CITS4403 Computational Modelling

Lecture 2: Graph I

Dr. Siwen Luo

Semester 2, 2024
School of Computer Science,
University of Western Australia

LECTURE PLAN

Lecture 2: Graph I

1. **Recap**
2. Different types of Graphs
   - Implementation of graph generation
3. Erdos Renyi Graph
4. Connectivity of the ER graph
5. Random Graph vs. Regular Graph

Recap

Computational Model and Simulation

A **computational model** contains numerous variables that characterize the system being studied.

Simulation is done by adjusting the variables alone or in combination and observing the outcomes.

Build a model or Build a simulator

Run the model or run a simulation

We run Simulation many times in actual experiments

Recap

Complex System

A **complex system** are composed of many **components** (or **systems**) which may interact with each other.

E.g. Society, Organ system, Ant colony, Global economy, transportation system etc.

Key Complex System Features

- Interactions
- Emergence
- Dynamic
- Self-organization
- Adaptation
- Interdisciplinary

LECTURE PLAN

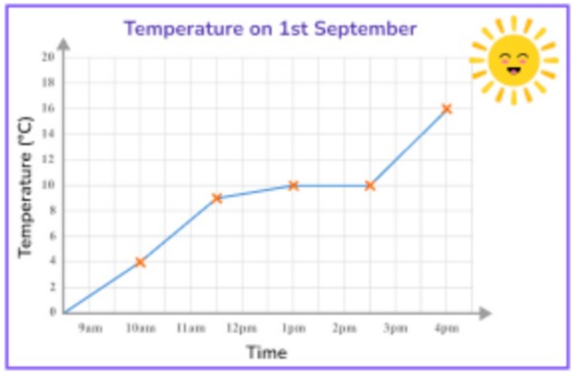
Lecture 2: Graph I

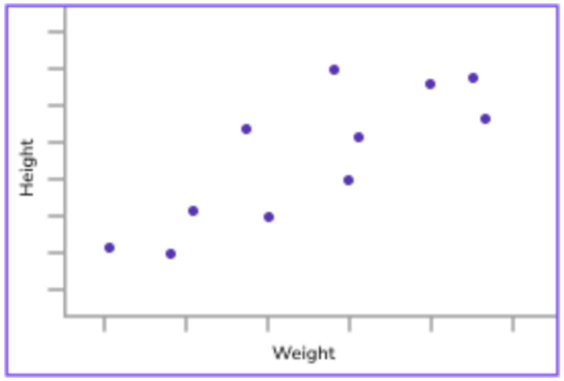
1. Recap
2. **Different types of Graphs**
   - Implementation of graph generation
3. Erdos Renyi Graph
4. Connectivity of the ER graph
5. Random Graph vs. Regular Graph

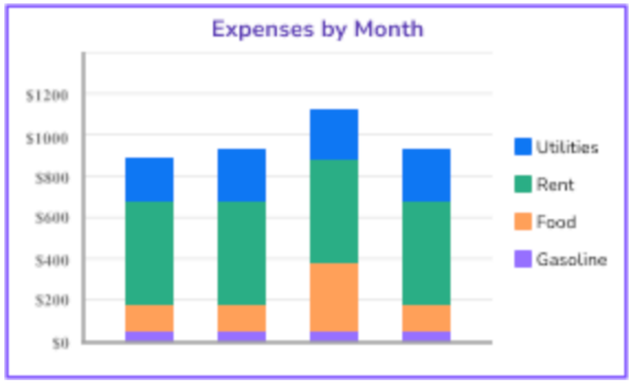
2 Graph

THE UNIVERSITY OF WESTERN AUSTRALIA

Normally, we know 'graph' as…







Graph

What is a graph?

Graph is a representation of a system that contains discrete, interconnected elements.

Formally:

$$G = (V, E)$$

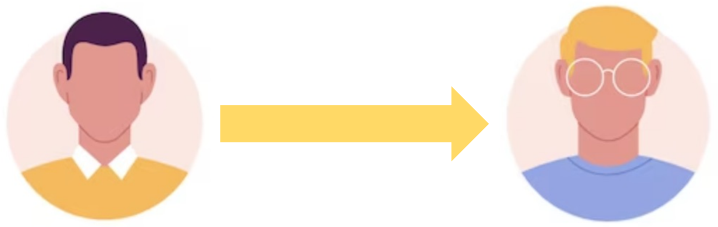
V: vertices or nodes, to *represent the elements in the system*
E*: edges or links, to represent the interconnections between elements*

Graph

Taking a social network as an example:

Node is the person in the social networks, and edge is connected between related person.

Directed Graph



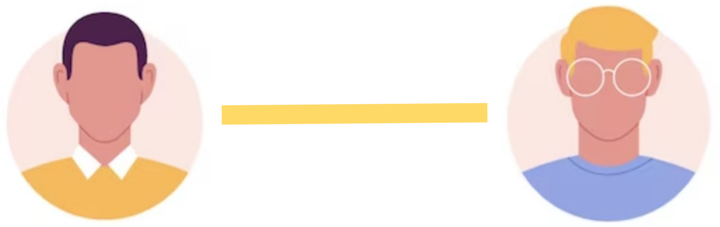
Asymmetric relationship

2 Graph

Taking a social network as an example:

Node is the person in the social networks, and edge is connected between related person.

Undirected Graph



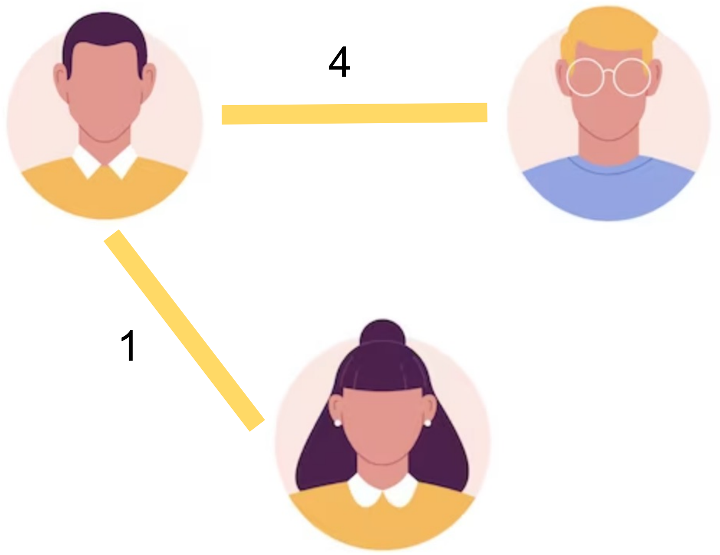
Symmetric relationship

Graph

Taking a social network as an example:

Node is the person in the social networks, and edge is connected between related person.

Weighted Graph



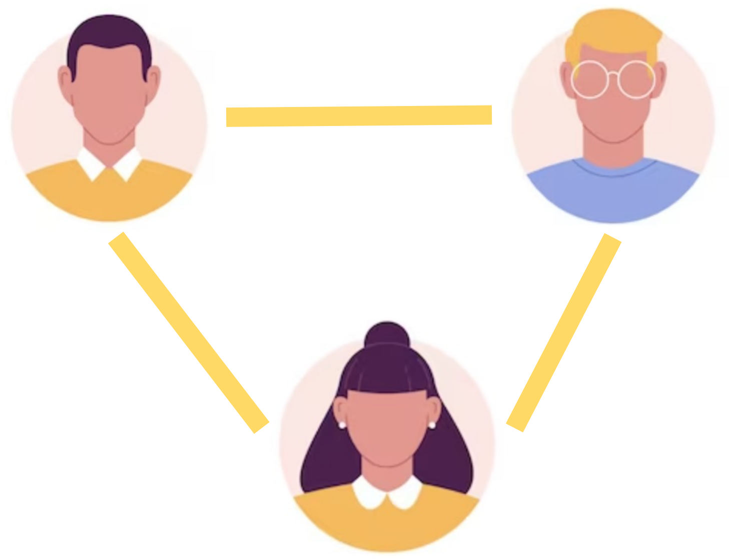
Edge can have weights.

Graph

Taking a social network as an example:

Node is the person in the social networks, and edge is connected between related person.

Complete Graph



Everyone is the friend list is the friend of each other.

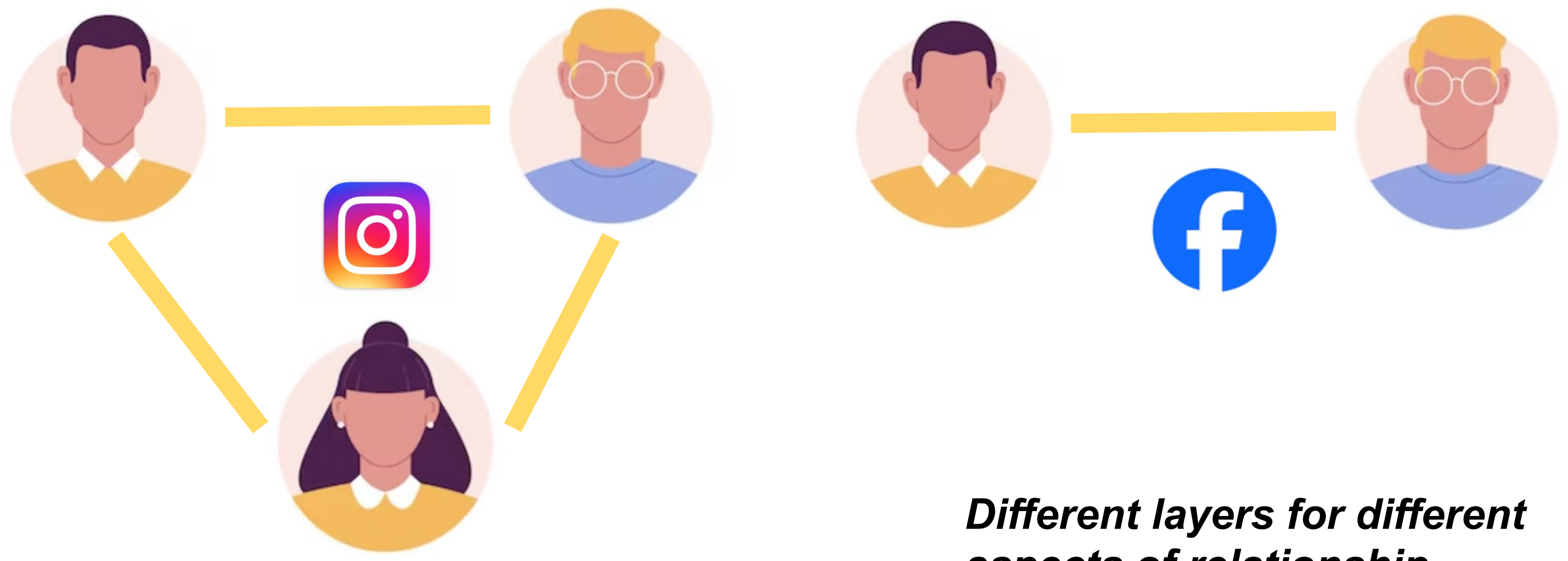
Graph

THE UNIVERSITY OF WESTERN AUSTRALIA

Taking a social network as an example:

Node is the person in the social networks, and edge is connected between related person.

Multi-Layered Graph



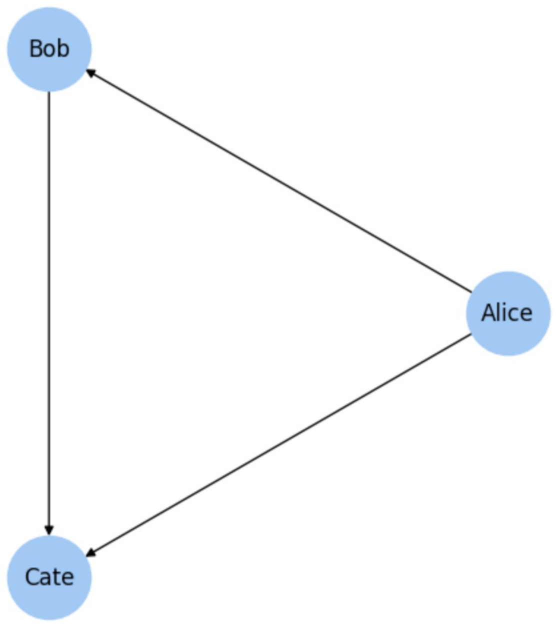
Different layers for different aspects of relationship.

Graph

Create different graphs with Python and NetworkX

Directed Graph

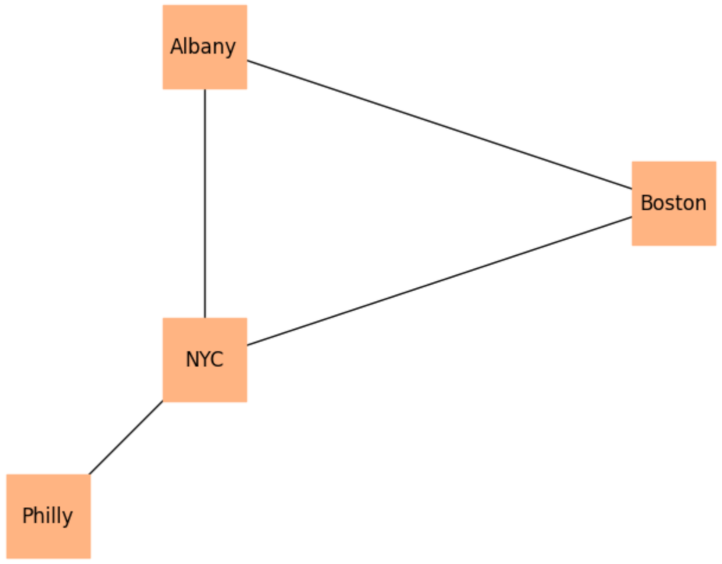
A graph where all the edges are directed from one node to another node.



Create different graphs with Python and NetworkX

Undirected Graph

An Undirected graph has edges that do not have a direction. The edges indicate a two-way relationship, in that each edge can be traversed in both directions.

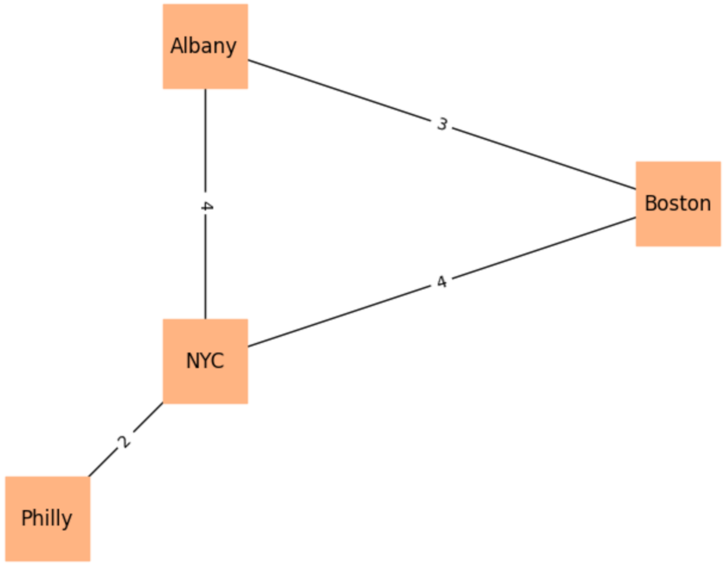


Graph

Create different graphs with Python and NetworkX

Weighted Graph

A graph in which each edge is given a numerical weight. A special type of labelled graph in which the labels are numbers (which are usually taken to be positive).

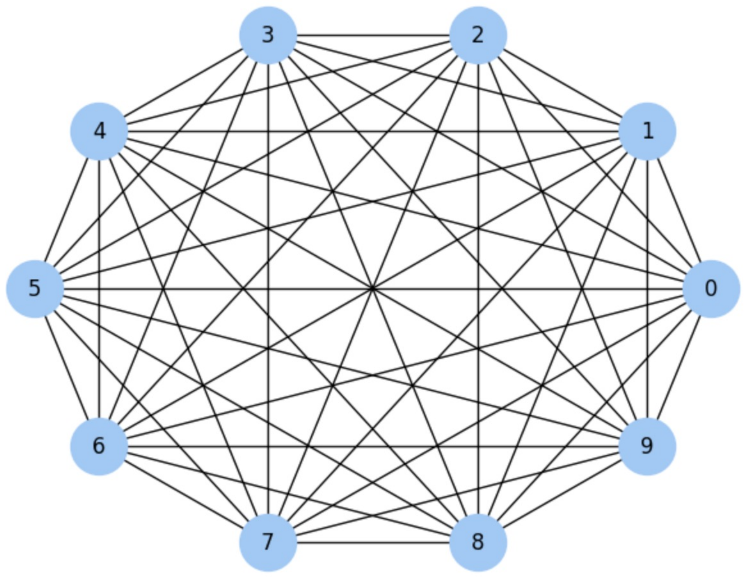


Graph

Create different graphs with Python and NetworkX

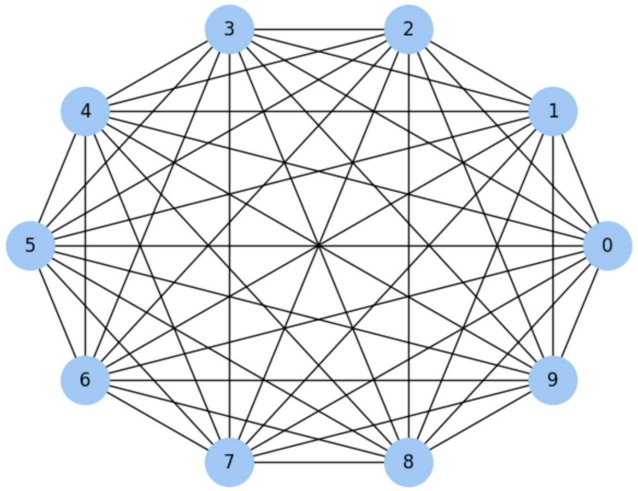
Complete Graph

A graph where every node is connected to every other node.



Complete Graph

A graph where every node is connected to every other node.

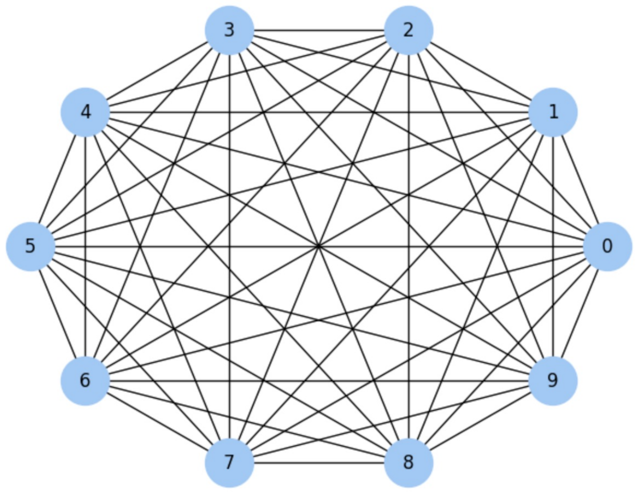


A complete graph of n nodes has $n(n-1)/2$ edges, with the degree of each node equals to $n-1$

Graph

Degree of a node

The number of edges connected to the node, or the number of neighbours of each node



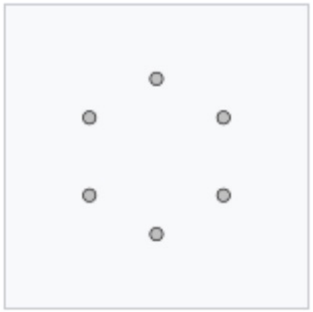
Each node has 9 neighbours

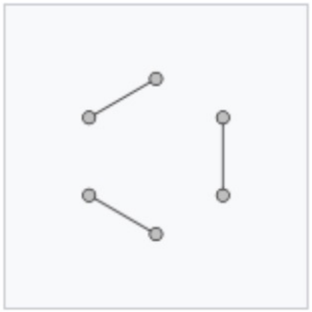
2 Graph

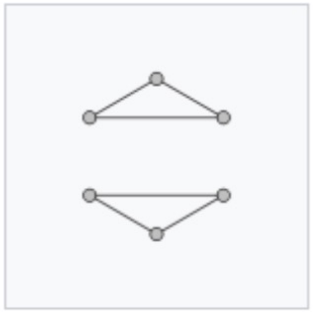
A complete graph is a special case of a regular graph.

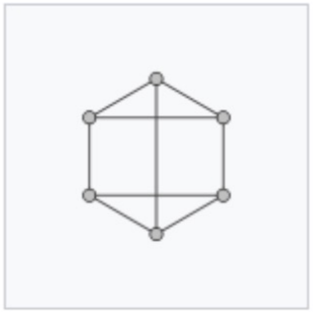
Regular graph is a graph that each node has the **same** number of neighbours, or every node has the same degree.

k*-regular graph or regular graph of degree *k: a regular graph with vertices of degree k.


0-regular graph


1-regular graph


2-regular graph


3-regular graph

0 LECTURE PLAN

Lecture 2: Graph I

1. Recap
2. Different types of Graphs
   - Implementation of graph generation
3. **Erdos Renyi Graph**
4. Connectivity of the ER graph
5. Random Graph vs. Regular Graph

ER Graph

Random Graph

A graph with nodes and edges generated from random.

Erdos Renyi (ER) Graph:

$$G(n, p)$$

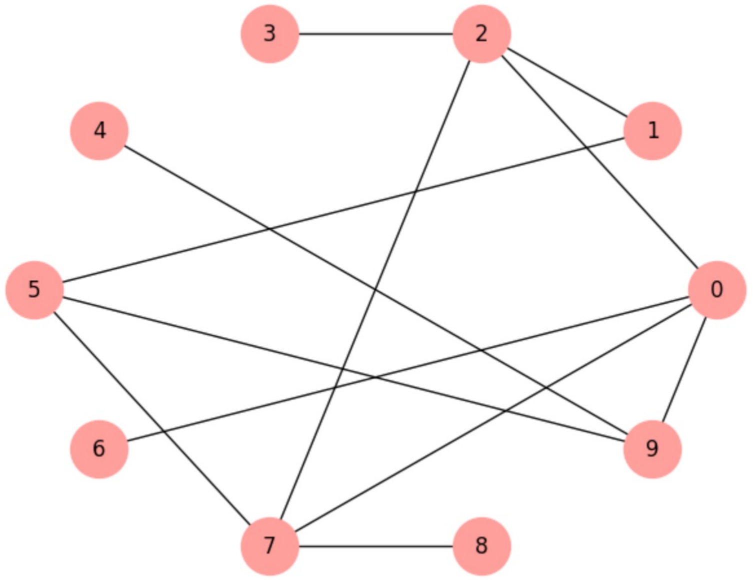
n is the number of nodes and p is the probability that there is an edge between any two nodes.

Note that this is not the same format as we just saw with $G = (V, E)$. This is saying that the Graph, G, is a function of n and p. i.e. it depends on these two parameters that we need to choose.

ER Graph

Create ER graphs with Python and NetworkX

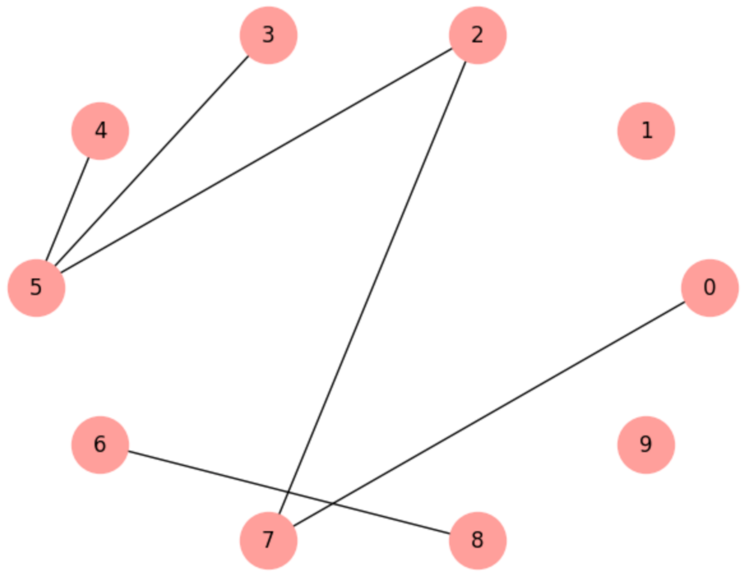
An ER random graph of $n = 10, \boldsymbol{p} = \mathbf{0.3}$



ER Graph

Create ER graphs with Python and NetworkX

An ER random graph of $n = 10, \boldsymbol{p} = \mathbf{0.1}$



ER Graph

Erdos Renyi (ER) Graph Property

Connectivity

*An undirected graph is **connected** if there is a path from every node to every other node.*

Path: a sequence of nodes with an edge between each consecutive pair.

Start at any node and whether you can reach all other nodes.

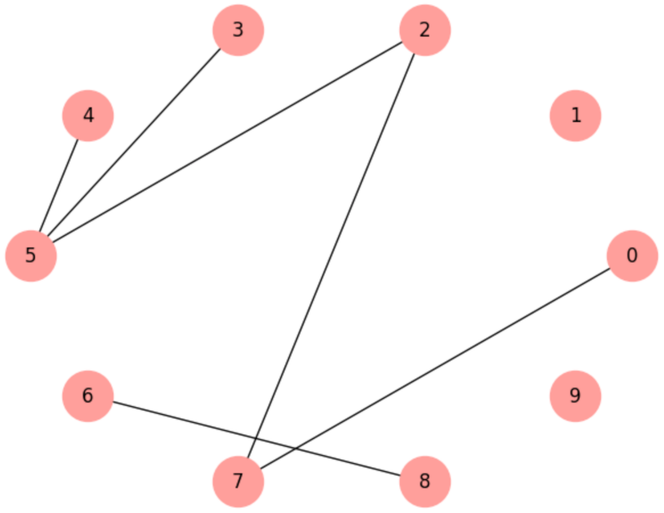
ER Graph

Erdos Renyi (ER) Graph Property

Connectivity

*An undirected graph is **connected** if there is a path from every node to every other node.*

Start at any node and whether you can reach all other nodes.



An ER random graph of $n = 10, \boldsymbol{p = 0.1,}$ *is this graph connected?*

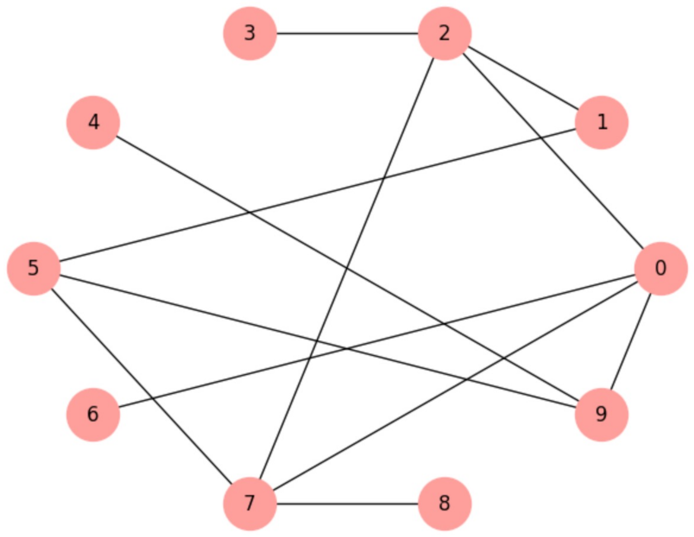
ER Graph

Erdos Renyi (ER) Graph Property

Connectivity

*An undirected graph is **connected** if there is a path from every node to every other node.*

Start at any node and whether you can reach all other nodes.



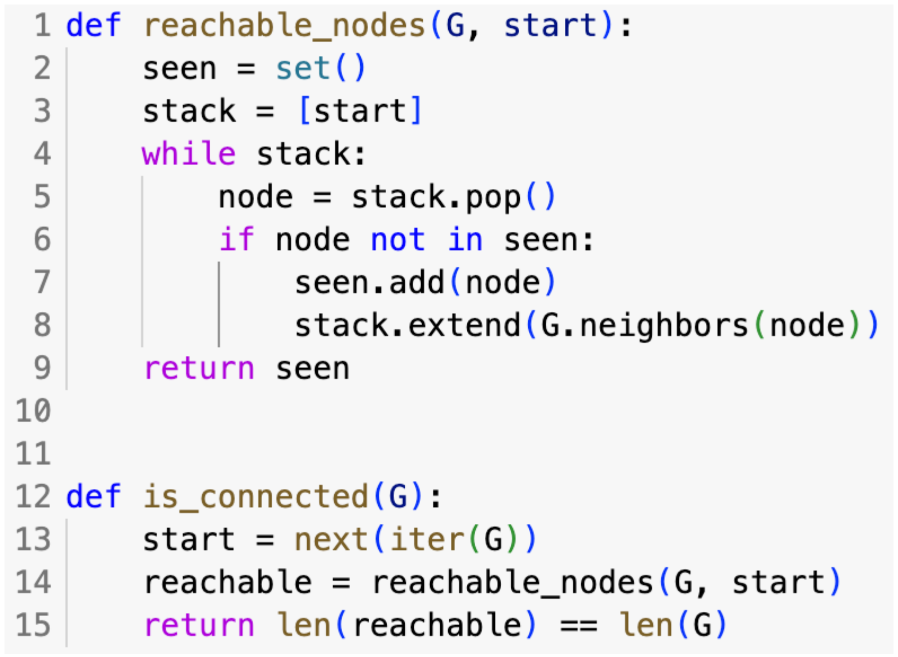
An ER random graph of $n = 10, \boldsymbol{p} = \mathbf{0.3},$ *is this graph connected?*

3 ER Graph

Erdos Renyi (ER) Graph Property

*An undirected graph is **connected** if there is a path from every node to every other node.*

Check graph connectivity with Python



```python
def reachable_nodes(G, start):
    seen = set()
    stack = [start]
    while stack:
        node = stack.pop()
        if node not in seen:
            seen.add(node)
            stack.extend(G.neighbors(node))
    return seen


def is_connected(G):
    start = next(iter(G))
    reachable = reachable_nodes(G, start)
    return len(reachable) == len(G)
```

ER Graph

Erdos Renyi (ER) Graph Property

Connectivity

the existence of **abrupt** changes in connectivity of random graphs as random edges are added.

In an ER graph, the probability that the graph is connected is very **low when $\boldsymbol{p}$ is small and nearly 1 when $\boldsymbol{p}$ is large**. Between these two regimes, there is a rapid transition at a particular value of p, denoted $\boldsymbol{p}^*$.

$$\boldsymbol{p}^* = (\ln n)/n$$

n is the number of nodes.

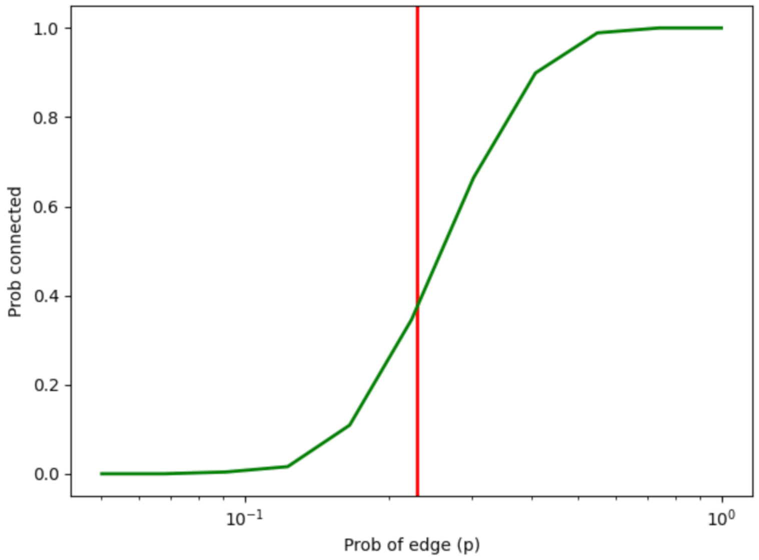
$\boldsymbol{G}(\boldsymbol{n}, \boldsymbol{p})$ is unlikely to be connected if $\boldsymbol{p} < \boldsymbol{p}^*$and very likely to be connected if $\boldsymbol{p} > \boldsymbol{p}^*$

Connectivity of ER Graph

How to test this claim?

$\boldsymbol{G(n,p)}$ is unlikely to be connected if $\boldsymbol{p} < \boldsymbol{p}^*$and very likely to be connected if $\boldsymbol{p} > \boldsymbol{p}^*$

Run experiments to estimate by generating a large number of random graphs and counting how many are connected.

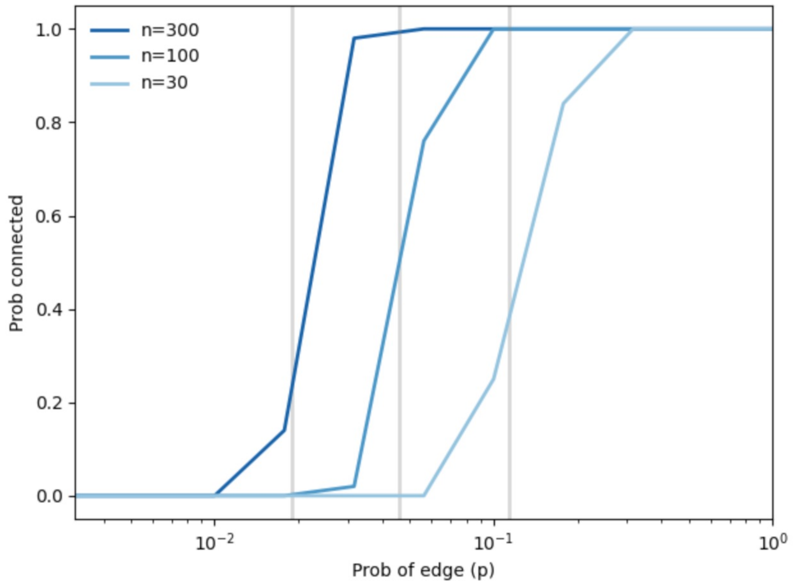


Connectivity of ER Graph

How to test this claim?

$\boldsymbol{G(n,p)}$ is unlikely to be connected if $\boldsymbol{p} < \boldsymbol{p}^*$and very likely to be connected if $\boldsymbol{p} > \boldsymbol{p}^*$

As n increases, the critical value $\boldsymbol{p}^*$ gets smaller, and the transition gets more abrupt.



LECTURE PLAN

Lecture 2: Graph I

1. Recap
2. Different types of Graphs
   - Implementation of graph generation
3. Erdos Renyi Graph
4. Connectivity of the ER graph
5. **Random Graph vs. Regular Graph**

5

Random Graph vs. Regular Graph

What's the point of modelling a random graph?

- Random graph is an important baseline for comparison with others
- It is important for understanding the stochastic process

When creating the ER graph, we also assume that:

- The edges are independent
- Each edge is equally alike

Relate the ER graph to the Complex System features…

Emergence

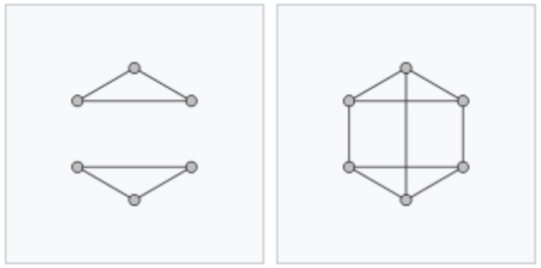
Local Interaction $\rightarrow$ Global Patterns

Random Graph vs. Regular Graph

Two other properties of graphs:

1. Clustering

- a measure of the “cliquishness” of the graph. In a graph, a clique is a subset of nodes that are all connected to each other.



2-regular graph   3-regular graph

- Clustering coefficient quantifies the likelihood that two nodes that are connected to the same node are also connected to each other (i.e. A friend of my friend is my friend).

Random Graph vs. Regular Graph

Clustering Coefficient Calculation

For a node u, has k neighbours. If all of the neighbours are connected to each other, there would be $k(k-1)/2$ edges among them.

The local clustering coefficient for u, denoted as C_u is the fraction of those edges that actually exist.

The graph average clustering coefficient, $\bar{C}$, is the average of C_u over all nodes.

Random Graph vs. Regular Graph

Two other properties of graphs:

2. Path Length

- A measure of the average distance between two nodes.

Average distance between nodes is measured in number of edges on the shortest path.

If two nodes are disconnected, the distance between them is infinite.

Random Graph vs. Regular Graph

Random Graph vs. Regular Graph

Random Graph	Regular Graph
Low clustering Low path lengths	High clustering High path length

Neither is good for modelling graphs of high clustering and short path length

NEXT WEEK PREVIEW…

Lecture 3: Graph II

- More in Regular graphs
- Small world property and Watts-Strogatz Graphs
- Scale–free network
- Barabasi and Albert (BA) model