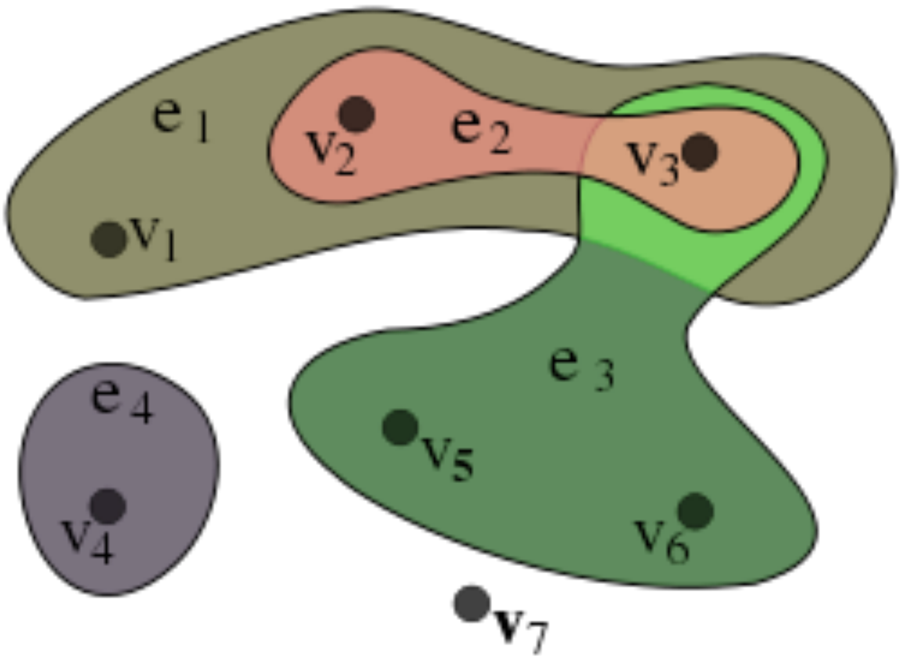
N=$\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$

E=$\{e_1, e_2, e_3, e_4\}$=

$\{\{v_1, v_2, v_3\}, \{v_2, v_3\}, \{v_3, v_5, v_6\}, \{v_4\}\}$

Graph Types- Hypergraphs- Cnt.

- Applications:
  - Recom. systems (communities as edges),
  - Image retrieval (correlations as edges),
  - Bioinformatics (interactions or semantic types as edges).



N=$\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$

E=$\{e_1, e_2, e_3, e_4\}$=

$\{\{v_1, v_2, v_3\}, \{v_2, v_3\}, \{v_3, v_5, v_6\}, \{v_4\}\}$

Weighted/Signed Graphs

- Edges may carry additional information
  - Tie strength → how good are two nodes as friends?
  - Distance → how long is the distance btw two cities?
  - Delay → how long does the transmission take btw two cities?
  - Signs → two nodes are friends or enemies?

Reading

- Ch.02 Graphs [NCM]
- Ch. 22 Elementary Graph Algorithms [CLRS]