Paths, transitivity

Network science (I606)

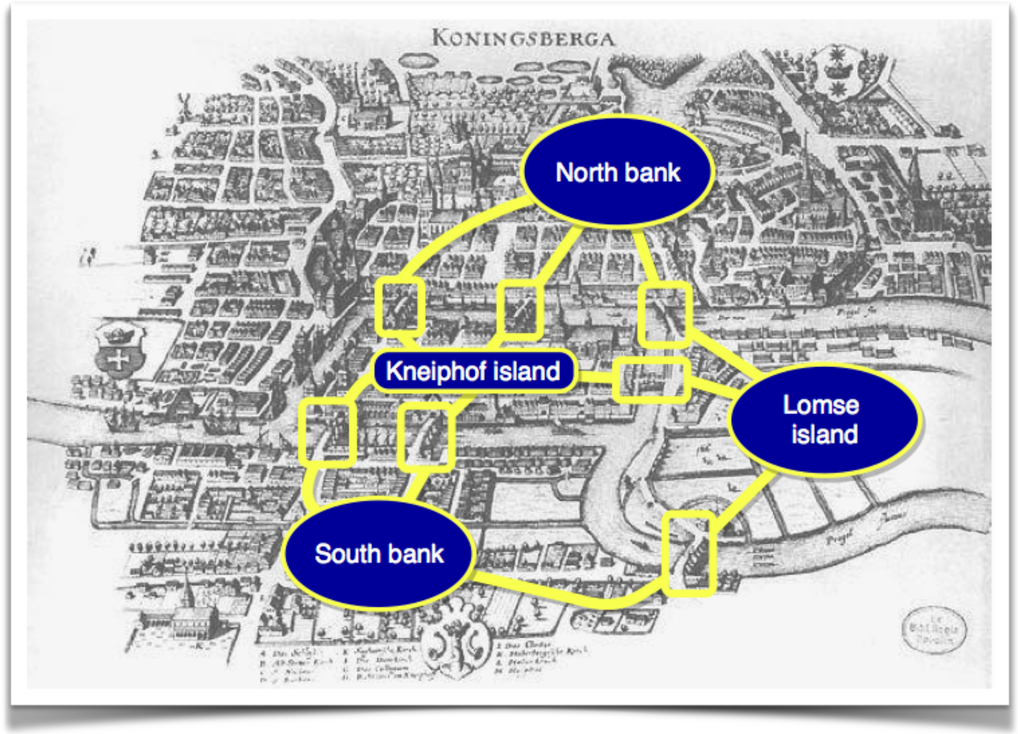
Outline

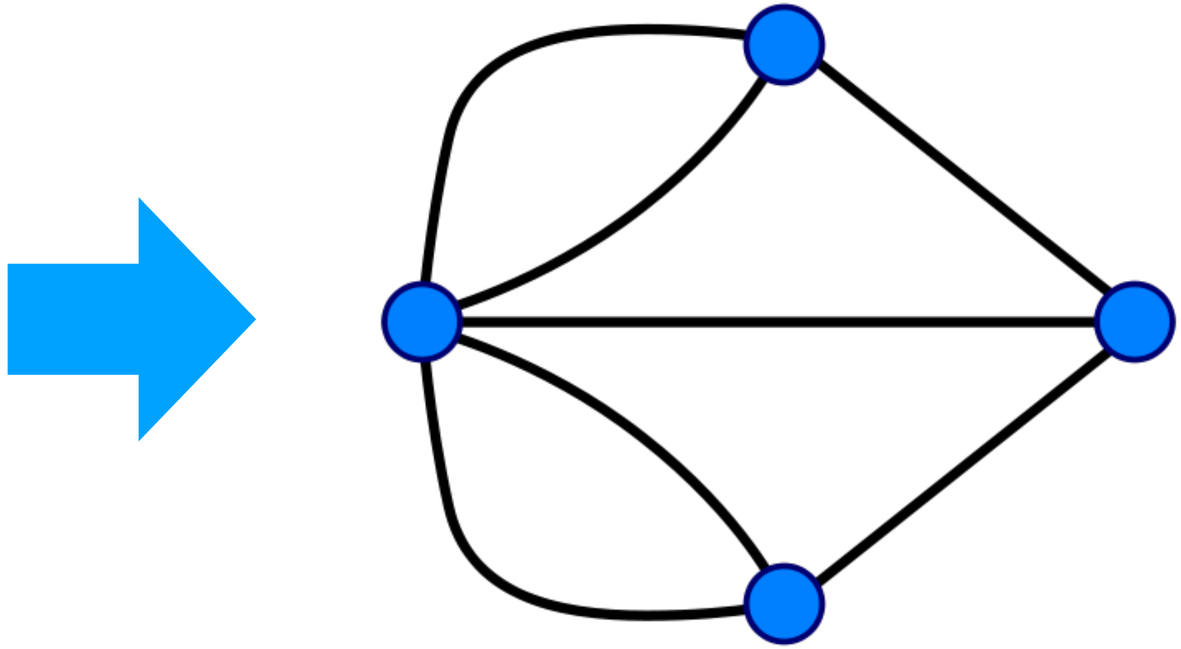
- Paths and distances
- Connectedness and components
- Trees
- Finding shortest paths
- Social distance
- Six degrees of separation
- Friend of a friend

Paths: definitions

- **Path**: sequence of links traversed to go from a **source** to a **target** node
  - In a directed network, links must be traversed according to their direction
  - There may not be a path
- **Cycle**: path where source and target node are the same
- **Simple path**: no traversing the same link more than once
  - We will only deal with simple paths
- **Path length**: number of links in path

Euler circa 1736: Koningsberg bridges



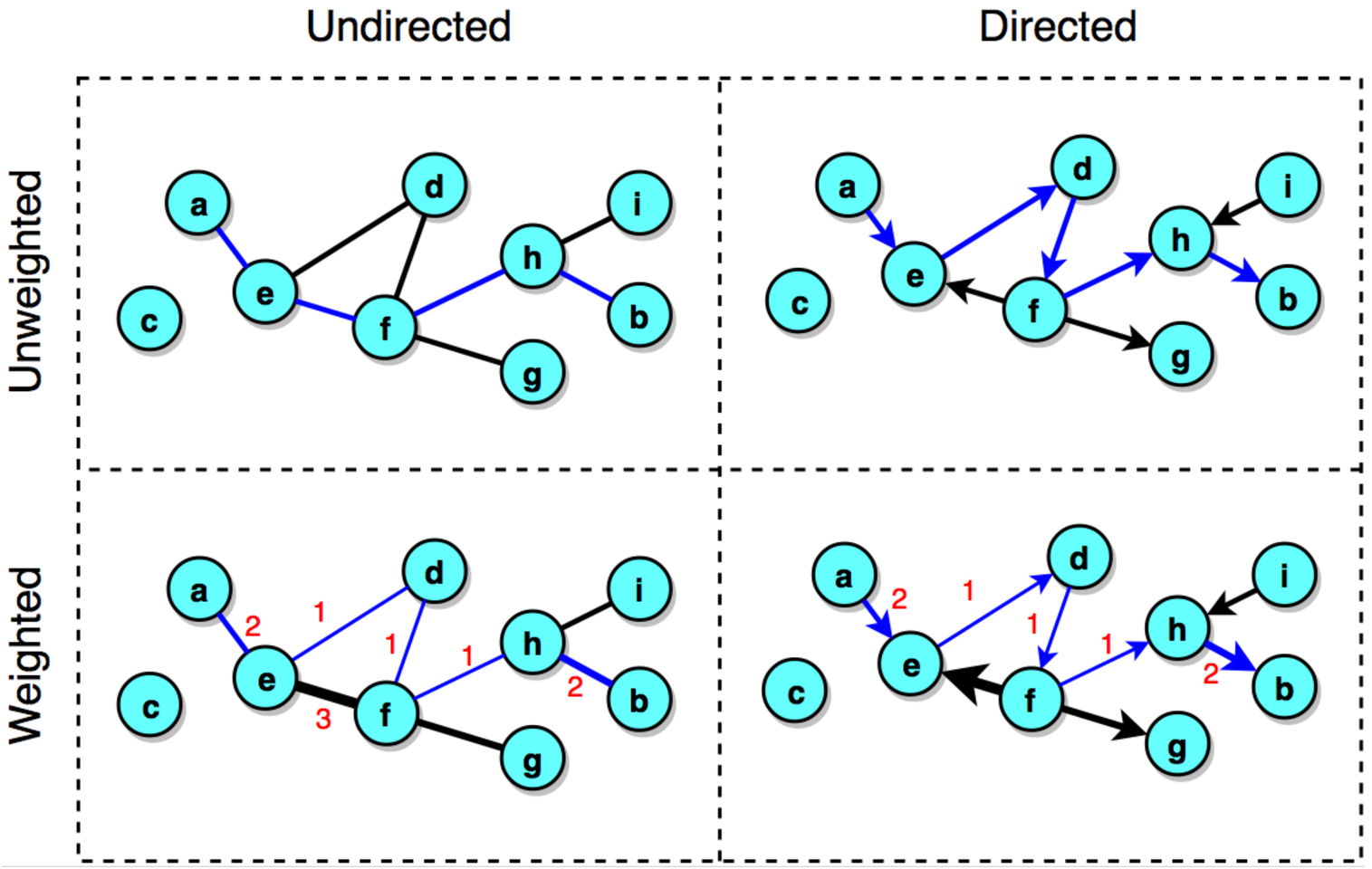


Q: Can you cross all 7 bridges just once each?

A: No. At most two nodes (start, end) may have odd degree

Shortest paths

- **Shortest path** between two nodes: minimal length (there may be more than one)
  - In weighted networks, weights may represent distances
- **Shortest path length** or **distance**: length of shortest path
  - Undefined (∞) if there is no path



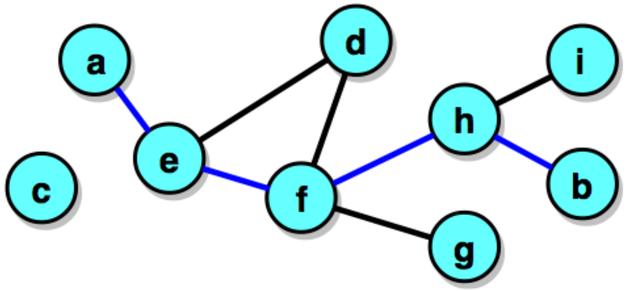
APL and diameter

- We can use the shortest paths to characterize a network:
  - The **diameter** is the longest shortest-path length, or the maximum of the shortest path lengths across all pairs of nodes: $\ell_{max} = \max_{i,j} \ell_{ij}$
  - The **average path length** (APL) is the average of the shortest path lengths across all pairs of nodes
    - Undirected network: $\langle \ell \rangle = \frac{\sum_{i,j} \ell_{ij}}{\binom{N}{2}} = \frac{2\sum_{i,j} \ell_{ij}}{N(N-1)}$
    - Directed network: $\langle \ell \rangle = \frac{\sum_{i,j} \ell_{ij}}{N(N-1)}$

APL and diameter

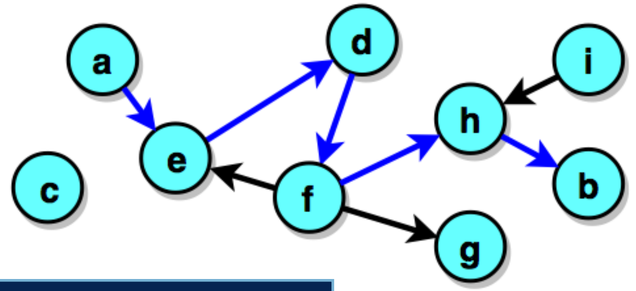
- What if there is no path between one or more pairs of nodes?
  - We can say APL and diameter are undefined (as NetworkX does)
  - We can measure APL and diameter within the largest connected component (defined later)
  - We can use a mathematical trick: $\langle \ell \rangle = \left(\frac{\sum_{i,j} \frac{1}{\ell_{ij}}}{\binom{N}{2}} \right)^{-1}$

Paths and APL

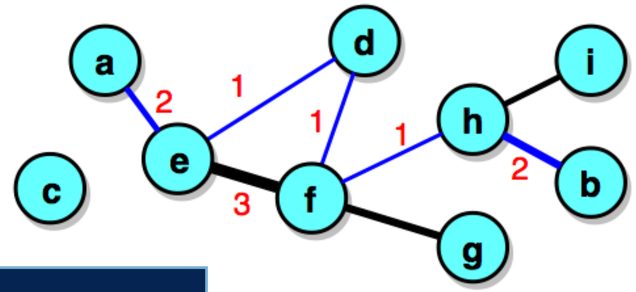



```
nx.has_path(G, 'a', 'c')                # False
nx.has_path(G, 'a', 'b')                # True
nx.shortest_path(G, 'a', 'b')           # ['a','e','f',h','b']
nx.shortest_path_length(G,'a','b') # 4
nx.shortest_path(G, 'a')                # dictionary
nx.shortest_path_length(G, 'a')     # dictionary
nx.shortest_path(G)                     # all pairs
nx.shortest_path_length(G)              # all pairs
nx.average_shortest_path_length(G) # error
G.remove_node('c')                      # make G connected
nx.average_shortest_path_length(G) # now okay
```

Paths and APL



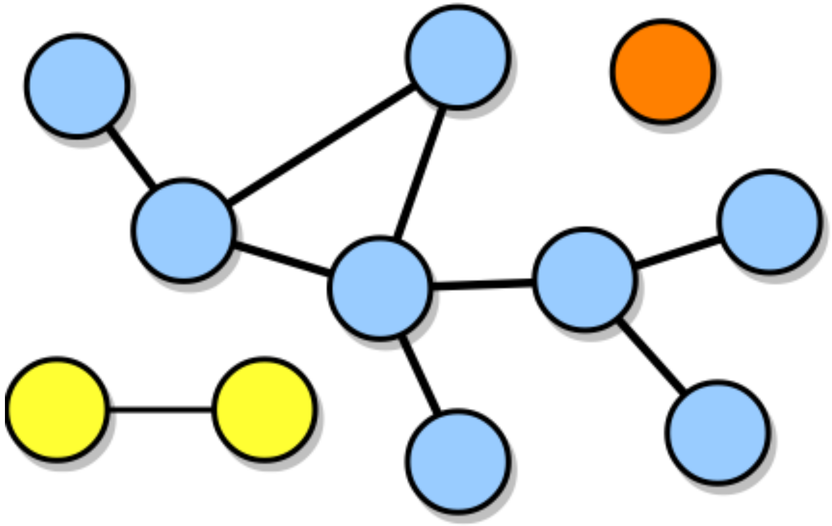

```
nx.has_path(D, 'b', 'a')       # False
nx.has_path(D, 'a', 'b')       # True
nx.shortest_path(D, 'a', 'b') # ['a','e','d','f',h','b']
```




```
nx.shortest_path_length(W, 'a', 'b')           # 4
nx.shortest_path_length(W, 'a', 'b', 'weight') # 7
```

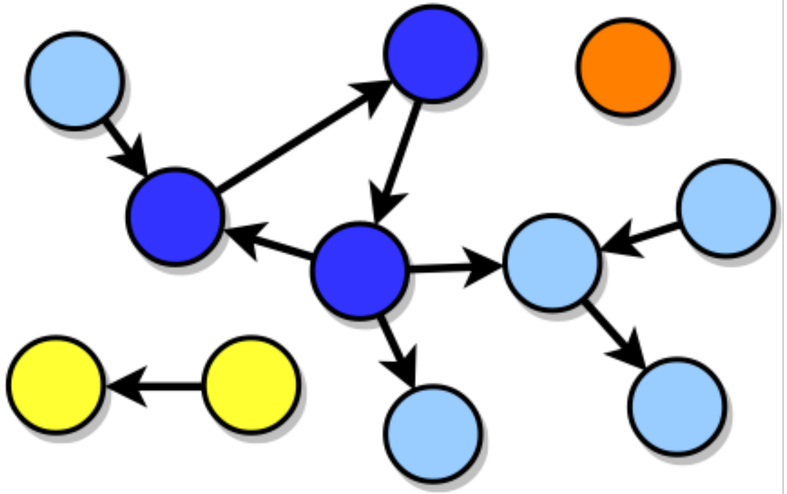
Connectedness and components

- A network is **connected** if there is a path between any two nodes
- If a network is not connected, it is **disconnected** and has multiple connected components
- A **connected component** is a connected subnetwork
  - The largest one is called **giant component**; it often includes a substantial portion of the network
  - A singleton is the smallest-possible connected component

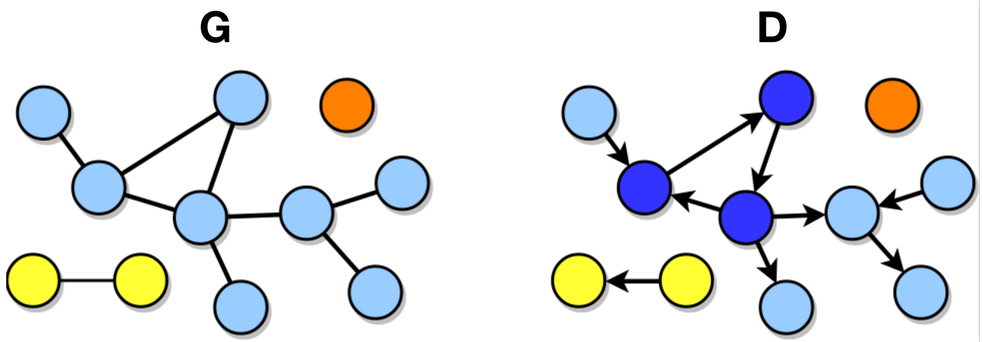


Connectedness and components

- A directed network can be **strongly connected** or **weakly connected** if there is a path between any two nodes, respecting or disregarding the link directions, respectively
- Similarly for **strongly connected** or **weakly connected** components
- The **in-component** of a strongly connected component *S* is the set of nodes from which one can reach *S*, but that cannot be reached from *S*
- The **out-component** of a strongly connected component *S* is the set of nodes that can be reached from *S*, but from which one cannot reach *S*



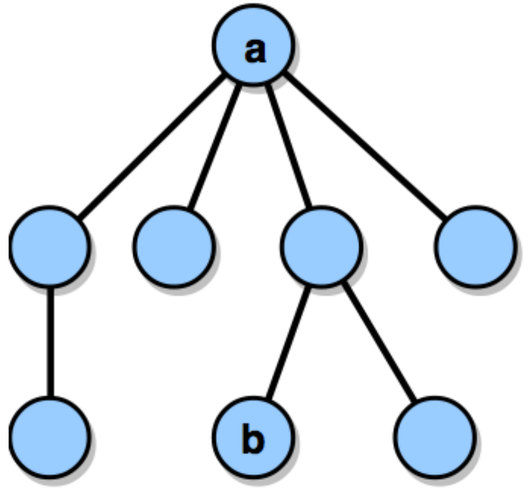
Connectedness and components

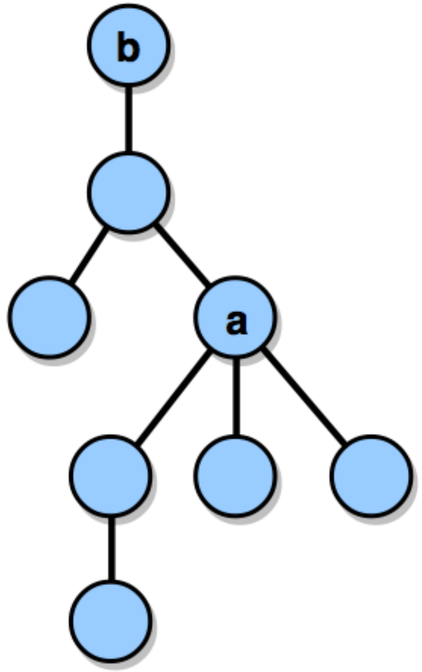



```
nx.is_connected(G)                                  # False
comps = sorted(nx.connected_components(G),
               key=len, reverse=True)
nodes_in_giant_comp = comps[0]
GC = nx.subgraph(G, nodes_in_giant_comp)
nx.is_connected(GC)                                 # True
nx.is_strongly_connected(D)                         # False
nx.is_weakly_connected(D)                           # False
list(nx.weakly_connected_components(D))
list(nx.strongly_connected_components(D))           # lots of
                                                    # singletons
```

Trees

- A **tree** is a **connected** network **without cycles**
- A **tree** is a **connected** network **with *N*-1 links**
- Exercise: prove that these two definitions are equivalent
- In a tree there is a single path between any two nodes
- Trees are **hierarchical**: you can pick a node as the **root**. Each node is connected to a **parent** node (toward the root) and to one or more **children** nodes (away from the root). Exceptions:
  - The root has no parent
  - The **leaves** have no children

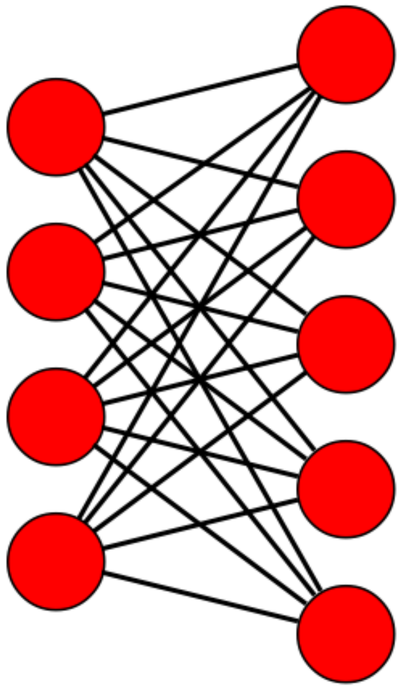




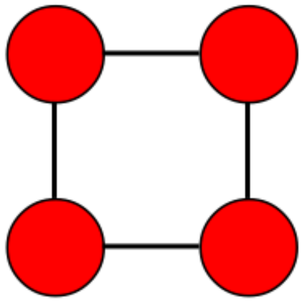
Trees

```
K4 = nx.complete_graph(4)
nx.is_tree(K4)                  # False

nx.is_tree(B)                   # False
nx.is_tree(C)                   # False

nx.is_tree(S)                   # True
nx.is_tree(P)                   # True
```

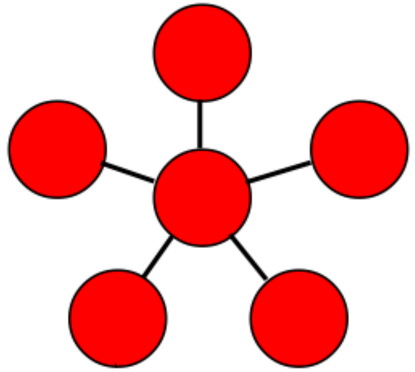
B = nx.complete_bipartite_graph(4,5)



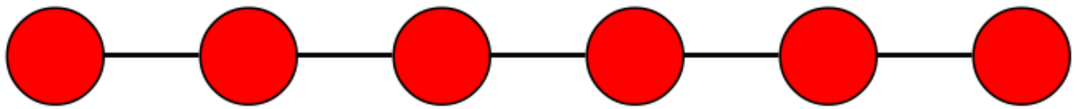
C = nx.cycle_graph(4)



S = nx.star_graph(6)

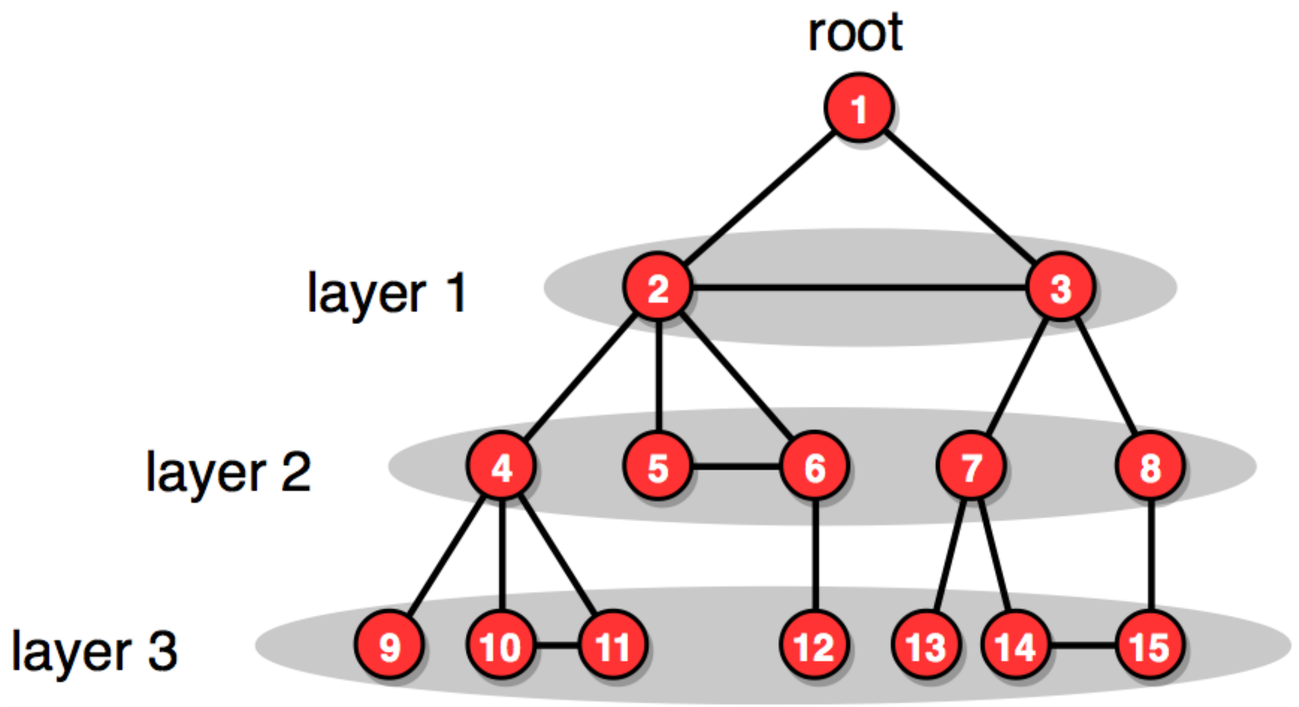


P = nx.path_graph(5)



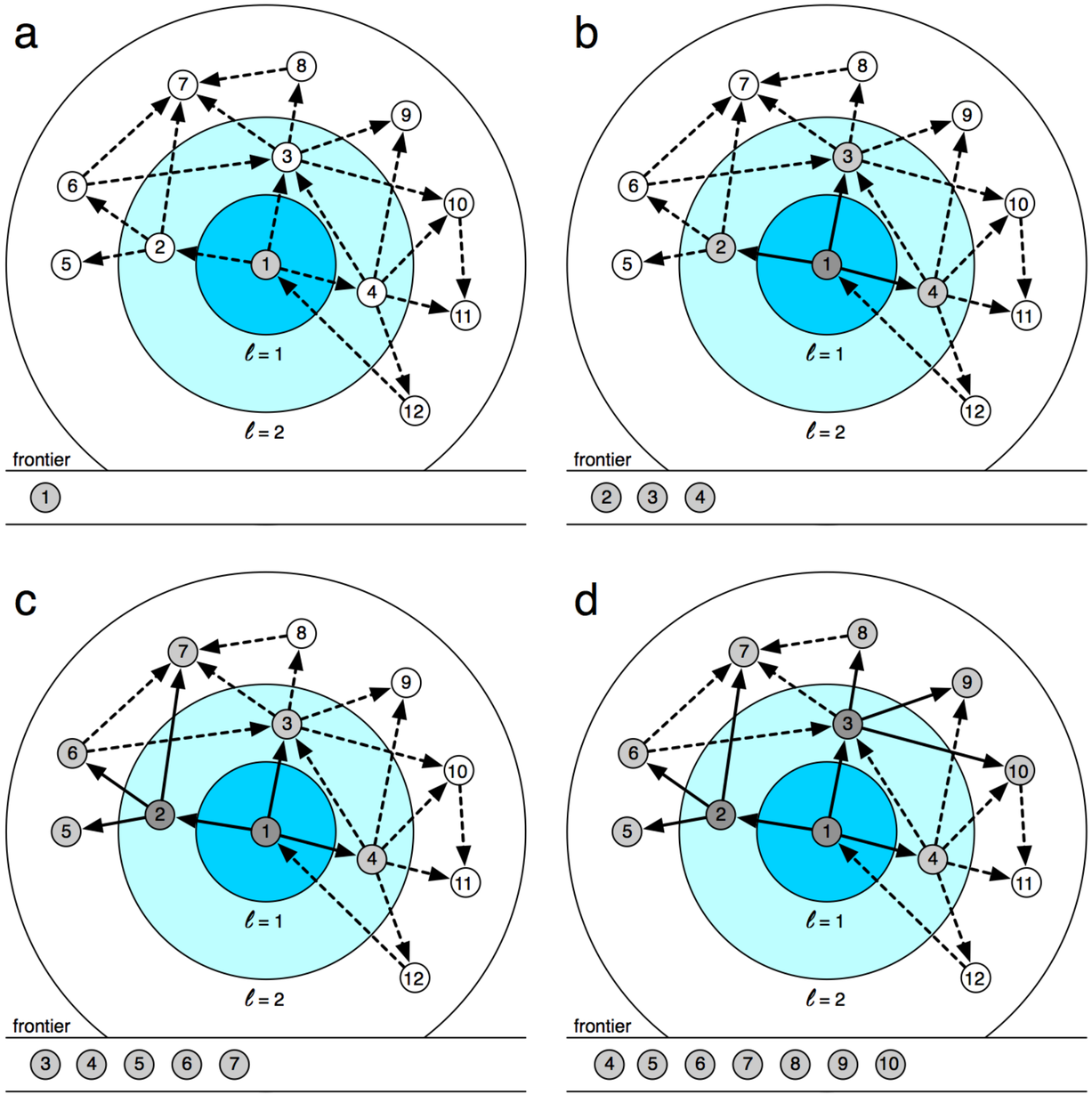
Finding shortest paths

- The algorithm used to find shortest paths is called **breadth-first search**
- Start from a source node (root)
- Visit the entire breadth of the network, within some distance from the source, before we move to a greater depth, farther away from the source
- Start from each node to find all-pairs-shortest-paths (slow: $O(N^2)$)



Breadth-first search (BFS)

- Each node has an attribute storing its **distance** ℓ from the source, initially $\ell = -1$ except $\ell(\text{source}) = 0$
- A queue (FIFO) holds the **frontier**, initially contains the source
- A directed **shortest path tree**, initially all the nodes and no links
- Iterate until the frontier is empty:
  1. Remove next node *i* in frontier
  2. For each neighbor/successor *j* of *i* with $\ell(j) = -1$:
     1. Queue *j* into frontier
     2. $\ell(j) = \ell(i) + 1$
     3. Add link $(i \rightarrow j)$ to shortest-path tree



Social distance

- How close or far are two nodes in a network?
- As we have seen, this is a question about the average path length
- The question has been explored extensively in social networks
- Let us start by considering coauthorship networks, in which nodes are scholars and links represent two people having coauthored one or more publications

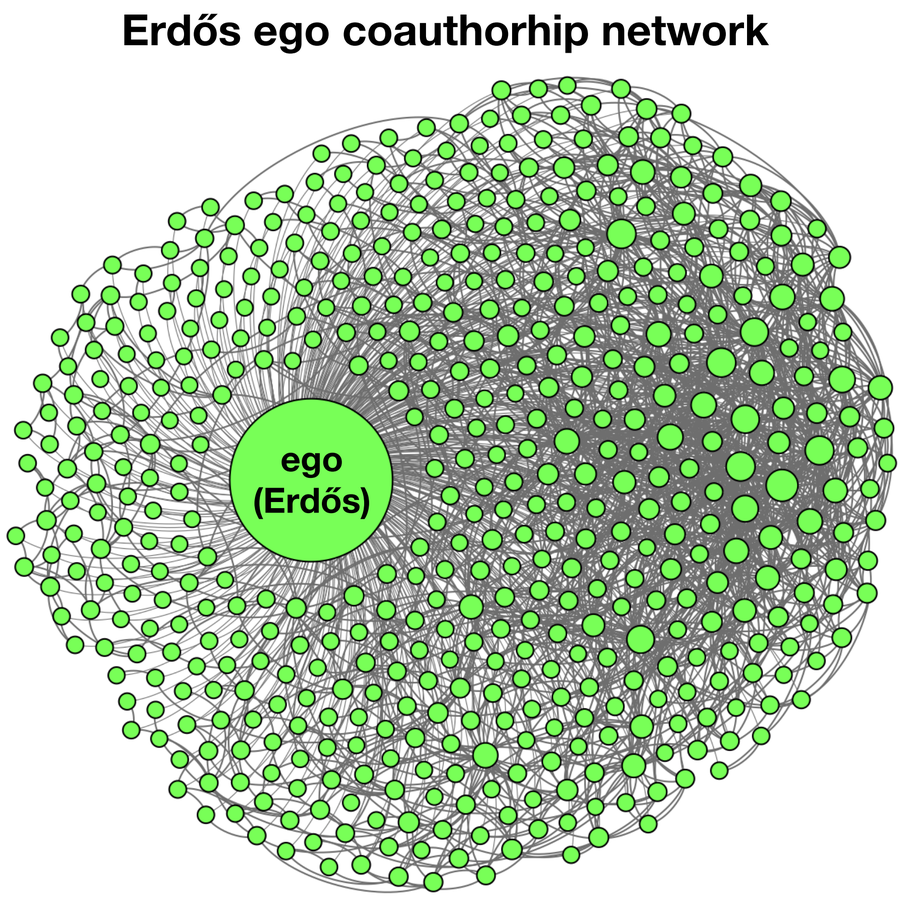
Paul Erdős

- One of the world's greatest mathematicians
- Considered the father of random graph theory together with Alfréd Rényi
- He collaborated with over 500 coauthors: a hub in the coauthorship network!

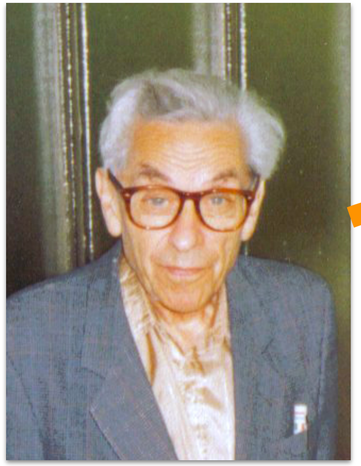
Erdős number

- An author's Erdős number is the length of the shortest path between them and Erdős in the coauthorship network
- Many mathematicians are proud to have a small Erdős number
- Tool to compute one's Erdős number:

mathscinet.ams.org/mathscinet/collaborationDistance.html



Erdős numbers



Paul Erdős
(Image by Kmhkmh - CC BY 3.0 commons.wikimedia.org/w/index.php?curid=38087162)

0

Fan Chung (Image by Che Graham - CC BY 3.0 commons.wikimedia.org/w/index.php?curid=19475582)

Journal of Graph Theory

Article | Full Access

Highly irregular graphs†

Yousef Alavi, Gary Chartrand, F. R. K. Chung, Paul Erdös, R. L. Graham, Ortrud R. Oellermann

1

The Workshop on Internet Topology (WIT) Report

Dmitri Krioukov CAIDA dima@caida.org	Fan Chung UCSD fan@math.ucsd.edu	kc claffy CAIDA kc@caida.org
Marina Fomenkov CAIDA marina@caida.org	Alessandro Vespignani Indiana University alexv@indiana.edu	Walter Willinger AT&T Research walter@research.att.com

2

Alessandro Vespignani - Image CC BY-SA 2.0 commons.wikimedia.org/w/index.php?curid=60299947

Topical interests and the mitigation of search engine bias

S. Fortunato, A. Flammini, F. Menczer, and A. Vespignani

PNAS August 22, 2006 103 (34) 12684-12689; https://doi.org/10.1073/pnas.0605525103

Communicated by Elinor Ostrom, Indiana University, Bloomington, IN, July 1, 2006 (received for review March 2, 2006)

3

BY EMILIO FERRARA, ONUR VAROL, CLAYTON DAVIS, FILIPPO MENCZER, AND ALESSANDRO FLAMMINI

The Rise of Social Bots

4

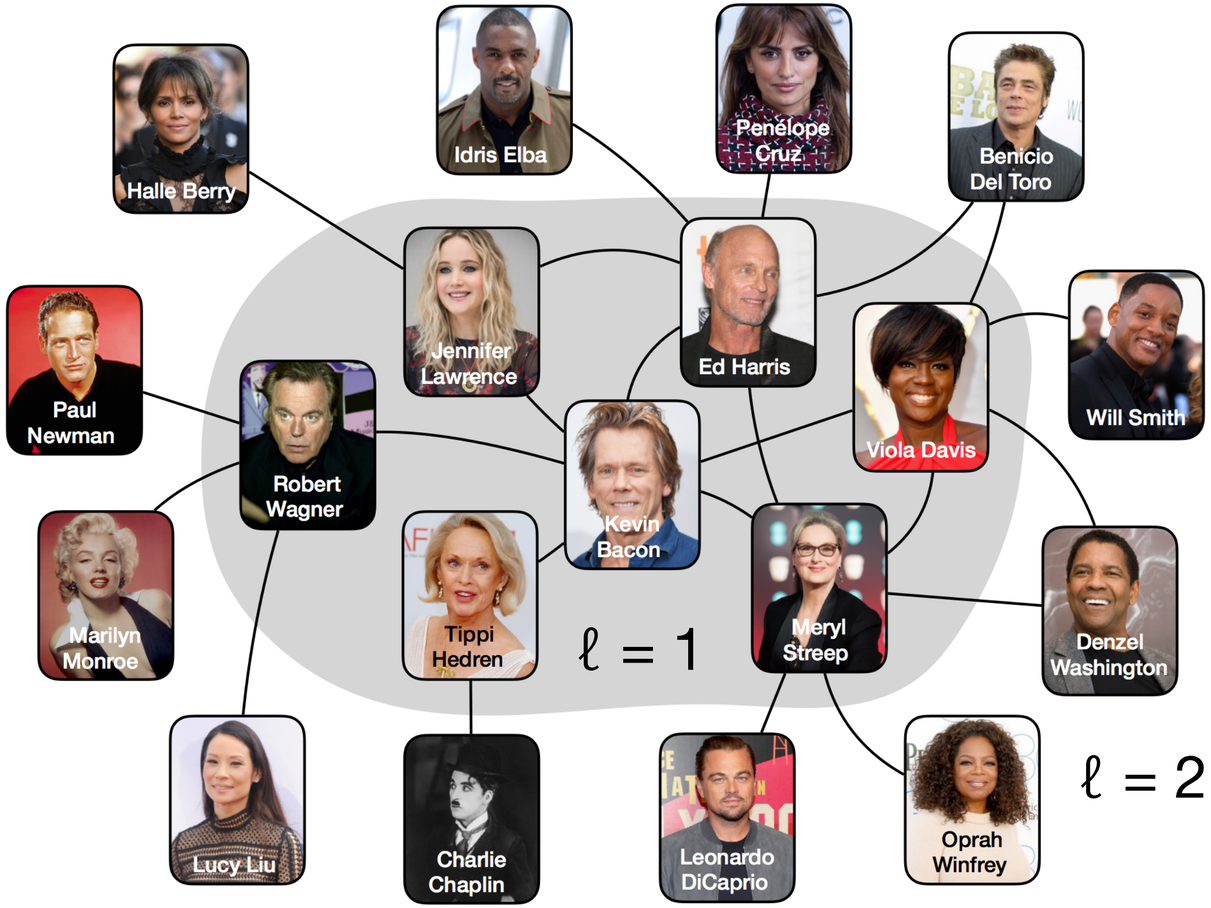
Davis

Menczer

Fortunato

Six Degrees of Kevin Bacon

- Short paths are found among all authors, not just Erdős…
- …And in all social networks, not just coauthorship
- Consider the movie co-star network as a second example
- Let's play the Oracle of Bacon game: oracleofbacon.org
  - Not just Kevin Bacon…
  - Can you find two stars separated by more that four links? Play the game and try!

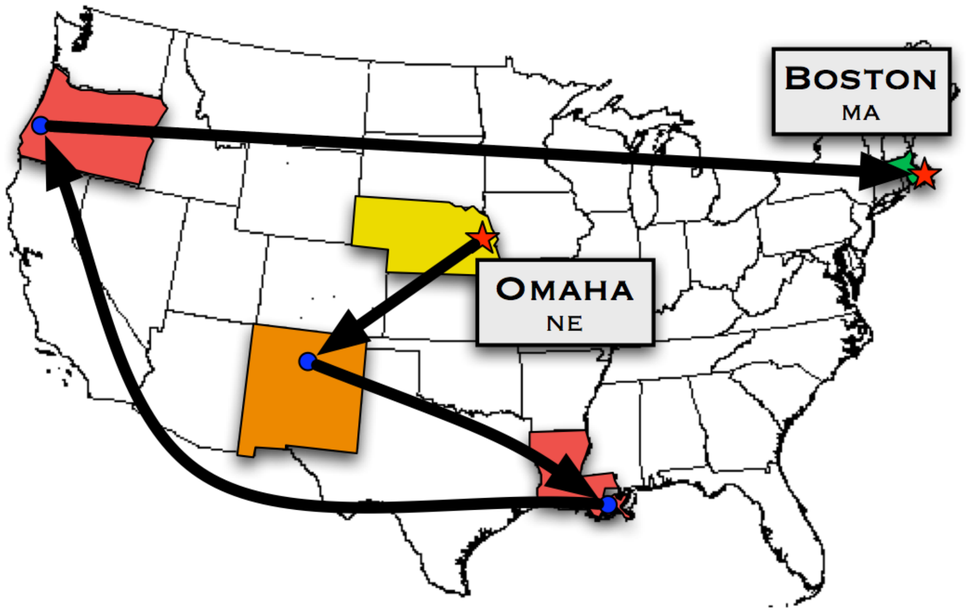


Small worlds

- What have we learned? Social networks tend to have very **short paths**
- **Six degrees of separation**: the idea that any two people are at most six steps away from each other in the social network
- First idea was in the short story “Chains” by Hungarian writer Frigyes Karinthy in 1929
- Psychologist **Stanley Milgram** provided first evidence in 1967 through a famous experiment to measure the **social distance** between any two people in the US
- John Guare coined the term “six degrees of separation” in a 1991 play (movie, too)

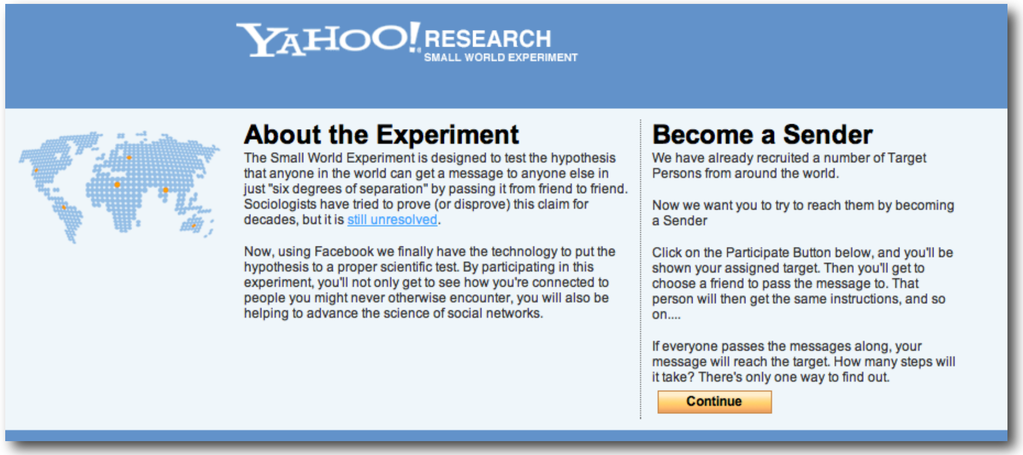
Milgram's experiment

- Instructions: send to personal acquaintance who is more likely to know target
- 160 letters to people in Omaha, NE and Wichita, KS
- 2 targets in Mass: the wife of a student in Sharon and a stockbroker in Boston
- 42 letters made it back (only 26%)
- Average: 6.5 steps (range: 3-12 steps)
- Much lower than most people expected!
- “Small world” effect is still surprising



More small world experiments

- Milgram's experiment was replicated in 2003 by Yahoo Research using email
  - 18 targets in 13 countries
  - 384 completed chains out of more than 24 thousand started
  - APL = 4 but when accounting for broken chains, estimated median PL of 5–7 steps
- Replicated by researchers at Facebook and University of Milan in 2011
  - 721 million active Facebook users
  - 69 billion friendships
  - APL = 4.74 steps: even shorter!



Short paths

- What do we mean by "short paths"? When can we call a path "short"?
- It depends on the size of the network!
- Observe the relationship between APL and network size when considering networks (or subnetworks) of different sizes
- We say that the average path length is **short** when it **grows very slowly** with the size of the network, say, logarithmically:

$$\langle \ell \rangle \sim \log N$$

Small worlds

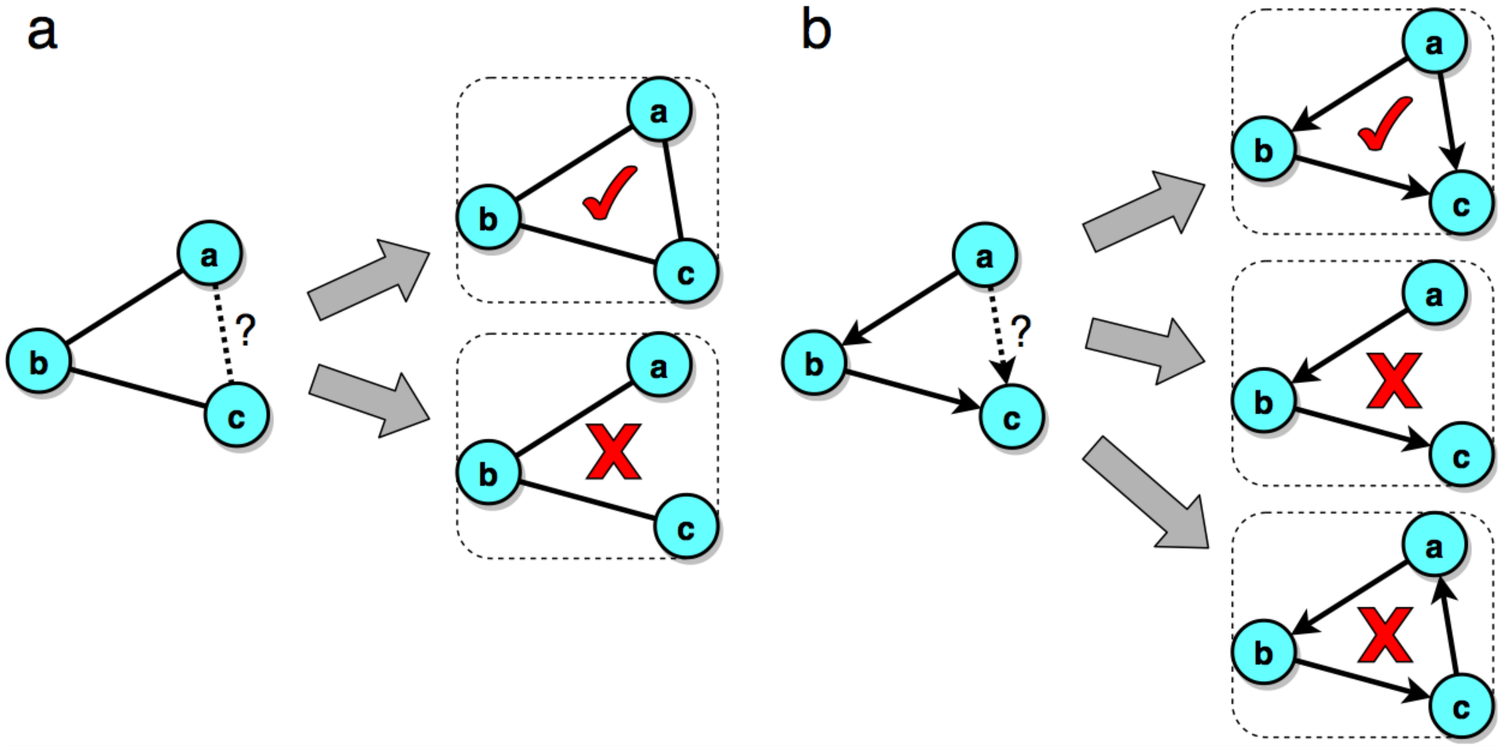
- Many other types of networks are small worlds, too
- Air transportation networks, the Internet, the Web, and Wikipedia, all have short paths
- Most real-world networks are small worlds
- Exceptions: grid-like networks

Table 2.1 Average path length and clustering coefficient of various network examples. The networks are the same as in Table 1.1, their numbers of nodes and links are listed as well. Link weights are ignored. The average path length is measured only on the giant component; for directed networks we consider directed paths in the giant strongly connected component. To measure the clustering coefficient in directed networks, we ignore link directions.

Network	Nodes (N)	Links (L)	Average path length ($\langle \ell \rangle$)	Clustering coefficient (C)
Facebook Northwestern Univ.	10,567	488,337	2.7	0.24
IMDB movies and stars	563,443	921,160	12.1	0
IMDB co-stars	252,999	1,015,187	6.8	0.67
Twitter US politics	18,470	48,365	5.6	0.03
Enron Email	87,273	321,918	3.6	0.12
Wikipedia math	15,220	194,103	3.9	0.31
Internet routers	190,914	607,610	7.0	0.16
US air transportation	546	2,781	3.2	0.49
World air transportation	3,179	18,617	4.0	0.49
Yeast protein interactions	1,870	2,277	6.8	0.07
C. elegans brain	297	2,345	4.0	0.29
Everglades ecological food web	69	916	2.2	0.55

Friend of a friend

- Another feature of social (and some other) networks is the presence of **triangles**: if Alice and Bob are both friends with Charlie, they are also likely friends of each other
- In other words, many friends of my friends are also my friends
- In directed networks, we can consider only certain types of directed triangles, like shortcuts in Twitter



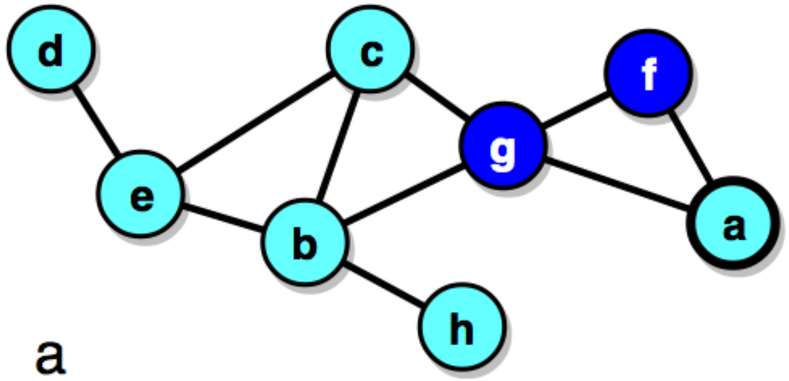
Clustering coefficient

- We can measure the number of triangles that a node actually has relative to how many it could have
- The **clustering coefficient** of a node is the **fraction of pairs of the node's neighbors that are connected to each other:**

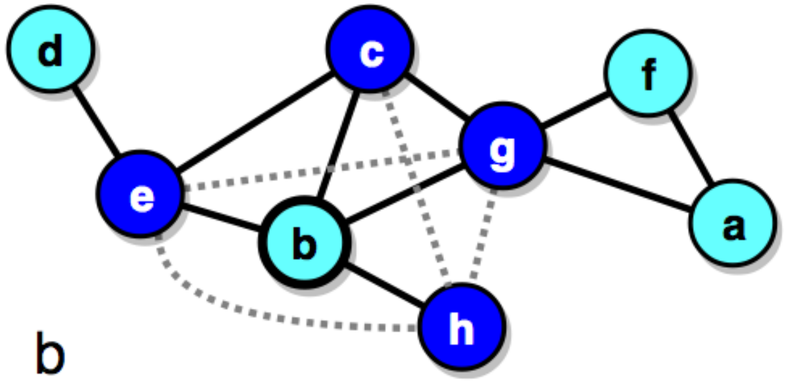
$$C(i) = \frac{\tau(i)}{\tau_{max}(i)} = \frac{\tau(i)}{\binom{k_i}{2}} = \frac{2\tau(i)}{k_i(k_i - 1)}$$

where $\tau(i)$ tau is the number of triangles involving *i.* Note that in this definition, the clustering coefficient is undefined if $k_i < 2$: a node must have at least degree 2 to have any triangles. However NetworkX assumes $C=0$ if $k=0$ or $k=1$.

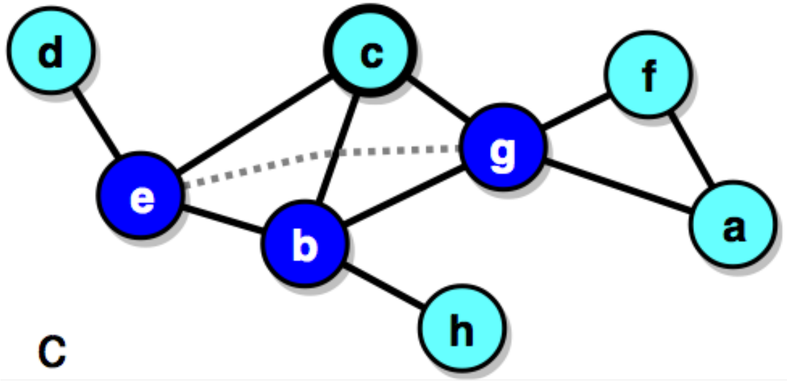
Clustering coefficient exercises



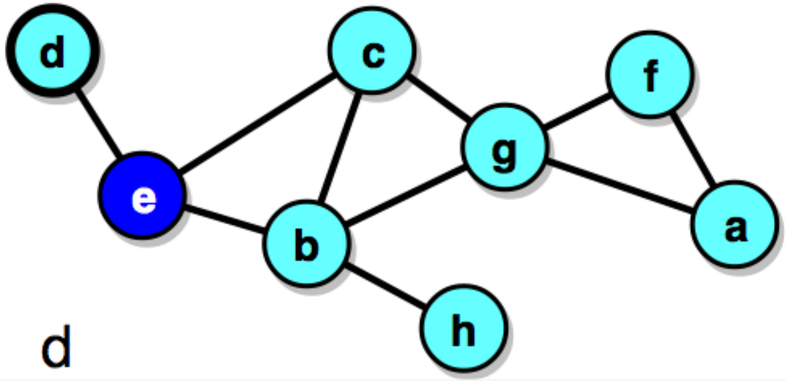
a



b



c



d

Network clustering coefficient

- The clustering coefficient of the network is the average of the clustering coefficients of the nodes:

$$C = \frac{\sum_{i:k_i>1} C(i)}{N_{k>1}}$$

  - Again, we should exclude singletons and nodes with k=1, but NetworkX assumes those have C=0

```
nx.triangles(G)            # dict node -> no. triangles
nx.clustering(G, node)     # clustering coefficient of node
nx.clustering(G)           # dict node -> clustering coefficient
nx.average_clustering(G) # network's clustering coefficient
```

Network clustering coefficient

- Some networks, e.g., social networks, tend to have high clustering coefficients because of **triadic closure**: we meet through common friends

- Other networks, e.g., bipartite and tree-like networks, have low clustering coefficient

Table 2.1 Average path length and clustering coefficient of various network examples. The networks are the same as in Table 1.1, their numbers of nodes and links are listed as well. Link weights are ignored. The average path length is measured only on the giant component; for directed networks we consider directed paths in the giant strongly connected component. To measure the clustering coefficient in directed networks, we ignore link directions.

Network	Nodes (N)	Links (L)	Average path length ($\langle \ell \rangle$)	Clustering coefficient (C)
Facebook Northwestern Univ.	10,567	488,337	2.7	0.24
IMDB movies and stars	563,443	921,160	12.1	0
IMDB co-stars	252,999	1,015,187	6.8	0.67
Twitter US politics	18,470	48,365	5.6	0.03
Enron Email	87,273	321,918	3.6	0.12
Wikipedia math	15,220	194,103	3.9	0.31
Internet routers	190,914	607,610	7.0	0.16
US air transportation	546	2,781	3.2	0.49
World air transportation	3,179	18,617	4.0	0.49
Yeast protein interactions	1,870	2,277	6.8	0.07
C. elegans brain	297	2,345	4.0	0.29
Everglades ecological food web	69	916	2.2	0.55