Graph Fundamentals

Sarunas Girdzijauskas

ID2211

March 2019

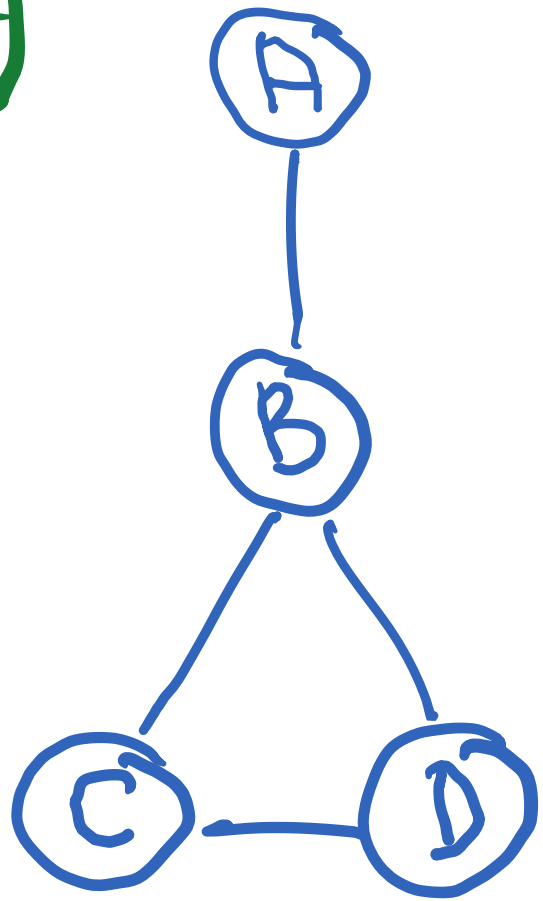
Basic Definitions

- A ***graph*** is a way of specifying relationships among a collection of items.
  - A ***graph*** consists of a set of objects called ***nodes*** (***vertices***)
  - With certain pairs of these objects connected by links called ***edges*** (***links***)
  - Two nodes are ***neighbors*** if they are connected by an ***edge***

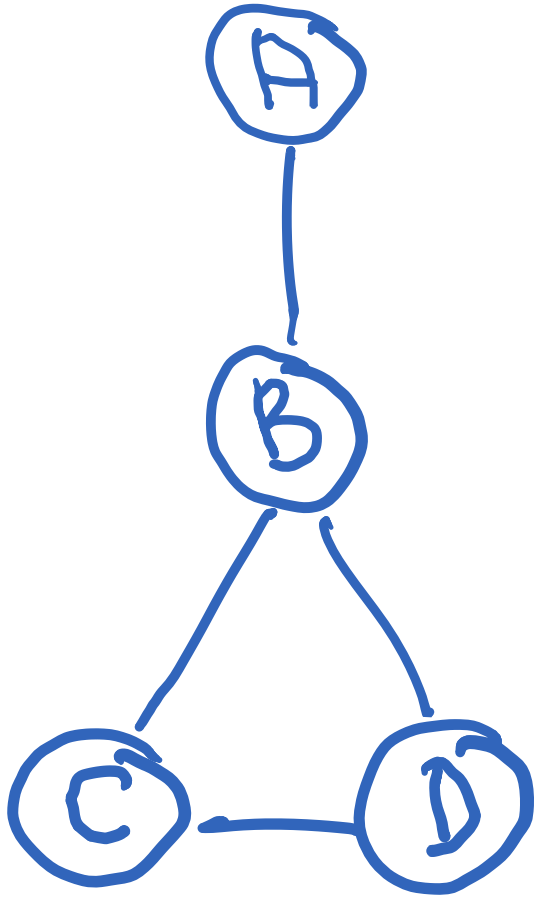
G(N,E)

N

E

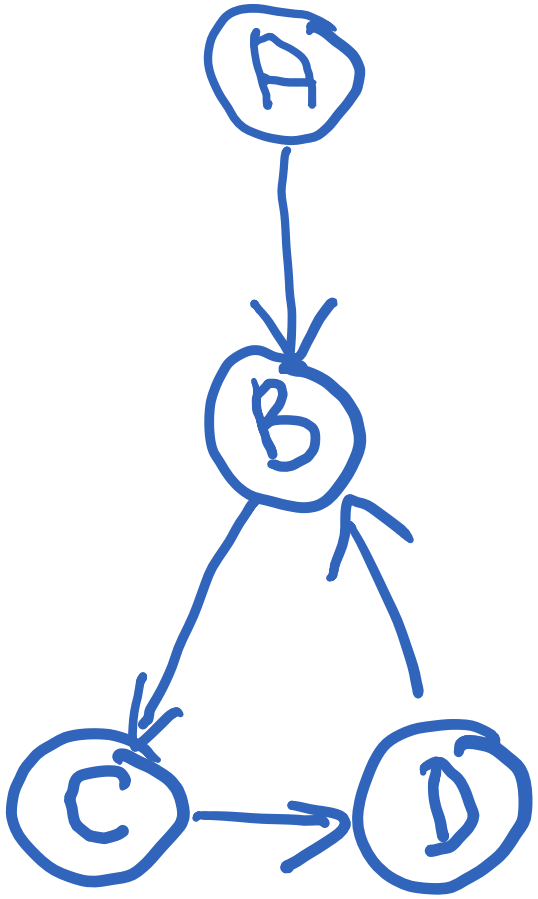


Directed vs. Undirected



- Edges have no orientation

(e.g, Facebook, Collaboration)

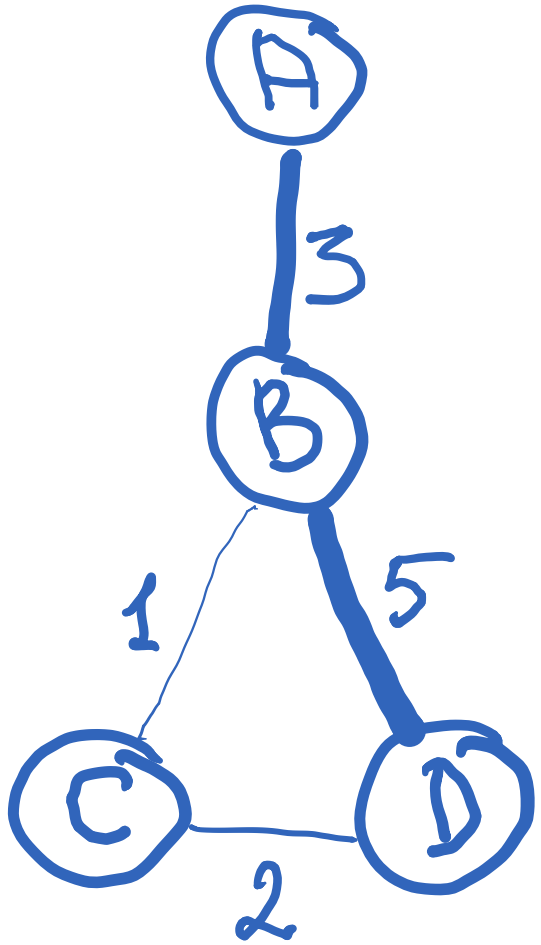


- Edges have orientation

(e.g., Twitter, Phone calls, Citations)

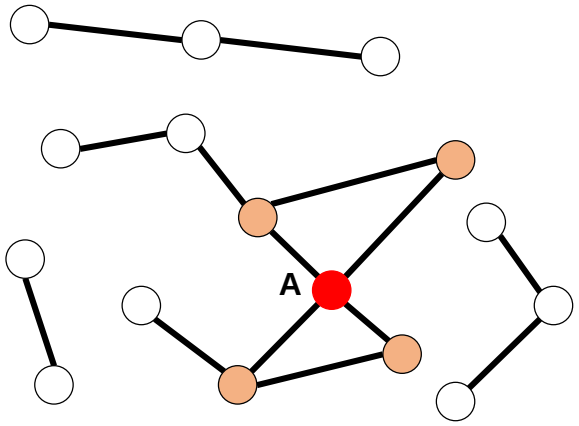
Weighted Graphs

- In a ***weighted*** graph every edge has an associated ***weight*** with it
  - What could weights represent?
  - E.g, distance, cost, frequency etc

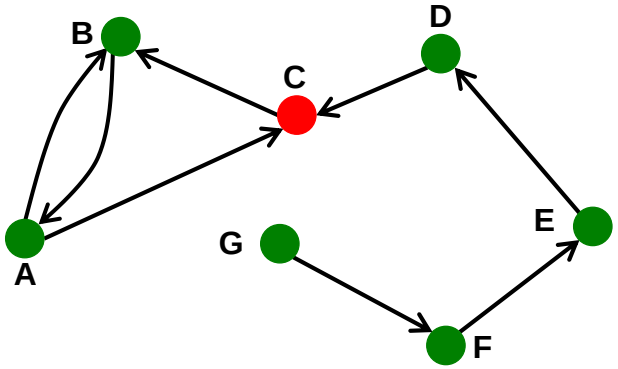


Node Degrees

Undirected



Directed



Source: Node with $k^{in} = 0$

Sink: Node with $k^{out} = 0$

Node degree, k_i: the number of edges adjacent to node ***i***

$$k_A = 4$$

Avg. degree: $\bar{k} = \langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i = \frac{2E}{N}$

In directed networks we define an **in-degree** and **out-degree.**

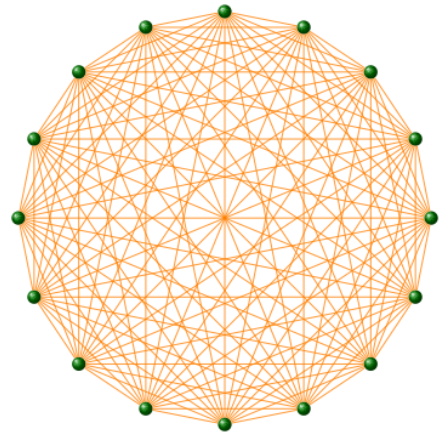
The (total) degree of a node is the sum of in- and out-degrees.

$$k_C^{in} = 2 \qquad k_C^{out} = 1 \qquad k_C = 3$$

$$\bar{k} = \frac{E}{N}$$

$$\overline{k^{in}} = \overline{k^{out}}$$

Dense vs. Sparse Graphs

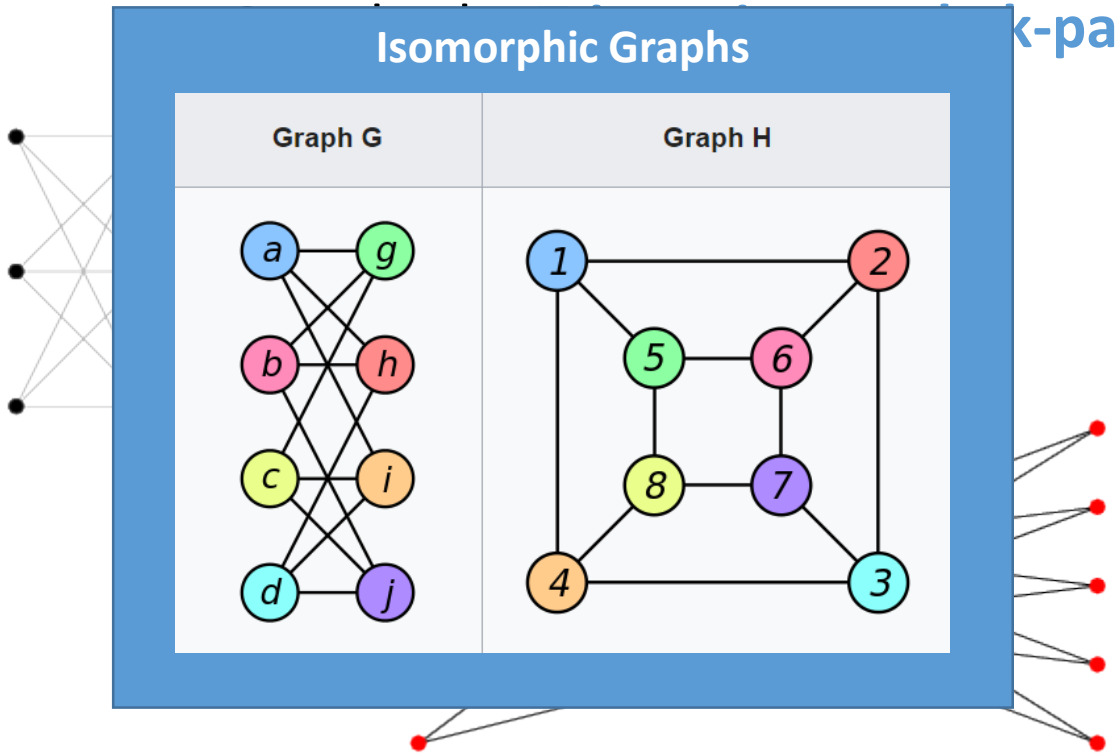


- Examples?
- The distinction between sparse and dense graphs is rather vague, and depends on the context.
  - The number of edges in **Dense graph** is close to the maximal number of edges.
    - Max
      -
      -
      -
  - The op
    - Ofte

NETWORK	NODES	LINKS	DIRECTED UNDIRECTED	N	L	⟨k⟩
Internet	Routers	Internet connections	Undirected	192,244	609,066	6.33
WWW	Webpages	Links	Directed	325,729	1,497,134	4.60
Power Grid	Power plants, transformers	Cables	Undirected	4,941	6,594	2.67
Mobile Phone Calls	Subscribers	Calls	Directed	36,595	91,826	2.51
Email	Email addresses	Emails	Directed	57,194	103,731	1.81
Science Collaboration	Scientists	Co-authorship	Undirected	23,133	93,439	8.08
Actor Network	Actors	Co-acting	Undirected	702,388	29,397,908	83.71
Citation Network	Paper	Citations	Directed	449,673	4,689,479	10.43
E. Coli Metabolism	Metabolites	Chemical reactions	Directed	1,039	5,802	5.58
Protein Interactions	Proteins	Binding interactions	Undirected	2,018	2,930	2.90

Bipartite graphs

- **Bipartite graph** (or **bigraph**) is a graph whose vertices can be divided into two disjoint sets U and V and such that every edge connects a vertex in U to one in V.
  - Complete Bipartite Graph

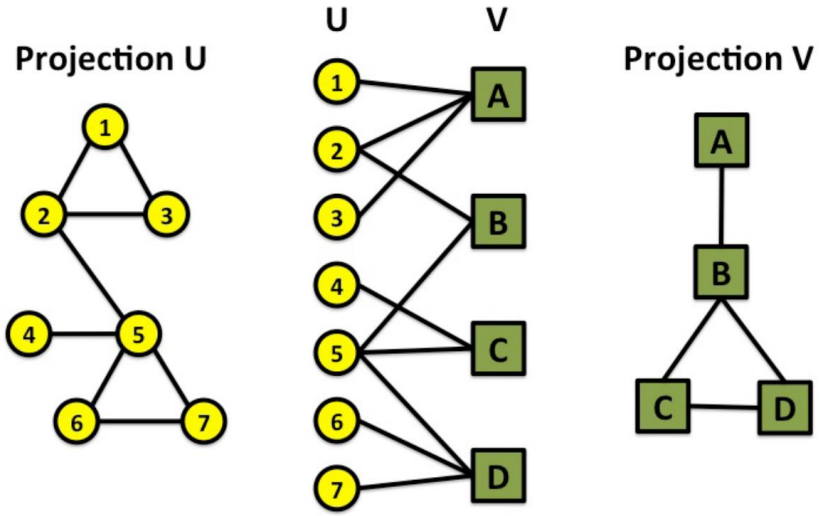


k-partite etc.

Examples?

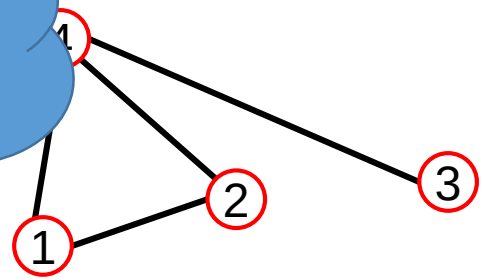
Authors-to-papers (they authored)
Actors-to-Movies (they appeared in)
Users-to-Movies (they rated)

Folded Networks

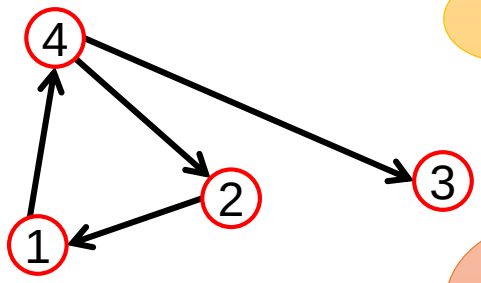


Representing Graphs: Adjacency Matrix

How to find out the node degree using A?



How to represent Weighted graphs?



When are the matrices symmetric?

$A_{ij} = 1$ if there is a link from node *i* to node *j*

$A_{ij} = 0$ otherwise

How would Adjacency Matrices of real-world graphs look like?

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

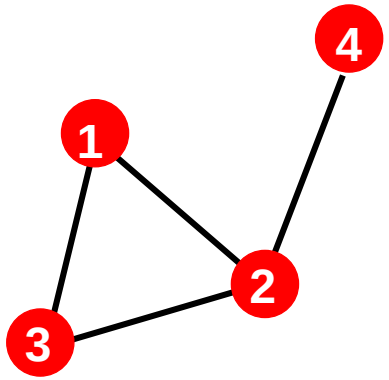
$$A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

indegree

outdegree

More Types of Graphs

- **Unweighted**
  (undirected)



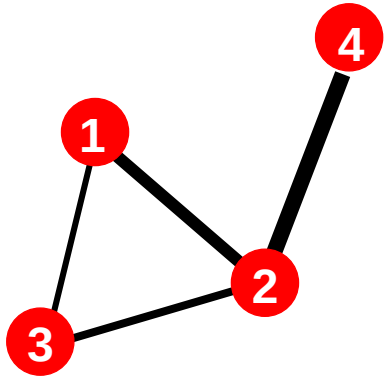
$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1}^{N} A_{ij} \qquad \bar{k} = \frac{2E}{N}$$

Examples: Friendship, Hyperlink

- **Weighted**
  (undirected)



$$A_{ij} = \begin{pmatrix} 0 & 2 & 0.5 & 0 \\ 2 & 0 & 1 & 4 \\ 0.5 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{pmatrix}$$

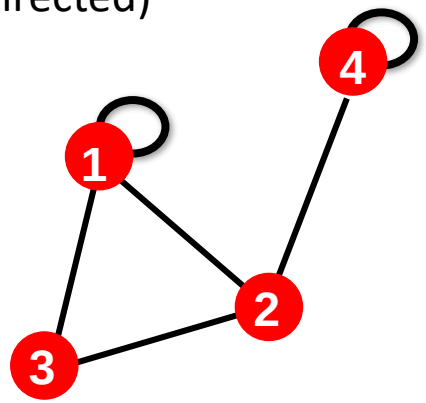
$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \bar{k} = \frac{2E}{N}$$

Examples: Collaboration, Internet, Roads

More Types of Graphs

- **Self-edges (self-loops)**
  (undirected)



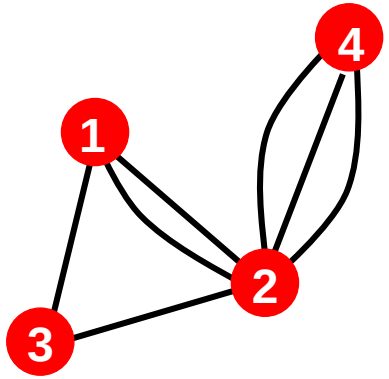
$$A_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$A_{ii} \neq 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1, i\neq j}^{N} A_{ij} + \sum_{i=1}^{N} A_{ii}$$

Examples: Proteins, Hyperlinks

- **Multigraph**
  (undirected)



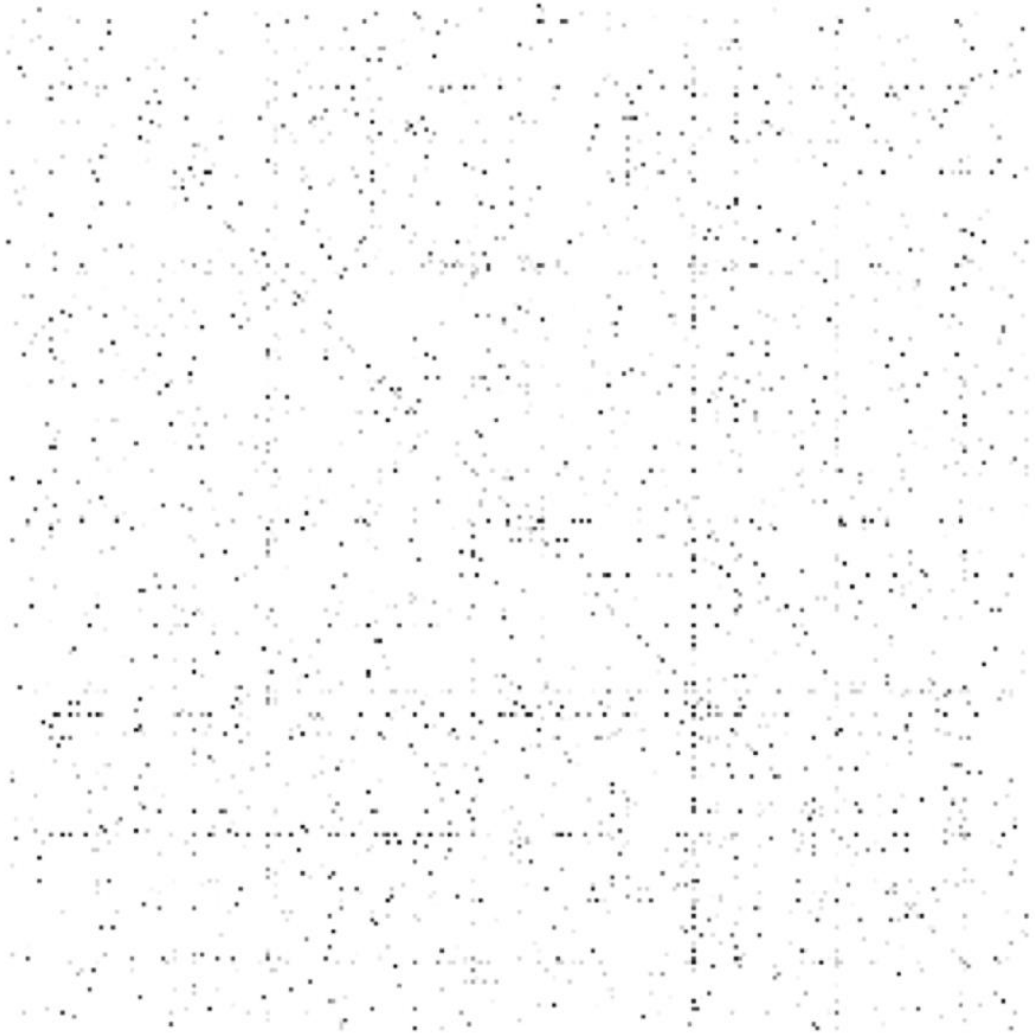
$$A_{ij} = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 3 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \bar{k} = \frac{2E}{N}$$

Examples: Communication, Collaboration

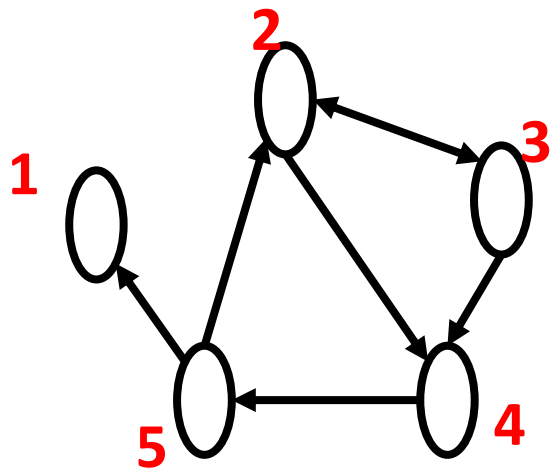
Example of an Adjacency Matrix



Can you identify ”important” node(s) here?

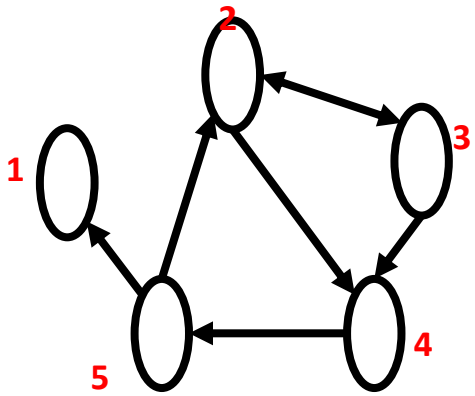
Representing Graphs: Edge list

- **Represent graph as a set of edges:**
  - (2, 3)
  - (2, 4)
  - (3, 2)
  - (3, 4)
  - (4, 5)
  - (5, 2)
  - (5, 1)



Representing Graphs: Adjacency list

- **Adjacency list:**
  - Easier to work with if network is
    - Large
    - Sparse
  - Allows us to quickly retrieve all neighbors of a given node
    - 1:
    - 2: 3, 4
    - 3: 2, 4
    - 4: 5
    - 5: 1, 2

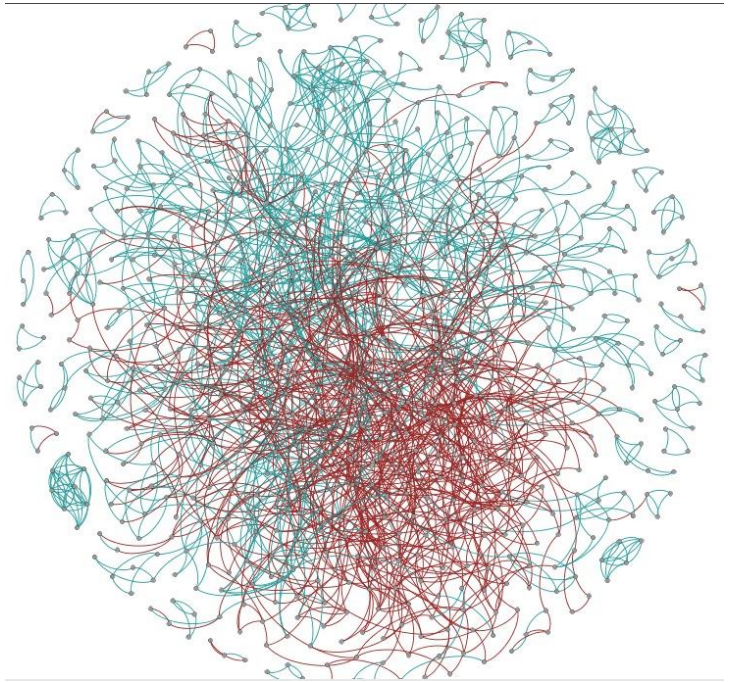


Edge Attributes

Possible options:

- Weight (e.g. frequency of communication)
- Ranking (best friend, second best friend…)
- Type (friend, relative, co-worker)
- Sign: Friend vs. Foe, Trust vs. Distrust
- Properties depending on the structure of the rest of the graph: number of common friends

- **Trust/Distrust Network**
  - Research challenge: How does one ‘propagate’ negative feelings in a social network? Is my enemy’s enemy my friend?



Example Network Representations

Email network >> directed multigraph with self-edges

Facebook friendships >> undirected, unweighted

Citation networks >> unweighted, directed, acyclic

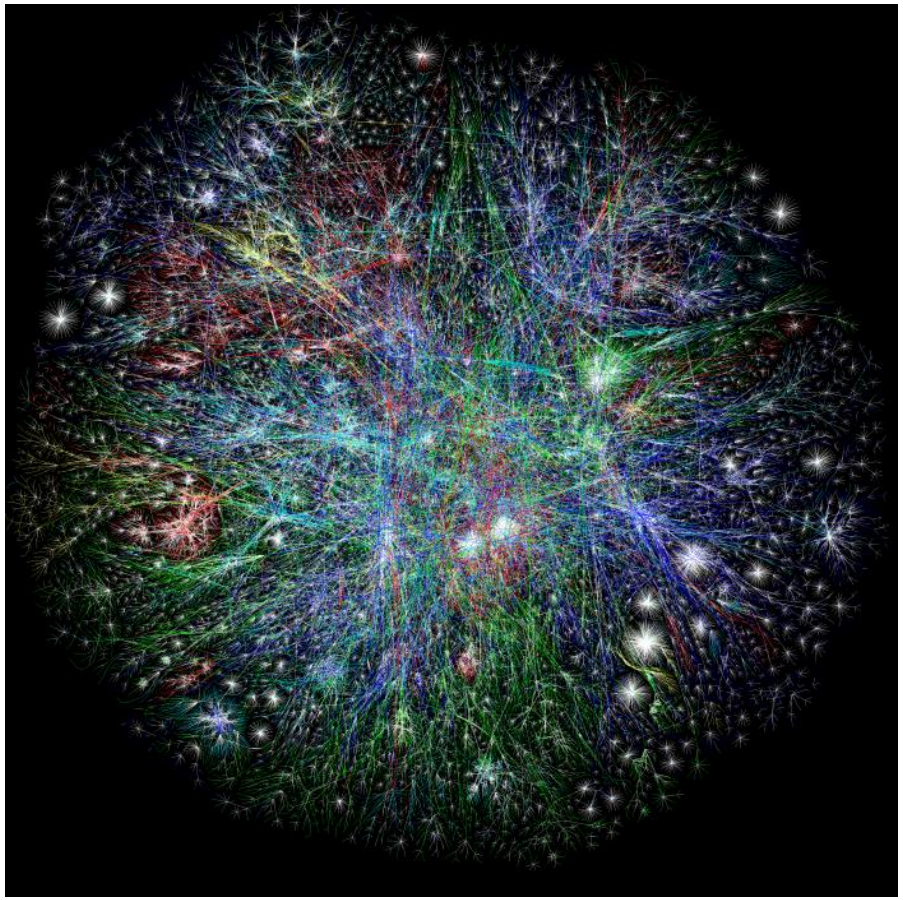
Collaboration networks >> undirected multigraph or weighted graph

Mobile phone calls >> directed, (weighted?) multigraph

Protein Interactions >> undirected, unweighted with self-interactions

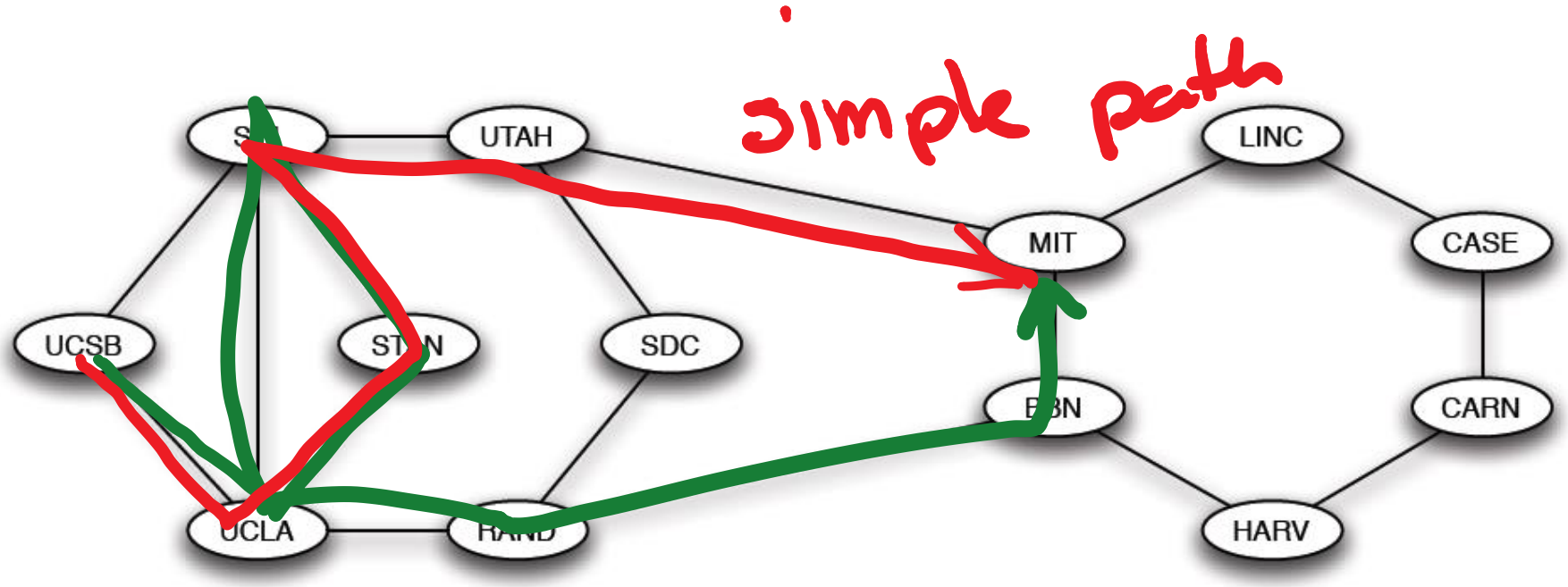
Main Concepts

- Paths
- Cycles
- Connectivity
- (Giant) Components
- Distance

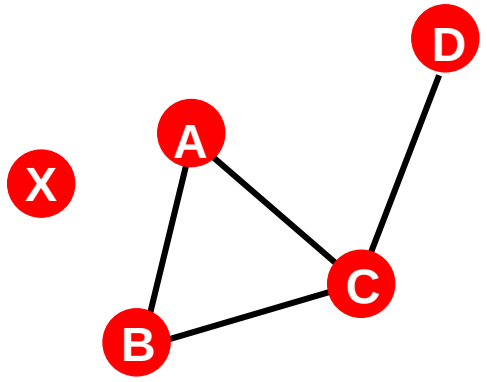


Paths

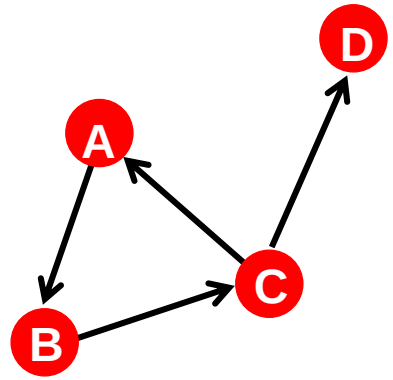
- A ***path*** in a graph is a sequence of nodes with the property that each consecutive pair in the sequence is connected by an edge
  - In a ***simple path*** nodes do not repeat
    - Sometimes can be referred to **Walks** and **Paths**



Distance?



$h_{B,D} = 2$
$h_{A,X} = \infty$



$h_{B,C} = 1$, $h_{C,B} = 2$

- **Distance (Shortest path)** between a pair of nodes is defined as the number of edges along the shortest path connecting the nodes
  - *If the two nodes are not connected, the distance is usually defined as infinite
- In **directed graphs** paths need to follow the direction of the arrows
  - Consequence: Distance is **not symmetric**: $h_{A,C} \neq h_{C,A}$

Breath-First Search

- (1) You first declare all of your actual friends to be at distance 1.
- (2) You then find all of their friends (not counting people who are already friends of yours), and declare these to be at distance 2.
- (3) Then you find all of their friends (again, not counting people who you've already found at distances 1 and 2) and declare these to be at distance 3.
- (...) Continuing in this way, you search in successive layers, each representing the next distance out. Each new layer is built from all those nodes that (i) have not already been discovered in earlier layers, and that (ii) have an edge to some node in the previous layer.

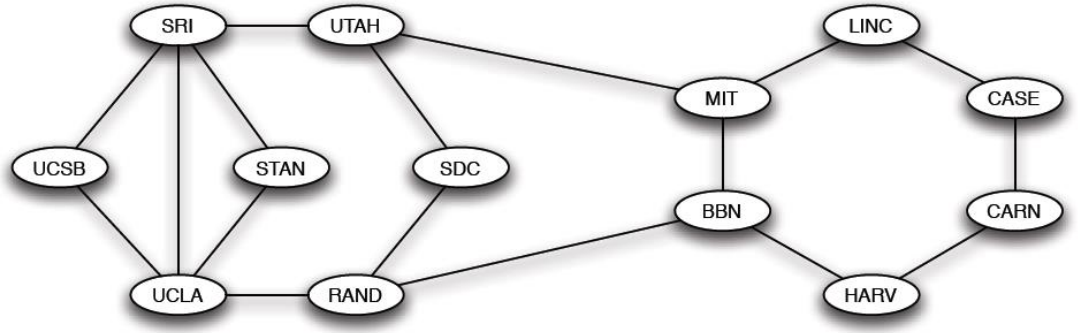
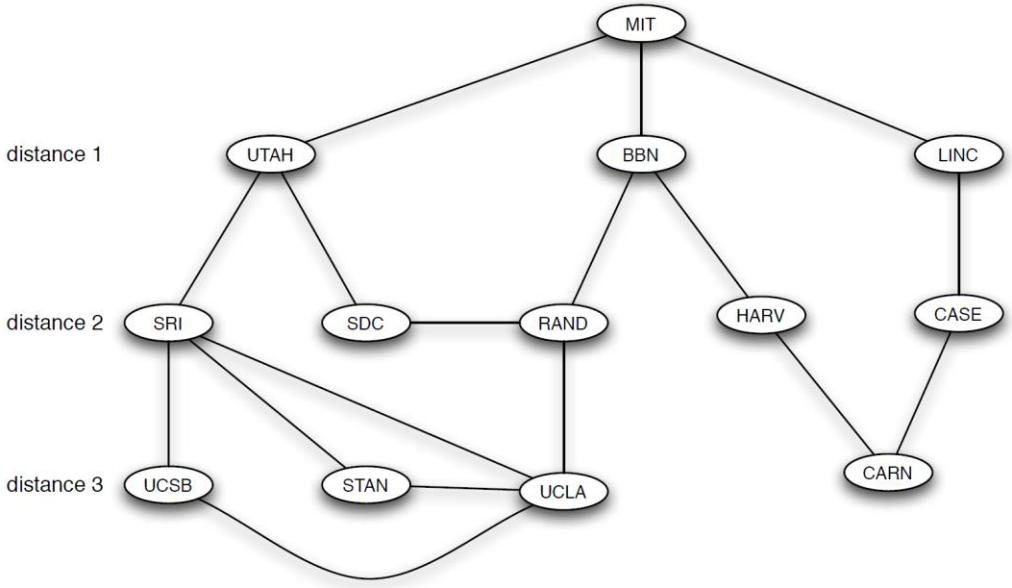


Figure 2.3: An alternate drawing of the 13-node Internet graph from December 1970.



Graph Diameter ?

- Graph **diameter** can be measured as the
  - Largest shortest path
    - Any issue with that?
    - Outliers?
  - Average of shortest paths among all the nodes
    - Less sensitive to outliers
    - Still any issues?
    - Disconnected nodes
- **Average path length** for a connected graph (component) or a strongly connected (component of a) directed graph

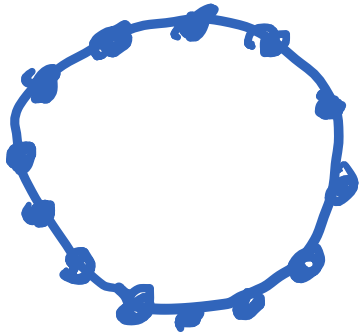
$$\bar{h} = \frac{1}{2E_{\max}} \sum_{i,\, j \neq i} h_{ij}$$

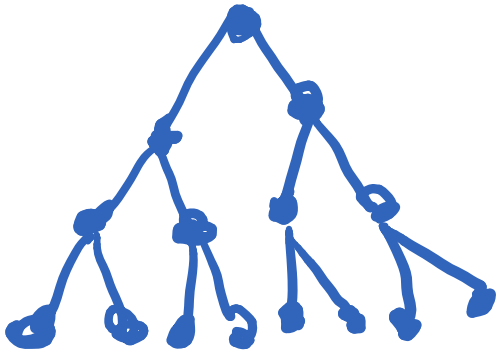
where h_{ij} is the distance from node i to node j

  - Often we compute the average only over the connected pairs of nodes (that is, we ignore "infinite" length paths)

Graph Diameter: Examples ?

- Examples (what are the diameters of these graphs?)





- O(N) O(Log N)
- WWW: 3.1, FB: 4.74, Co-authorship graphs: 5-10
- Usually diameter is considered “small” if it is O(logN).
  - We’ll see later that it will depend on the degree of the network

Cycles

- A ***cycle*** is a closed path with at least three edges
  - All nodes are distinct except the first and the last
  - Why cycles are useful?
    - Every edge in 1970 network belongs to a cycle: design choice for making network connected even if one link failed.

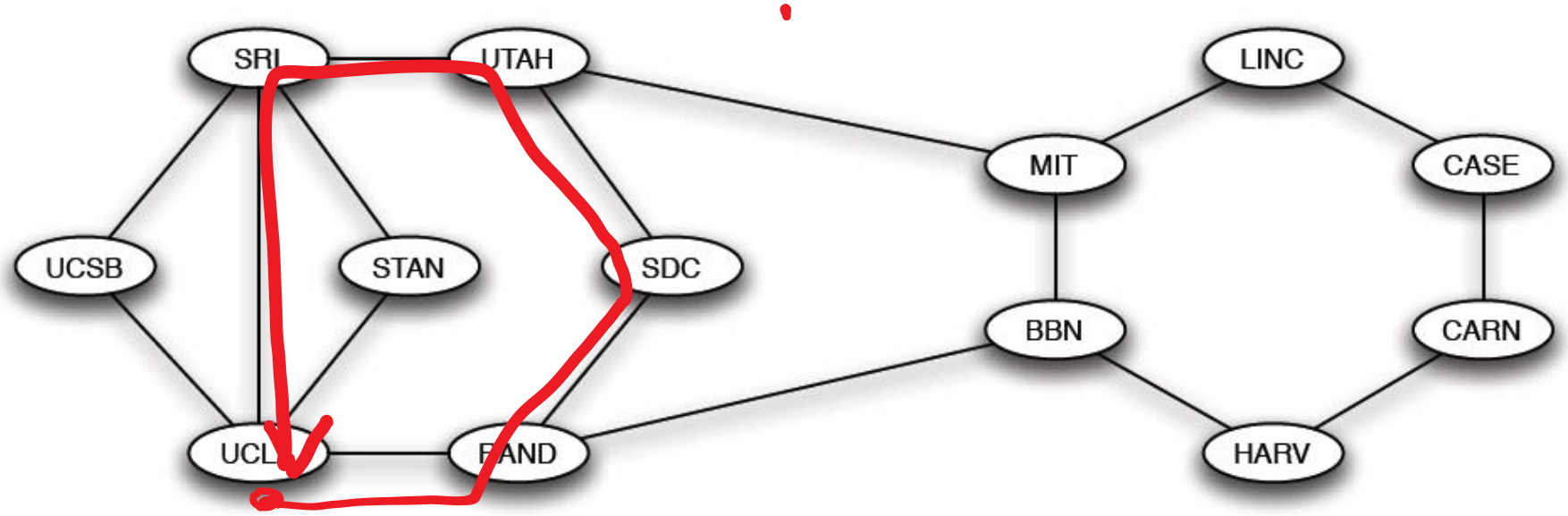
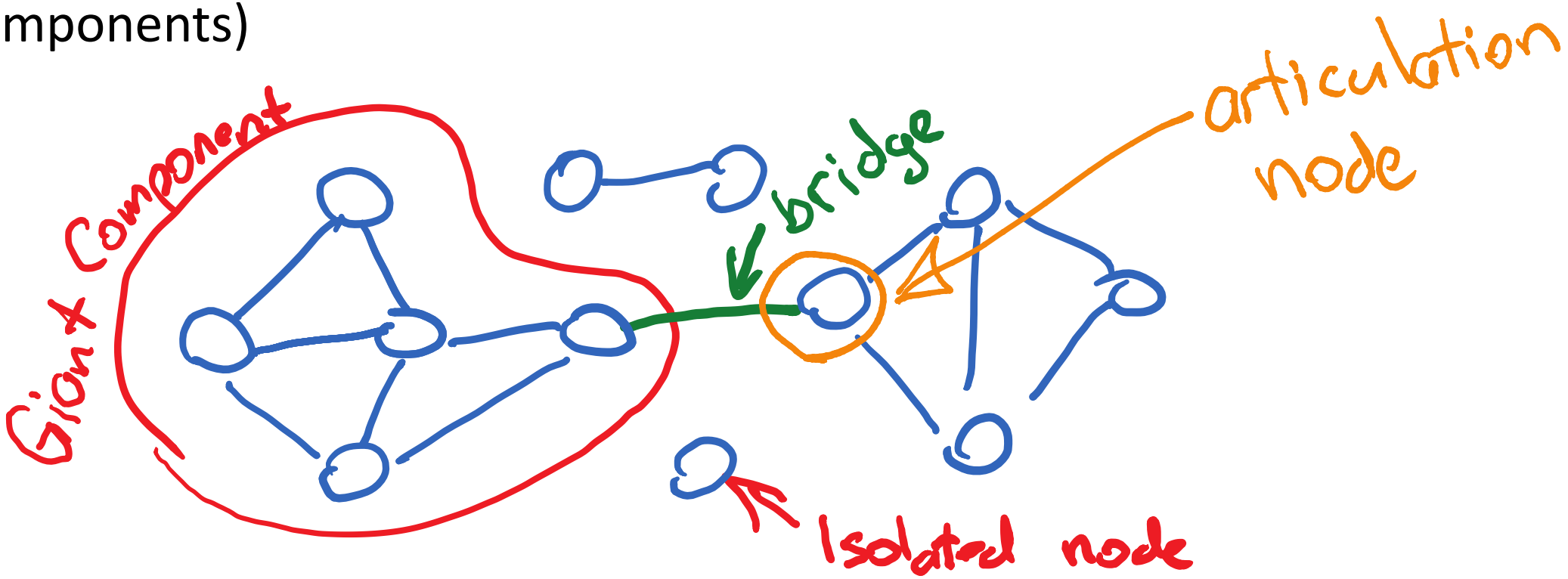


Figure 2.3: An alternate drawing of the 13-node Internet graph from December 1970.

Connectivity

- A graph is ***connected*** if for every pair of nodes there is a path between them.
- A ***disconnected*** graph is made of at least two connected sub-graphs (components)



Connectivity (cont.)

- Local bridge
  - AB edge is a local bridge if A and B have no neighbors in common, but there exist another path from A to B.

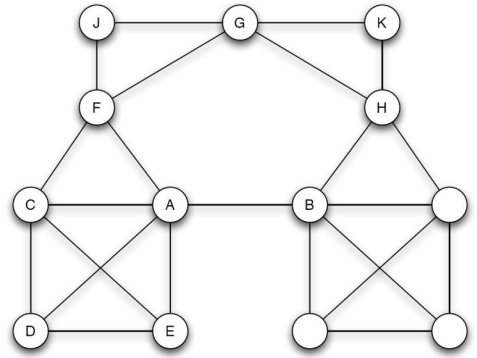


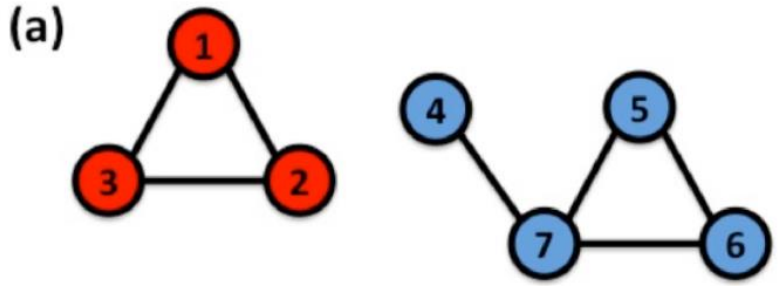
Figure 3.4: The A-B edge is a local bridge of span 4, since the removal of this edge would increase the distance between A and B to 4.

- Embeddedness of the edge
  - number of mutual friends that the endpoints of the edge have in common.
  - What's the embeddedness of a local bridge?
    - Bridges have embeddedness of zero.

Connectivity:Adjecency Matrix example

- How would the adjecency matrices look like for these graphs?:

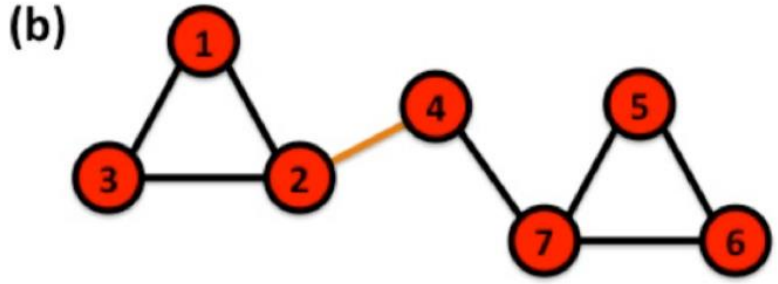
Disconnected



No links from Red to Blue

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

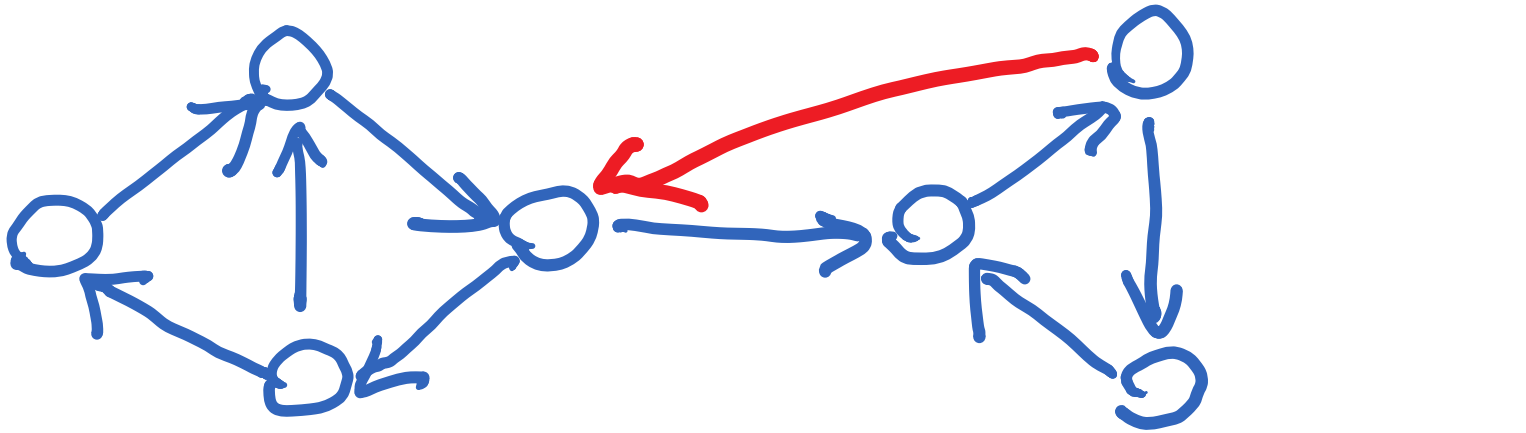
Connected



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

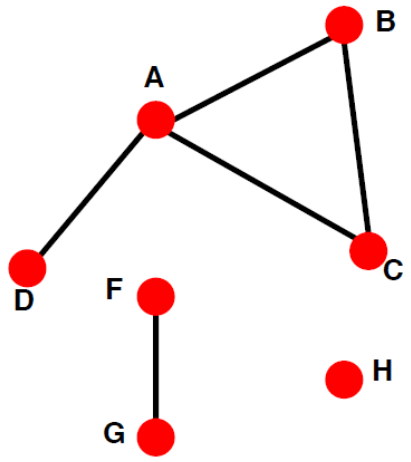
Connectivity (Directed Graphs)

- A directed graph is ***strongly connected*** if for every pair of nodes there is a path between them. A ***weakly connected*** graph is connected if we disregard edge directions



Giant Component

- A ***connected component*** of a graph is a subset of the nodes such that:
  - (i) every node in the subset has a path to every other;
  - and (ii) the subset is not part of some larger set with the property that every node can reach every other.
- ***Giant component***: a connected component with the largest number of nodes



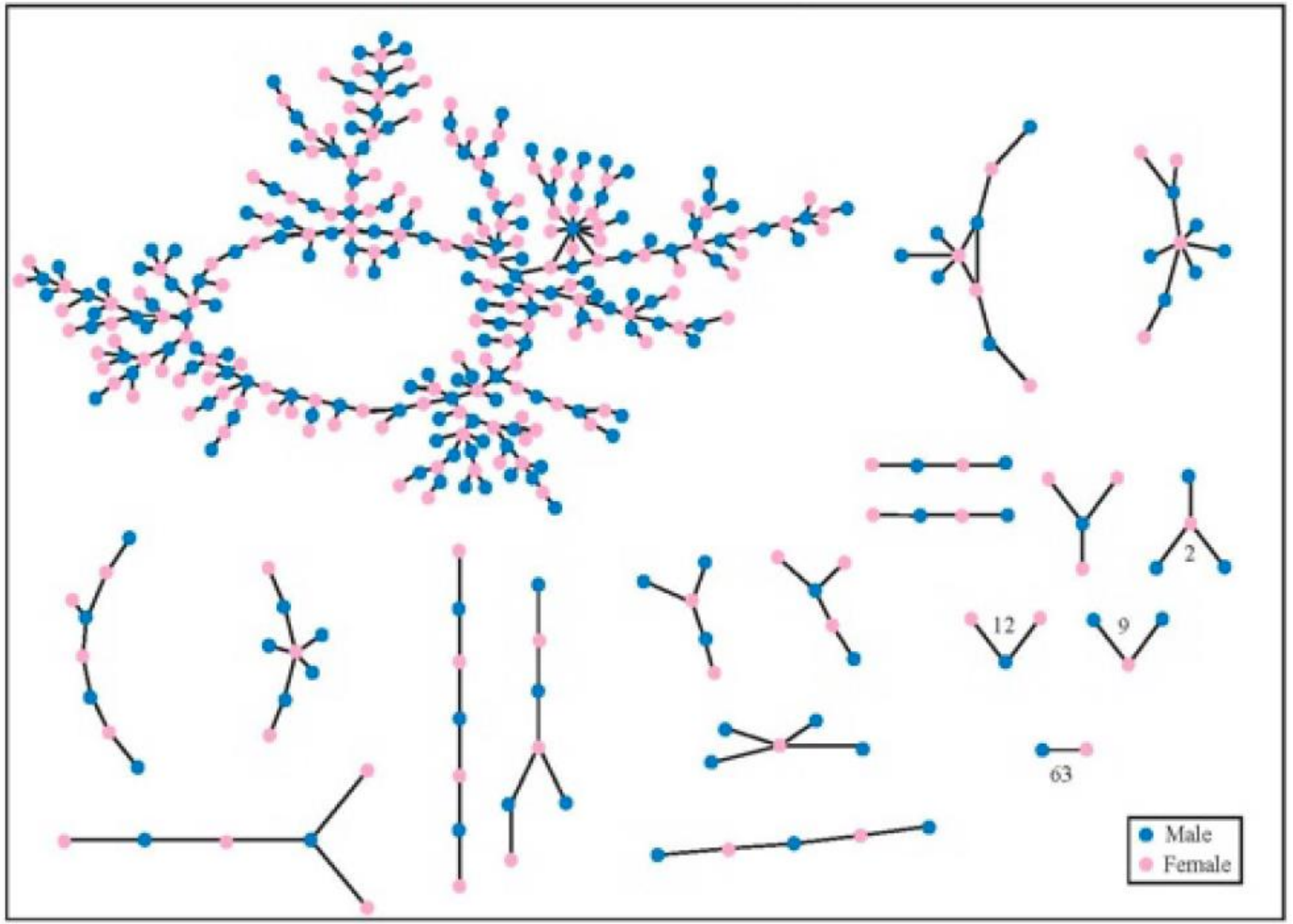
How to find connected components:

- Start from random node and perform Breadth First Search (BFS)
- Label the nodes BFS visited
- If all nodes are visited, the network is connected
- Otherwise find an unvisited node and repeat BFS

Giant Components

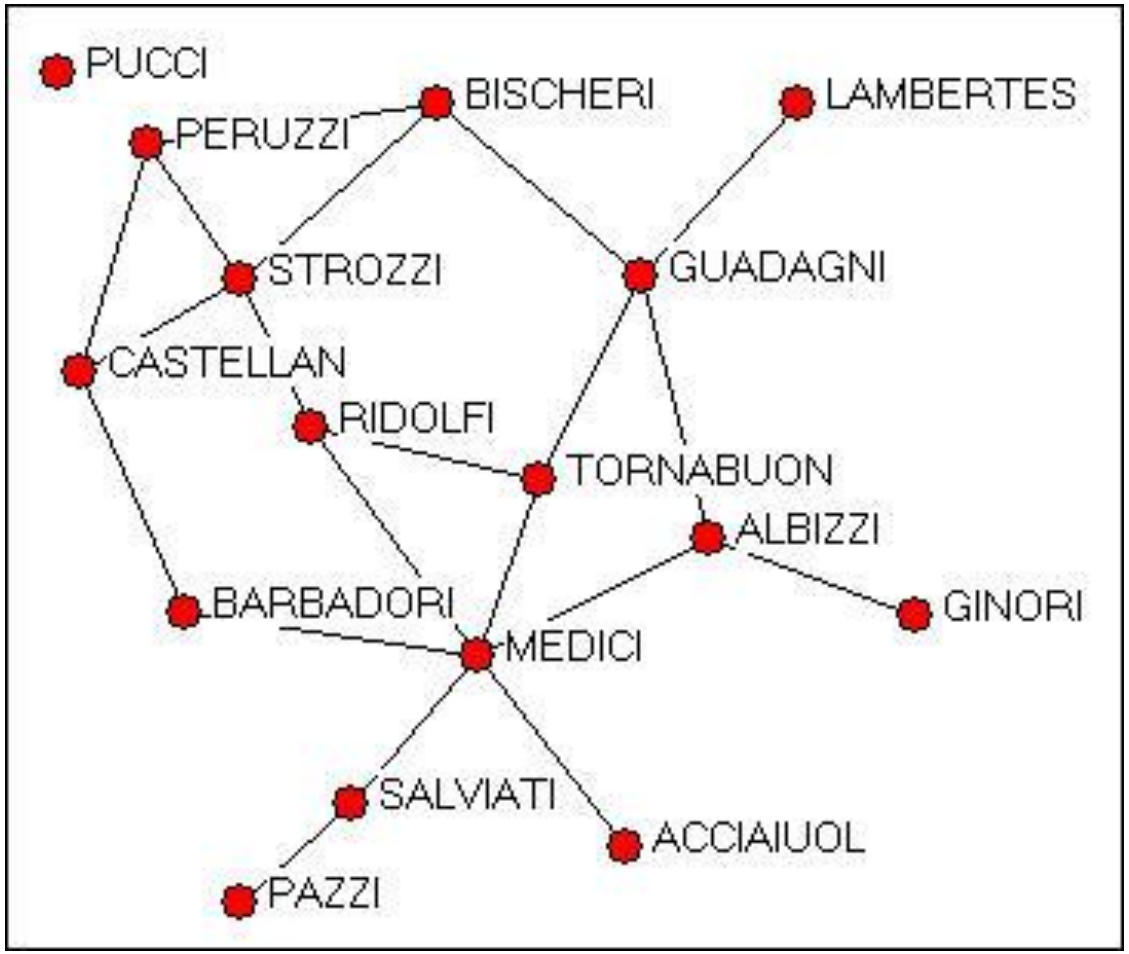
- Real World networks often contain only a specific number of **largest** components that are **similar** in size.
  - Think what could this number be?
- Real world networks often contain only **one giant component**

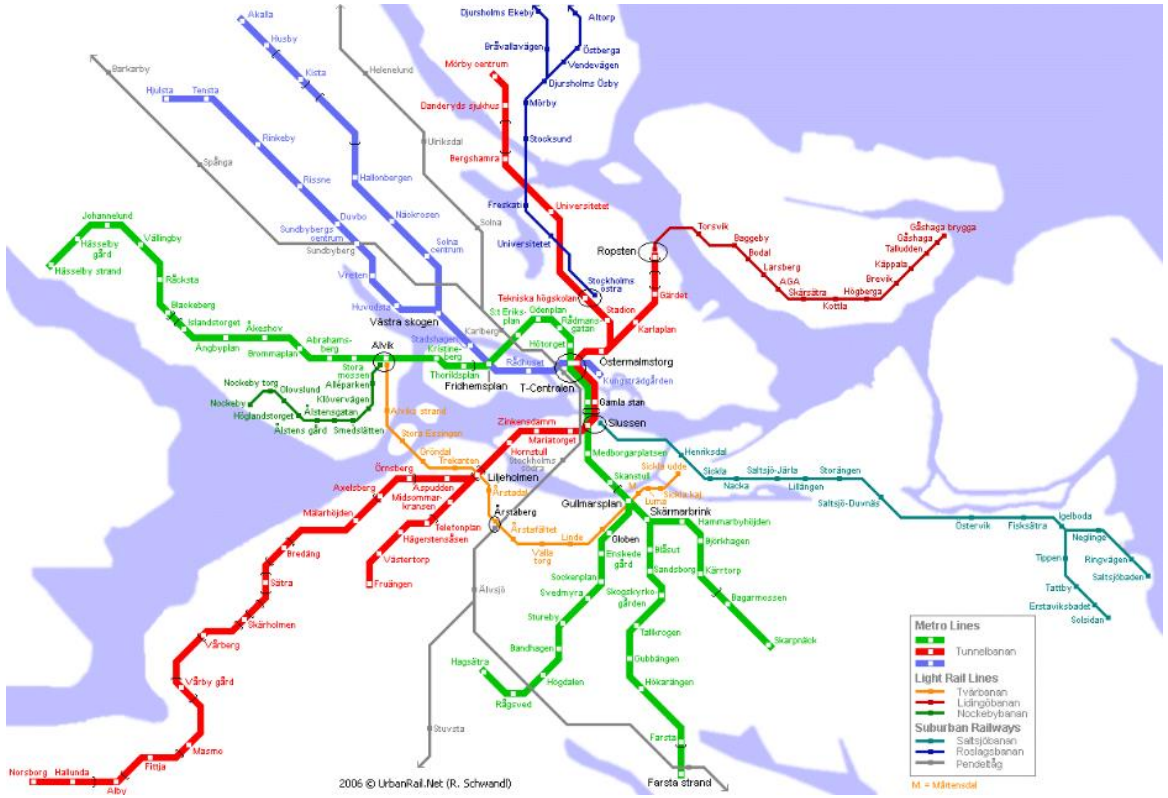
why?



- http://ccl.northwestern.edu/netlogo/models/GiantComponent

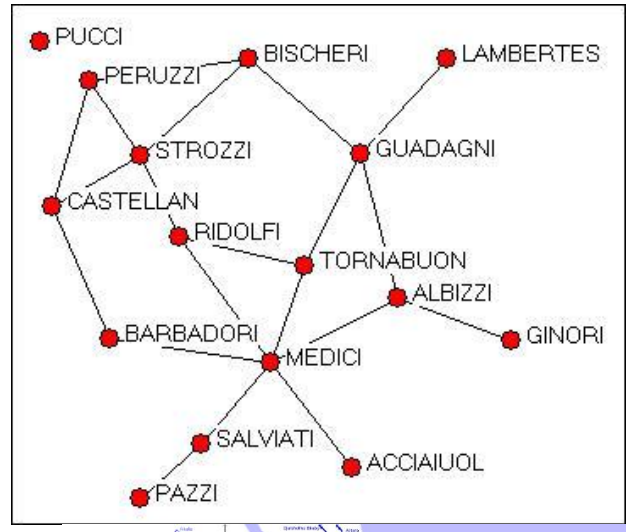
Padgett' s Florentine Families

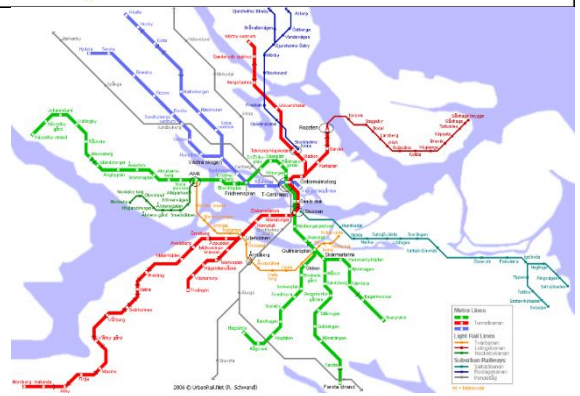




Which vertices are important?

- What if you can't draw the network (like metro map, or Medicci graph)?
- What if you want to rank the nodes?
  - **Centrality measures**
- Which centrality measures do you know?
- There are many centrality measures
  - **Page rank** the most prominent approach that deals with directed graphs.





Which vertces are important? (cont.)

- Most obvious cases:
  - Star, Line, circle
  - In star case – the central node should always be selected no matter what measure we use.
    - Similarly with line graph.
  - In circle ase – all nodes are equivalent and all centrality measures should give equal value to all the nodes.

Degree Centrality

- Number of nearest neighbours $C_D(i) = k(i)$
  - What if I tell you degree centralities of two nodes from two graphs? E.g, 45 and 67. What does it say to you?
  - Problem: you can not compare nodes across graphs (some grpahs can be much denser than others)
  - How would you fix it?
- Normalized degree centrality $C_D^*(i) = \frac{1}{n-1} C_D(i)$
  - Normalize centrality by the maximal possible degree (i.e,. For a central node in a star)
- Problem with Degree centrality?
  - Does not take graph topology into account!
  - Any ideas how to fix it?

Closeness centrality

- How close a node is to all other nodes in the network

$$C_c(i) = \frac{1}{\sum_j d(i,j)}$$

- Normalized closeness centrality

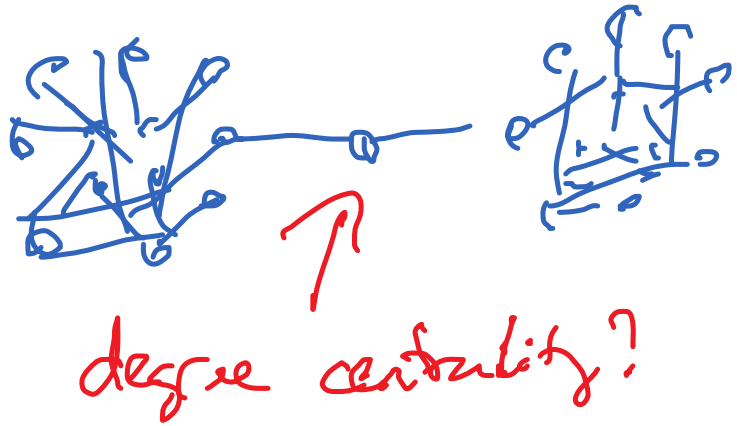
$$C_c^*(i) = (n-1)\, C_c(i)$$

  - (inverse) average distance to all the nodes
- There is a problem though. Anyone sees it`?
  - What if the graph is disconnected. Closeness centrality – only for connected components.
- Remedy:
  - Harmonic centrality

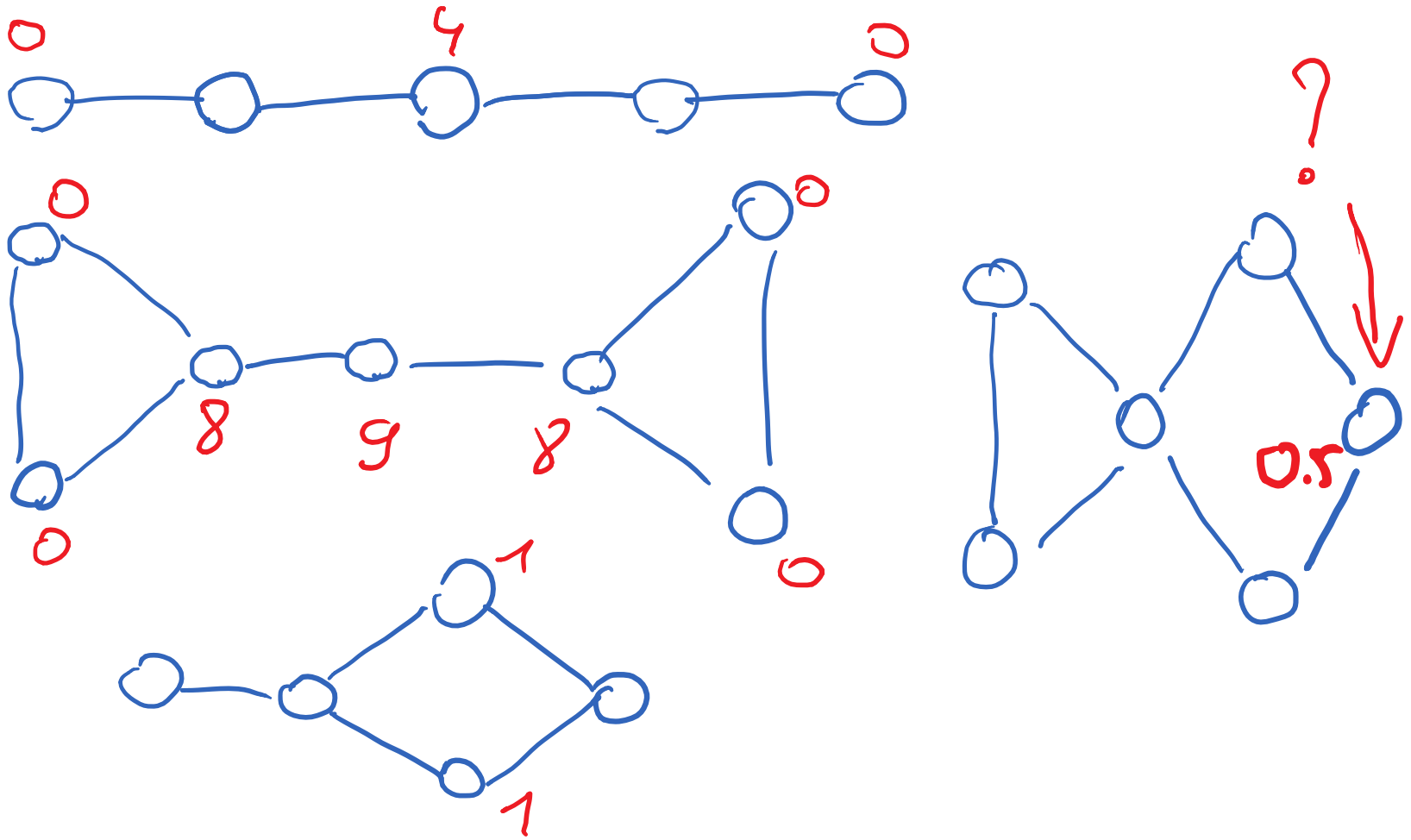
$$C_H(i) = \sum_j \frac{1}{d(i,j)}$$

Betweenness Centrality

- Intuition: ***how many pairs of nodes have a shortest path through you?***
- Betweenness centrality:
  - $C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$
  - σ_{st} number of shortest paths between s and t
  - $\sigma_{st}(v)$ number of shortest paths between s and t via v.
- Can be normalized:
  - $C'_B(v) = \frac{C_B(v)}{(n-1)(n-2)/2}$
- Similarly ***Edge Betweenness*** is the number of shortest paths between pairs of vertices that run along it



Examples



How to find clusters in the network?

- **The Girvan-Newman Method**: Successively Deleting Edges of High Betweenness
  - (1) **Find the edge of highest betweenness** — or multiple edges of highest betweenness, if there is a tie — and **remove these edges from the graph**. This may cause the graph to separate into multiple components. If so, this is the first level of regions in the partitioning of the graph.
  - (2) **Now recalculate all betweennesses**, and again **remove the edge or edges** of highest betweenness. This may break some of the existing components into smaller components; if so, these are regions nested within the larger regions.
  - (...) **Proceed in this way as long as edges remain in graph**, in each step recalculating all betweennesses and removing the edge or edges of highest betweenness.

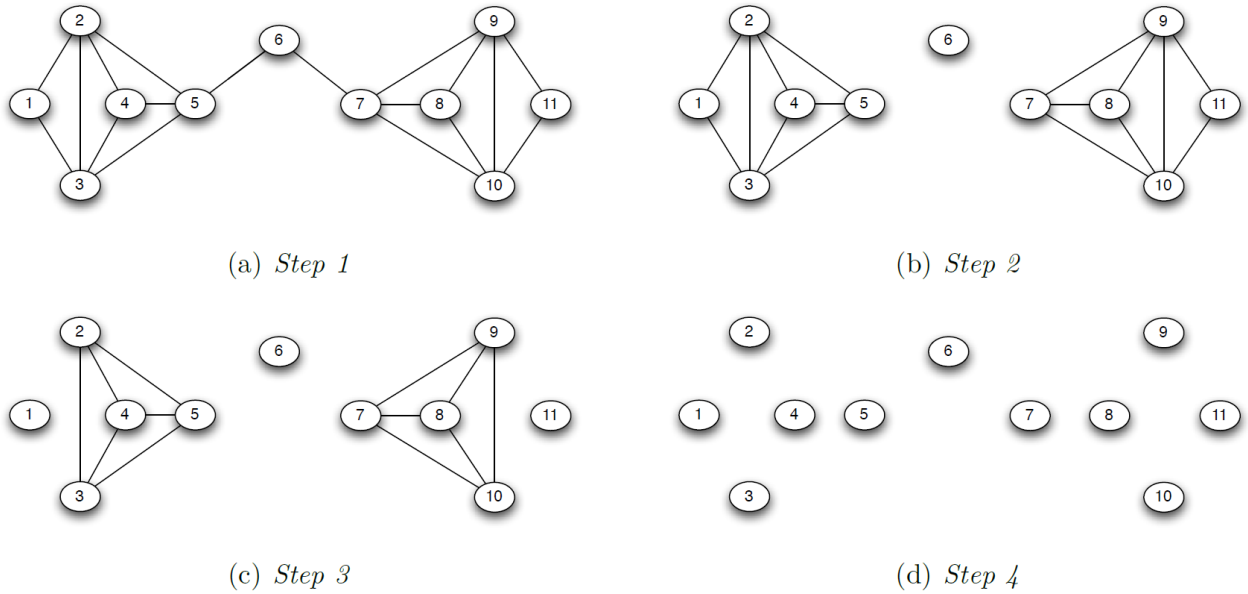


Figure 3.17: The steps of the Girvan-Newman method on the network from Figure 3.15.