

---

# COMP 102: Computers and Computing

## Lecture 12: Graphs

---

Instructor: Kaleem Siddiqi (siddiqi@cim.mcgill.ca)

Class web page: [www.cim.mcgill.ca/~siddiqi/102.html](http://www.cim.mcgill.ca/~siddiqi/102.html)

---

---

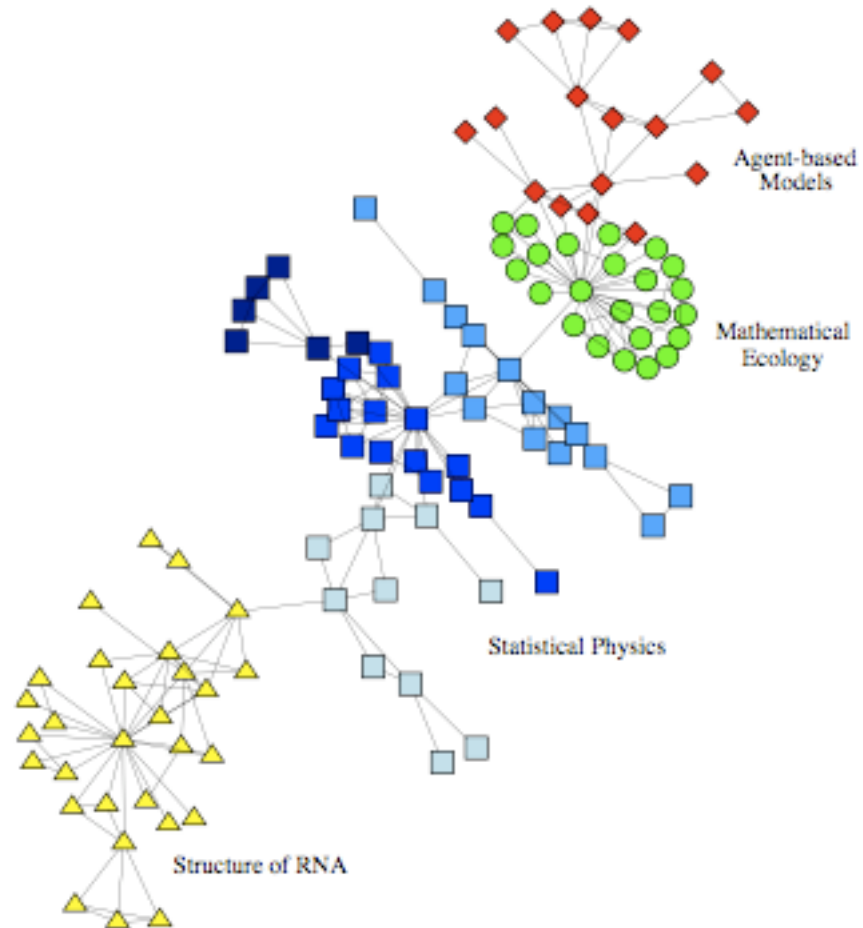
# Quick Review of Graphs

---

- A graph is a pair  $(N, E)$ , where
  - $N$  is a set of nodes.
  - $E$  is a collection of pairs of nodes, called edges.
- What is a path? What is a path length?
- What is an adjacency matrix?
- What is the difference between directed and undirected graphs?
- What is a cycle? What is a tree?

# Example: Scientific collaborations

- Nodes correspond to scientists in residence at the Santa Fe Institute in 1999-2000, and their collaborators.
- An edge is drawn between a pair of scientists if they coauthored one or more articles during this time period.
- The research topics are shown as different colours. These are identified automatically using a *clustering* algorithm.

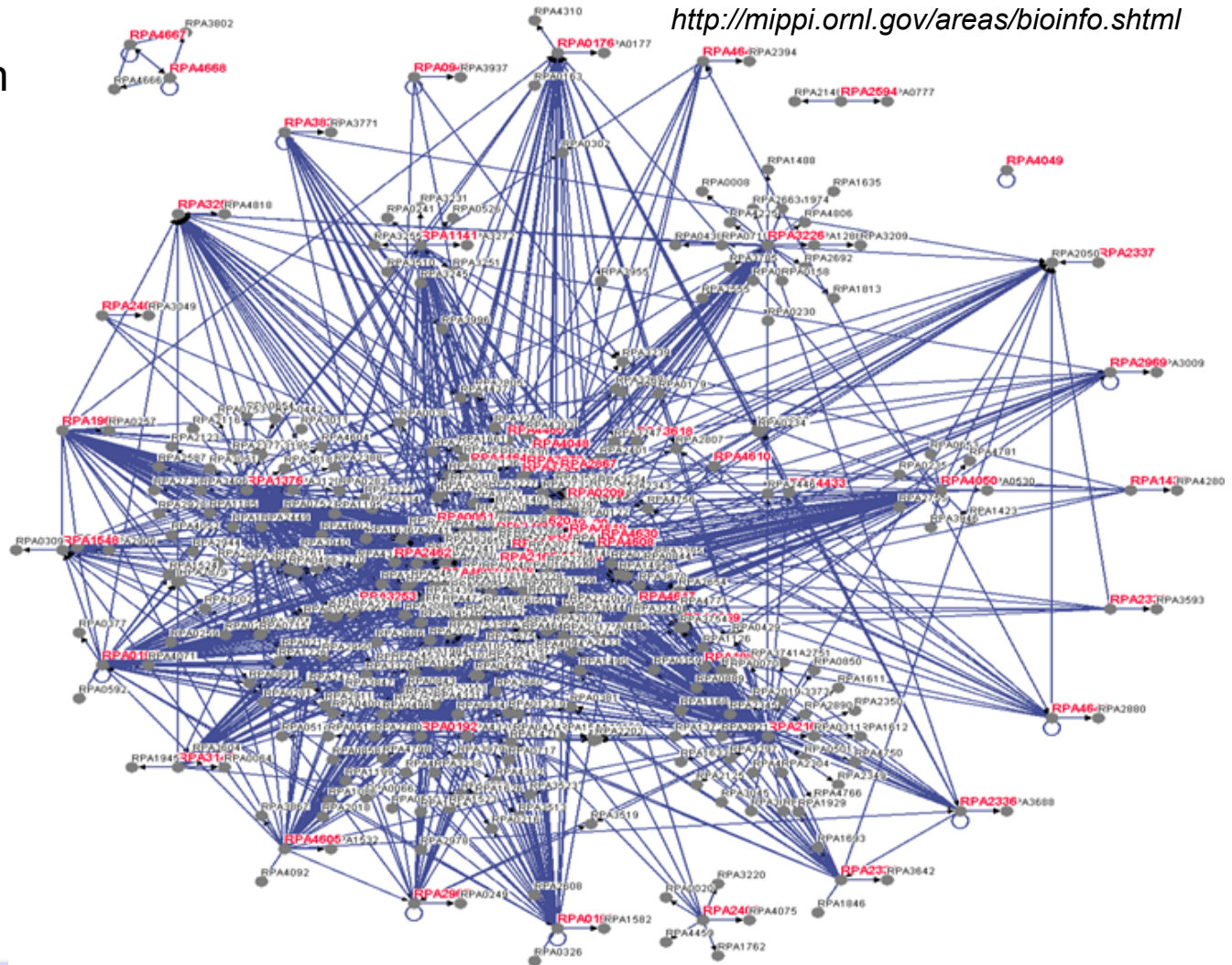


<http://arxiv.org/pdf/cond-mat/0112110v1>

## Example: Protein-protein interaction network

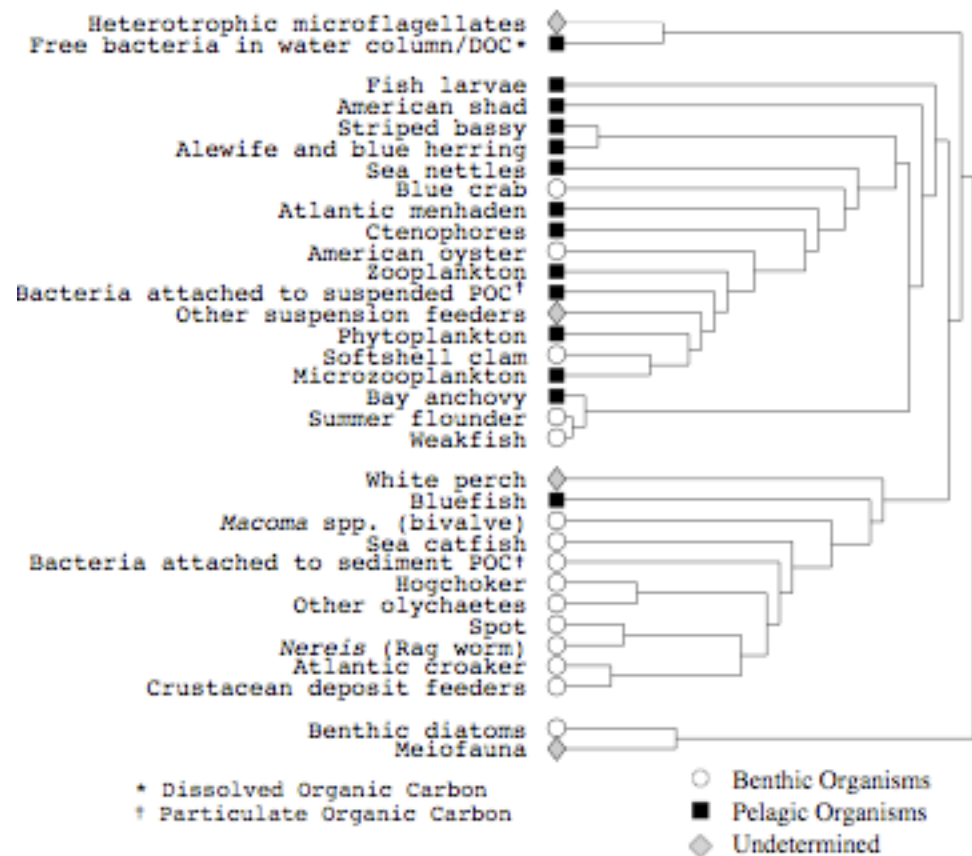
Interaction of protein molecules from the perspective of biochemistry and signal transduction.

E.g. *R. Palustris*  
protein-protein  
interaction  
network.



# Example: Food web

- Nodes correspond to the most prevalent marine organisms living in the Chesapeake Bay (USA).
- An edge is drawn between a pair if one of the organisms eats the other.
- Graph suggests there are two well-defined communities.
- These correspond quite closely to **pelagic** organisms (those that live near the surface) and **benthic** organisms (those that live near the bottom).



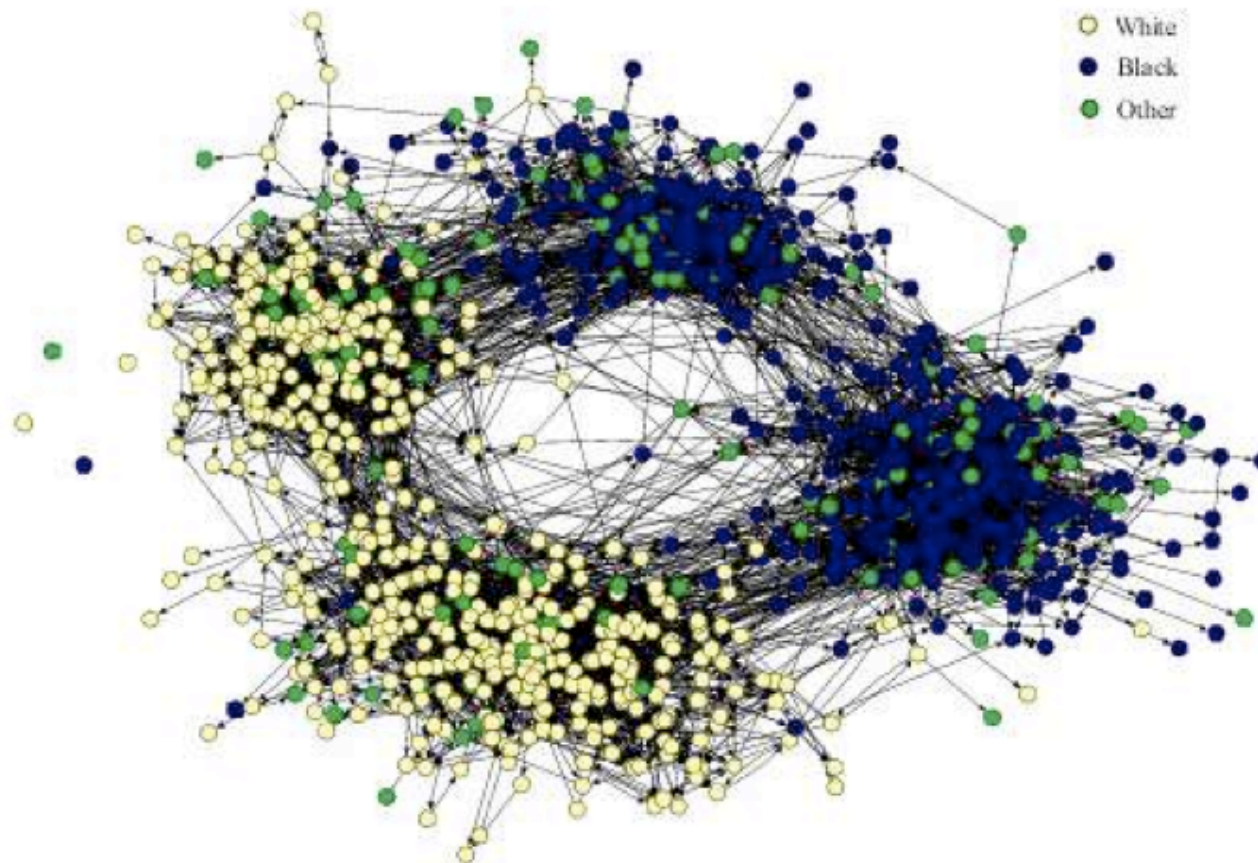
<http://arxiv.org/pdf/cond-mat/0112110v1>



---

## Example: A friendship network

---



[http://aps.arxiv.org/PS\\_cache/cond-mat/pdf/0303/0303516v1.pdf](http://aps.arxiv.org/PS_cache/cond-mat/pdf/0303/0303516v1.pdf)

FIG. 8 Friendship network of children in a US school. Friendships are determined by asking the participants, and hence are directed, since A may say that B is their friend but not *vice versa*. Vertices are color coded according to race, as marked, and the split from left to right in the figure is clearly primarily along lines of race. The split from top to bottom is between middle school and high school, i.e., between younger and older children. Picture courtesy of James Moody.

---

# Aside

---

- **Question:** How should we display a graph (nodes and edges) such that the information is interpretable for a human reader?
- This is the problem of graph **visualization**.
- This is a hard problem! Especially for graphs with many nodes and edges.
- Many people interested in finding good graph drawing algorithms to automatically generate an image of a graph, given its adjacency matrix.
- Nothing more on this problem in this course. But interesting techniques in graph theory and computer graphics for this.

---

# Back to work

---

So many questions we could ask of these graphs!

E.g. How do we find the shortest path between two given nodes?

Task: Find the shortest path between the U. de Montréal and McGill stations.

Time to think about  
**SEARCHING!**





---

# Searching over Graphs

---

- Your graph is defined by a set of nodes and an adjacency matrix.
- You also need to know the start node and the end node.
- The goal is to explore all possible paths and return the shortest one.

Warning! Need to be **systematic** about the order in which you explore these paths.

---

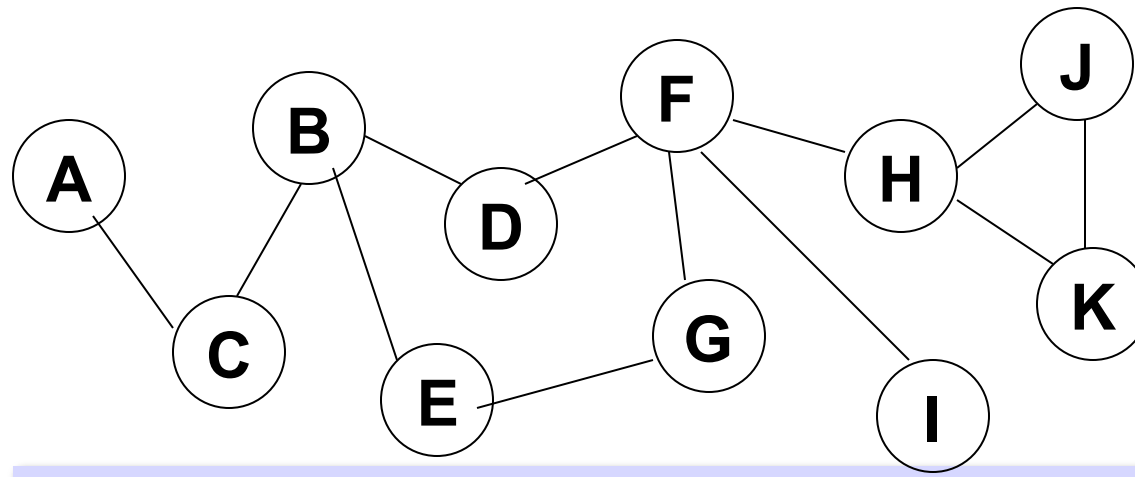
# Breadth-first search

---

- Start at some node n. Say we start with F.
- Explore all the neighbors of n. Explore D, G, I and H.
- Then explore all the unvisited neighbours of the neighbours of n. B, E, K, J
- Then visit unvisited neighbours of those. C
- Continue until no more unvisited nodes remain. A

Visitation order: F, D, G, I, H, B, E, K, J, C, A

Visitation path: F - D - F - G - F - I - F - H - F - D - B - D - F - G  
- E - G - F - H - K - H - J - H - F - D - B - C - A



---

# Comments on breadth-first search

---

- Breadth-first search explores the graph **layer by layer**.
  - E.g. For web-browsing, all n-away links are explored.
  - Need to decide before-hand on the order of neighbours (e.g. clockwise)
  - Need to keep track of nodes you've already explored.
- **Pro:** Good algorithm if you want to find the **shortest path** between the start node, and another node. (As soon as you find that node, you know you have found the shortest path to it.)
- **Con:** Often requires a lot of **backtracking** (= visitation path goes through visited nodes again and again.)

**Can we avoid all this backtracking?**

---

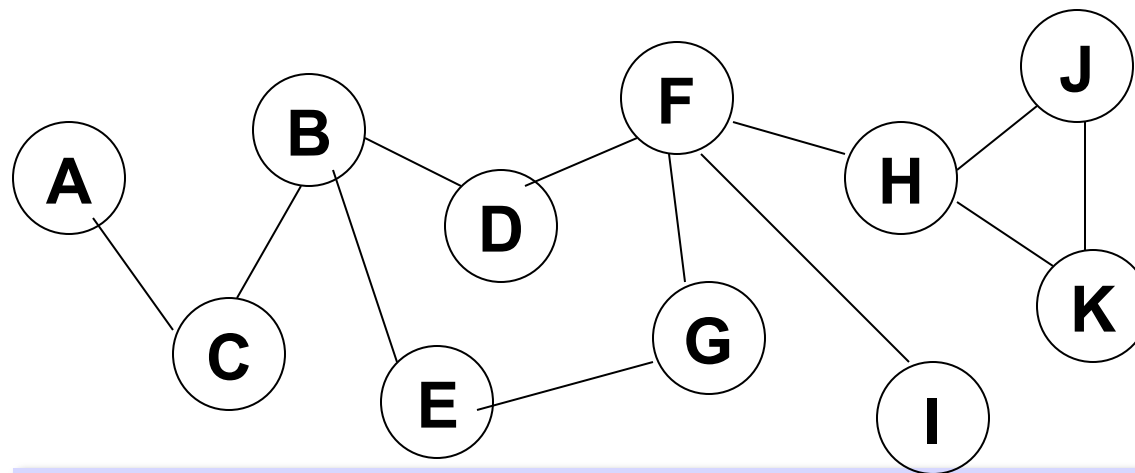
# Depth-first search

---

- Start at some node n. Say we start with F.
- Then explore the first unvisited neighbour of n (call this n'). Explore D.
- Then explore the first unvisited neighbour n', and so on until you hit a node with no unexplored neighbours. B, C, A
- Then backtrack 1 level to explore the next unvisited neighbour. E, G, etc.

Visitation order: F, D, B, C, A, E, G, I, H, K, J

Visitation path: F - D - B - C - A - C - B - E - G - E - B - D - F  
- I - F - H - K - J



---

# Comments on depth-first search

---

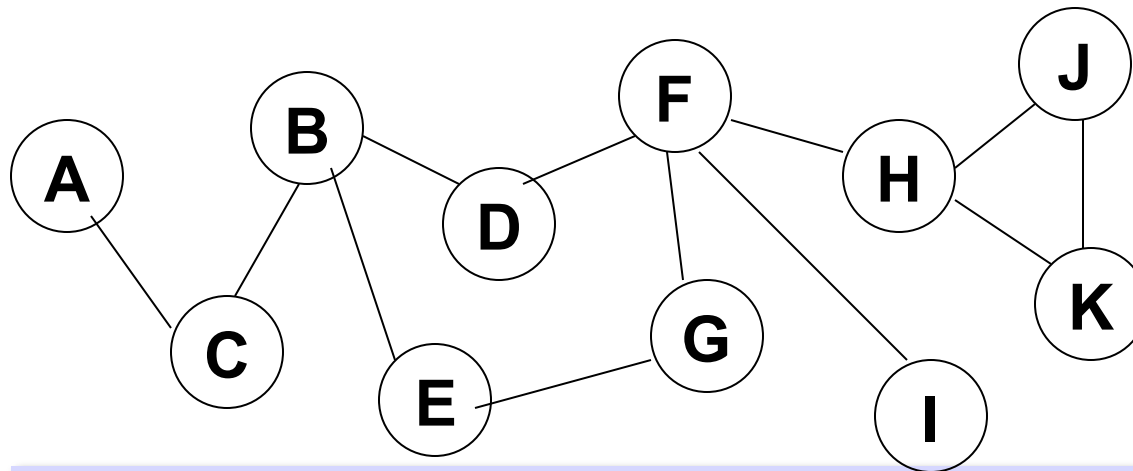
- Depth-first search explores graph by going **deeper whenever possible**.
  - E.g. For web-browsing, always click on 1<sup>st</sup> link until you hit a dead-end.
  - Need to decide before-hand on the order of neighbours (e.g. clockwise)
  - Need to keep track of nodes you've already explored.
- **Pro:** Usually uses much **less backtracking** to explore the full graph than breadth-first search. How much less depends on neighbourhood ordering (sometimes lucky, sometimes not)
- **Con:** Not guaranteed to find the shortest path, unless you explore the full graph.

---

# Quick review of best-first search

---

- Start at some node n. Say we start with F.
  - Pick a score function. Say score = alphabetical order.
  - Add its neighbours to the list of candidate nodes. Add D(=4), G(=7), I(=9), H(=8).
  - Pick candidate node with highest score. Pick D.
  - Add its neighbours to the list of candidate nodes. Add B(=2).
  - Continue until no more unexplored nodes. Pick B, Add C(=3) and E(=5), etc.
- Exploration order: F, D, B, C, A, E, G, H, I, J, K  
Candidate list: D, G, I, H, B, C, E, A, K, J





---

# Comments on best-first search

---

- Best-first search explores graph by according to **priority order**.
  - E.g. For web-browsing, always explore link with highest PageRank.
  - Need to have a score function, which can be calculated for each node.
  - Need to keep track of candidate nodes.
- **Pro:** Usually much faster to reach a goal node (e.g. let's say we stop when we reach "A".)
- **Con:** Not an advantage if you want to explore the full graph.

---

# Graph Topologies

---

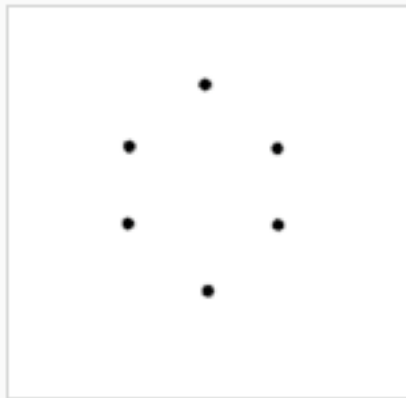
- **Topology** = The arrangement in which the nodes of a graph are connected to each other.
- Common types of graphs:
  - **Regular graph**
  - **Complete graph**
  - **Random graph**

---

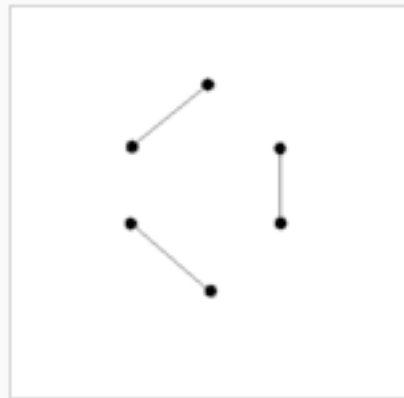
# Regular graph

---

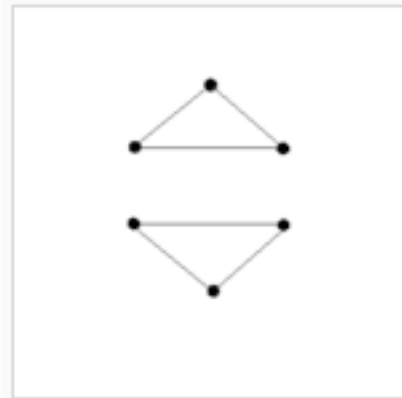
- Main characteristic: Each node has same number of neighbours.



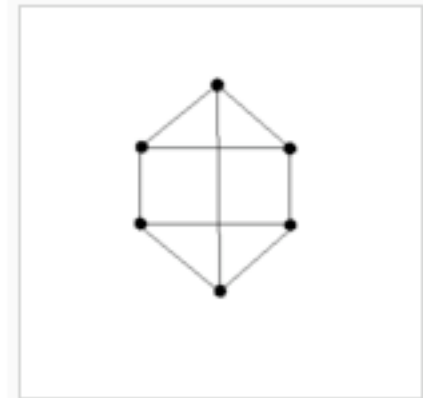
0-regular graph



1-regular graph



2-regular graph



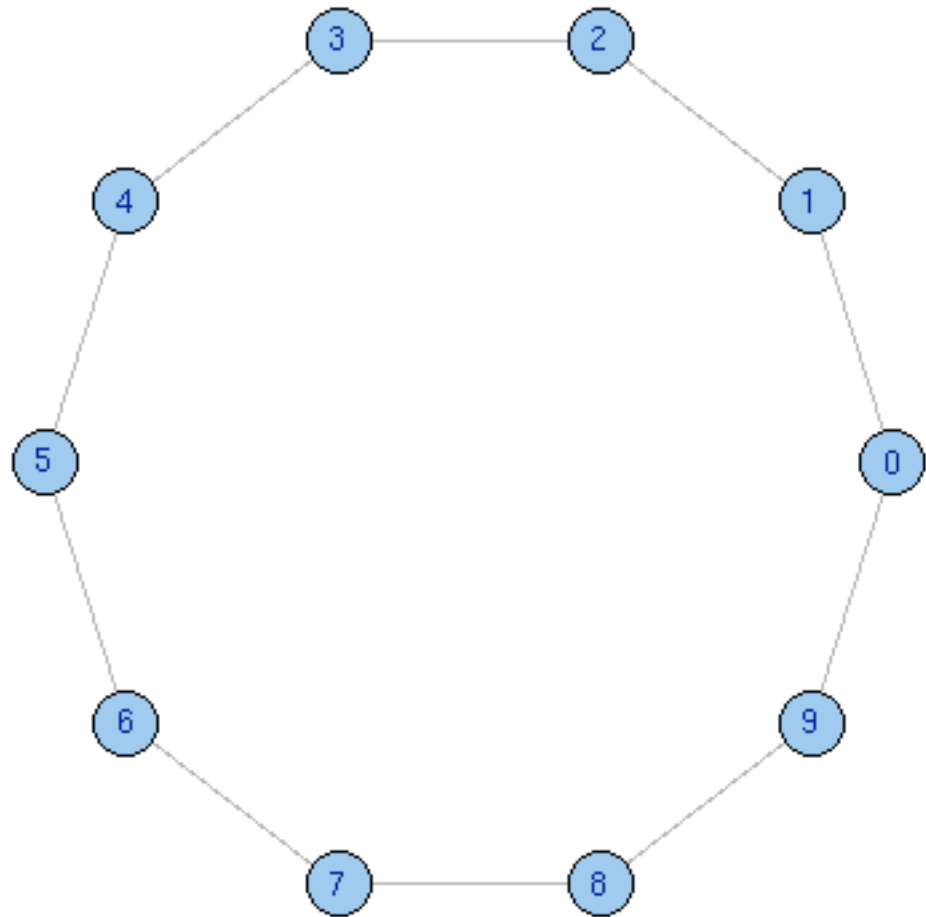
3-regular graph

[http://en.wikipedia.org/wiki/Regular\\_graph](http://en.wikipedia.org/wiki/Regular_graph)

---

# Special regular graph: the Ring

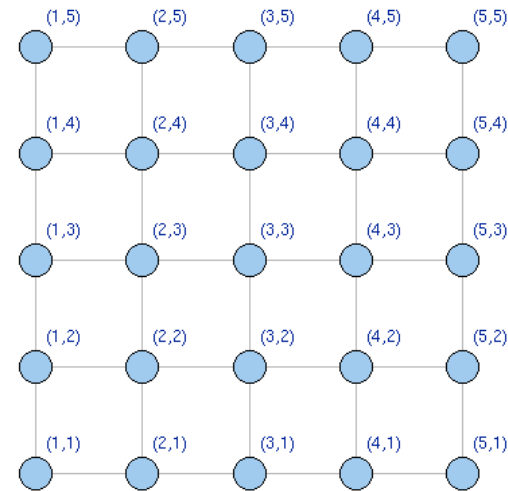
---



<http://geza.kzoo.edu/~csardi/module/html/>

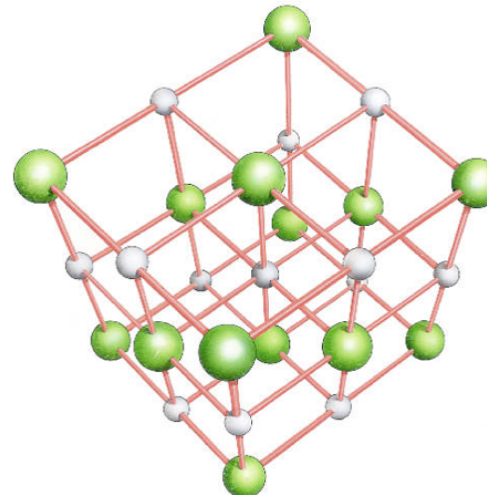
# Special regular graph: the Lattice

- This is a common topology to model road networks (in 2-D).



<http://geza.kzoo.edu/~csardi/module/html/>

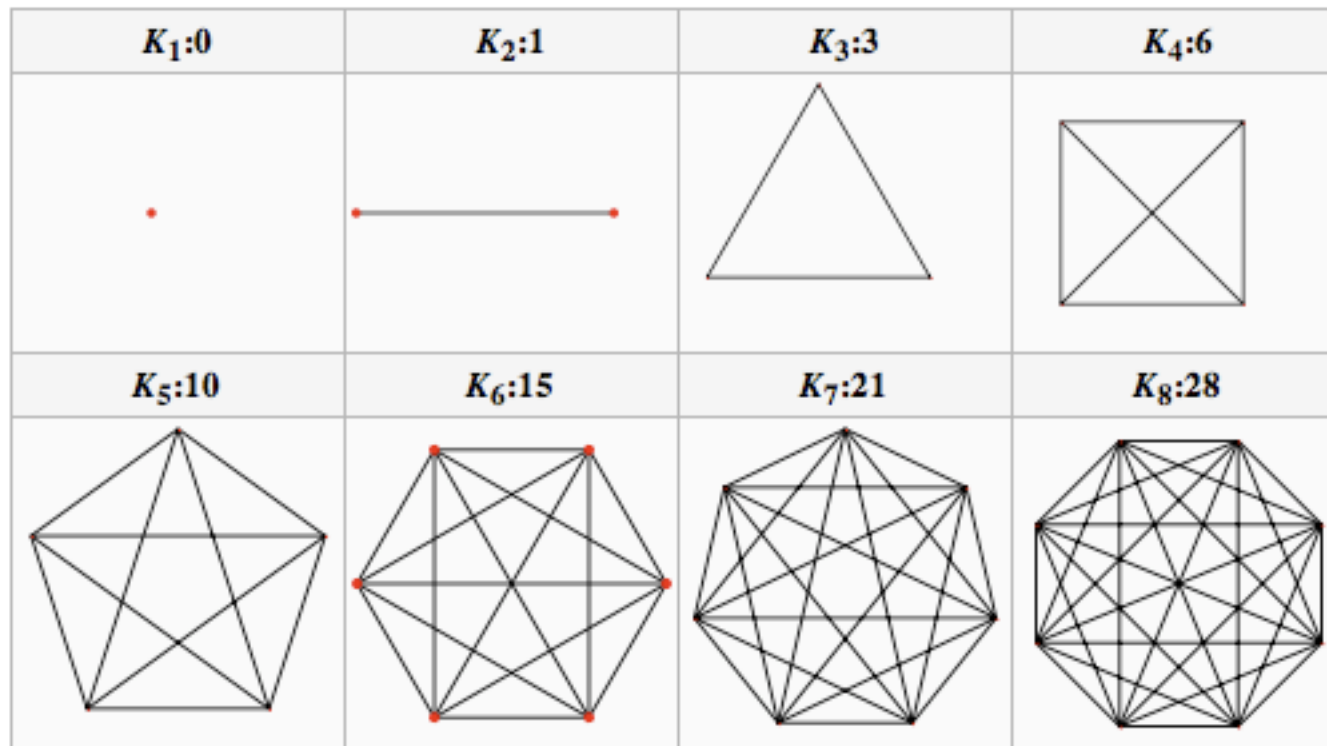
- Also common for molecular diagrams (in 3-D).



<http://www.dkimages.com>

# Complete graph (also a regular graph)

- Main characteristic: Each pair of nodes is connected by an edge.



[http://en.wikipedia.org/wiki/Complete\\_graph](http://en.wikipedia.org/wiki/Complete_graph)

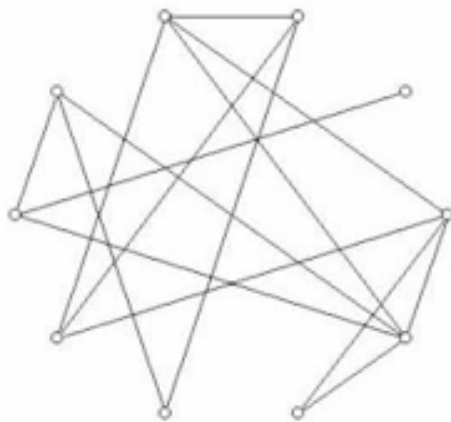


---

# Random graph

---

- Basic construction: Start with a set of nodes, randomly add edges with probability  $p$  between each pair of nodes.
- Graph is denoted  $G(n,p)$ , where  $n$  is the number of nodes and  $p$  is the probability of a pairwise connection.



<http://epress.anu.edu.au/cs/html/ch05s03.html>

[http://aps.arxiv.org/PS\\_cache/cond-mat/pdf/0007/0007235v2.pdf](http://aps.arxiv.org/PS_cache/cond-mat/pdf/0007/0007235v2.pdf)

---

# Graphs in the real world

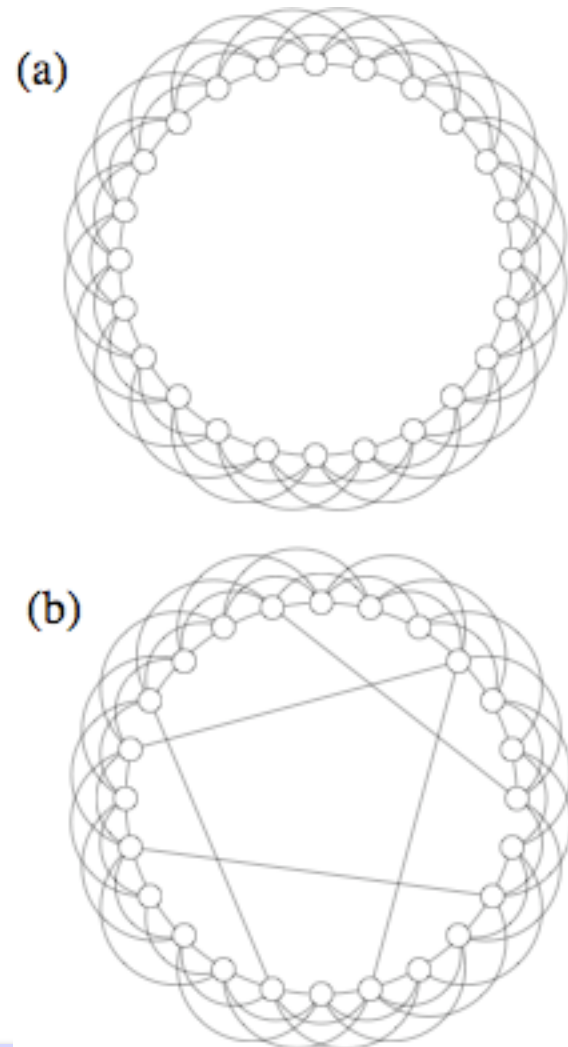
---

- Think back to our examples:
  - Montreal metro system.
  - Roadmap of a city.
  - The internet.
  - A maze.
- Most biological, technological and social graphs are not exactly regular, complete or random.
- Graphs are sometimes also called **networks**.
- Next: explore a special class of graphs.

# Small-world networks

- A small-world network is a mix of a regular graph and a random graph.
- Simple construction:
  - Start with a ring made of  $n$  nodes and  $k$  edges per node.
  - Wire the  $k$  edges as for a regular graph.
  - With probability  $p$ , re-wire each edge to another random node.

[http://aps.arxiv.org/PS\\_cache/cond-mat/pdf/0303/0303516v1.pdf](http://aps.arxiv.org/PS_cache/cond-mat/pdf/0303/0303516v1.pdf)



---

# Characteristics of a small-world network

---

- Key parameters:
  - $n$  controls the size of the graph (= number of nodes)
  - $k$  controls the degree of connectedness (e.g. if  $k=n$  then we have a complete graph.
  - $p$  controls the trade-off between “regular” ( $p=0$ ) and “random” ( $p=1$ )
- Measures of the graph:
  - Characteristic path length
  - Clustering coefficient

---

# Characteristic path length

---

- Characteristic path length measures the typical separation between any 2 nodes.
- This is a global property.
- $L(p)$  = # edges in the shortest path between 2 nodes, averaged over all pairs of nodes.

---

# Clustering coefficient

---

- Clustering coefficient measures the “cliquiness” of a typical neighbourhood.
- This is a local property.
- To calculate:
  - Suppose node  $n$  has  $k_n$  neighbours, then at most  $k_n * (k_n - 1) / 2$  edges can exist between them.
  - Let  $C_n(p)$  be the fraction of these allowable edges that actually exist.
  - Then the clustering coefficient is:  $C(p) = \sum_n C_n(p) / n$



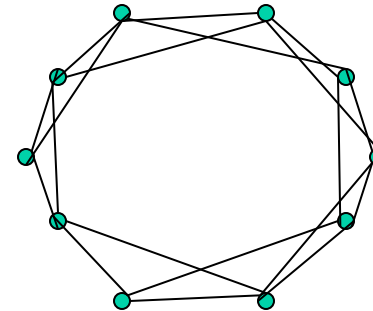
---

# Intuition

---

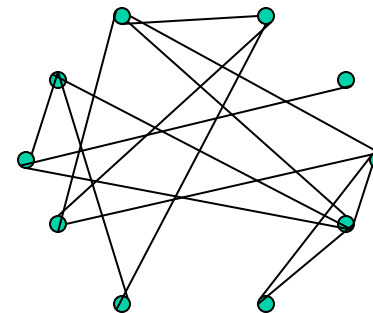
- At  $p=0$ : graph is a regular lattice, highly clustered, “large” world.

$L(p)$  grows linearly with  $n$ .



- At  $p=1$ : graph is a random graph, poorly clustered, “small” world.

$L(p)$  grows logarithmically with  $n$ .

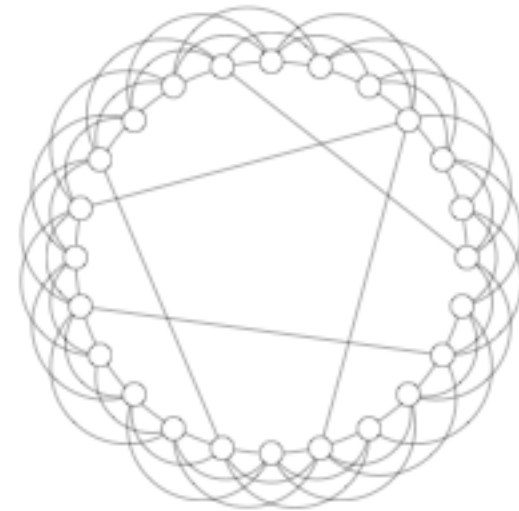


---

## Question

---

- Is it always the case that a large characteristic path length ( $L$ ) implies a large clustering coefficient ( $C$ )?
- **No!** There are many values of  $p$  for which  $C(p)$  is large and  $L(p)$  is small.
- These are the 2 required properties of a small-world network.
  - A few long-distance connections are enough to reduce  $L(p)$ .

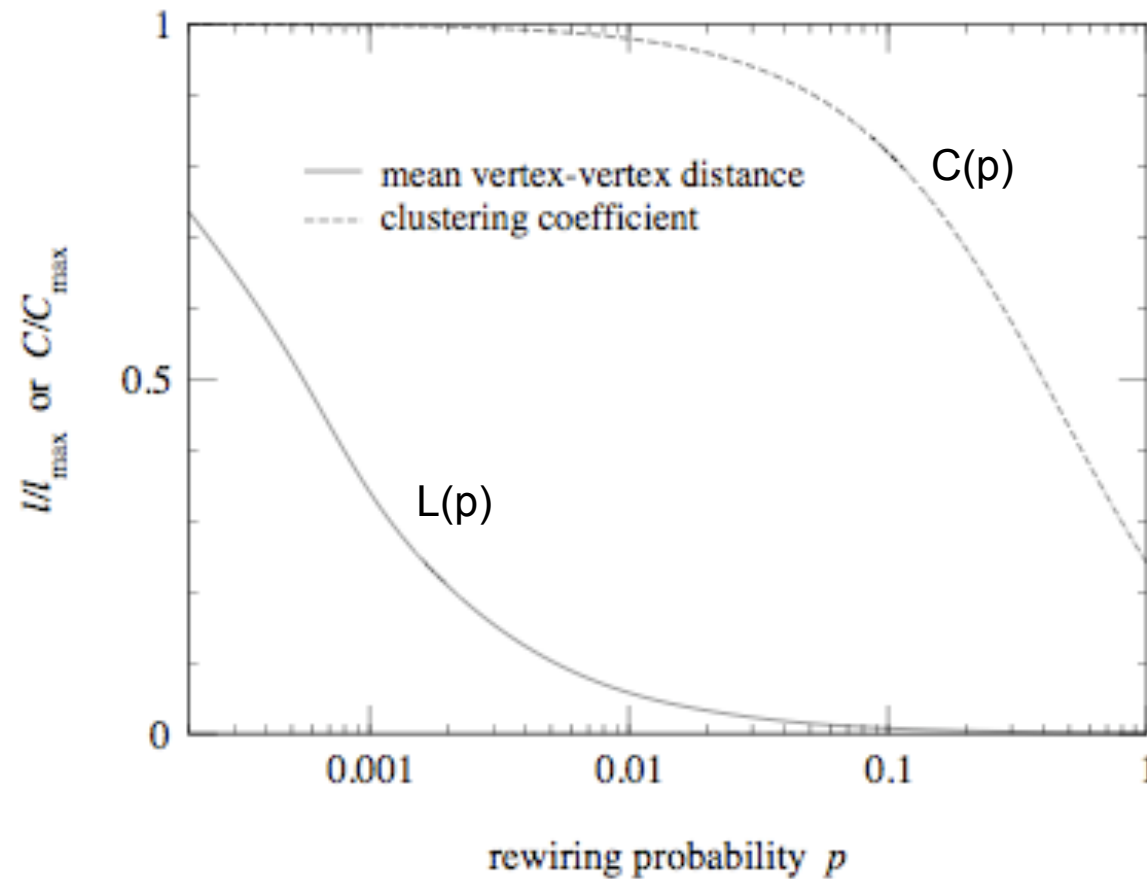


**Why do we care? This correctly models real-life networks!**

---

How do  $C(p)$  and  $L(p)$  change as a function of  $p$ ?

---



# Example: Internet Movie Database (IMDB)



The Internet Movie Database

[NOW PLAYING](#) [MOVIE / TV NEWS](#) [MY MOVIES](#) [DVD & BLU-RAY™](#) [IMDb TV](#) [MESSAGE BOARDS](#) [SHOWTIMES & TICKETS](#) [IMDbPro](#) [IMDb Resume](#)

[Home](#) [Top Movies](#) [Photos](#) [Independent Film](#) [GameBase](#) [Browse](#) [Help](#)

search   [more](#) [tips](#)

### Tops at the Box Office

1. [Beverly Hills Chihuahua](#)
2. [Eagle Eye](#)
3. [Nick and Norah's Infinite Playlist](#)
4. [Nights in Rodanthe](#)
5. [Appaloosa](#)

[more](#)

### Opening This Week

- [Quarantine](#)
- [Body of Lies](#)
- [RocknRolla](#)
- [City of Ember](#)
- [Happy-Go-Lucky](#)

[more](#)

### Coming Soon

- [W.](#)
- [Sex Drive](#)
- [Morning Light](#)
- [Max Payne](#)
- [The Secret Life of Bees](#)

[more](#)

### New DVDs This Week

- [The Visitor](#) - [DVD](#)
- ["30 Rock": Season 2](#) - [DVD](#)
- [Sleeping Beauty](#) - [DVD](#)
- [The Happening](#) - [DVD](#)
- ["The Simpsons": Season 11](#) - [DVD](#)

### New on Blu-ray HD-DEF

## The Internet Movie Database

Visited by over 57 million movie and TV lovers each month!

Welcome to the Internet Movie Database, the biggest, best, most award-winning movie and TV site on the planet. Want to make IMDb your home page? Drag [this link](#) onto your  Home button.

### Today's IMDb Poll Question Is:

 Have you ever gone to a movie theater in a different country than your own? (Suggested by "PinilakangTabing") ([vote](#))

### IMDb Snapshot: [New Photos](#) and [More](#)




Movie and TV Stills: [Confessions of a Shopaholic](#), ["My Own Worst Enemy"](#), [Quarantine](#)

Celebrity and Event Photos: [Nick & Norah's Infinite Playlist Premiere](#), [You Vote Campaign Shoot](#), [High School Musical 3 - Paris Premiere](#)

Top Trivia: [The Dark Knight](#), [Burn After Reading](#), [Tropic Thunder](#) ([more trivia ...](#))

More Tops: [Top Quotes](#) - [Top Goofs](#)



### Truly Trivial

She made her screen debut in [Once Upon a Time in America](#). ([answer](#))

[Click for more trivia!](#)



### Movie/TV Quote of the Day

### Movie and TV News

**Mon 6 October 2008**

[WENN](#)

- [Mamma Mia! Keeps Singing Atop Global Box Office Chart](#)
- [Keener 'Dates Toyboy Sports Star'](#)
- [Ambrosio Reunited With Lost Pooch](#)

[Studio Briefing](#)

- [Ay Chihuahua!](#)
- [Movie Reviews: Appaloosa](#)
- [Movie Reviews: An American Carol](#)

### Born Today

Monday, 6 October 2008:



[Elisabeth Shue \(45\)](#)

COMP-102: Computers and Computing

30

(thanks to Joelle Pineau!)

---

# IMDB as a graph

---

- Consider a graph restricted to the connected components of IMDB (roughly 90% of the database).
  - Nodes: actors ( $n = 225,226$ )
  - Edges: 2 actors are joined by an edge if they have acted in a film together.
  - Characteristics of this graph:  $L_{\text{actual}} = 3.65$   $C_{\text{actual}} = 0.79$
  - Compare to a random graph (same size):  $L_{\text{random}} = 2.99$   $C_{\text{random}} = 0.00027$
- We observe that:
  - L is the same as a random graph : short path between any pair of actors.
  - C is much larger than in a random graph : high level of “cliquiness”.

**All this is interesting! But not very surprising. Also not very useful.**

---

## Example: Model the spread of an infectious disease

---

Consider a population of  $n$  individuals, connected as a small-world networks.

### Basic model:

- On day 1: a single individual is infected.
- On day 2 and subsequent days: we see the effect of that infection
  - Each infected individual infects each of its neighbours with probability  $r$ .
  - Each infected individual is then removed (immunity or death).
- Termination:
  - At some time, the network reaches a stable state.
  - Either everyone is infected, or the disease dies out after having infected some.

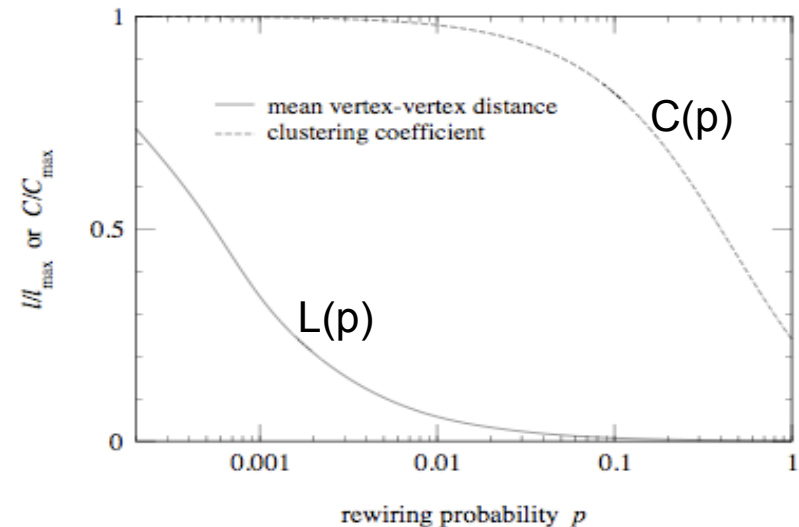


---

# Analysis of the model

---

- Now we can ask lots of interesting questions!
- For what values of  $r$  and  $p$  does everyone get infected?
  - Recall  $r$  = infection rate and  $p$  = percentage of long-distance edges.
- Results:
  - The critical infection rate ( $r_{half}$ ) at which the disease infects half of  $n$  decreases rapidly for small  $p$ .
  - For a disease with  $r$  high enough to infect everyone, then the time to global infection follows the  $L(p)$  curve.



**How do we get these results?**

---

# Simulating a graph

---

- Need to simulate our graph, to capture the change of state in the infected population.
- What do you remember about finite-state machines?
  - States + Transition graph. Use this here!
- Pick values for  $n$ ,  $p$  and  $r$ .
- The state is described by a separate variable,  $n_i = \{healthy, infected, dead\}$  for each node.
- The transition graph expresses the effect of the infection.

---

# Simulation of our model of infectious disease

---

---

## Graph-based simulation (beyond small-world networks)

---

- What is the impact of the graph topology on the spread of a disease?
- What is the impact of a specific intervention strategy (e.g. through setting  $r$ ) on the spread of the disease?

---

# Take-home message

---

- Main searching algorithms for graphs: Breadth-first search, Depth-first search, Best-first search.
  - Know the steps of each algorithm, and the pros/cons for each.
- Characteristics of the basic types of graphs (regular, complete, random).
- Definition and characteristics of small-world networks.
- Understand how graphs and finite-state machines can be combined to simulate real-world phenomena.