ACDS Lecture Series

Lecture - 3

CSIR

Network Science

G. N. Sastry and Team

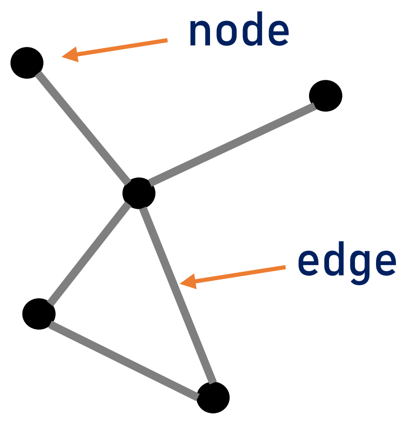
ADVANCED COMPUTATION AND DATA SCIENCES (ACDS) DIVISION

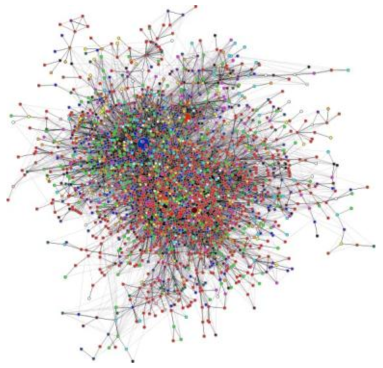
CSIR-North East Institute of Science and Technology, Jorhat, Assam, India

3. Network Science

3.1 Introduction to Network Science

- Network Science is an academic field which studies complex networks such as telecommunication networks, computer networks, biological networks, cognitive and semantic networks, and social networks considering distinct elements or actors represented by nodes and the connections between the elements or actors as links.
- The United States National Research Council defines network science as "the study of network representations of physical, biological, and social phenomena leading to predictive models of these phenomena."



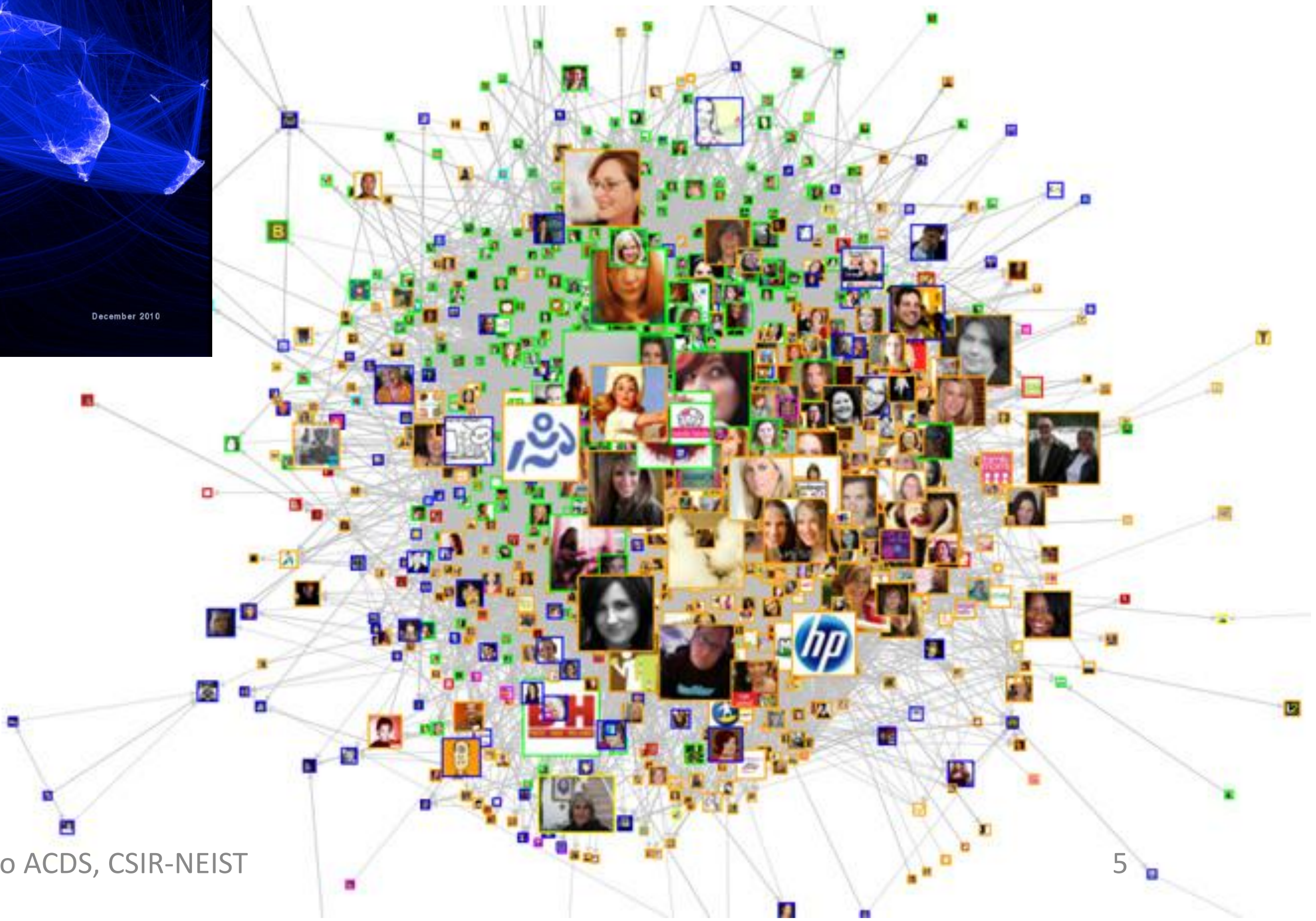


3.2 Networks at the heart of complex systems

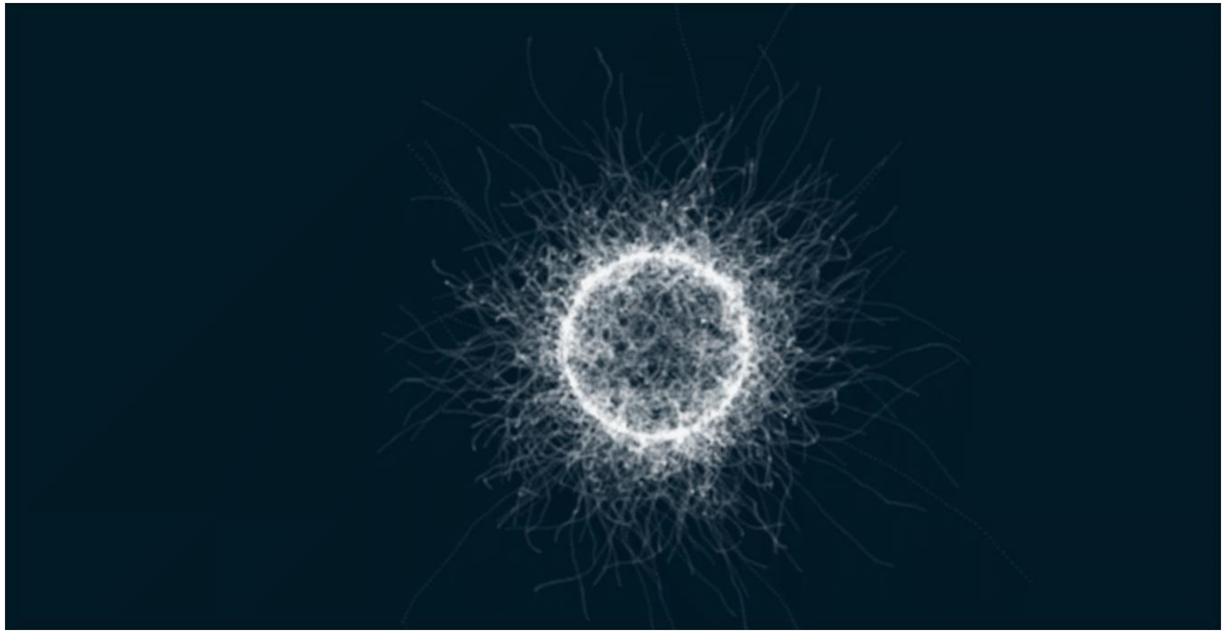
Behind each complex system there is a **network**, that defines the interactions between the component.

3.2.1 Society Facebook: The Social graph

Connectivity of Network

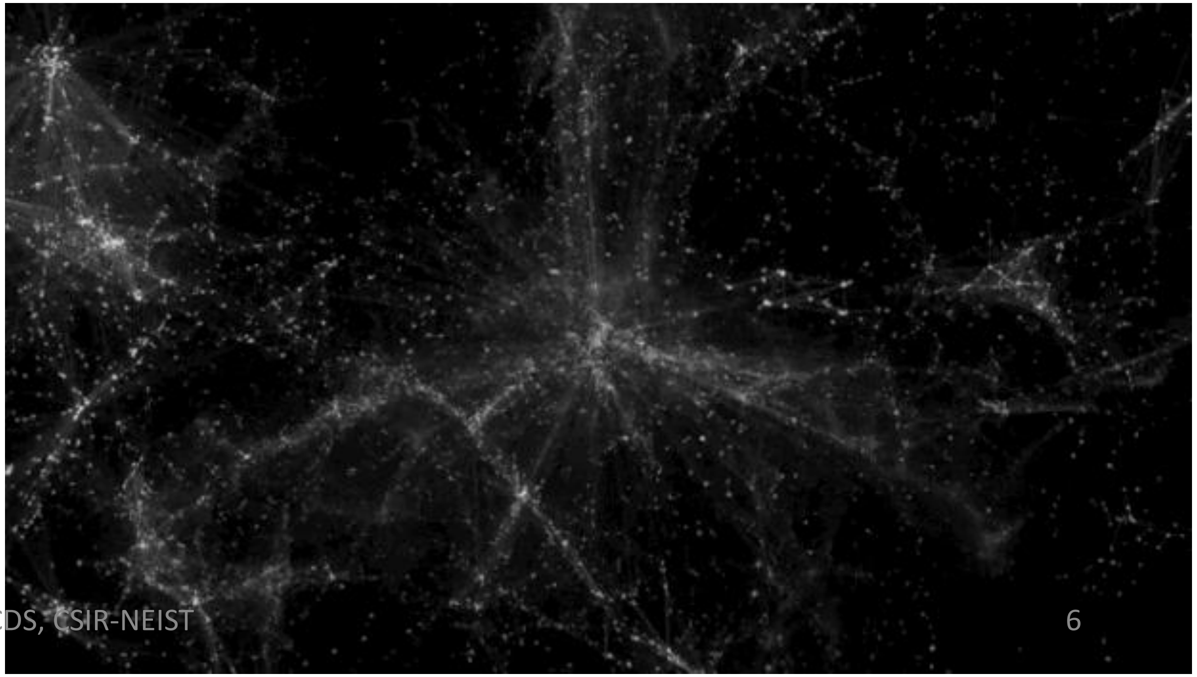


3.2.2 Biological Network



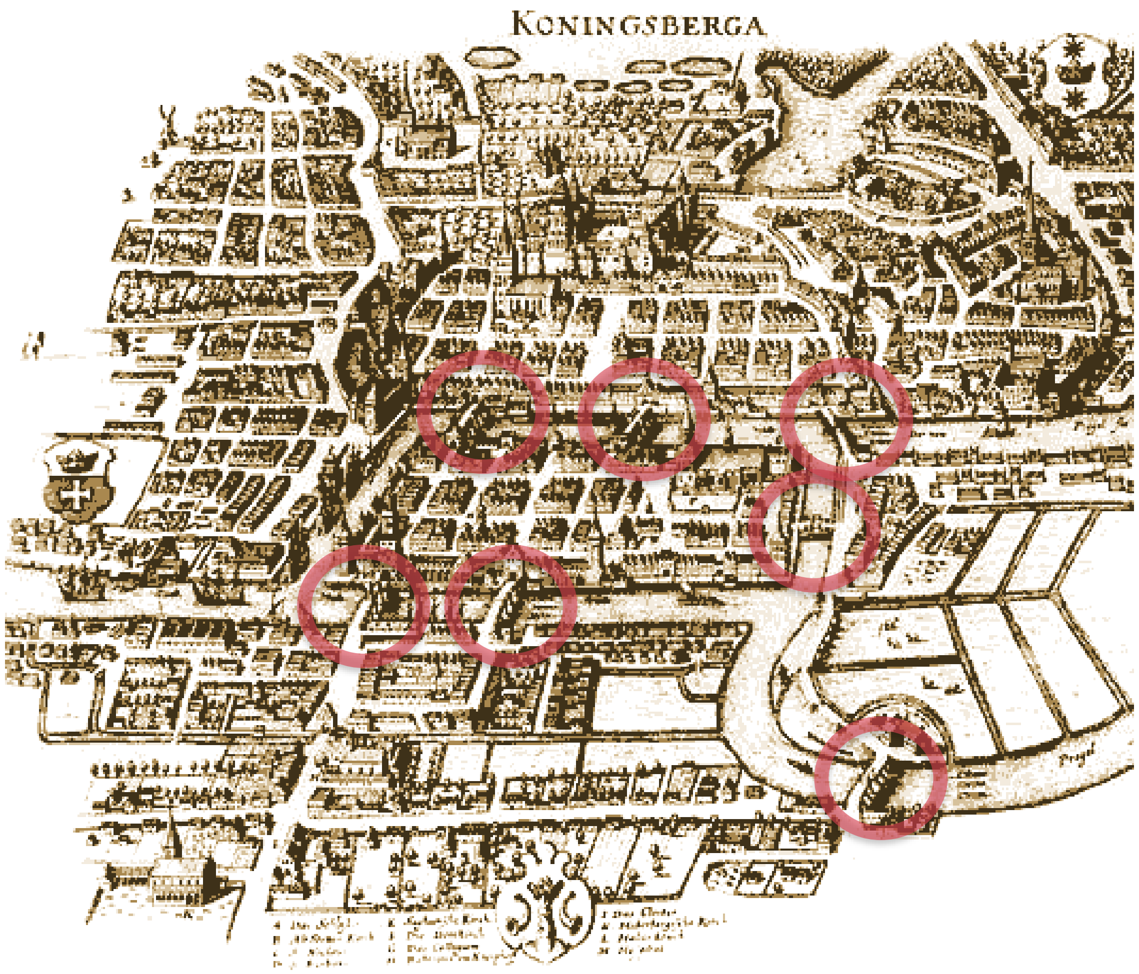
Nodes- Proteins
Links- The interaction between proteins

The Universe as a Network..

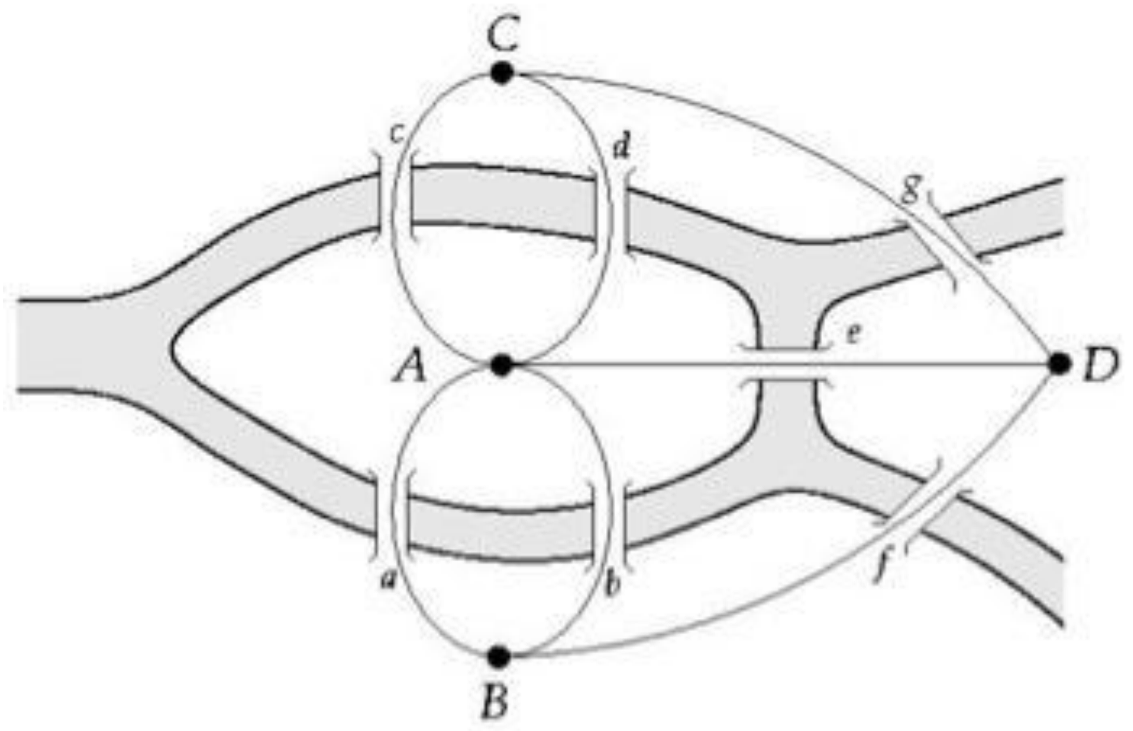


3.3.1 Origin of the Theory

- Graph theory is the mathematical scaffold behind network science.
- The roots go back to 1736 to **Konigsberg**, a thriving merchant city in Eastern Prussia. Its busy fleet of ships and the trade they brought allowed city officials to build seven bridges across the river **Pregel** that surrounded the town.
- Five of these connected the elegant island Kneiphof, caught between the two branches of the Pregel, to the mainland; two crossed the two branches of the river.
- This peculiar arrangement gave birth to a contemporary puzzle: Can one walk across all seven bridges and never cross the same one twice?
- The problem remained unsolved until 1736, when **Leonard Euler**, a Swiss born mathematician, offered a rigorous mathematical proof that such path does not exist.



3.3.2 The bridges of Konigsberg



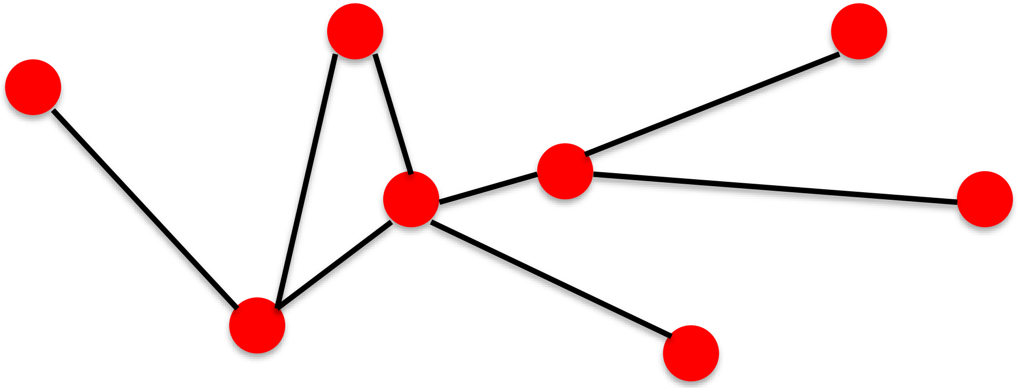
Can one walk across the seven bridges and never cross the same bridge twice?

1735: Euler's theorem:

(a) If a graph has more than two nodes of odd degree, there is no path.

(b) If a graph is connected and has no odd degree nodes, it has at least one path.

3.3.3 Components of a complex system



- **components**: nodes, vertices
  N
- **interactions**: links, edges
  L
- **system:** network, graph
  (N,L)

3.3.4 Networks or graphs?

Network often refer to real systems

- WWW
- Social Network
- Metabolic Network
- Biological Network

Language: (Network, node, link)

Graph: mathematical representation of a network

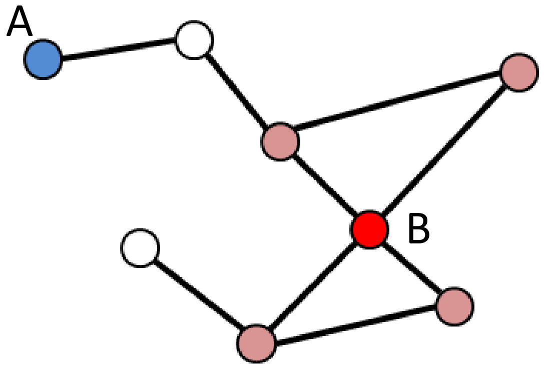
- Web graph
- Social graph(a Facebook term)

Language: (Graph, Vertex, edge)

3.3.5 Degree and Degree distribution

Node Degrees

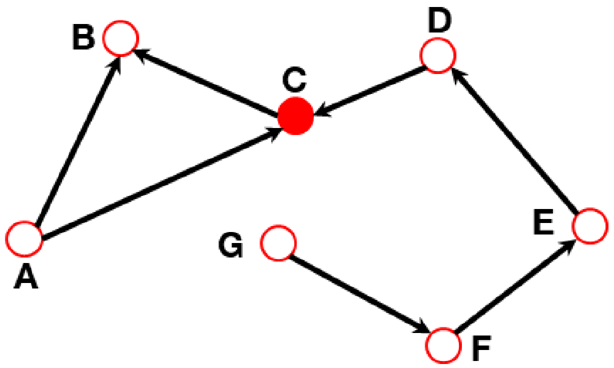
Undirected



Node degree: the number of links connected to the node.

$$k_A = 1 \qquad k_B = 4$$

Directed



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

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

Source: a node with $k^{in} = 0$; Sink: a node with $k^{out} = 0$.

3.3.6 A bit of statistics

BRIEF STATISTICS REVIEW

Four key quantities characterize a sample of N values $x_1, \ldots, x_N$:

Average (mean):

$$\langle x \rangle = \frac{x_1 + x_2 + \ldots + x_N}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

The n^{th} moment:

$$\langle x^n \rangle = \frac{x_1^n + x_2^n + \ldots + x_N^n}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i^n$$

Standard deviation:

$$\sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \langle x \rangle\right)^2}$$

Distribution of x:

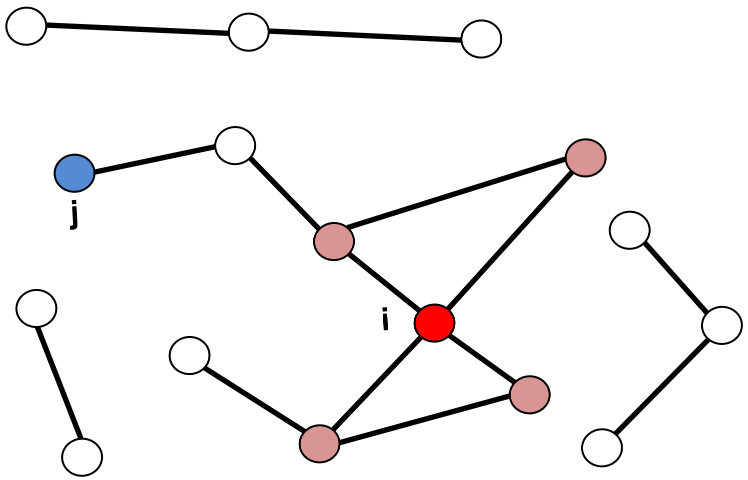
$$p_x = \frac{1}{N}\sum_i \delta_{x,x_i}$$

where p_x follows

$$\sum_i p_x = 1 \left(\int p_x \, dx = 1 \right)$$

3.3.7 Average Degree

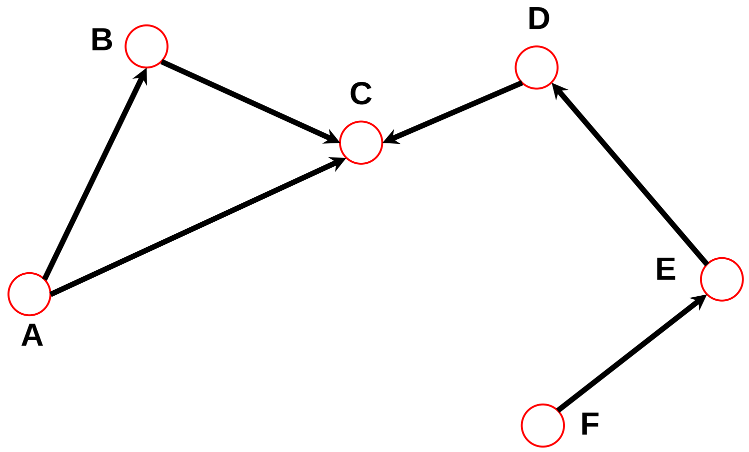
Undirected



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

N – the number of nodes in the graph

Directed



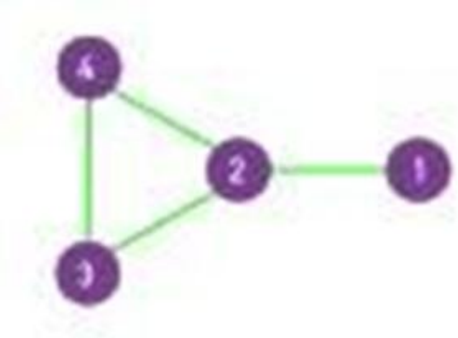
$$\langle k^{in} \rangle \equiv \frac{1}{N}\sum_{i=1}^{N} k_i^{in}, \quad \langle k^{out} \rangle \equiv \frac{1}{N}\sum_{i=1}^{N} k_i^{out}, \quad \langle k^{in} \rangle = \langle k^{out} \rangle$$

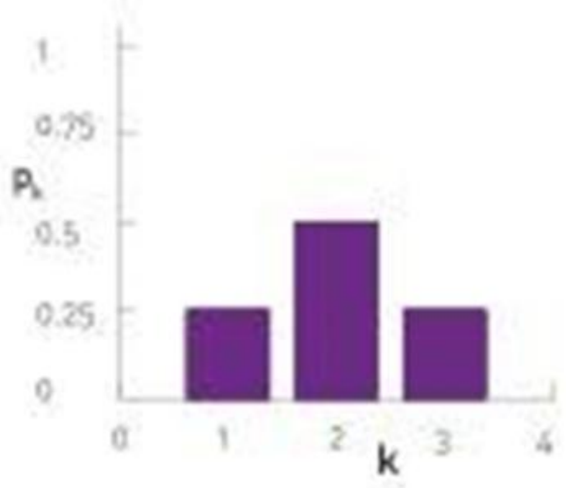
$$\langle k \rangle \equiv \frac{L}{N}$$

3.3.7 Degree Distribution

Degree Distribution

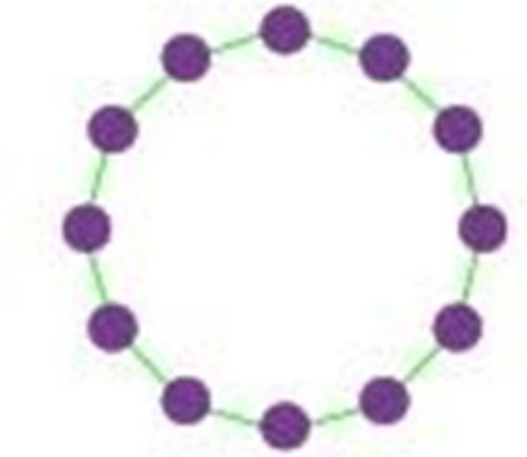
P(k): probability that a randomly chosen node has degree k

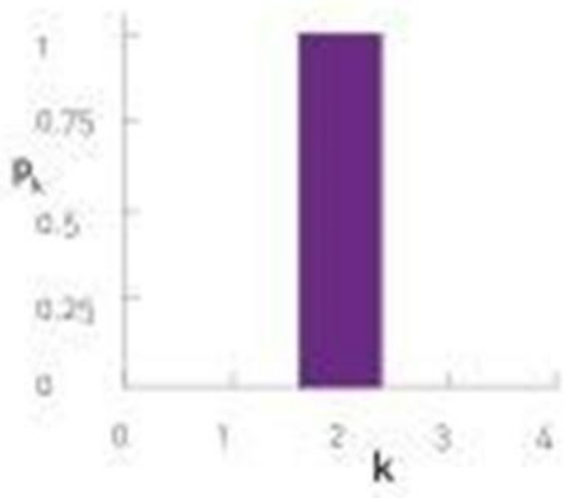




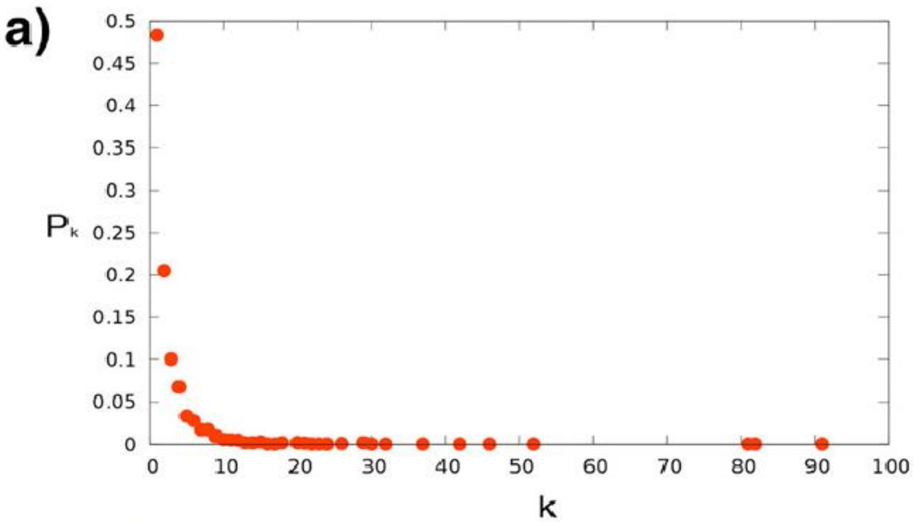
N_k= # nodes with degree k

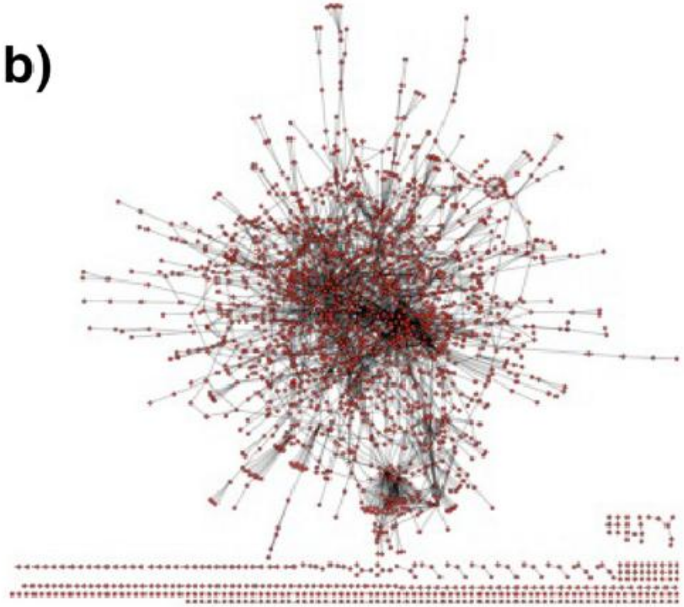
$P(k) = N_k / N$

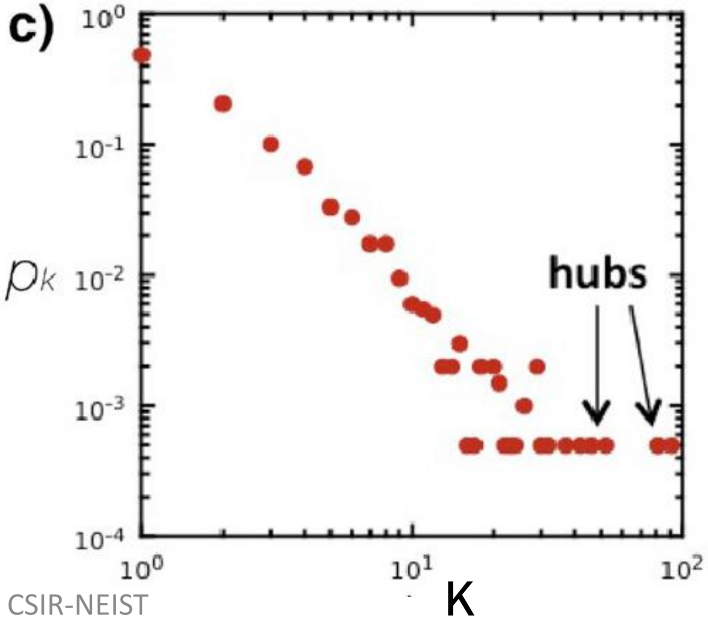




3.3.7 Degree Distribution







3.3.7 Degree Distribution

Discrete Representation: $\mathbf{p_k}$ is the probability that a node has degree **k**.

Continuum Description: **p(k)** is the pdf of the degrees, where

$$\int_{k_1}^{k_2} p(k)\,dk$$

represents the probability that a node's degree is between $\mathbf{k_1}$ and $\mathbf{k_2}$.

Normalization condition:

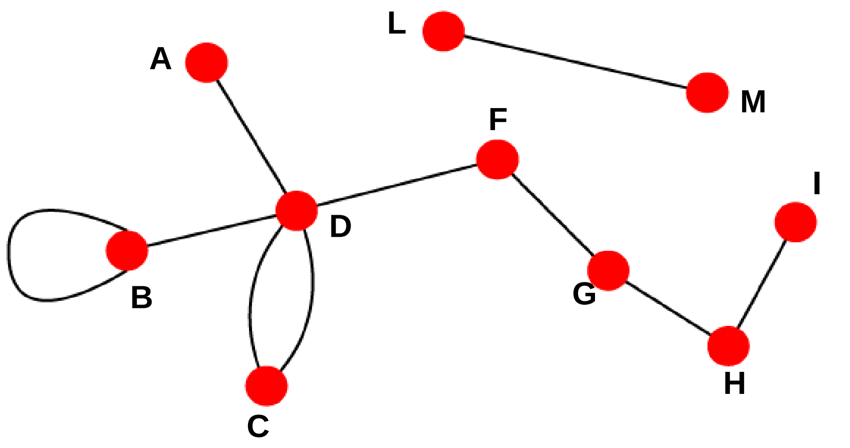
$$\sum_{0}^{\infty} p_k = 1 \qquad\qquad \int_{K_{\min}}^{\infty} p(k)\,dk = 1$$

where K_{min} is the minimal degree in the network.

Undirected VS. Directed Networks

Undirected

Links: undirected (*symmetrical*)
Graph:

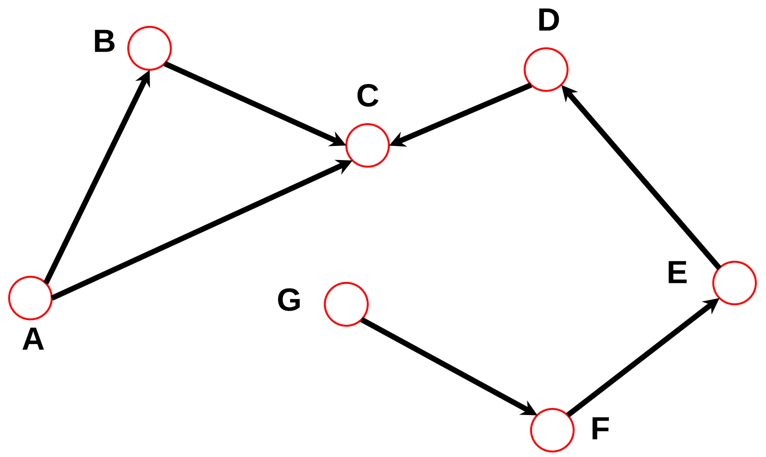


Undirected links :
coauthorship links
Actor network
protein interactions

Directed

Links: directed (*arcs*).

Digraph = directed graph:

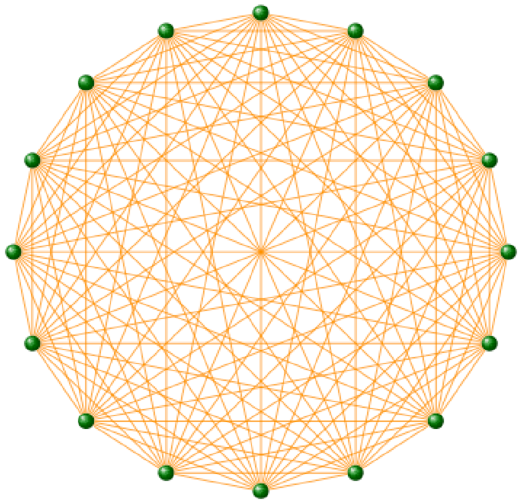


An undirected link is the superposition of two opposite directed links.

Directed links :
URLs on the www
phone calls
metabolic reactions

Real networks are sparse

Complete Graph



The maximum number of links a network of N nodes can have is:

A graph with degree $L=L_{max}$ is called a complete graph, and its average degree is **$\langle k \rangle = N-1$**

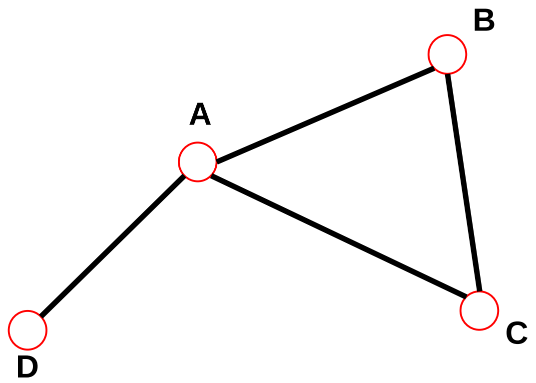
Most networks observed in real systems are sparse:

$$L \ll L_{max}$$

or

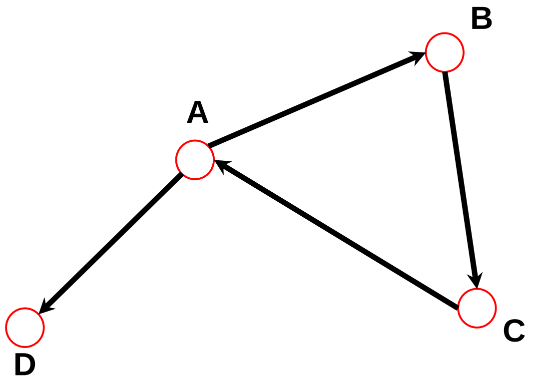
$$\langle k \rangle \ll N-1.$$

Distance In a Graph



The *distance (shortest path, geodesic path)* 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.



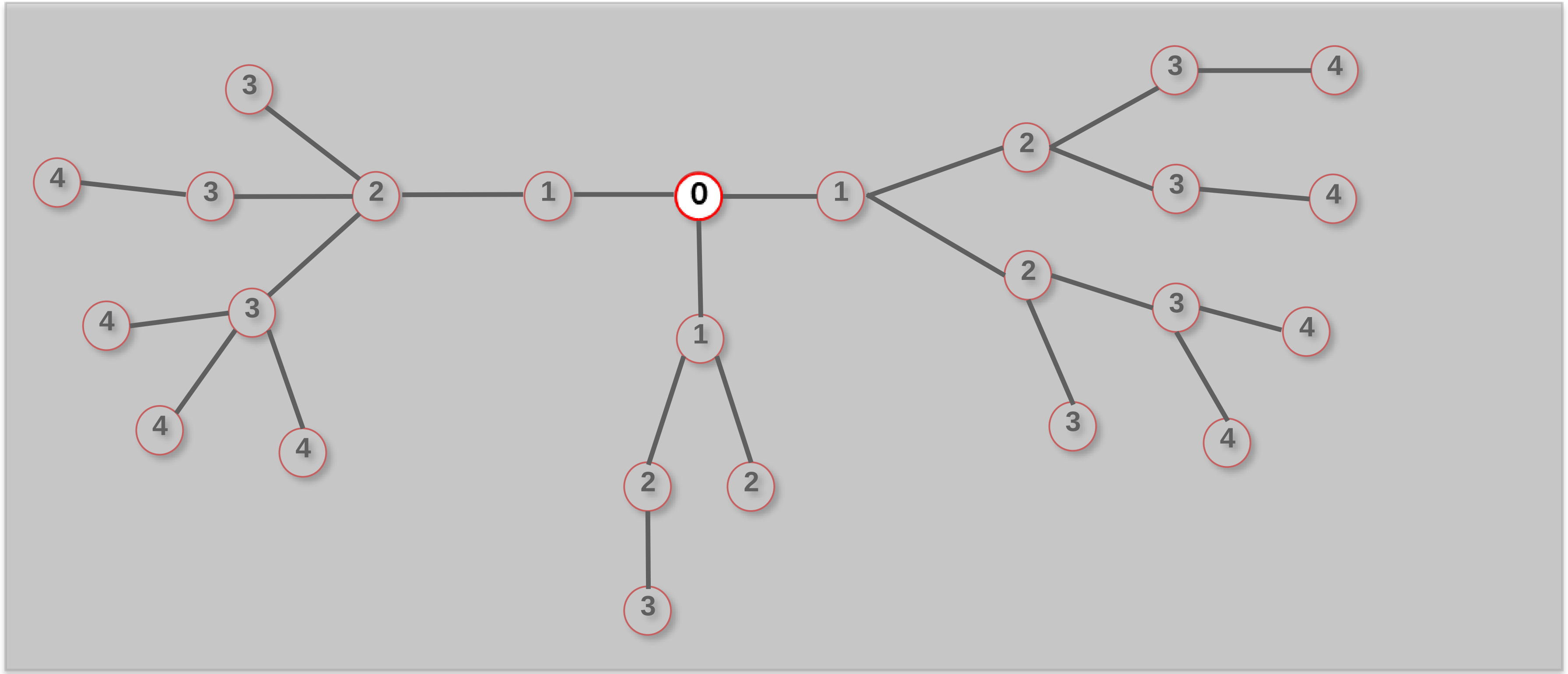
In directed graphs each path needs to follow the direction of the arrows.

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 BCA path).

Finding Distances: Breadth first search

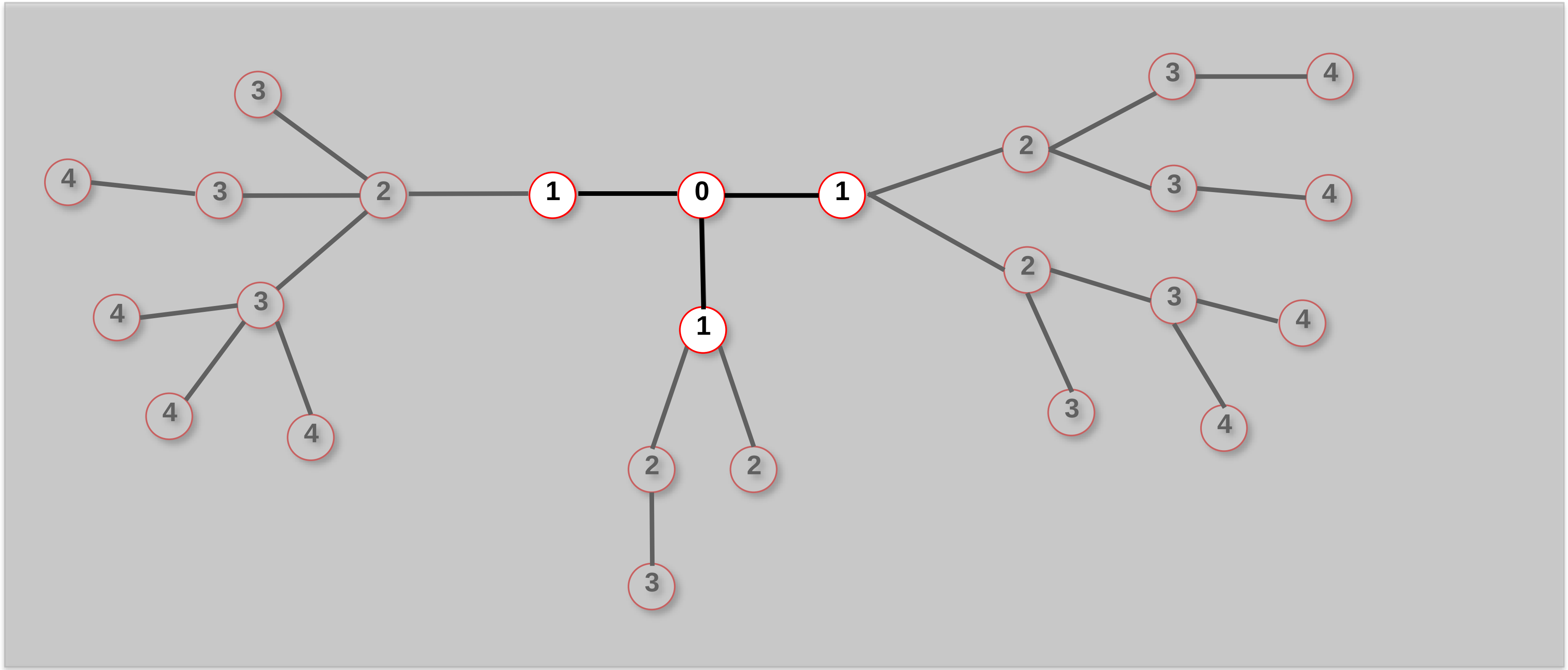
Distance between node 0 and node 4:

1. Start at 0.



Distance between node 0 and node 4:

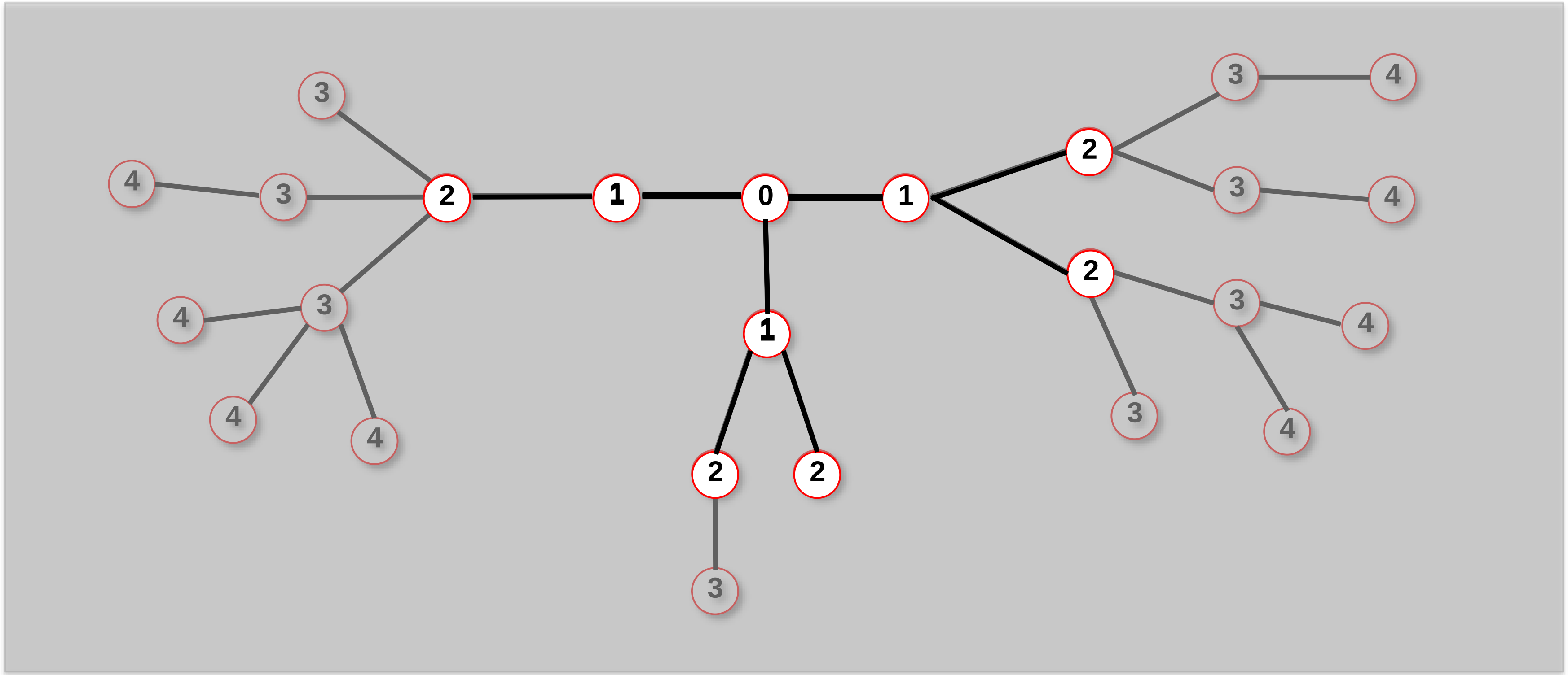
1. Start at 0.
2. Find the nodes adjacent to 1. Mark them as at distance 1. Put them in a queue.



Finding Distances: Breadth first search

Distance between node 0 and node 4:

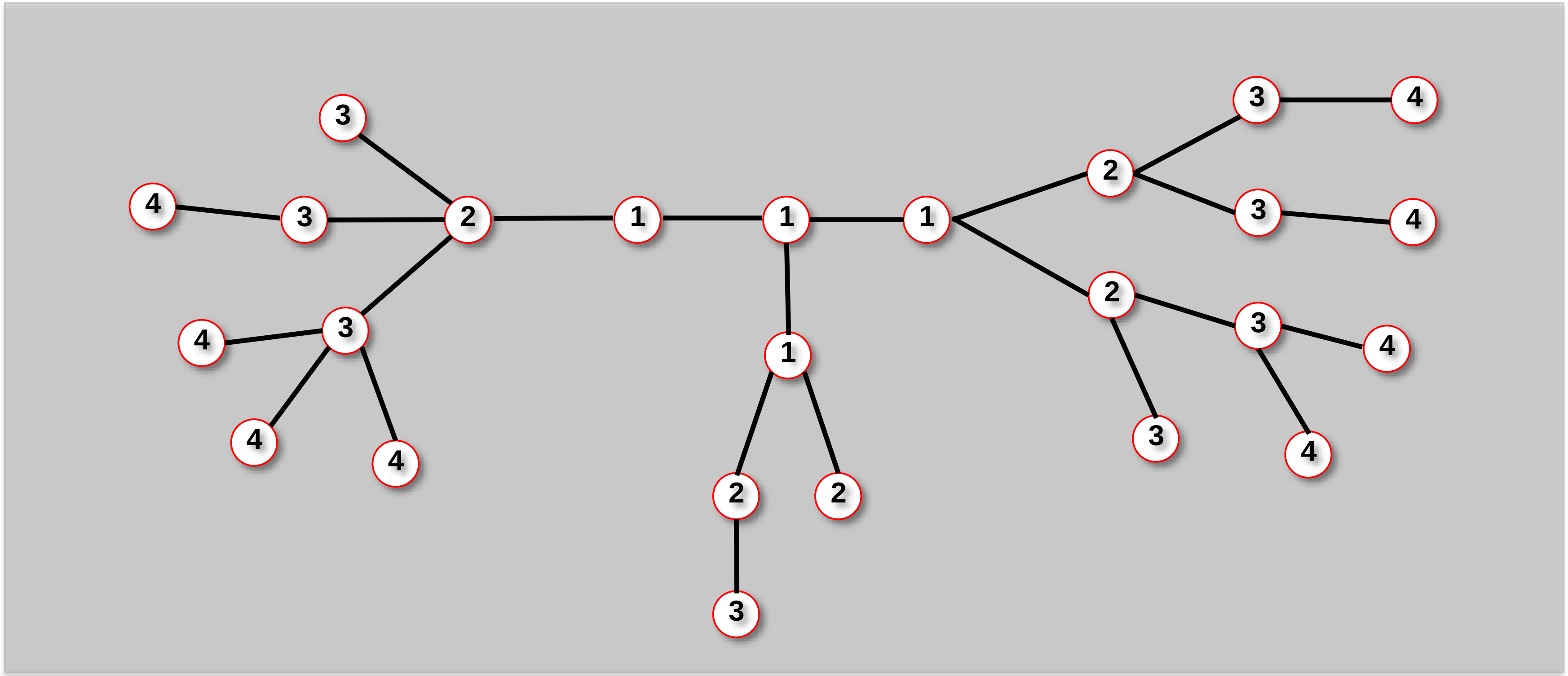
1. Start at 0.
2. Find the nodes adjacent to 0. Mark them as at distance 1. Put them in a queue.
3. Take the first node out of the queue. Find the unmarked nodes adjacent to it in the graph. Mark them with the label of 2. Put them in the queue.



Finding Distances: Breadth first search

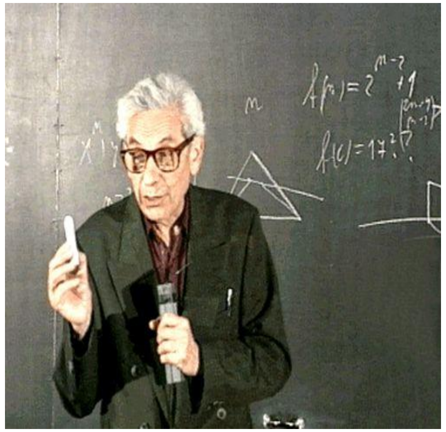
Distance between node 0 and node 4:

1. Repeat until you find node 4 or there are no more nodes in the queue.
2. The distance between 0 and 4 is the label of 4 or, if 4 does not have a label, infinity.



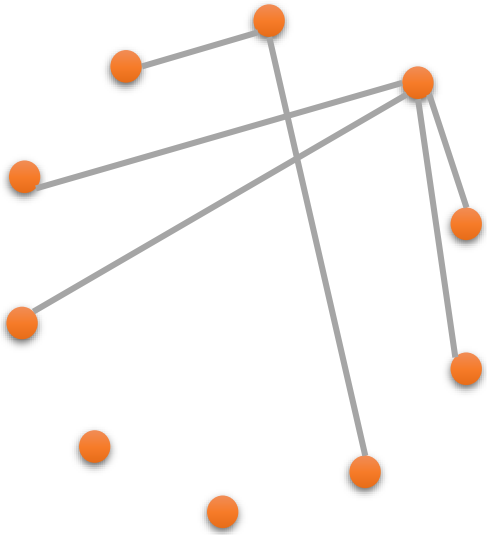
3.4 Random Network Model

Erdős Paul
(1913-1996)



Rényi Alfréd
(1921-1970)

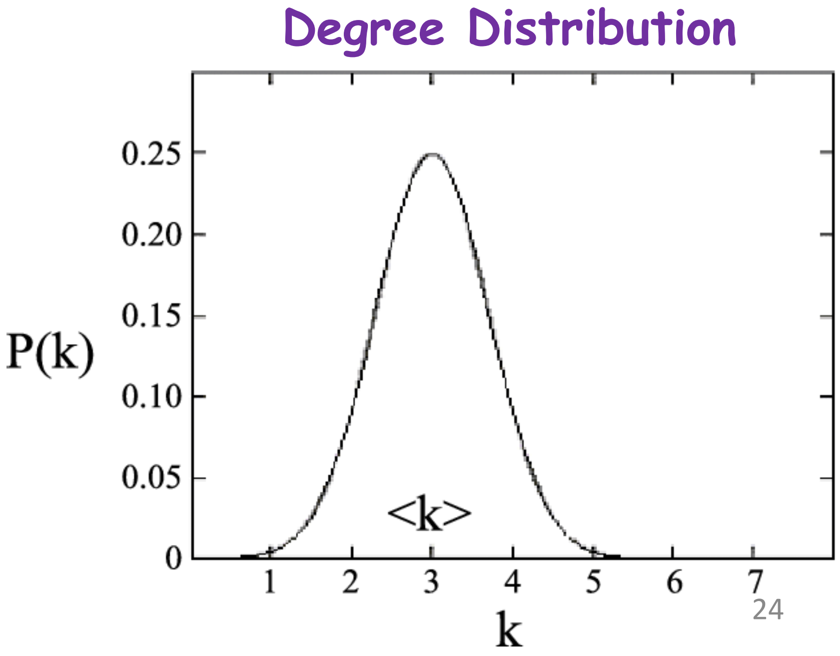
Erdős- Rényi Model(1960)



Connect with probability p

p= 1/6 N=10

$\langle k \rangle \approx 1.5$



3.4.1 Random Network Model

Definition:

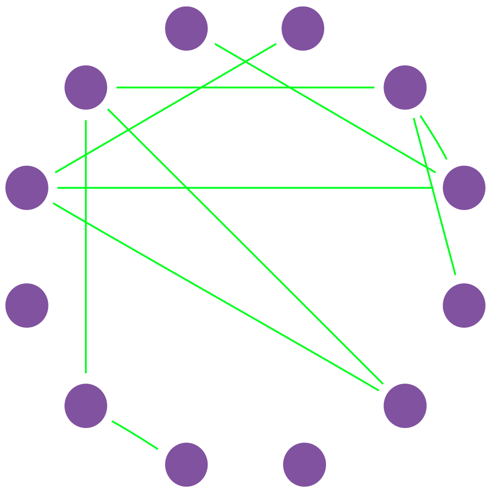
A **random graph** is a graph of N nodes where each pair of nodes is connected by probability **p**.

***G(N,L)* Model**
N labeled nodes are connected with L randomly placed links. Erdos and Renyi used this definition in their string of papers on random networks.

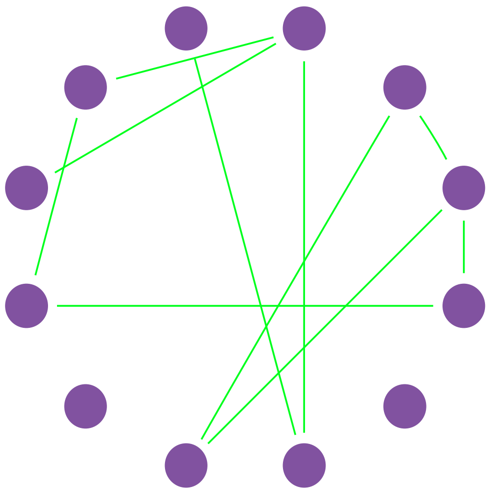
G(N,p)Model
Each pair of n labeled nodes is connected with probability p, a model introduced by Gilbert.

3.4.2 Random Network Model

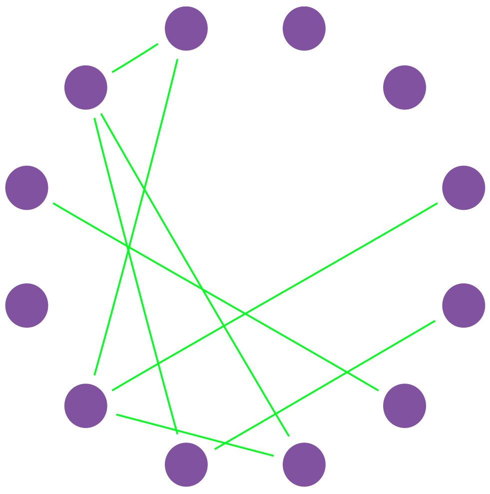
p=1/6
N=12



L=8



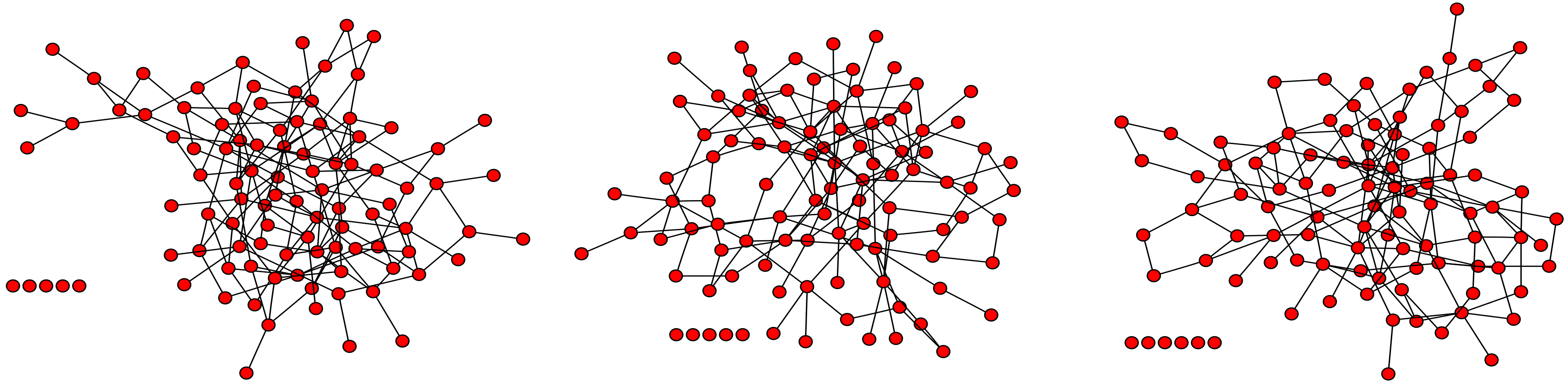
L=10



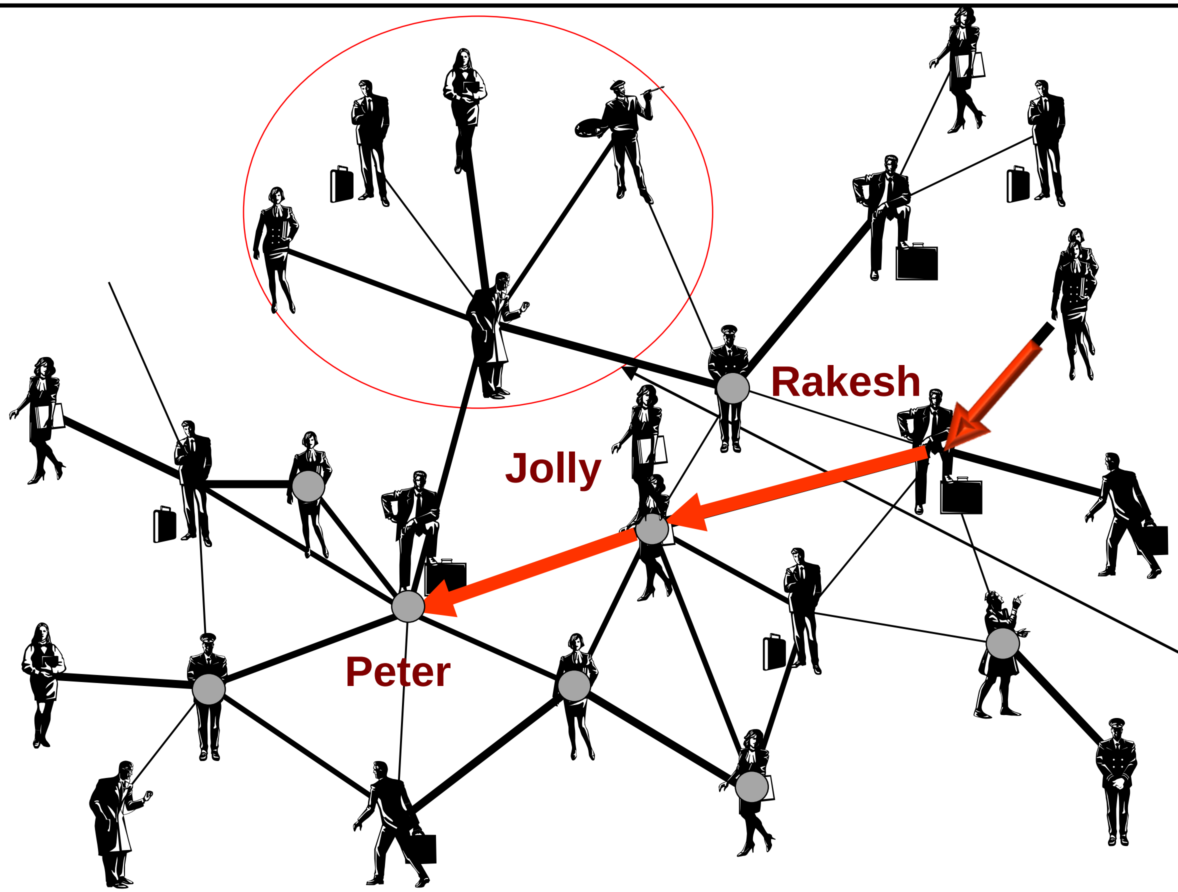
L=7

3.4.3 Random Network Model

p=0.03
N=100



3.4.4 Small worlds



A small-world is a social network in which most nodes are not neighbors of one another, but most nodes can be reached from every other by small number of steps.

When a group of nodes is more interconnected with each other than they are individually with the rest of the network, they form a network cluster.

The small world phenomenon- human society is a small world type network characterized by short path lengths.

Frigyes Karinthy, 1929
Stanley Milgram, 1967

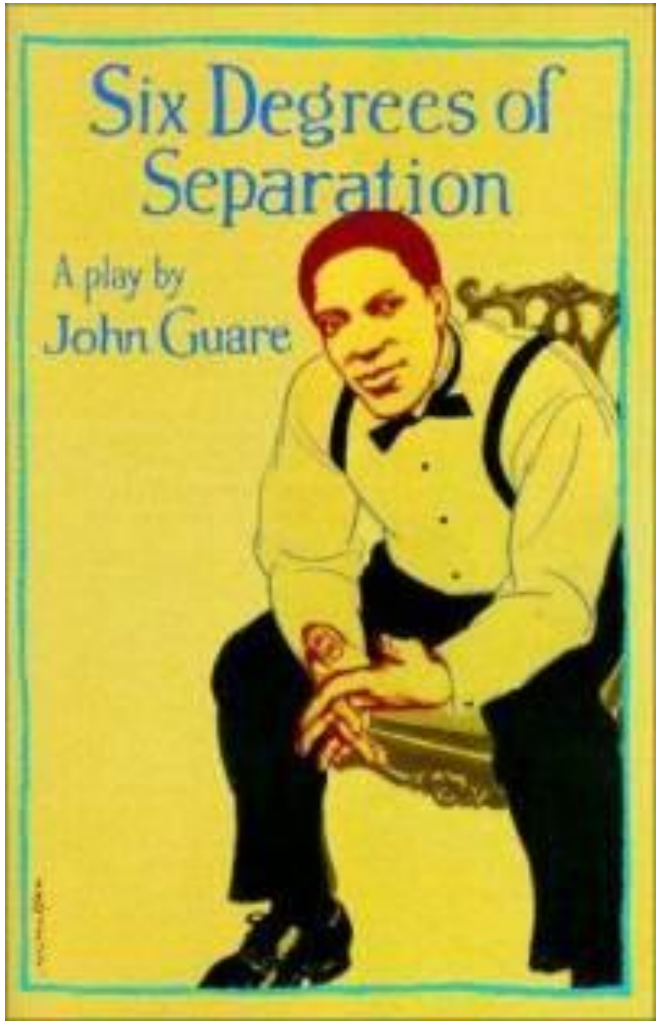
Six degrees 1929: Frigyes Kartinthy

Frigyes Karinthy (1887-1938)
Hungarian Writer

1929: *Minden másképpen van* (Everything is Different)
Láncszemek (Chains)

"The worker knows the manager in the shop, who knows Ford; Ford is on friendly terms with the general director of Hearst Publications, who last year became good friends with Arpad Pasztor, someone I not only know, but to the best of my knowledge a good friend of mine. So I could easily ask him to send a telegram via the general director telling Ford that he should talk to the manager and have the worker in the shop quickly hammer together a car for me, as I happen to need one."

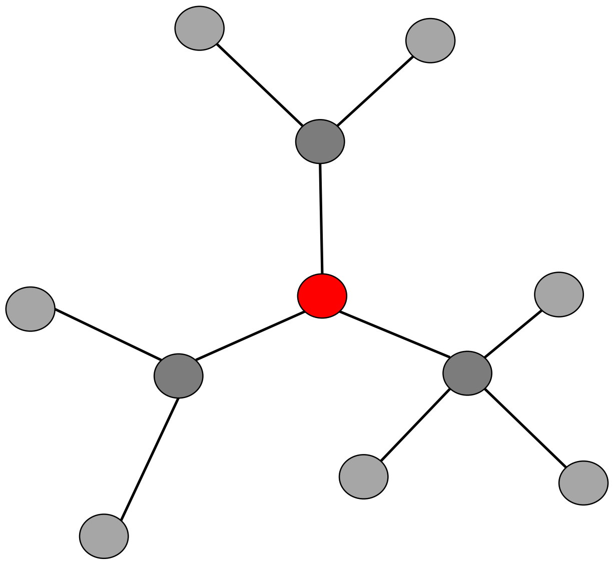
Six degrees 1991: John Gaure



"Everybody on this planet is separated by only six other people. Six degrees of separation. Between us and everybody else on this planet. The president of the United States. A gondolier in Venice…. It's not just the big names. It's anyone. A native in a rain forest. A Tierra del Fuegan. An Eskimo. I am bound to everyone on this planet by a trail of six people. It's a profound thought. How every person is a new door, opening up into other worlds."

Distances in Random Graphs

Random graphs tend to have a tree-like topology with almost constant node degrees.



$\langle k \rangle$ nodes at distance one (d=1).

$\langle k \rangle^2$ nodes at distance two (d=2).

$\langle k \rangle^3$ nodes at distance three (d =3).

...

$\langle k \rangle^d$ nodes at distance d.

$$N = 1 + \langle k \rangle + \langle k \rangle^2 + \ldots + \langle k \rangle^{d_{\max}} = \frac{\langle k \rangle^{d_{\max}+1} - 1}{\langle k \rangle - 1} \approx \langle k \rangle^{d_{\max}} \Rightarrow d_{\max} = \frac{\log N}{\log \langle k \rangle}$$

Distances in Random Graphs

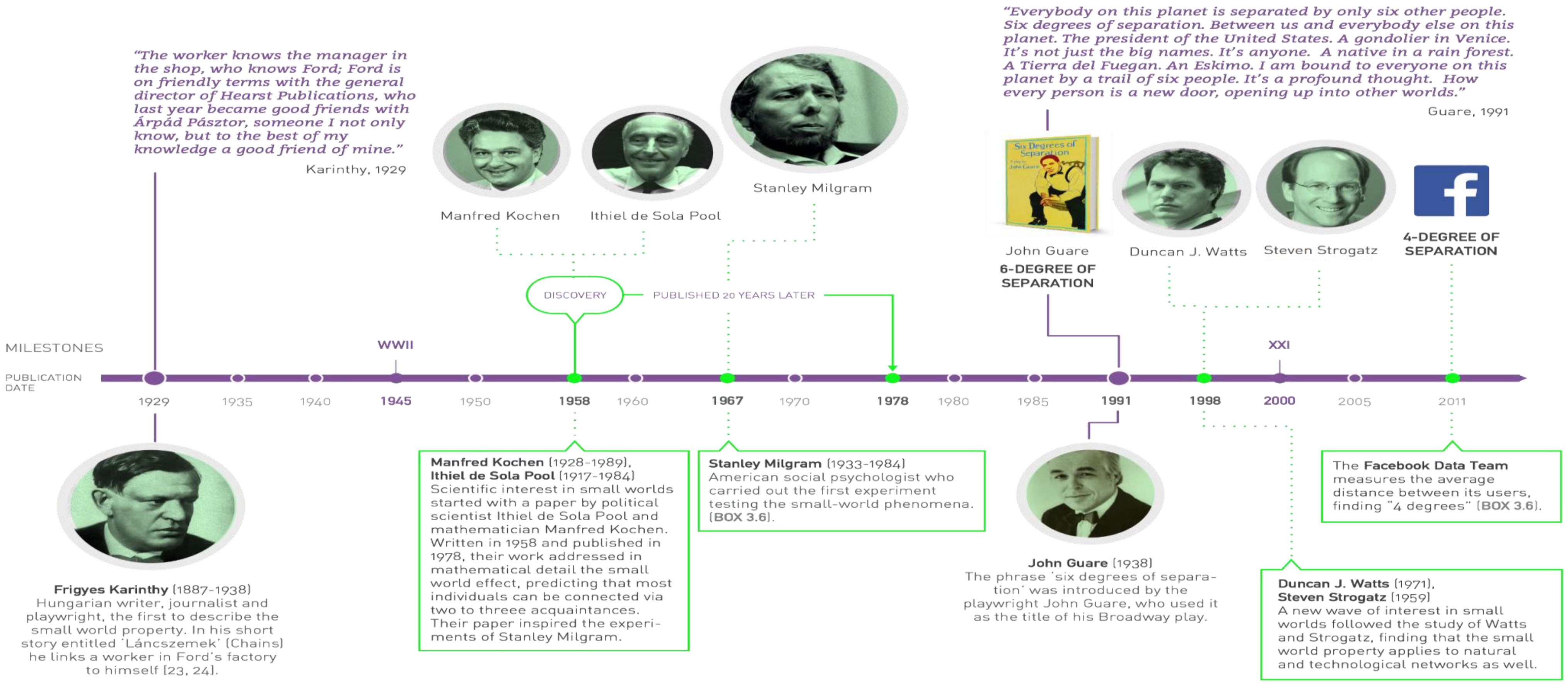
$$d_{max} = \frac{\log N}{\log \langle k \rangle}$$

In most networks this offers a better approximation to the average distance between two randomly chosen nodes, $\langle d \rangle$, than to d_{max} .

$$< d > = \frac{\log N}{\log \langle k \rangle}$$

We will call the *small world phenomena* the property that the average path length or the diameter depends logarithmically on the system size. Hence, ”small” means that $\langle d \rangle$ is proportional to log N, rather than N.

The 1/log$\langle k \rangle$ term implies that denser the network, the smaller will be the distance between the nodes.


"The worker knows the manager in the shop, who knows Ford; Ford is on friendly terms with the general director of Hearst Publications, who last year became good friends with Árpád Pásztor, someone I not only know, but to the best of my knowledge a good friend of mine."
Karinthy, 1929
Manfred Kochen
Ithiel de Sola Pool
Stanley Milgram
"Everybody on this planet is separated by only six other people. Six degrees of separation. Between us and everybody else on this planet. The president of the United States. A gondolier in Venice. It's not just the big names. It's anyone. A native in a rain forest. A Tierra del Fuegan. An Eskimo. I am bound to everyone on this planet by a trail of six people. It's a profound thought. How every person is a new door, opening up into other worlds."
Guare, 1991
John Guare
6-DEGREE OF SEPARATION
Duncan J. Watts
Steven Strogatz
4-DEGREE OF SEPARATION
DISCOVERY
PUBLISHED 20 YEARS LATER
MILESTONES
WWII
XXI
PUBLICATION DATE
1929
1935
1940
1945
1950
1958
1960
1967
1970
1978
1980
1985
1991
1998
2000
2005
2011
Frigyes Karinthy (1887-1938)
Hungarian writer, journalist and playwright, the first to describe the small world property. In his short story entitled 'Láncszemek' (Chains) he links a worker in Ford's factory to himself [23, 24].
Manfred Kochen (1928-1989),
Ithiel de Sola Pool (1917-1984)
Scientific interest in small worlds started with a paper by political scientist Ithiel de Sola Pool and mathematician Manfred Kochen. Written in 1958 and published in 1978, their work addressed in mathematical detail the small world effect, predicting that most individuals can be connected via two to threee acquaintances. Their paper inspired the experi-ments of Stanley Milgram.
Stanley Milgram (1933-1984)
American social psychologist who carried out the first experiment testing the small-world phenomena. (**BOX 3.6**).
John Guare (1938)
The phrase 'six degrees of separa-tion' was introduced by the playwright John Guare, who used it as the title of his Broadway play.
Duncan J. Watts (1971),
Steven Strogatz (1959)
A new wave of interest in small worlds followed the study of Watts and Strogatz, finding that the small world property applies to natural and technological networks as well.
The **Facebook Data Team** measures the average distance between its users, finding "4 degrees" (**BOX 3.6**).

Are real Networks like Random Graphs?

As quantitative data about real networks became available, we can compare their topology with the predictions of random graph theory.

Note that once we have N and <k> for a random network, from it we can derive every measurable property. Indeed, we have:

Average path length:

$$< l_{rand} > \approx \frac{\log N}{\log\langle k\rangle}$$

Clustering Coefficient:

$$C_i = \frac{2\langle L_i\rangle}{k_i(k_i - 1)} = p = \frac{\langle k\rangle}{N}.$$

Degree Distribution:

$$P(k) = e^{-<k>}\frac{<k>^k}{k!}$$

Network Data: Science collaboration networks

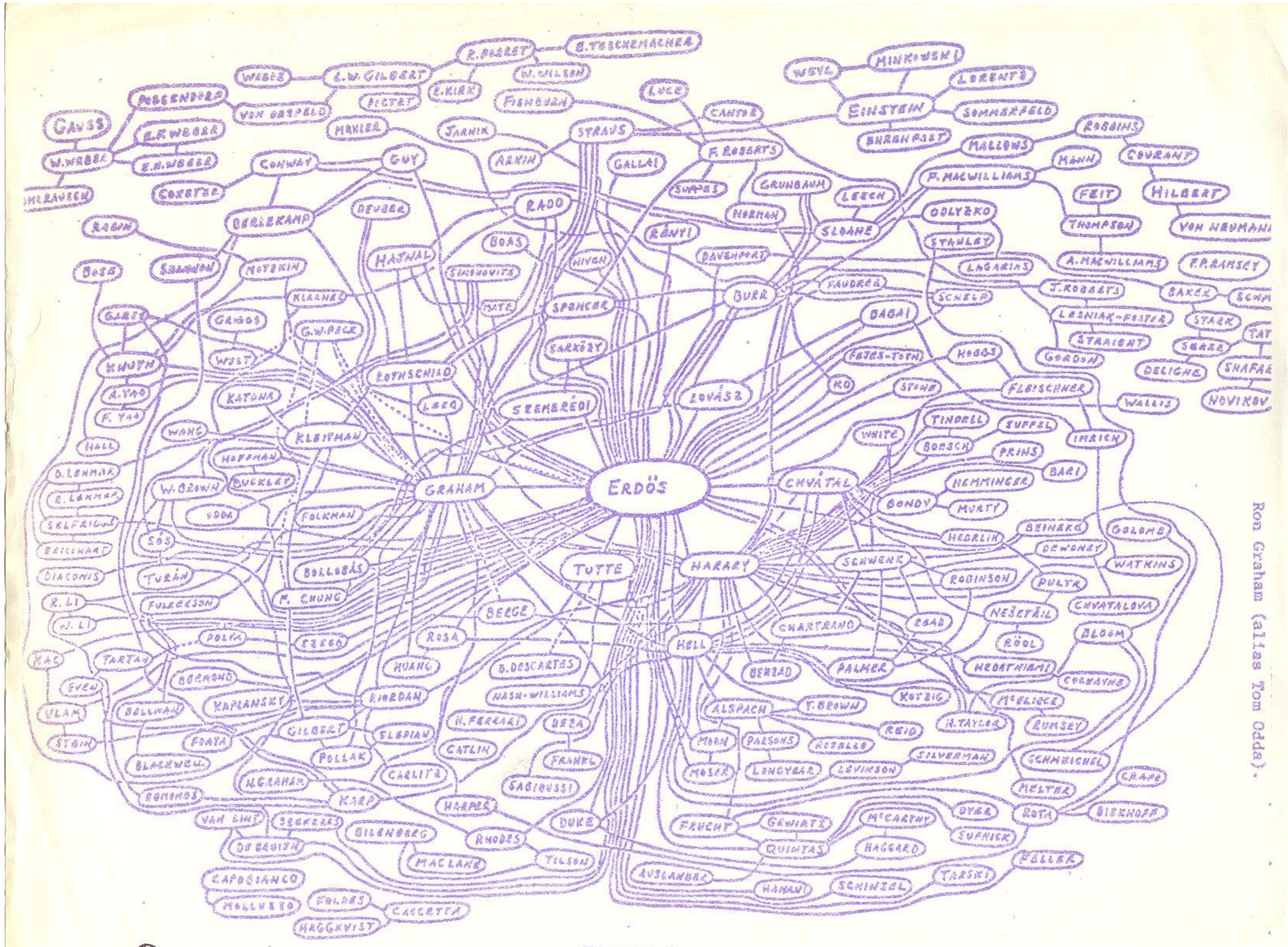


Figure 1
To appear in Topics in Graph Theory (F. Harary, ed.) New York Academy of Sciences (1979).

Erdos Number: distance from the mathemaHcian (most people are 4-5 hops away) based on collaboration.

1,400 papers

507 coauthors

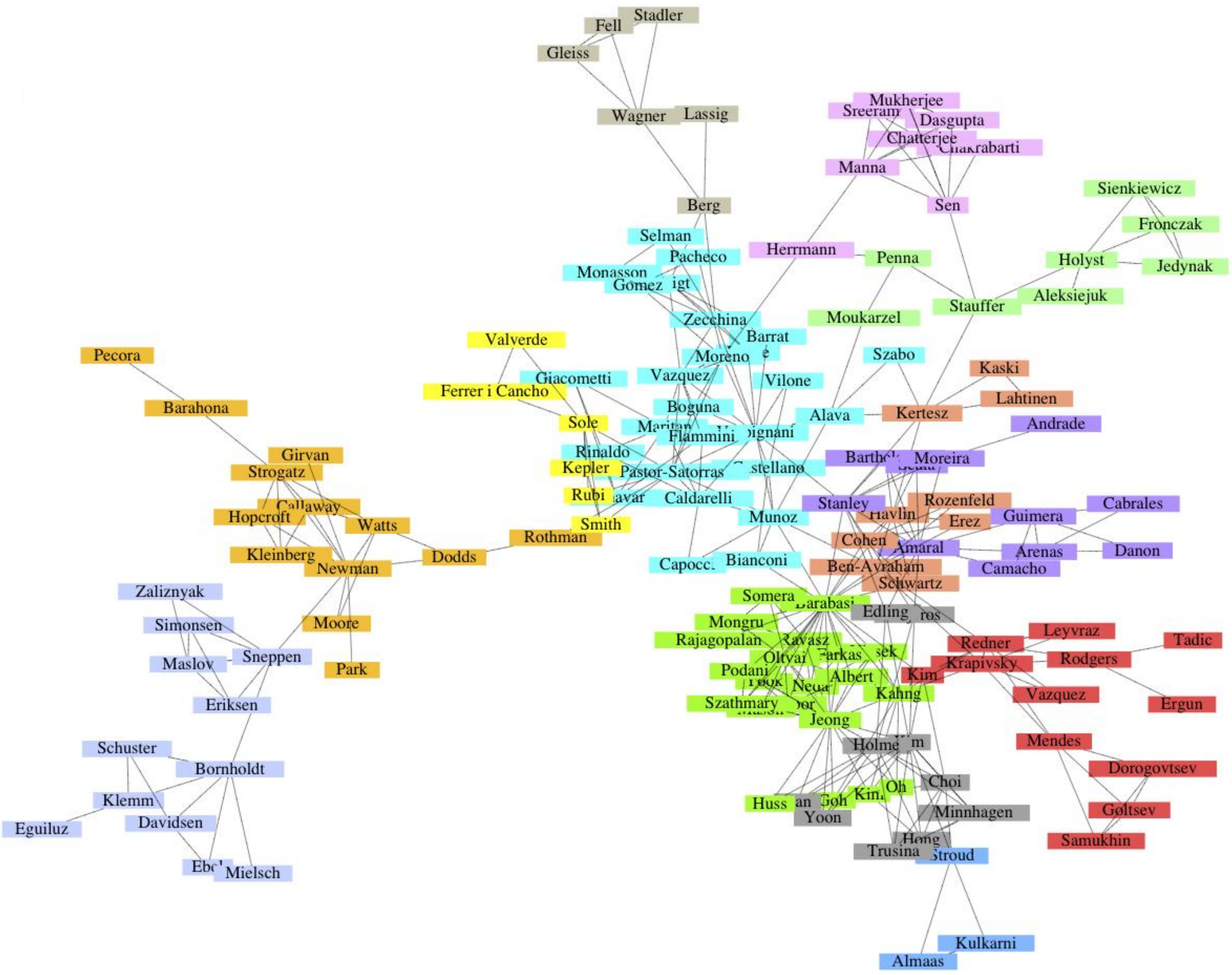
Einstein: EN=2

Paul Samuelson EN=5

....

ALB: EN: 3

Network Data: Science collaboration networks

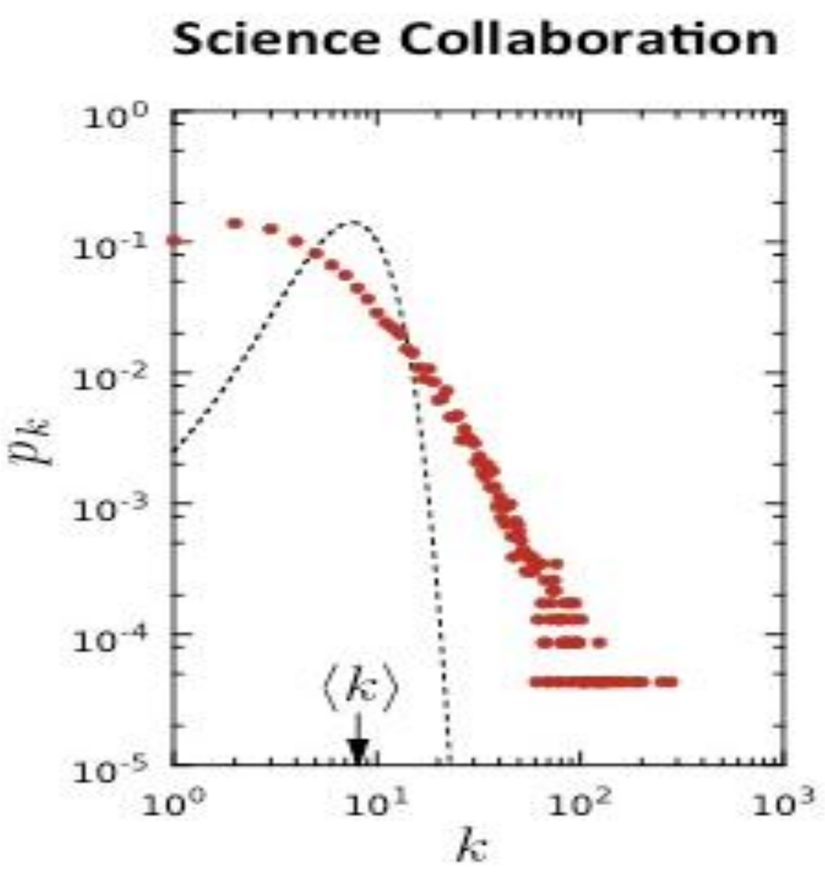


Collaboration Network:
Nodes: Scientists
Links: Joint publications

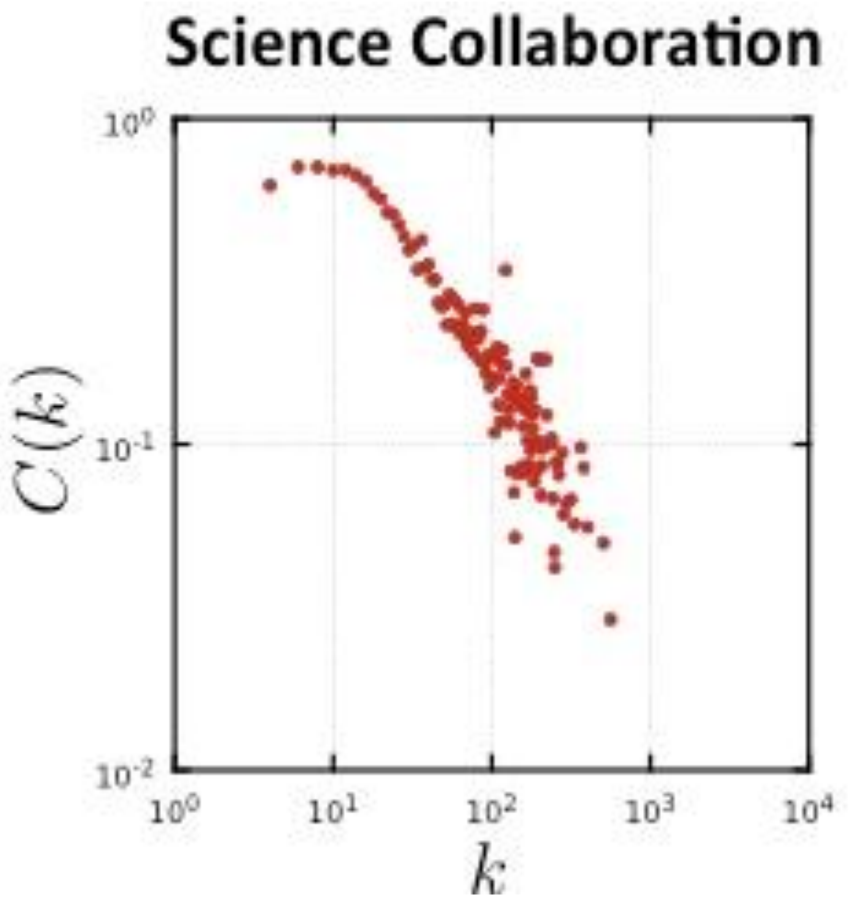
Physical Review:
1893 – 2009.

N=449,673
L=4,707,958

See also Stanford Large Network database http://snap.stanford.edu/data/#canets.



Scale-free



Hierarchical

Networks are:

- Large → Use computational resources for analysis
- Sparse → Efficient ways of storing the information
- Objects to analyze → Need representations amenable to efficient algorithms

Introduction: Why Python?

Python is an interpreted, general-purpose high-level programming language whose design philosophy emphasizes code readability.

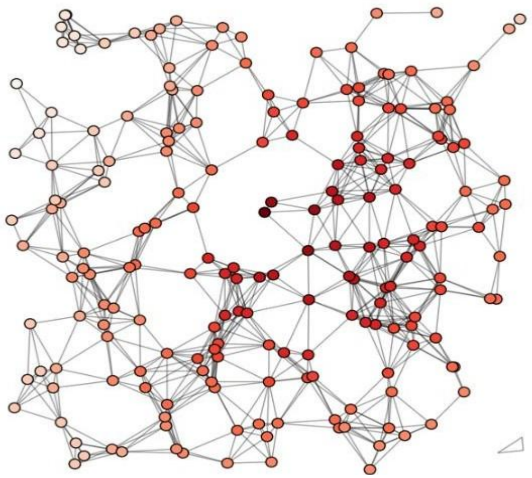
Clear syntax
Multiple programming paradigms
Dynamic typing
Strong on-line community
Rich documentation
Numerous libraries
Expressive features
Fast prototyping

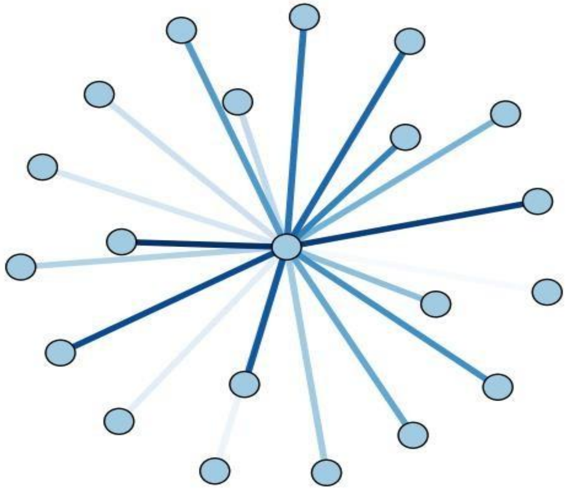
Can be slow
Beware when you are analysing very large networks

Introduction: NetworkX

A "high-productivity software for complex networks" analysis

- Data structures for representing various networks(directed, undirected, multigraphs).
- Extreme flexibility: nodes can be any hashable object in Python, edges can contain arbitrary data
- A treasure trove of graph algorithms.
- Multi-platform and easy-to-use.





When to use NetworkX

When to use

Unlike many other tools, it is designed to handle data on a scale relevant to modern problems

Most of the core algorithms rely on extremely fast legacy code

Highly flexible graph implementations

(a node/edge can be anything!)

When to avoid

Large-scale problems that require faster approaches (i.e. massive networks with 100M/1B edges)

Better use of memory/threads than Python (large objects, parallel computation)

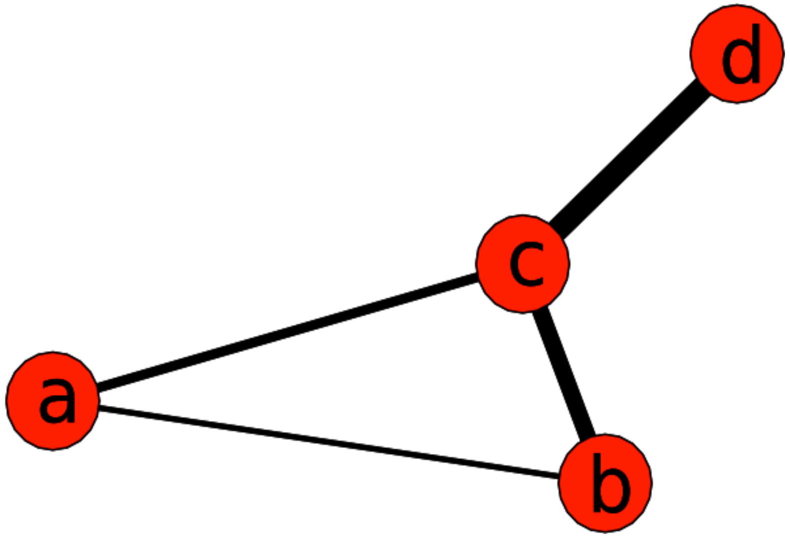
Visualization of networks is better handled

by other professional tools

Example

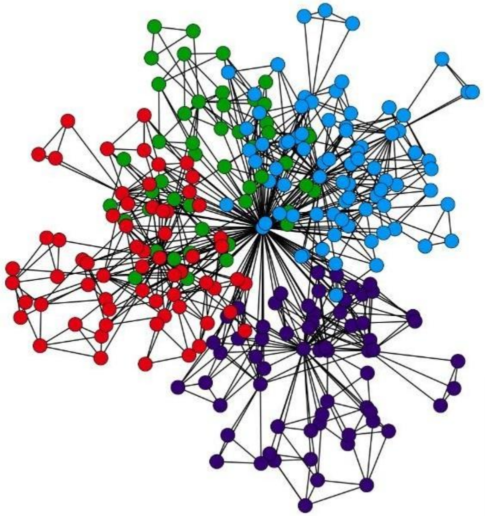
- Use Dijkstra's algorithm to find the shortest path in a weighted and unweighted network.

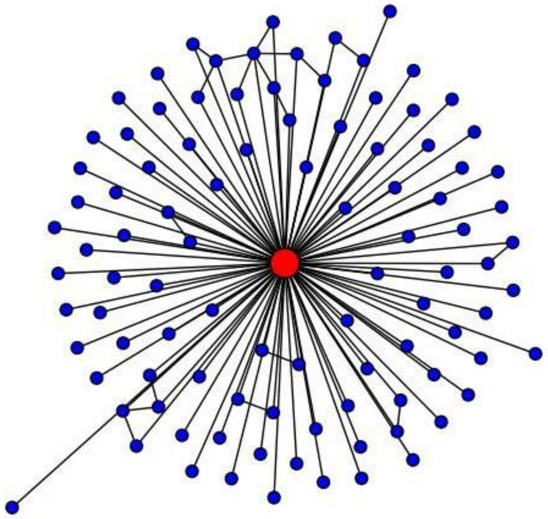
```
>>> import networkx as nx
>>> g = nx.Graph()
 >>> g.add_edge('a', 'b', weight=0.1)
>>> g.add_edge('b', 'c', weight=1.5)
>>> g.add_edge('a', 'c', weight=1.0)
>>> g.add_edge('c', 'd', weight=2.2)
>>> print nx.shortest_path(g, 'b', 'd') ['b', 'c', 'd']
>>> print nx.shortest_path(g, 'b', 'd', weight='weight')
['b', 'a', 'c', 'd']
```

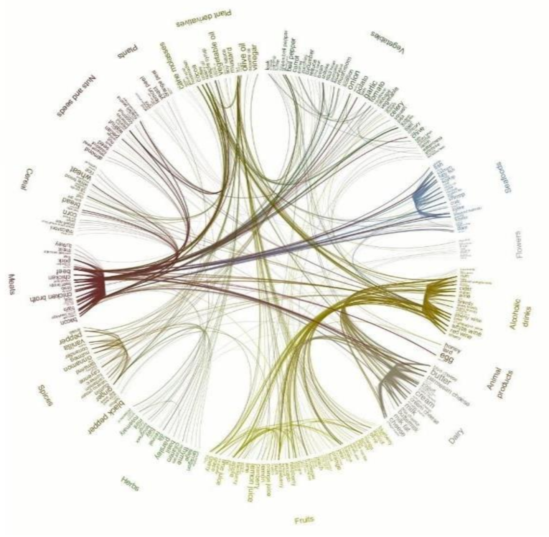


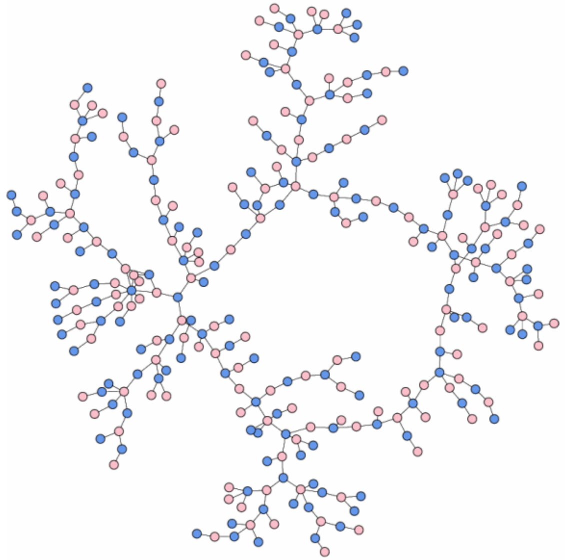
Drawing and plotting

It is possible to draw small graphs with NetworkX. You can export network data and draw with other programs (GraphViz, Gephi, etc.).



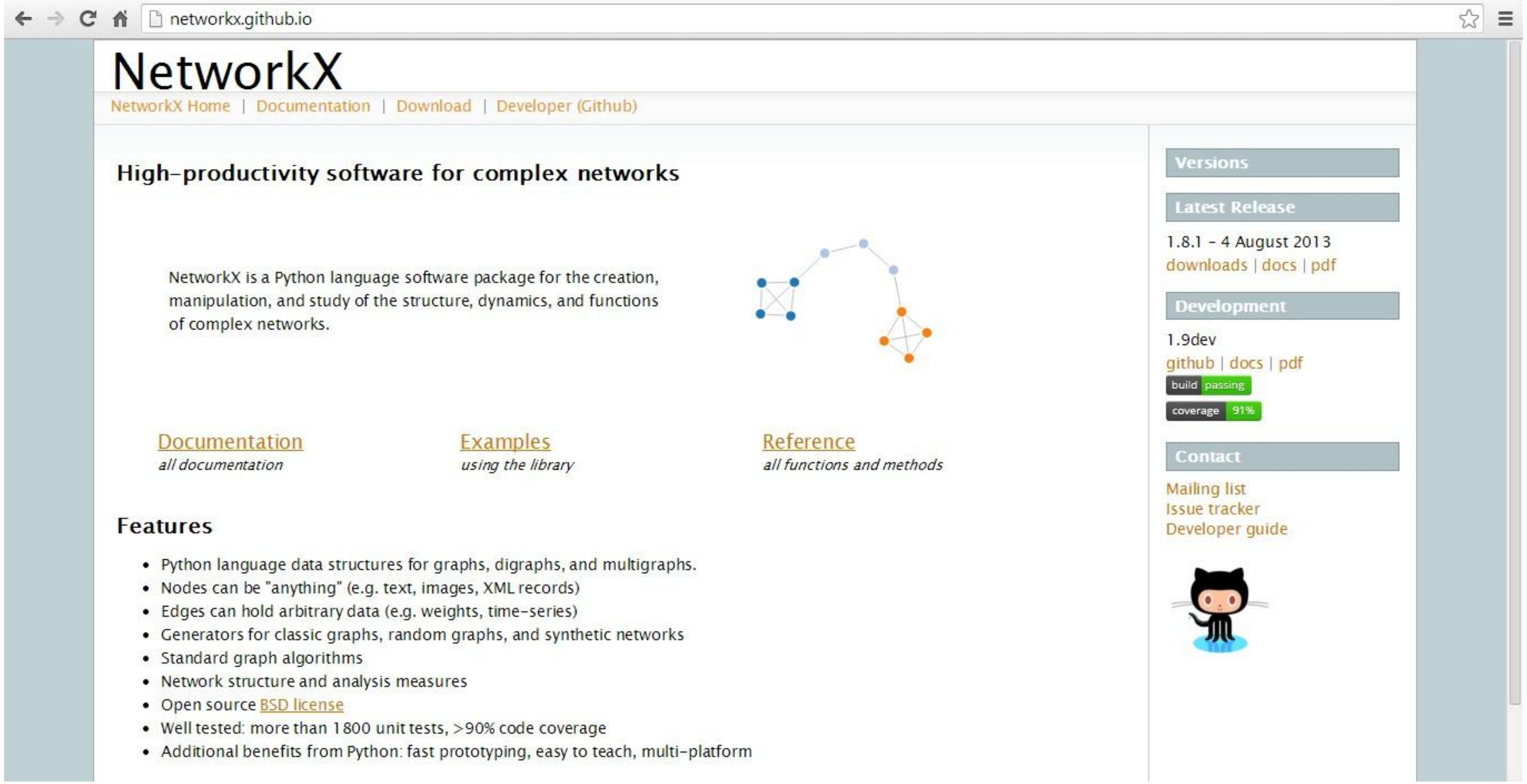






NetworkX official website

http://networkx.github.io/



Getting started with Python and NetworkX

- Start Python (interactive or script mode) and import NetworkX

```
$ python
>>> import networkx as nx
```

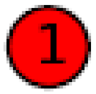
- Different classes exist for directed and undirected networks. Let’s create a basic undirected Graph:

```
>>> g = nx.Graph() # empty graph
```

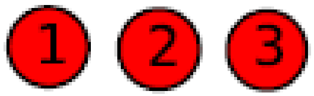
- The graph **g** can be grown in several ways. NetworkX provides many generator functions and facilities to read and write graphs in many formats.

Addition of nodes

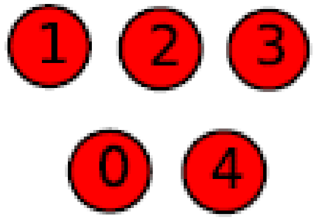

```
# One node at a time
>>> g.add_node(1)
```



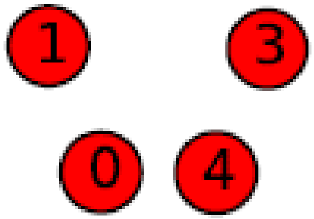

```
# A list of nodes
>>> g.add_nodes_from([2, 3])
```




```
# A container of nodes
>>> h = nx.path_graph(5)
>>> g.add_nodes_from(h)
```




```
# You can also remove any node of
the graph
>>> g.remove_node(2)
```



Node objects

- A node can be **any hashable object** such as a string, a function, a file and more.

```
>>> import math
>>> g.add_node('string')
>>> g.add_node(math.cos) # cosine
function
>>> f = open('temp.txt', 'w') #
file handle
>>> g.add_node(f)
>>> print g.nodes()
['string', <open file 'temp.txt',
mode 'w' at 0x000000000589C5D0>,
<built-in function cos>]
```

Adding edges

```
# Single edge
>>> g.add_edge(1, 2)
>>> e = (2, 3)
>>> g.add_edge(*e) # unpack tuple
```

```
# List of edges
>>> g.add_edges_from([(1, 2), (1,
3)])
```

```
# A container of edges
>>> g.add_edges_from(h.edges())
```

```
# You can also remove any edge
>>> g.remove_edge(1, 2)
```

Accessing nodes and edges


```
>>> g.add_edges_from([(1, 2), (1, 3)])
>>> g.add_node('a')
>>> g.number_of_nodes() # also g.order()
4
>>> g.number_of_edges() # also g.size()
2
>>> g.nodes() ['a', 1, 2, 3]
>>> g.edges() [(1, 2), (1, 3)]
>>> g.neighbors(1) [2, 3]
>>> g.degree(1)
2
```

Python dictionaries

- NetworkX takes advantage of Python dictionaries to store node and edge measures. The dict type is a data structure that represents a key-value mapping.

```
# Keys and values can be of any data type
>>> fruit_dict = {'apple': 1, 'orange': [0.12, 0.02], 42: True}
# Can retrieve the keys and values as Python lists (vector)
>>> fruit_dict.keys() ['orange', 42, 'apple']
# Or (key, value) tuples
>>> fruit_dict.items() [('orange', [0.12, 0.02]), (42, True),
('apple', 1)]
```

Directed Graphs


```
>>> dg = nx.DiGraph()

>>> dg.add_weighted_edges_from([(1, 4, 0.5), (3, 1, 0.75)])

>>> dg.out_degree(1, weight='weight') 0.5 >>> dg.degree(1,
weight='weight') 1.25

>>> dg.successors(1) [4]

>>> dg.predecessors(1) [3]
```

- Some algorithms work only for undirected graphs and others are not well defined for directed graphs. If you want to treat a directed graph as undirected for some measurement you should probably convert it using **Graph.to_undirected()**

Graph operators

- subgraph(G, nbunch) - induce subgraph of G on nodes in nbunch
- union(G1, G2) - graph union, G1 and G2 must be disjoint
- cartesian_product(G1, G2) - return Cartesian product graph
- compose(G1, G2) - combine graphs identifying nodes common to both
- complement(G) - graph complement
- create_empty_copy(G) - return an empty copy of the same graph class
- convert_to_undirected(G) - return an undirected representation of G
- convert_to_directed(G) - return a directed representation of G

Graph generators


```
# small famous graphs
>>> petersen = nx.petersen_graph()
>>> tutte = nx.tutte_graph()
>>> maze = nx.sedgewick_maze_graph()
>>> tet = nx.tetrahedral_graph()
```

```
# classic graphs
>>> K_5 = nx.complete_graph(5)
>>> K_3_5 = nx.complete_bipartite_graph(3, 5)
>>> barbell = nx.barbell_graph(10, 10)
>>> lollipop = nx.lollipop_graph(10, 20)
```

```
# random graphs
>>> er = nx.erdos_renyi_graph(100, 0.15)
>>> ws = nx.watts_strogatz_graph(30, 3, 0.1)
>>> ba = nx.barabasi_albert_graph(100, 5)
>>> red = nx.random_lobster(100, 0.9, 0.9)
```

Graph input/output

- General read/write

```
>>> g = nx.read_<format>('path/to/file.txt',...options...)
>>> nx.write_<format>(g,'path/to/file.txt',...options...)
```

- Read and write edge lists

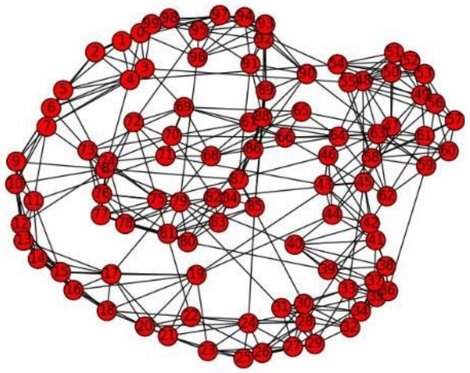
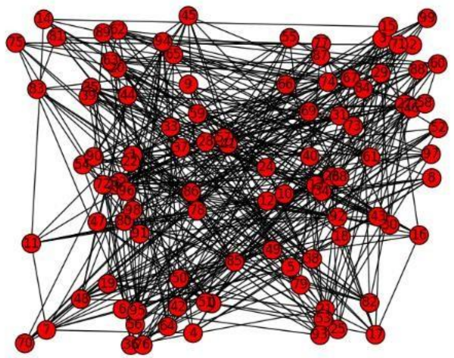
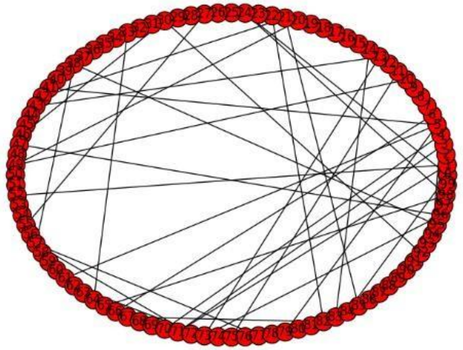
```
>>> g = nx.read_edgelist(path, comments='#', create_using=None,
delimiter=' ', nodetype=None, data=True, edgetype=None,
encoding='utf-8')
>>> nx.write_edgelist(g, path, comments='#', delimiter=' ',
data=True, encoding='utf-8')
```

- Data formats
  - Node pairs with no data: 1 2
  - Python dictionaries as data: 1 2 {'weight':7, 'color':'green'}
  - Arbitrary data: 1 2 7 green

Drawing graphs

- NetworkX is not primarily a graph drawing package but it provides basic drawing capabilities by using matplotlib. For more complex visualization techniques it provides an interface to use the open source GraphViz software package.

```
>>> import pylab as plt #import Matplotlib plotting interface
>>> g = nx.watts_strogatz_graph(100, 8, 0.1)
>>> nx.draw(g) >>> nx.draw_random(g)
>>> nx.draw_circular(g) >>> nx.draw_spectral(g)
>>> plt.savefig('graph.png')
```

Degree Distribution

- Calculate in (and out) degrees of a directed graph

```
in_degrees = cam_net.in_degree() # dictionary node:degree
in_values = sorted(set(in_degrees.values()))
in_hist = [in_degrees.values().count(x) for x in in_values]
```

- Then use matplotlib (pylab) to plot the degree distribution

```
plt.figure() # you need to first do 'import pylab as plt'
plt.grid(True)
plt.plot(in_values, in_hist, 'ro-') # in-degree
plt.plot(out_values, out_hist, 'bv-') # out-degree
plt.legend(['In-degree', 'Out-degree']) plt.xlabel('Degree')
plt.ylabel('Number of nodes') plt.title('network of places in
Cambridge') plt.xlim([0, 2*10**2])
plt.savefig('./output/cam_net_degree_distribution.pdf')
plt.close()
```

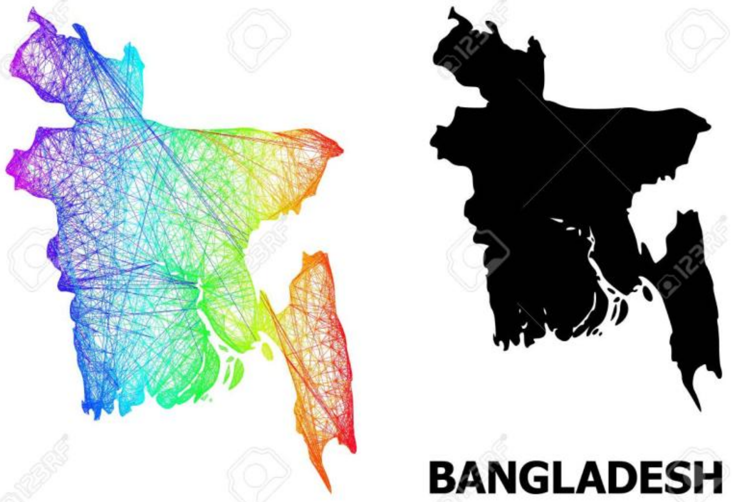
Clustering coefficient

- We can get the clustering coefficient of individual nodes or all the nodes (but first we need to convert the graph to an undirected one)

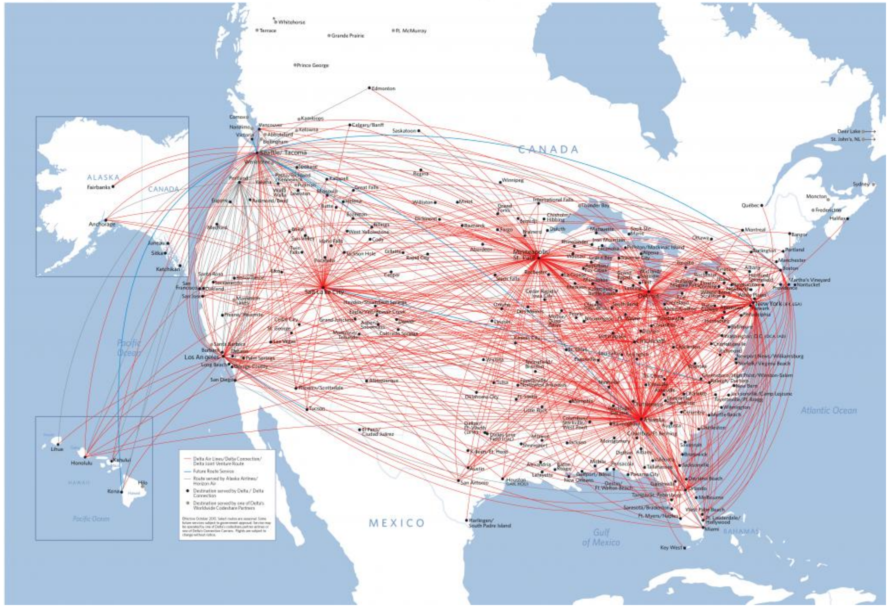
```
cam_net_ud = cam_net.to_undirected()
# Clustering coefficient of node 0 print nx.clustering(cam_net_ud, 0)
# Clustering coefficient of all nodes (in a dictionary)
clust_coefficients = nx.clustering(cam_net_ud)
# Average clustering coefficient avg_clust =
sum(clust_coefficients.values()) / len(clust_coefficients) print
avg_clust
# Or use directly the built-in method print
nx.average_clustering(cam_net_ud)
```

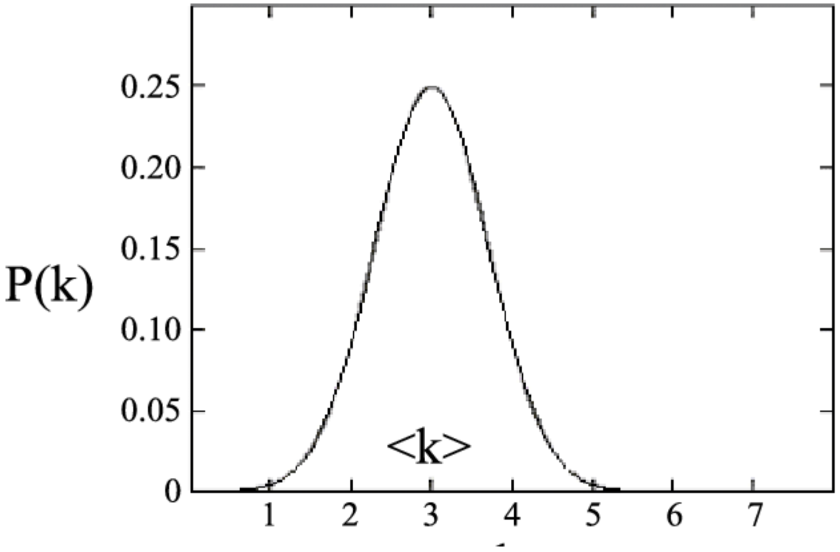
3.7 World Wide Web

Random Network

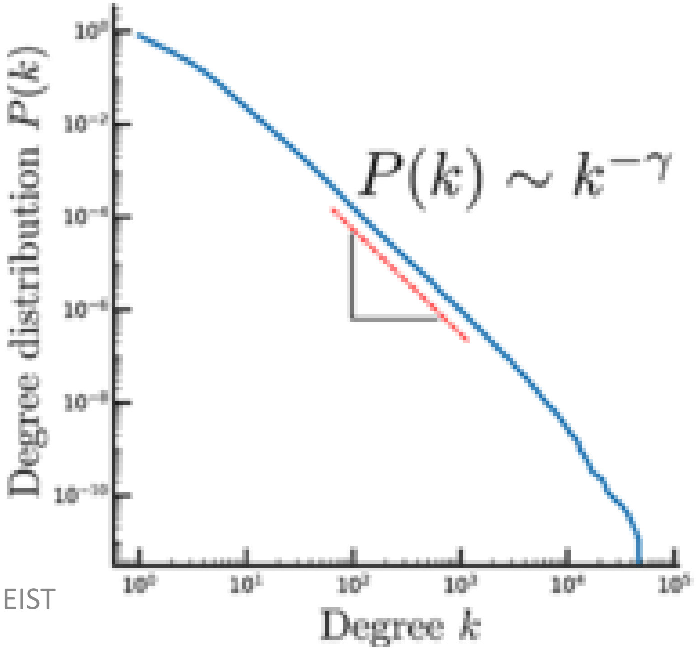


Scale-free Network





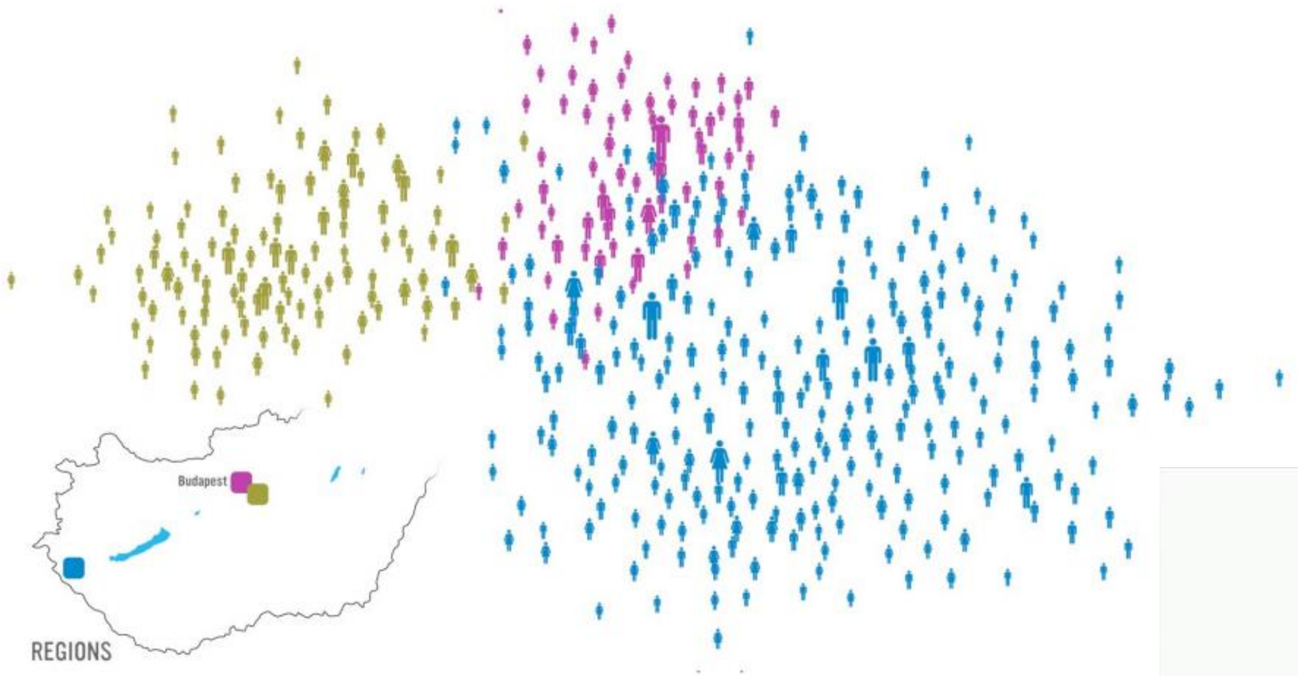
Expected

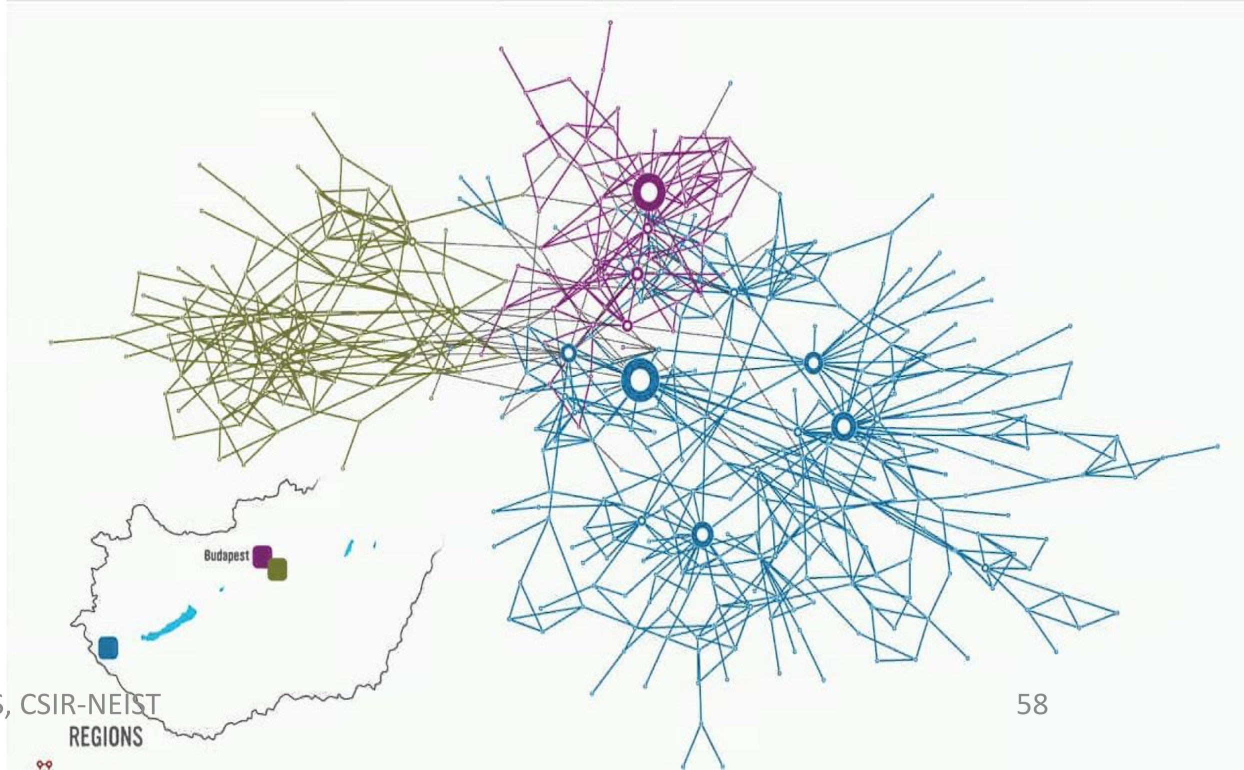


Found

R. Albert, H. Jeong, A-L Barabasi, Nature 401 130 (1999)

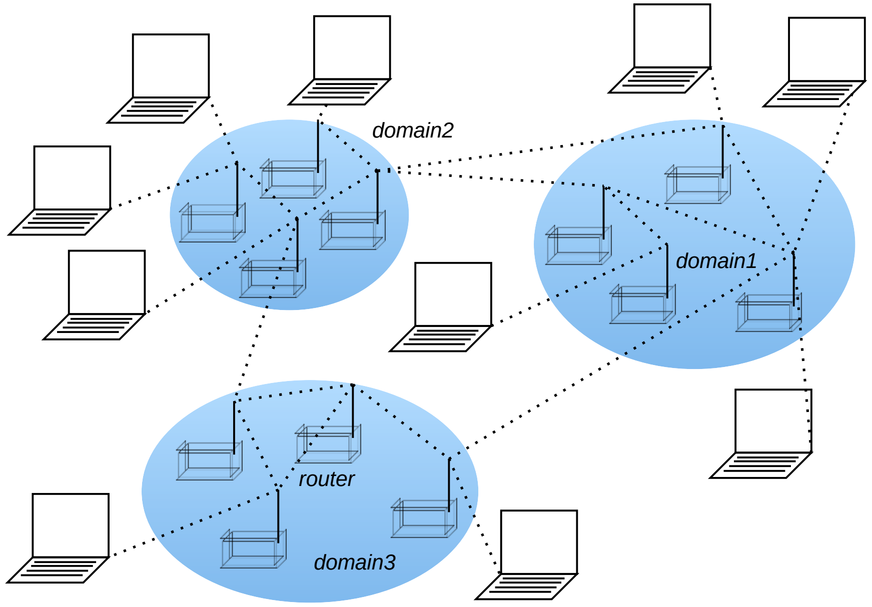
3.8 Network Behind an Organization

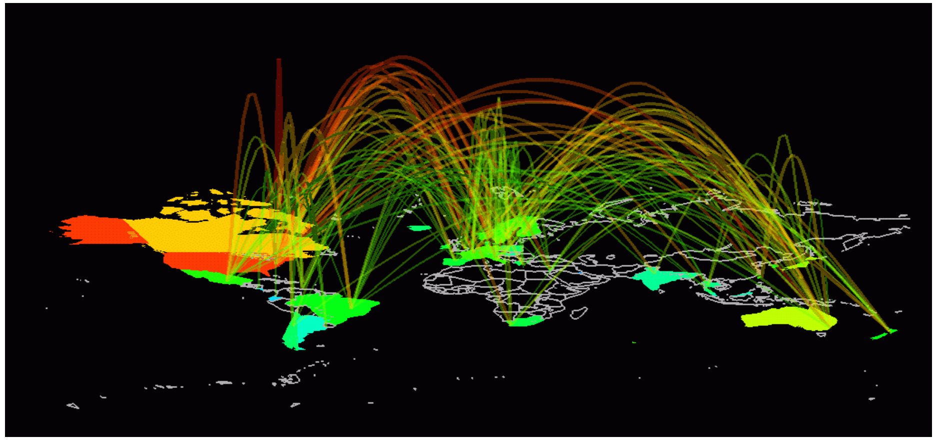


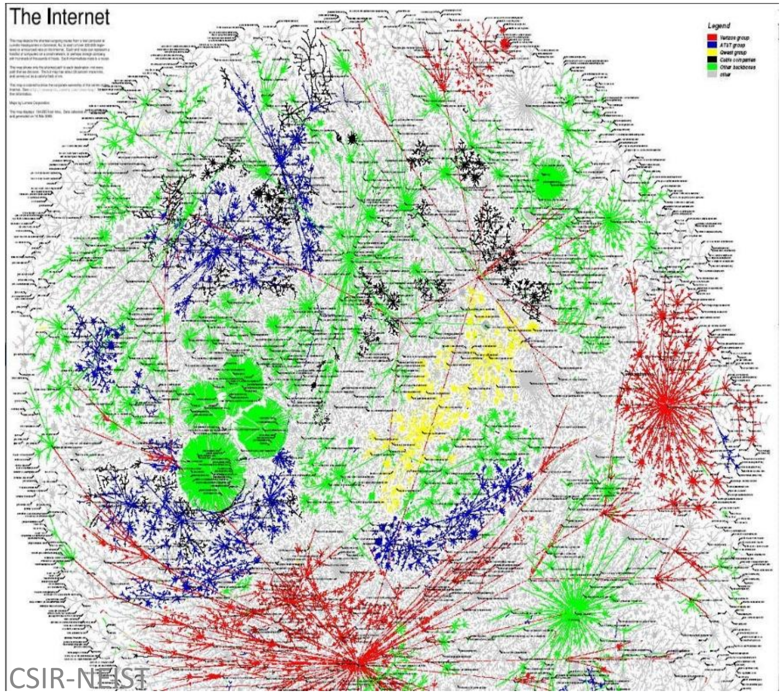


The real Network behind an Organization

3.9 The Communication Network



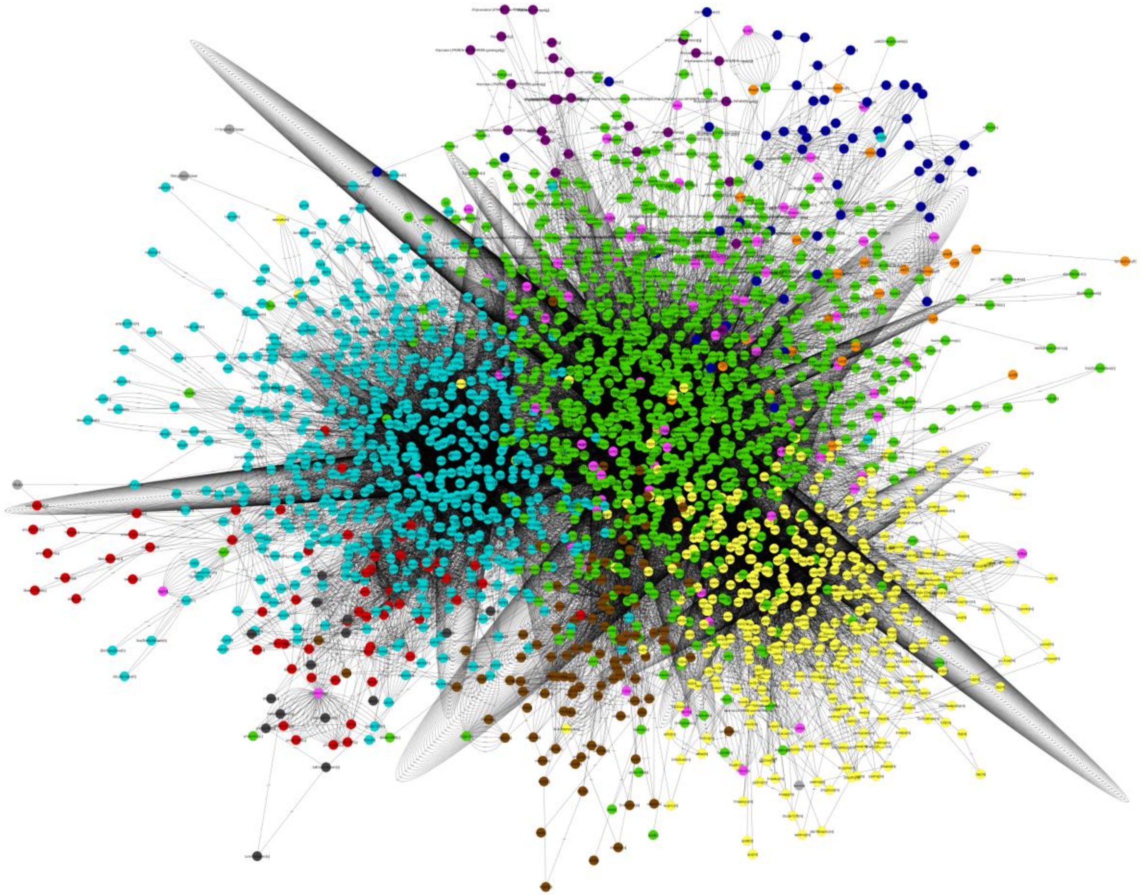




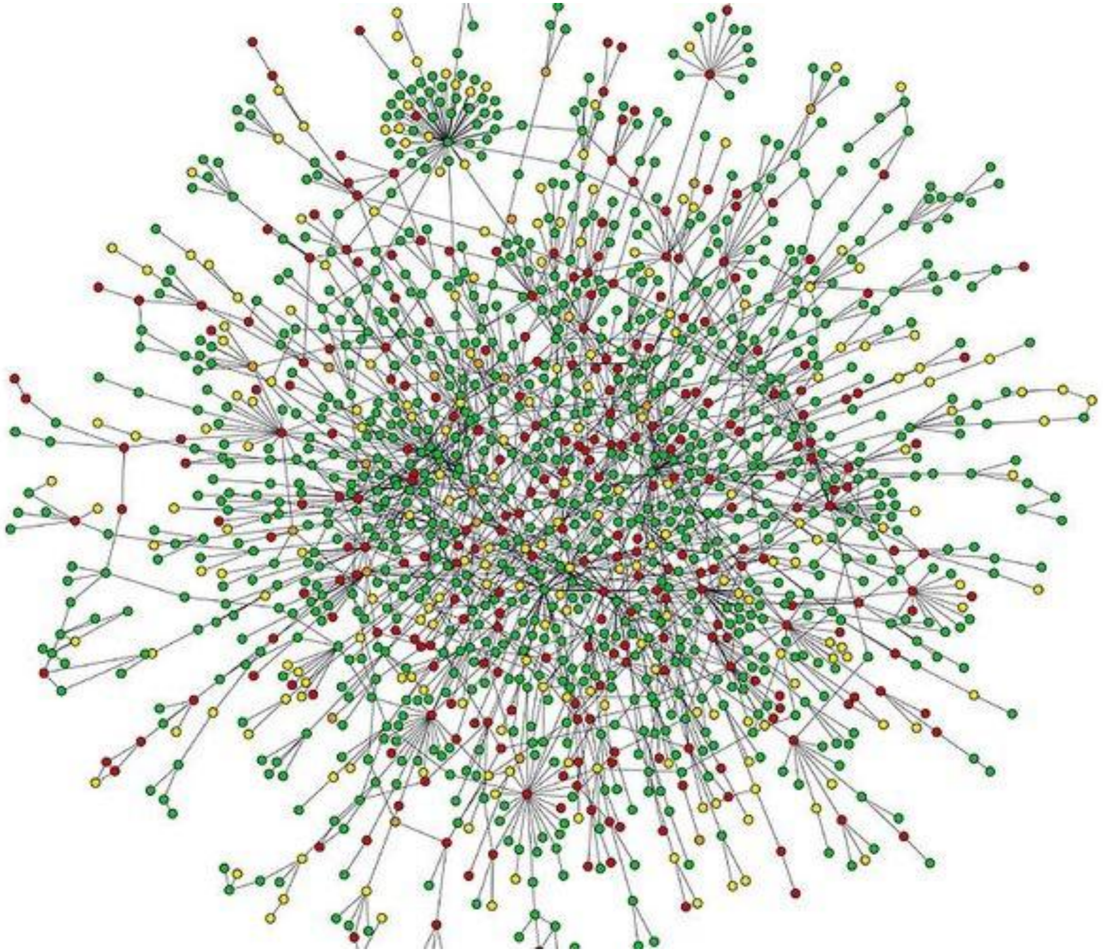
Nodes - Routers
Links - Physical links

The Internet is the key point for communication and link with each other.

3.10 Metabolic Network & Protein Interactions



Metabolic Network



Protein Interaction Network

3.11 Origin of SF Networks: Growth & preferential attachment

1. Networks continuously expand by addition of new nodes.

WWW: addition of new documents

2. New nodes prefer to link to highly connected nodes.

WWW: linking to well known sites

Growth:

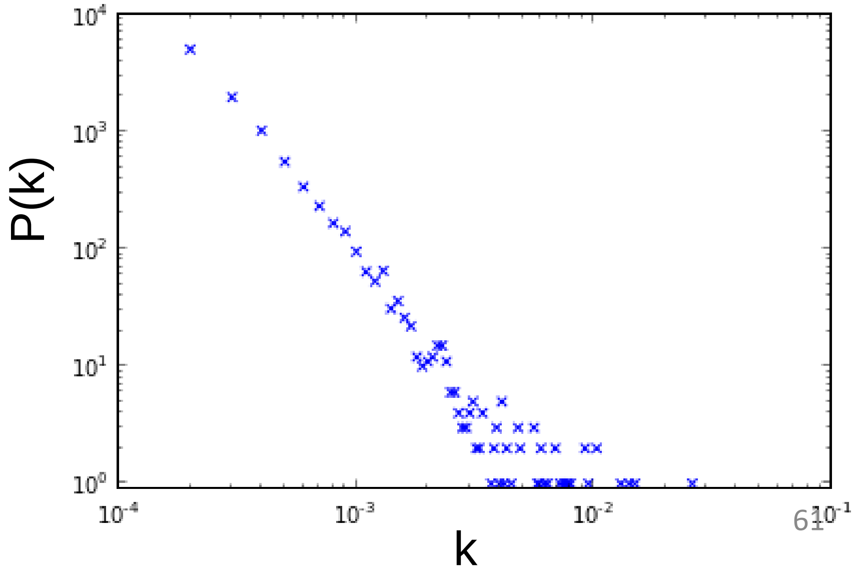
add new node with m links

Preferential attachment:

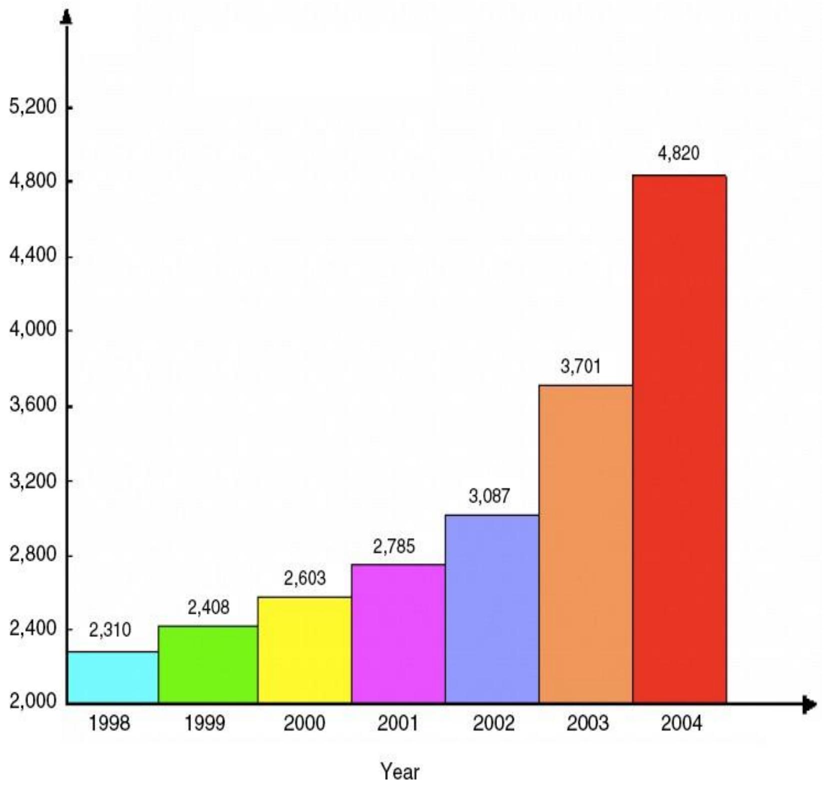
The probability that a node connects to a node with K links is proportional to k.

$$p_i = \frac{k_i}{\sum_j k_j}$$

Where k_i is the degree of node i and the sum is made over all pre-existing nodes j



3.12 Growth of network science as measured by publications



Papers with "complex networks" in the title
National Academy of Science Report,2007

Journal special issue on Network Science

3.13 Viewpoints on Network Science

Scientists forming co-evolving networks World

Engineers
- Understand how infrastructure networks work
- Design and control of these networks

Physicists and Mathematicians
- Interest and methods in complex systems
- Theories of macroscopic behavior (phase transitions)

Computer Scientists
- Understand and design complex, distributed networks
- algorithmic view: design of a system and inferring its semantics

Biologists
- Neural networks, gene regulatory networks,…
- Understanding the evolution of networks

Social Scientists, Behavioral Psychologists, Economists
- Understand human behavior in “simple” settings
- Revised views of economic rationality in humans

3.14 Key Research Challenges (Report on Network science)

1. Dynamics: Better understanding between structure and function
2. Modeling and Analysis of large networks: Tools, abstractions, approximations
3. Design and Synthesis of Networks
4. Increasing level of rigor and mathematical structure
5. Abstracting common concepts across Mields
6. Better experiments and measurements of network structure
7. Robustness and Security

3.15 Network Analysis

- **Social Network analysis :** Social network analysis examines the structure of relationships between social entities.

  These entities are often persons, but may also be groups, organizations, nation states, web sites, scholarly publications.

- **Dynamic Network Analysis:** Dynamic network analysis examines the shifting structure of relationships among different classes of entities in complex socio-technical systems effects, and reflects social stability and changes such as the emergence of new groups, topics, and leaders.
- **Biological Network Analysis:** The type of analysis in this content are closely related to social network analysis, but often focusing on local patterns in the network.
- **Link Analysis:** Link analysis is a subset of network analysis, exploring associations between objects.
- **Network Robustness:** The structural robustness of networks is studied using percolation theory.
- **Pandemic Analysis:** The SIR model is one of the most well known algorithms on predicting the spread of global pandemics within an infectious population.
- **Web link Analysis, PageRank etc**…

3.16 The Impact of Network Science

- Economic Impact
- Network Biology/Network Medicine
- Human Disease Network
- Drug Design, Metabolic Engineering
- Fighting Terrorism and Military
- Epidemic Forecast
- Brain Research

3.16.1 Economic Impact

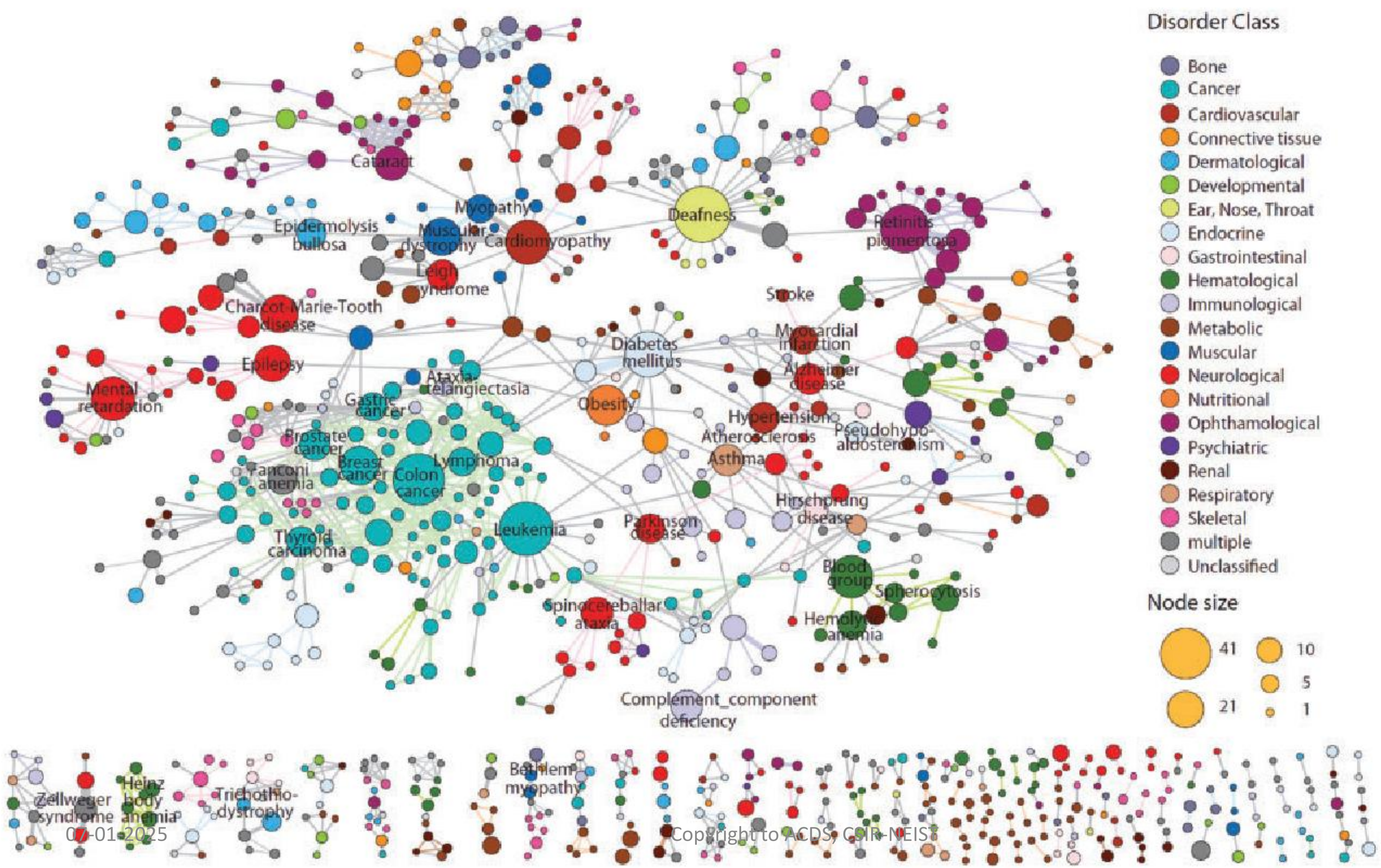
Google
Market Cap(Jan 1, 2010):
$189 billion

Cisco Systems
networking gear Market cap (Jan 1, 2010):
$112 billion

Facebook
market cap:
$50 billion

www.bizjournals.com/austin/news/2010/11/15/facebooks... - Cached

3.16.2 Human Disease Network



3.17 Characteristics of Network Science

1. **Structure :** Networks have structure-they are not random collections of nodes and links.

   Example- The structures of electrical power grids, online social networks are not random, but instead have a distinct format or topology.

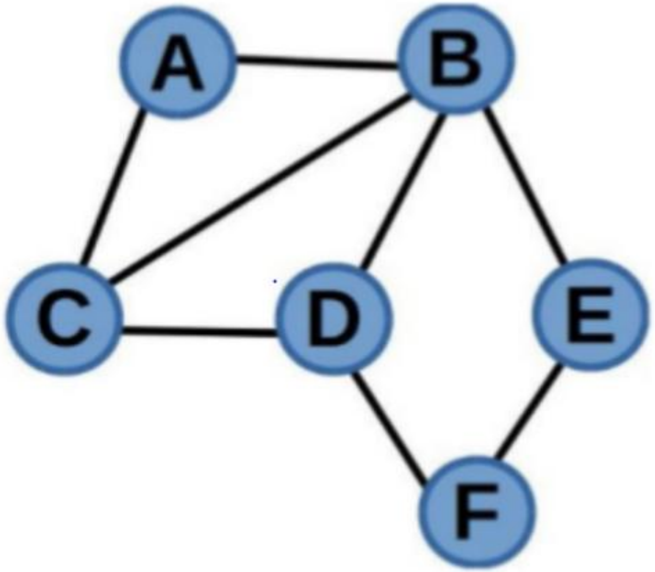
   Function follows form-many real-world phenomena behave the way they do because of their network structure.

2. **Emergence:** Emergence is a network synchronization issue—stable networks transition from one state to another until they reach a fixed point, and stay there.
3. **Dynamism:** Network science is concerned with both structure and dynamic behavior of networks. Dynamic behavior is often the result of emergence or a series of small evolutionary steps leading to a fixed-point final state of the system.
4. **Autonomy:** A network forms by the autonomous and spontaneous action of independent nodes that "volunteer" to come together (link), rather than through central control or central planning.
5. **Topology:** The architecture or topology of a network is a property that emerges over time as a consequence of distributed-and often subtle-forces or autonomous behaviors of its nodes.

   A network is dynamic if its topology or other properties change as a function of time.

Worked out Examples

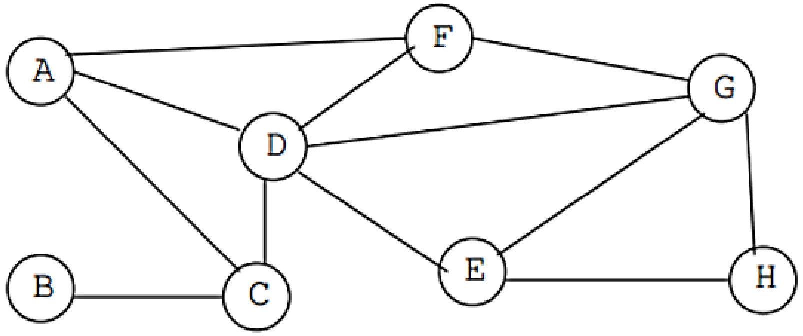
1. Given a network being generated by 'rich get richer' phenomenon. The following figure shows the snapshot of the network at time t. A new node 'u' enters the network at time t+1 and makes an edge with one of the existing nodes . The probability of 'U' making an edge with an existing node 'w' is defined as p(w). Which of the following equations is correct?



- P(A) < p(c) < p(B)
- P(E) < p(D) < p(B)
- P(F) < p(c) < p(B)
- All of the above

Answer: All of the above

2. Characterize the structure of the network showed below, using classic metrics.



a) What is the number of nodes and edges in this network?

Answer: Nodes = 8; Edges = 12

b) Explain the concept of density and calculate the graphs' density.

Answer: Density is a metric that varies from 0 to 1 and tells how far apart the actual number of edges is from the possible number of edges.

This graphs' density is calculated like this: 2*(number of nodes)/(number of edges)*(number of edges1) ~ 0.121212...

c) Calculate this networks' mean degree.

Answer: 23/8 = 2.875

Worked out Examples

d) Which vertices are the most far apart from each other? What is the longest path?

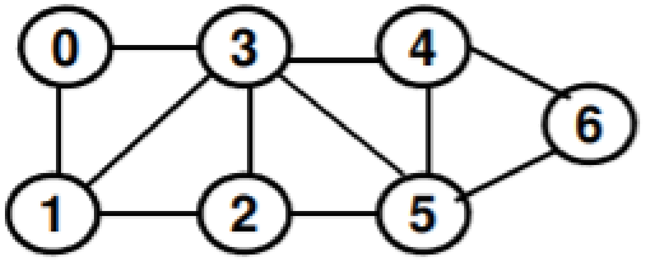
Hint: do not overestimate the path, use always the shortest longest one. For instance: from B to D, use B >> C >> D, instead of B >> C >> A >> F >> D.

Answer: B and H. The longest path is B >> C >> D >> E >> H.

3. For the graph given, find whether or not the links happened by chance ?

For this, do the following:

a) Find the frequency (probability) distribution of the degree of the vertices in the actual graph.



Solution:

Node	Degree
0	2
1	3
2	3
3	5
4	3
5	4
6	2

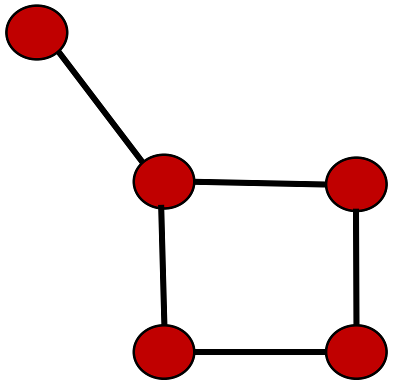
Degree	# nodes	P(k)
2	2	2/7
3	3	3/7
4	1	1/7
5	1	1/7

Avg. Degree = 2*2/7 + 3*3/7 + 4*4/7 +4*1/7 + 5* 1/7
= 3.14

Example: $p(2) = \exp(-3.14) * (3.14)^2 / 2!$
= 0.2154

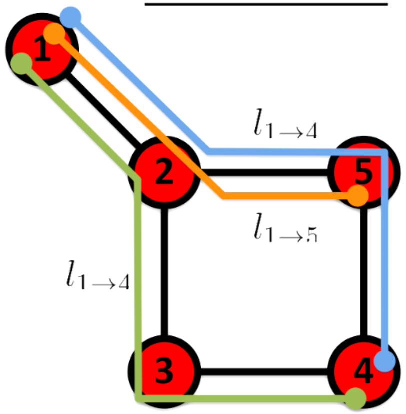
Worked out Examples

4.For the graph given, find the shortest path, diameter and average path length.



Solution:

Shortest Path

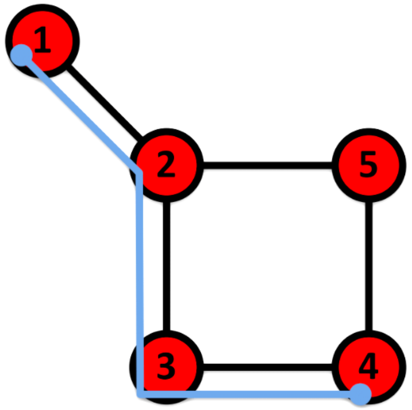


$l_{1\to4} = 3$

$l_{1\to5} = 2$

The path with the shortest length between two nodes (distance).

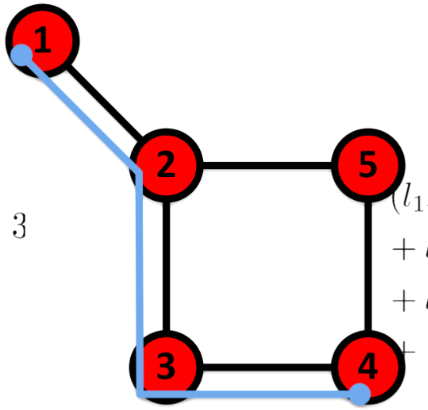
Diameter



$l_{1\to4} = 3$

The longest shortest path in a graph

Average Path Length

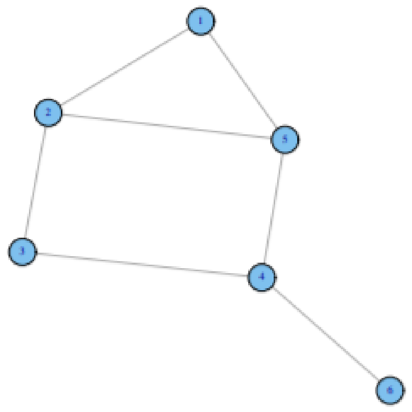


$$(l_{1\to2} + l_{1\to3} + l_{1\to4} + l_{1\to5} + l_{2\to3} + l_{2\to4} + l_{2\to5} + l_{3\to4} + l_{3\to5} + l_{4\to5})/10 = 1.6$$

The average of the shortest paths for all pairs of nodes.

Worked out Examples

5. In the following undirected graph find the degree of distribution.



Solution:

- Degree distribution: A frequency count of the occurrence of each degree. First the degrees are listed below:

Node	Degree
1	2
2	3
3	2
4	3
5	3
6	1

- The degree distribution therefore is:

degree	frequency
1	1/6
2	2/6
3	3/6

Worked out Examples

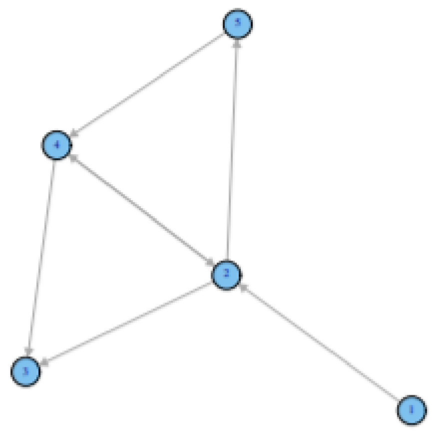
- Average degree: let N = |V | be the number of nodes, and L = |E| be the number of edges:

$$\langle K \rangle = \frac{\sum_{i=1}^{n} deg(i)}{N} = \frac{2L}{N}$$

- <k> = 2(7)/6 = 7/3 for the above graph.

Worked out Examples

6. In the following directed graph find the degree and degree distribution.



Solution:

- Indegree of any node i: the number of nodes destined to i.
- Outdegree of any node i: the number of nodes originated at i.
- Every loop adds one degree to each of the indegree and outdegree of a node.

node	In degree	Out degree
1	0	1
2	2	3
3	2	0
4	2	2
5	1	1

Worked out Examples

- Degree distribution: A frequency count of the occurrence of each degree

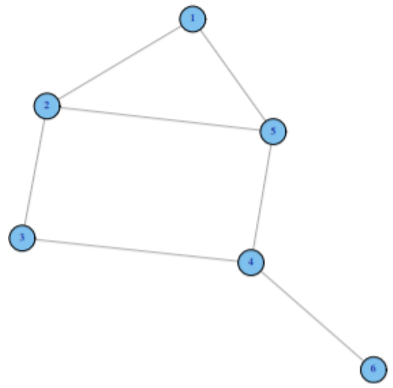
indegree	frequency	outdegree	frequency
0	1/5	0	1/5
1	1/5	1	2/5
2	3/5	2	1/5
		3	1/5

- Average degree: let N = |V | be the number of nodes, and L = |E| be the number of arcs:

$$\langle K^{in} \rangle = \frac{\sum_{i=1}^{n} deg_{in}(i)}{N} = \frac{\sum_{i=1}^{n} deg_{out}(i)}{N} = \frac{L}{N}$$

- $\langle K^{in} \rangle = \langle K^{out} \rangle = 7/5$ for the above graph.

7. In the following undirected graph find the path distance distribution



Solution:

Path distribution: A frequency count of the occurrence of each path distance.
First the path distances are listed below:

	1	2	3	4	5	6
1	0	1	2	2	1	3
2	1	0	1	2	1	3
3	2	1	0	1	2	2
4	2	3	1	0	1	1
5	1	1	2	1	0	2
6	3	3	2	1	2	0

Worked out Examples

- The path distance distribution D therefore is:

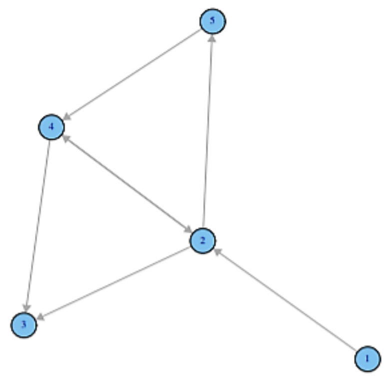
distance	frequency
1	7/15
2	6/15
3	2/15

- Average path distance: let N = |V | be the number of nodes:

$$\langle D \rangle = \frac{\sum_{i=1}^{n} dist(i, j)}{\binom{N}{2}}$$

- <D> = E[D] = 5/3 for the above graph.

8. In the following directed graph find the path distance distribution.



Solution:

- Path distribution: A frequency count of the occurrence of each path distance.
- First the path distances are listed below:

	1	2	3	4	5
1	0	1	2	2	2
2	inf	0	1	1	1
3	inf	inf	0	inf	inf
4	inf	1	1	0	2
5	inf	2	2	1	0

Worked out Examples

- The path distance distribution D therefore is:

Distance	Frequency
1	7/13
2	6/13

- Average path distance: let N = |V | be the number of nodes:

$$\langle D \rangle = \frac{\sum_{i<1} dist(i,j)}{\binom{N}{2}}$$

- <D> = E[D] = 19/13 for the above graph.

Worked out Examples

9. Consider a random network generated according to the G(N, p) model where the total number of nodes is 12 and the probability that there are links between any two nodes is 0.20. Determine the following:

1. The probability that there are exactly 60 links in the network
2. The average number of links in the network
3. The average node degree
4. The average path length (distance between any two nodes in the network)

Solution:

- There are N = 12 nodes
- Prob[link between any two nodes] = p = 0.2

Max. possible number of links between any two nodes is (N)(N-1)/2 = (12*11/2) = 66

1. Prob[there are exactly 60 links in the network] = C(66, 60) * p60 * (1-p)(66-60)
   C(66, 60) = 66! / (60! * 6!)
   = 60! *61*62*63*64*65*66 / (60!*1*2*3*4*5*6)
   = 90858768
   Prob[there are exactly 60 links in the network]
   = 90858768 * (0.2)60 * (0.8) 6
   = 2.75 * 10-35

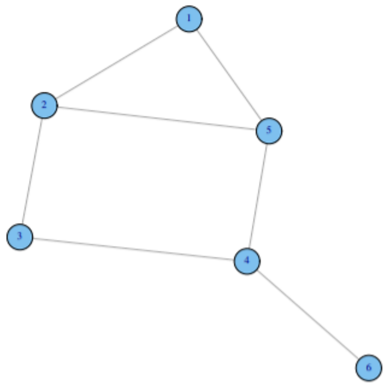
Worked out Examples

2. The average number of links in the network = p * N(N-1)/2 = 0.2 * 66 = 13.2

3. Average node degree = p*(N-1) = 0.2 * 11 = 2.2

4. Average path length = ln N / ln = ln(12) / ln(2.2) = 3.15

Q.10 For the given undirected graph find out the clustering coefficient distribution.



Solution:

Clustering coefficient distribution: A frequency count of the occurrence of each clustering coefficient.

First the clustering coefficient are listed below:

node	clustering coefficient
1	1
2	1/3
3	0
4	0
5	1/3
6	NaN

The Clustering coefficient Distribution therefore is:

Clustering coefficient C	Frequency
0	2/5
1/3	2/5
1	1/5

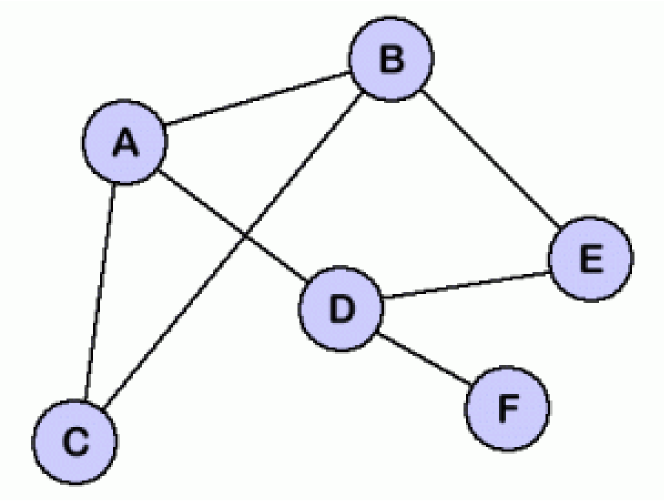
Average Clustering coefficient: let N = |V | be the number of nodes:

$$\langle C \rangle = \frac{\sum_{i=1}^{n} CC(I)}{N}$$

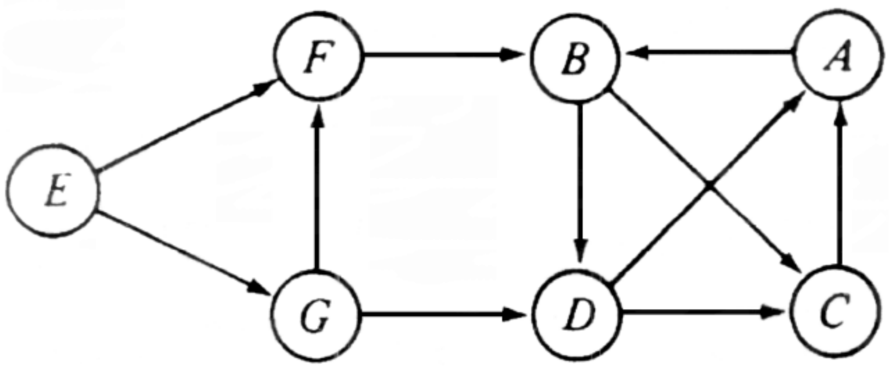
<C>= E[C] = 1/3 for the above graph.

Worked out Examples

11. Find out the adjacency matrix of the following two network graphs.



Undirected Graph



Directed Graph

Worked out Examples

Solution:

	A	B	C	D	E	F
A	0	1	1	1	0	0
B	1	0	1	0	1	0
C	1	1	0	0	0	0
D	1	0	0	0	1	1
E	0	1	0	1	0	0
F	0	0	0	1	0	0

	A	B	C	D	E	F	G
A	0	1	0	0	0	0	0
B	0	0	1	1	0	0	0
C	1	0	0	0	0	0	0
D	1	0	1	0	0	0	0
E	0	0	0	0	0	1	1
F	0	1	0	0	0	0	0
G	0	0	0	1	0	1	0

Worked out Examples

12. A project schedule has the following chart

Activity	1-2	1-3	2-4	3-4	3-5	4-9	5-6	5-7	6-8	7-8
Time	4	1	1	1	6	5	4	8	1	2

1. Construct network graph.
2. Compute earliest event time and latest event time.
3. Determine the critical path and total project duration.
4. Compute total and free float for each activity.

Worked out Examples

□ → Latest time
□ → Earliest time

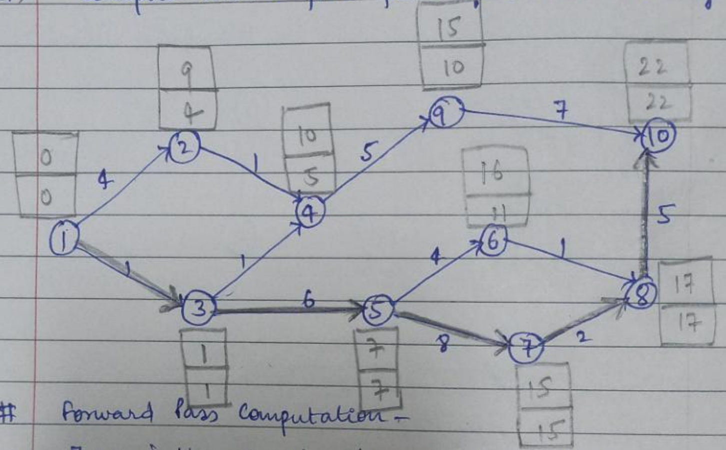
23/2/2021

Critical Path Method (CPM)

Qu- A project schedule has the following char-

Activity	1-2	1-3	2-4	3-4	3-5	4-9	5-6	5-7	6-8	7-8	8-10	9-10
Time (days)	4	1	1	1	6	5	4	8	1	2	5	7

(1) construct network graph
(2) compute earliest event time & latest event time
(3) determine the critical path & total project duration
(4) compute total & free float for each activity



Forward Pass Computation-

- Zero is the starting time for the project
- $(EF)_{ij} = (ES)_{ij} + t_{ij}$
- $E_j = Max_i \, (E_i + t_{ij})$

Backward Pass Computation-

- For ending event assume, $E = L$
- $(LF_{ij}) = (LF)_{ij} - t_{ij}$
  $= L_j - t_{ij}$
- $L_j = Min_i \, (L_j - t_{ij})$

Critical Activity- If total Float (TF_{ij}) for any activity (i,j) is zero then its called Critical Act.

- $ES_i = LF_i$
- $ES_j = LF_j$
- $ES_j - ES_i = LF_j - LF_i = t_{ij}$

Now following table gives us other information-

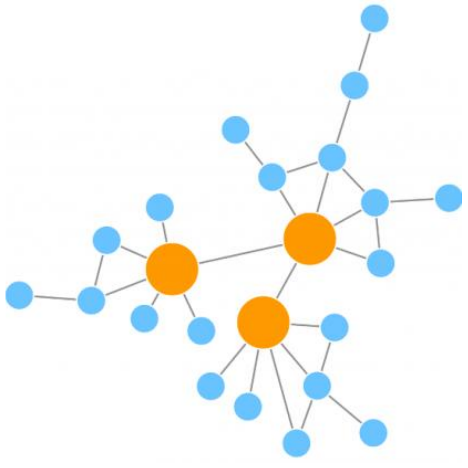
	Activity	Time (day) t_{ij}	Earliest Start (ES) E_i	Earliest Finish (EF) $E_i + t_{ij}$	Latest (LF) Finish L_j	Latest (LF) Start $L_j - t_{ij}$	(TF) Total Float LS-ES	E_j
0	1-2	4	0	4	9	5	5	4
0	1-3	1	0	1	1	0	0	1
0	2-4	1	4	5	10	9	5	5
3	3-4	1	1	2	10	9	8	5
0	3-5	6	1	7	7	1	0	7
0	4-9	5	5	10	15	10	5	10
0	5-6	4	7	11	16	12	5	11
0	5-7	8	7	15	15	7	0	15
5	6-8	1	11	12	17	16	5	17
0	7-8	2	15	17	17	15	0	17
0	8-10	5	17	12	22	17	0	22
5	9-10	7	10	17	22	15	5	22

Critical Path ⇒ 1 → 3 → 5 → 7 → 8 → 10

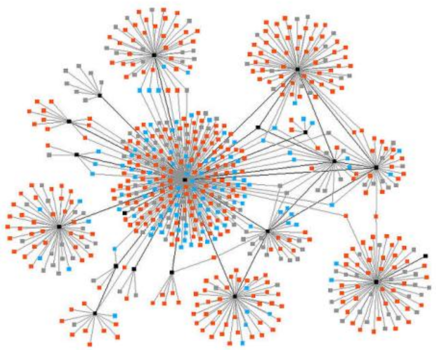
1 + 6 + 8 + 2 + 5 = 22 → Total Project Duration

Worked out Examples

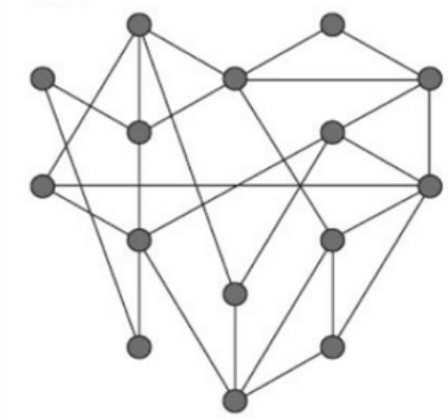
13. Different types of graphs are given below. Identify the type of graph with explanation.



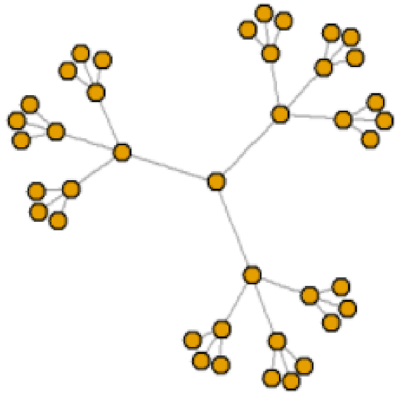
1



2



3



4

Solution:

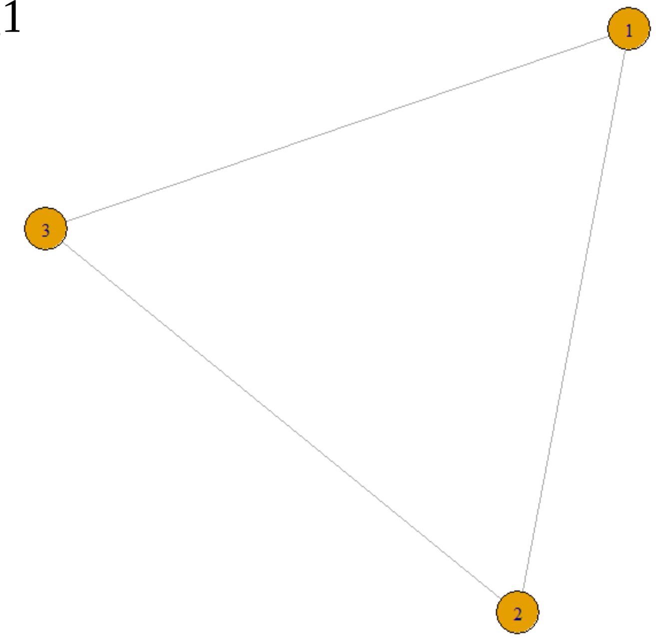
1. Scale-free network
2. Complex Network
3. Random Network
4. Tree Network

Worked out Examples

14. Write code in R to draw a network graph having vertices-(1,2,3) and edges-3

Solution:

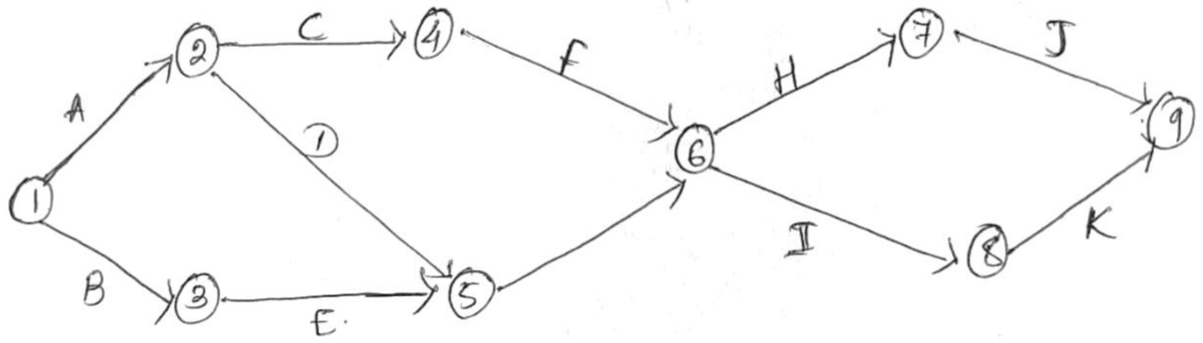
```
install.packages("igraph")
library("igraph")
g1 <- graph( edges=c(1,2, 2,3, 3, 1), n=3, directed=F )
plot(g1
```



15. Construct a network diagram for the following chart.

Activity	Preceding Act
A	-
B	-
C	A
D	A
E	B
F	C
G	D,E
H	F,G
I	F,G
J	H
K	I

Solution:

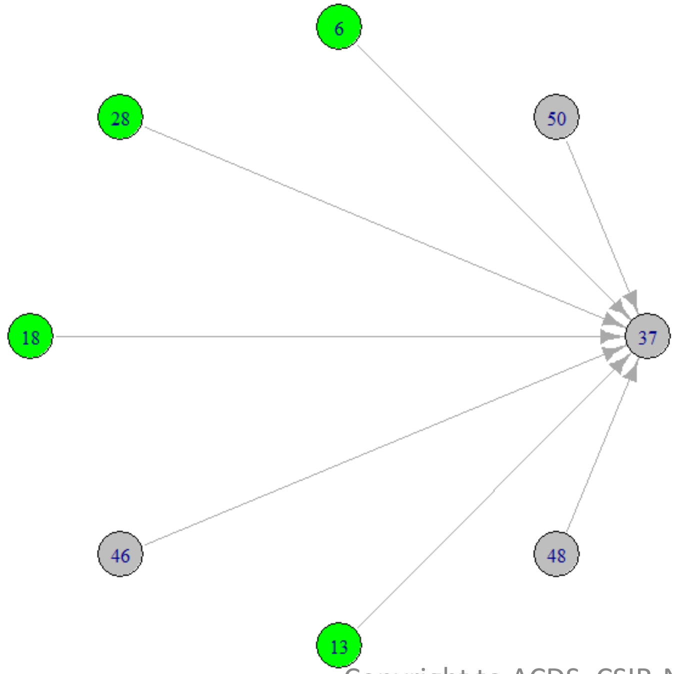


Worked out Examples

16. Create a star graph with 8 edges. Next, assign random numbers between 1 and 50 to vertices and select vertices for which this value is less than 30, set the color of these vertices to green.

Solution:

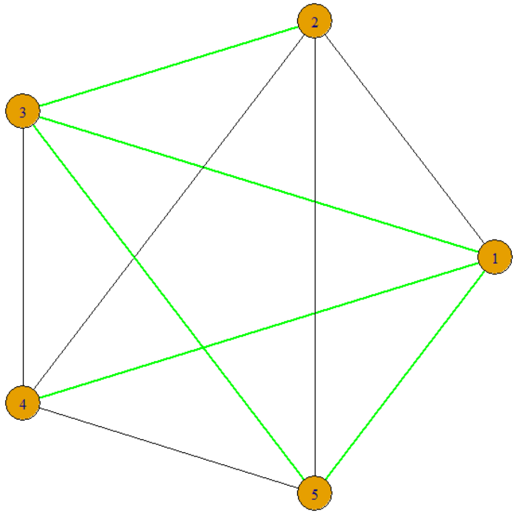
```
install.packages("igraph")
library("igraph")
g <- graph.star(8)
V(g)$number <- sample(1:50, vcount(g), replace=TRUE)
V(g)$color <- "grey"
V(g)[ number < 30 ]$color <- "green"
plot(g, layout=layout.circle, vertex.color=V(g)$color, vertex.label=V(g)$number)
```



17. Create a complete graph with 5 edges. Next, assign random weights between 0 and 1 to all of edges. Set width of edges to 2 and color to green for those of them that have weight less than 0.5, set the width value to 1, color to red for others.

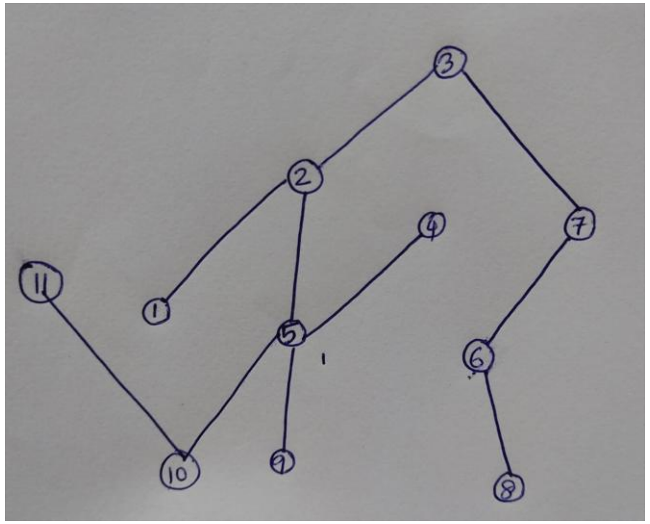
Solution:

```
install.packages("igraph")
library("igraph")
g <- graph.full(5)
E(g)$weight <- runif(ecount(g))
E(g)$width <- 1
E(g)$color <- "black"
E(g)[ weight < 0.5 ]$width <- 2
E(g)[ weight < 0.5 ]$color <- "green"
plot(g, layout=layout.circle, edge.width=E(g)$width, edge.color= E(g)$color)
```



Worked out Examples

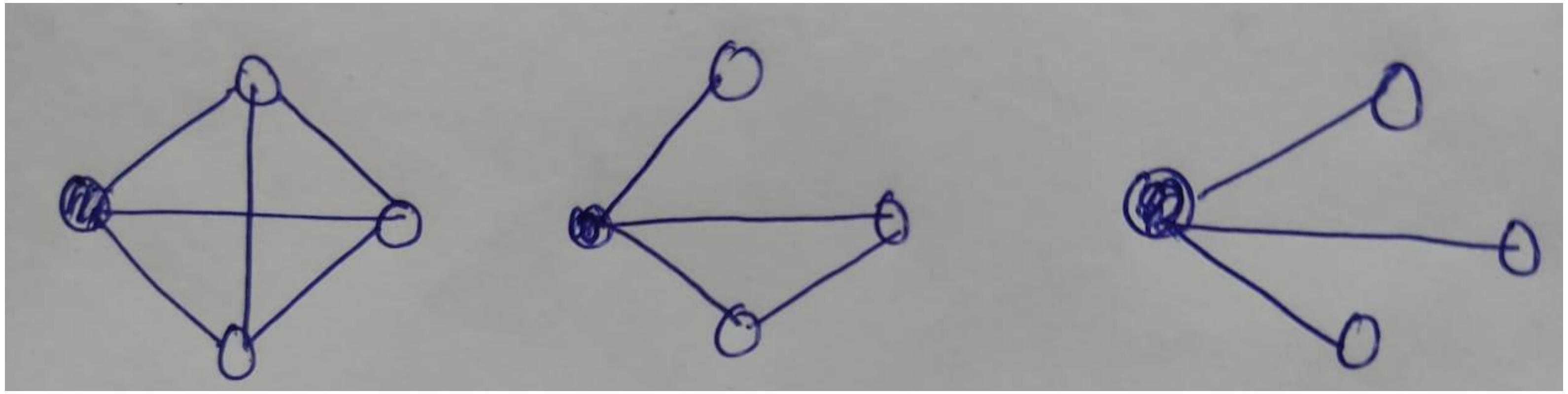
18. The following is a social network, find out the which is the hub node?



Solution: Vertex (5), is the hub node. Most of the nodes are connected to node 5 (which has highest degree among all)

Worked out Examples

19. Find the average clustering coefficient?



Graph 1 Graph 2 Graph 3

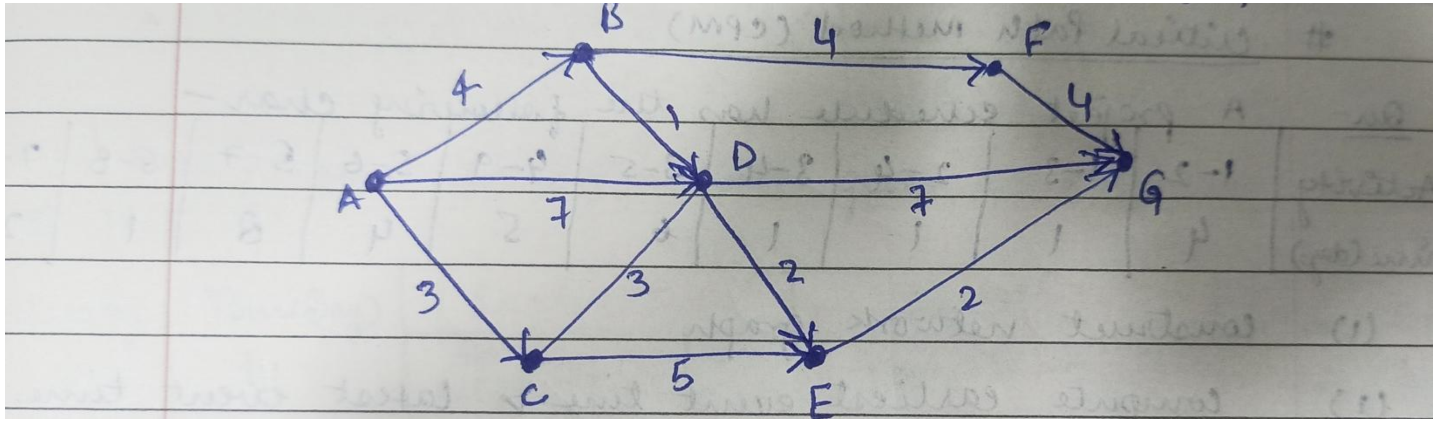
Solution: Using the formula we get

$$C_i = \frac{\text{number of triangles connected to node } i}{\text{number of triples centered around node } i},$$

C = 1 for graph 1
C = 1/3 for graph 2
C= 0 for graph 3

Worked out Examples

20. Construct an activity graph for the above network.

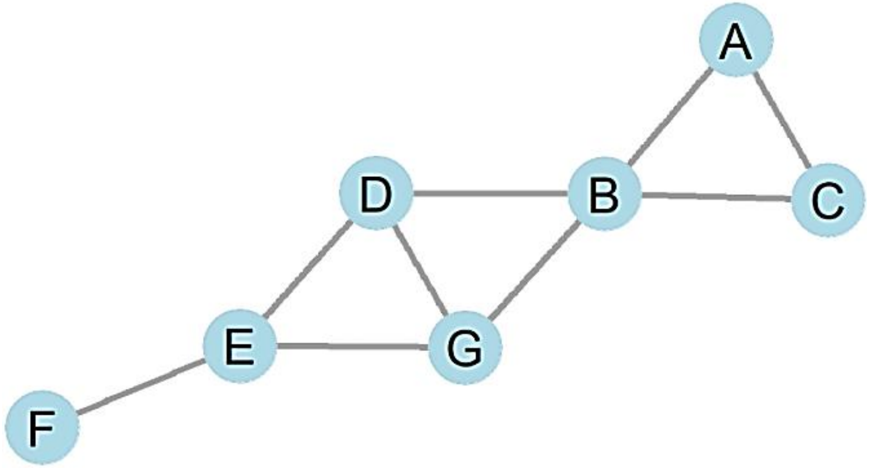


Solution:

Activity	A→B	A→D	A→C	B→D	B→F
Length	4	7	3	1	4
Activity	D→G	D→E	C→D	C→E	F→G
Length	7	2	3	5	4
Activity	E→G				
Length	2				

Exercise

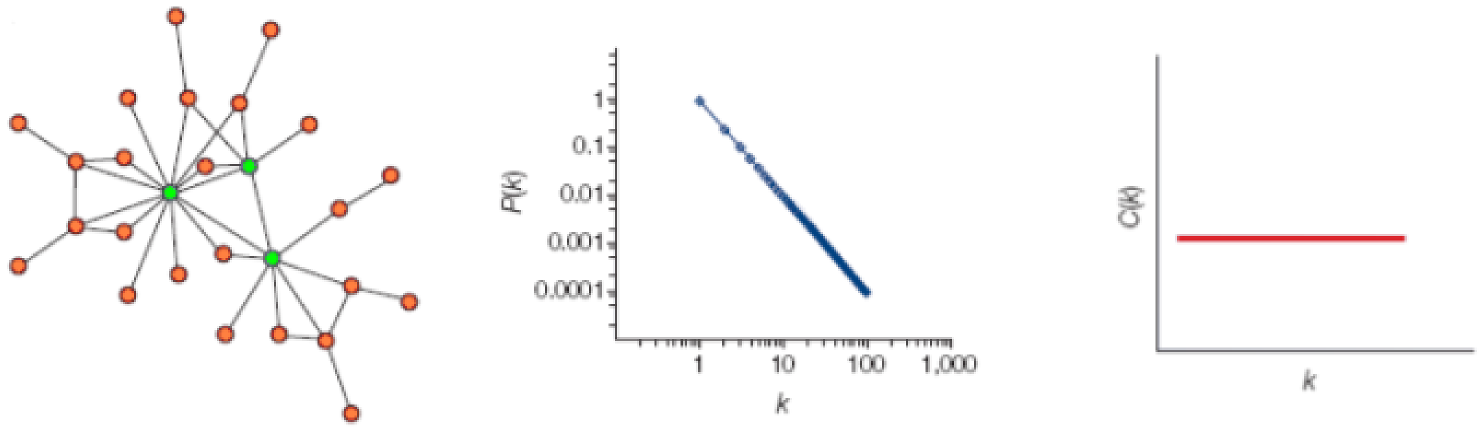
1.



a) What type of graph is pictured above? What type of relationships might the graph represent?
b) How many nodes and edges are in the above graph?
c) What is the average degree of the above graph?
d) What is the degree distribution of the above graph?
e) What is the density of the above graph?

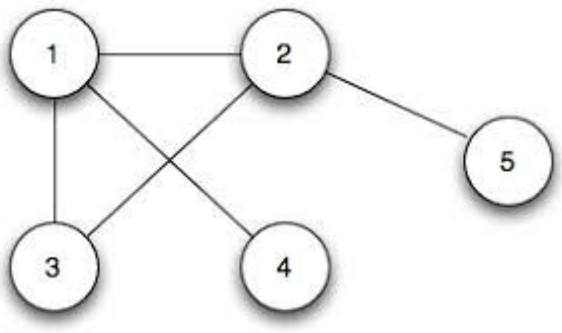
Exercise

2. Describe the network shown in the figure below (left panel), focusing on how well network nodes are connected. Describe what is shown in the two graphs to the right of it and discuss what analysis method has been used to produce these graphs, and how they can help to understand the network.

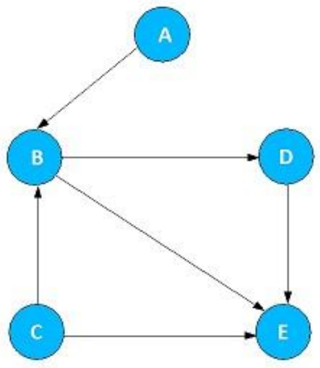


Exercise

3. In the following undirected graph find the degree of distribution.

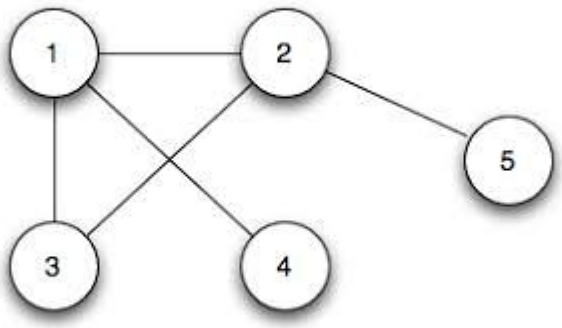


4. In the following directed graph find the degree and degree distribution.

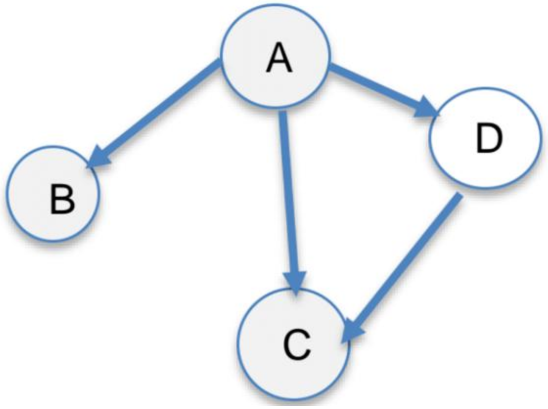


Exercise

5. In the following undirected graph find the path distance distribution



6. In the following directed graph find the path distance distribution.

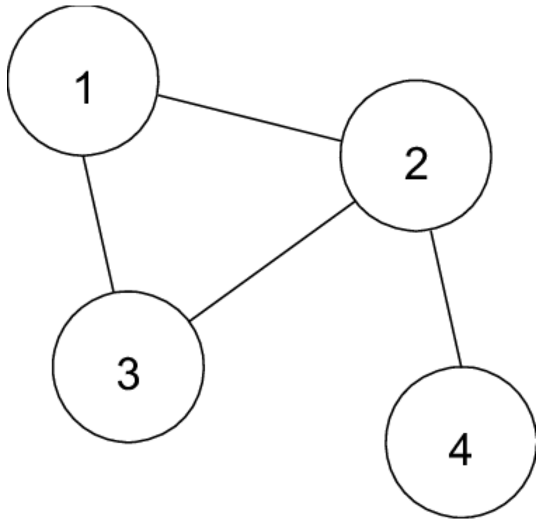


Exercise

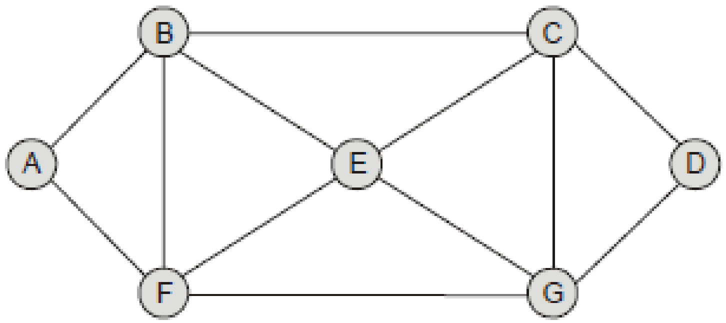
7. Consider a random network generated according to the G(N, p) model where the total number of nodes is 12 and the probability that there are links between any two nodes is 0.20. Determine the following:

    1. The probability that there are exactly 60 links in the network
    2. The average number of links in the network
    3. The average node degree
    4. The standard deviation of node degree in the network.

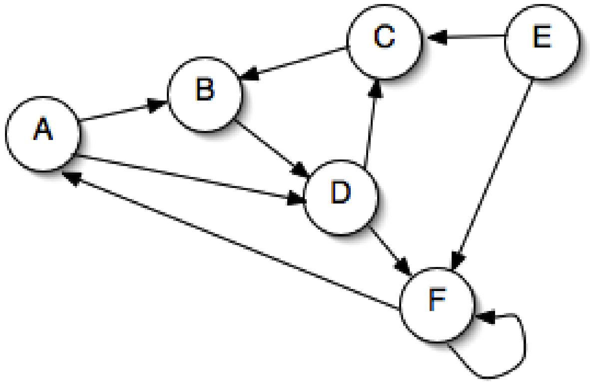
8. For the given undirected graph find out the clustering coefficient distribution.



9. Find out the adjacency matrix of the following network graphs:



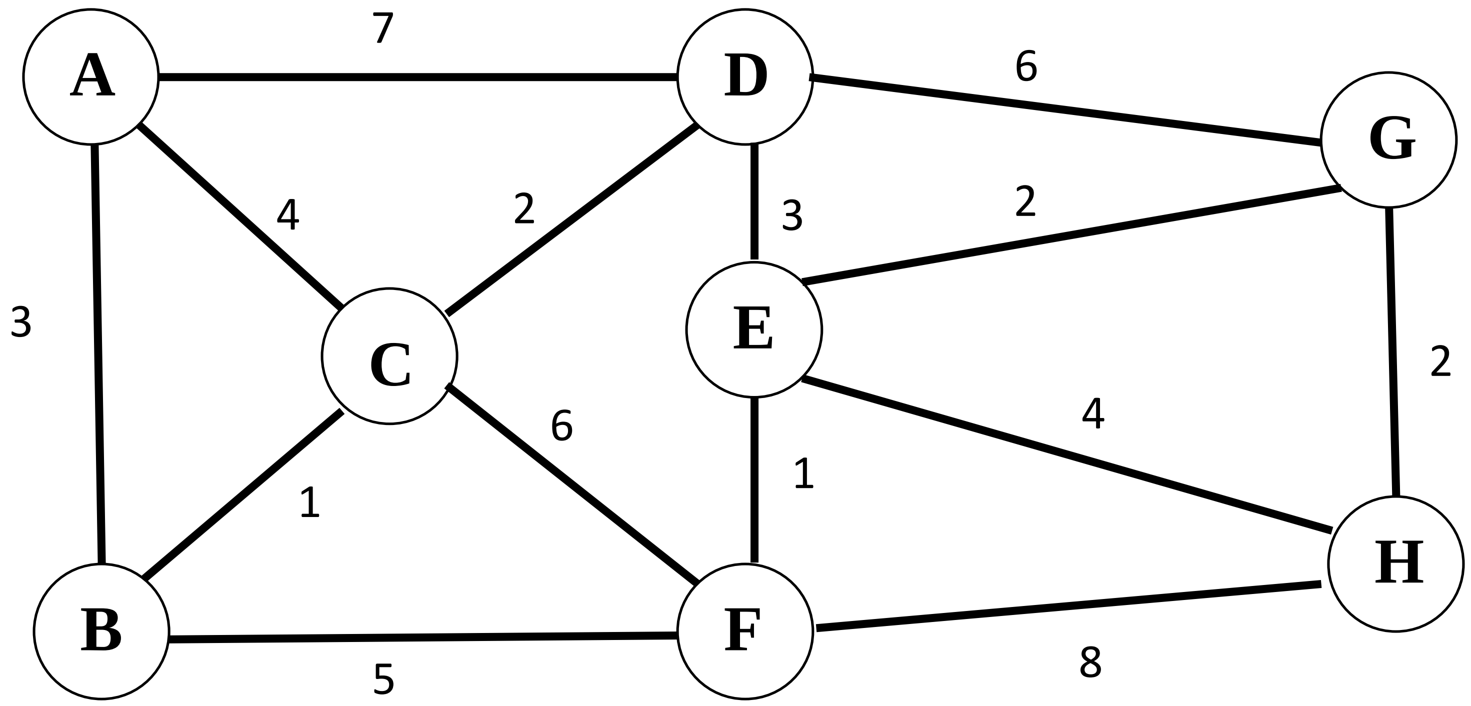
Undirected Graph



Directed Graph

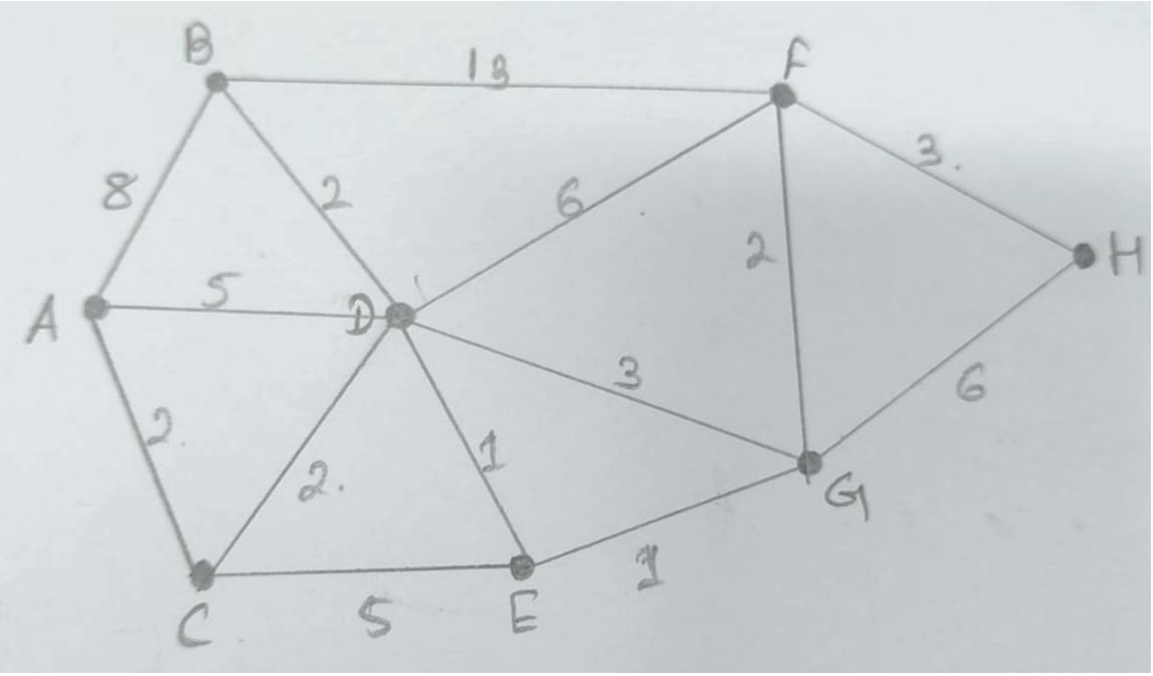
Exercise

10. Find the least cost path in the following graph.



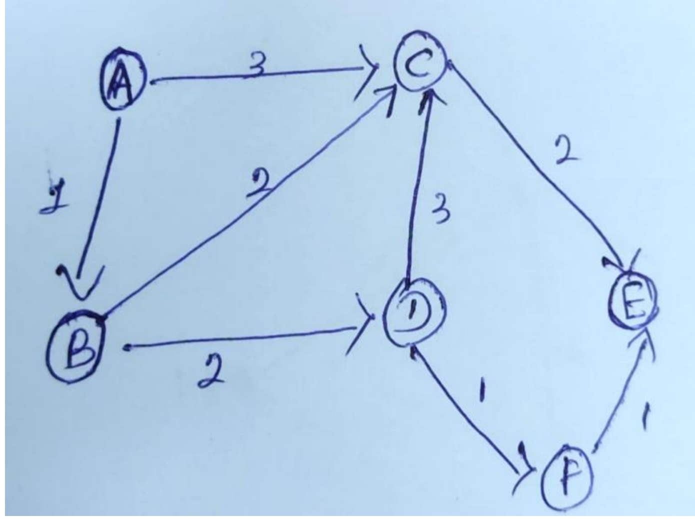
Exercise

11. Find the minimum weight path in the following graph.

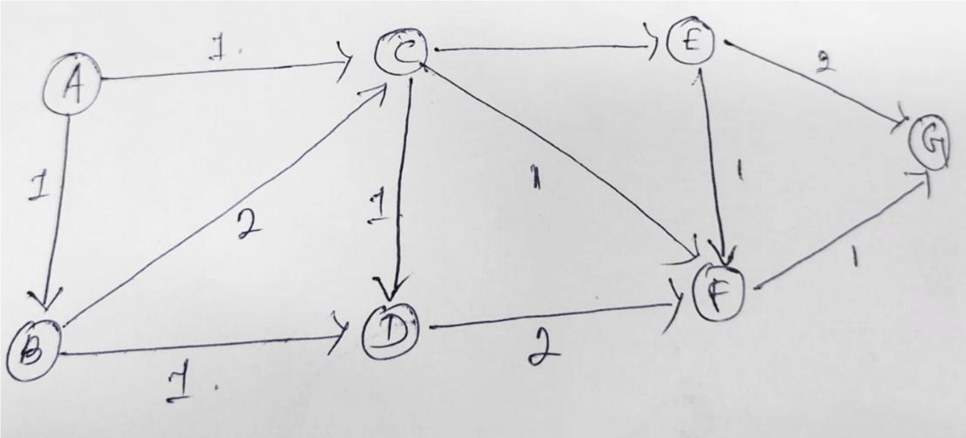


Exercise

12. Determine free float and total float for following:-



13. Determine (Early Start, Early Finish) and (Late Start, Late Finish) for the following graph:



Exercise

14. For the following activity construct the network graph.

Activity	1-2	1-3	2-4	3-5	4-9	4-5	5-6	5-7	6-8	7-8	8-9	6-9
Weight	3	2	2	1	1	2	3	2	1	2	3	2

15. For the following activity construct the network graph.

Activity	A-D	A-B	A-C	B-C	B-D	D-C	D-F	D-E	C-F	F-E	B-F
Weight	1	3	2	4	2	3	1	1	4	3	3

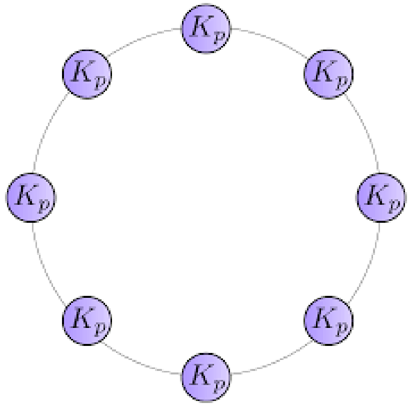
16. Write code in R to draw a directed network graph having vertices-A,B,C, and D.

Exercise

17. Create a star graph with 10 edges. Next, assign random numbers between 1 and 50 to vertices and select vertices for which this value is less than 20, set the color of these vertices to blue.

18. Create a complete graph with 8 edges. Next, assign random weights between 0 and 2 to all of edges. Set width of edges to 3 and color to purple for those of them that have weight less than 1, set the width value to 2, color to yellow for others.

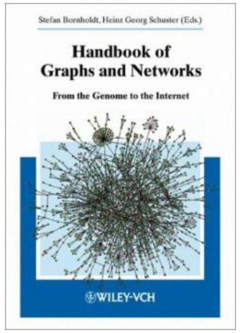
19. Find a closeness centrality of a node in the graph shown in the following figure.



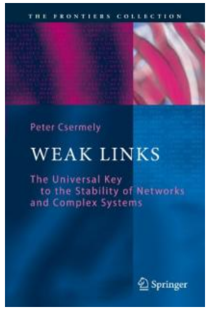
20. From the following networks given below, identify scale free and small world network

a) Author citation
b) Protein-protein
c) Friendship
d) Power grid

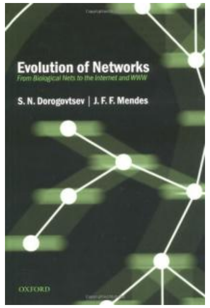
3.18 Books



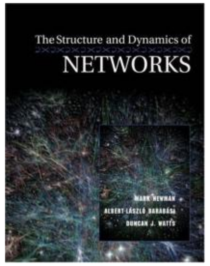
Handbook of Graphs and Networks: From the Genome to the Internet (Wiley-VCH, 2003).



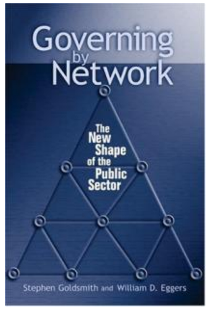
P. Csermely, Weak Links: The Universal Key to the Stability of Networks and Complex Systems (The Frontiers Collection) (Springer, 2006), rst edn.



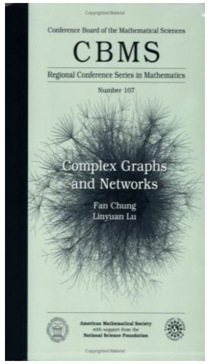
S. N. Dorogovtsev and J. F. F. Mendes, Evolution of Networks: From Biological Nets to the Internet and WWW (Oxford University Press, 2003).



M. Newman, A.-L. Barabasi, D. J. Watts, The Structure and Dynamics of Networks: (Princeton Studies in Complexity) (Princeton University Press, 2006), rst edn.

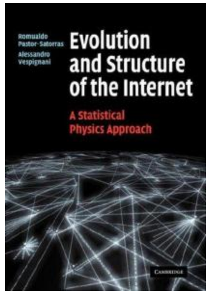


S. Goldsmith, W. D. Eggers, Governing by Network: The New Shape of the Public Sector (Brookings Institution Press, 2004).

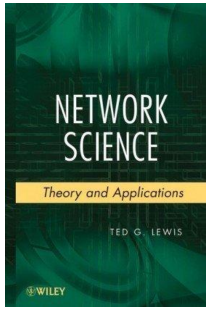


L. L. F. Chung, Complex Graphs and Networks (CBMS Regional Conference Series in Mathematics) (American Mathematical Society, 2006).

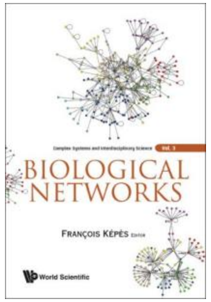
Books



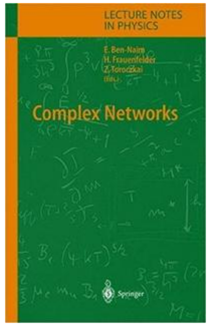
R. Pastor-Satorras, A. Vespignani, Evolution and Structure of the Internet: A Statistical Physics Approach (Cambridge University Press, 2007), rst edn.



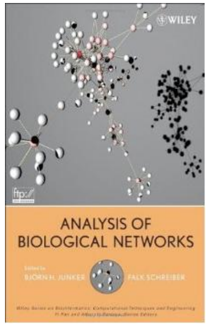
T. G. Lewis, Network Science: Theory and Applications (Wiley, 2009).



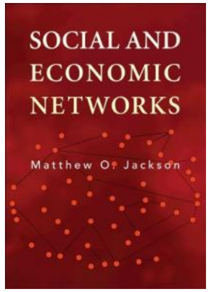
F. Kopos, Biological Networks (Complex Systems and Interdisciplinary Science) (World Scientic Publishing Company, 2007), rst edn.



E. Ben Naim, H. Frauenfelder, Z.Torotzai, Complex Networks (Lecture Notes in Physics) (Springer, 2010), rst edn.



B. H. Junker, F. Schreiber, Analysis of Biological Networks (Wiley Series in Bioinformatics) (Wiley-Interscience, 2008).



M. O. Jackson, Social and Economic Networks (Princeton University Press, 2010).

References

- https://slideplayer.com/slide/3809855/
- https://www.slideshare.net/NDSSL_VT/network-science-theory-modeling-and-applications
- https://www.barabasilab.com/
- Erdős, P.; Rényi, A. (1959). "On Random Graphs. I". Publications Mathematica. 6: 290–297.★★★★

  This book briefly explains probable structure of a random graph. All the mathematical expressions has been explained very clearly.

- Erdős, P.; Rényi, A. (1960). "On the evolution of random graphs". Magyar Tudományos Akadémia Matematikai Kutató Intézetének Kőzleményei [Publications of the Mathematical Institute of the Hungarian Academy of Sciences]. 5: 17–61. ★★★★

  This book is very systematically arranged. The evolution of Random Graphs has been explained through different theorems.

- Barab´asi, A.-L., Albert, R. and Jeong, H., http://xxx.lanl.gov/abs/cond-mat/990768 ★★★

  This paper presents a detailed information about the World Wide Web. Local connectivity measurements are used to construct topological model of World Wide Web.