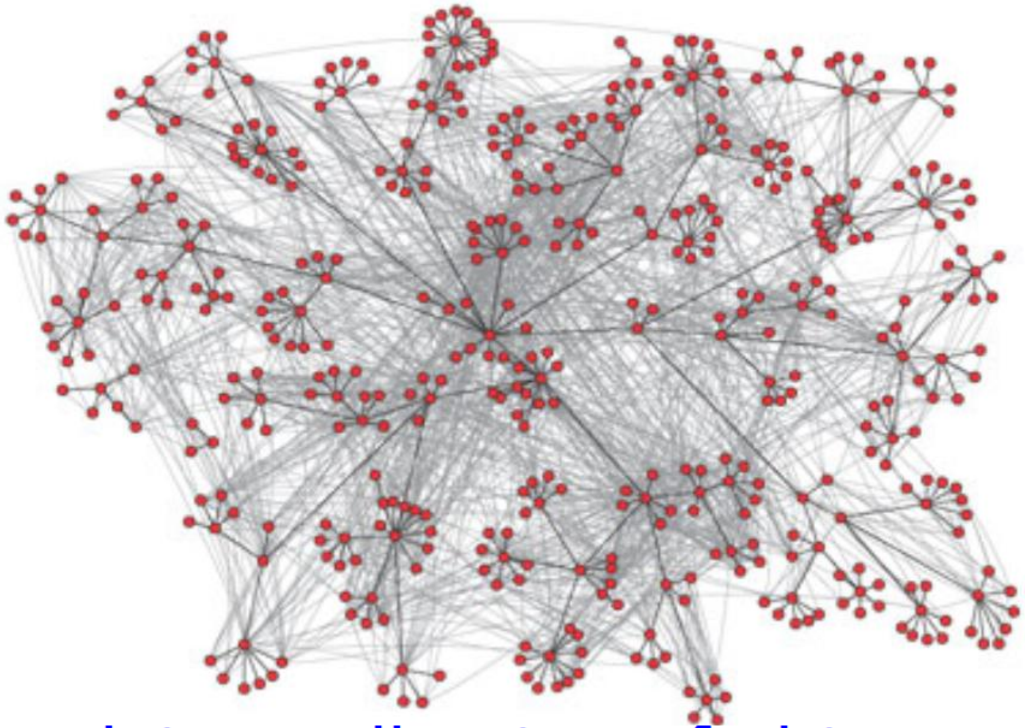
Graph Theory

CE642: Social and Economic Networks
Maryam Ramezani
Sharif University of Technology
maryam.ramezani@sharif.edu

01

Introduction

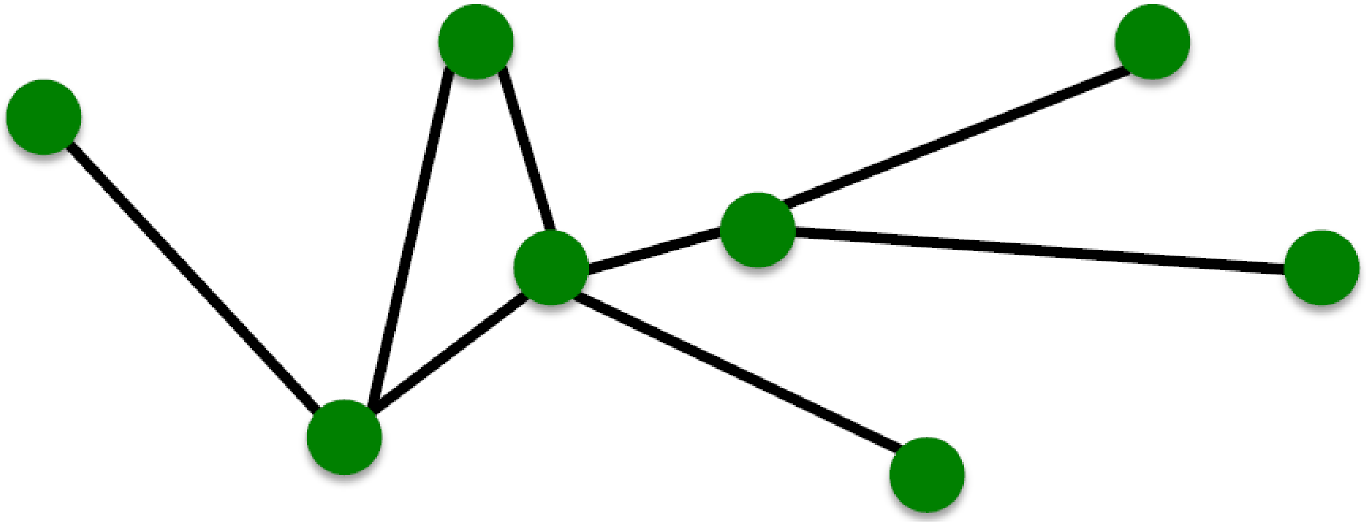
Structure of Networks



A network is a collection of objects where some pairs of objects are connected by links

What is the structure of the network?

Components of a Network



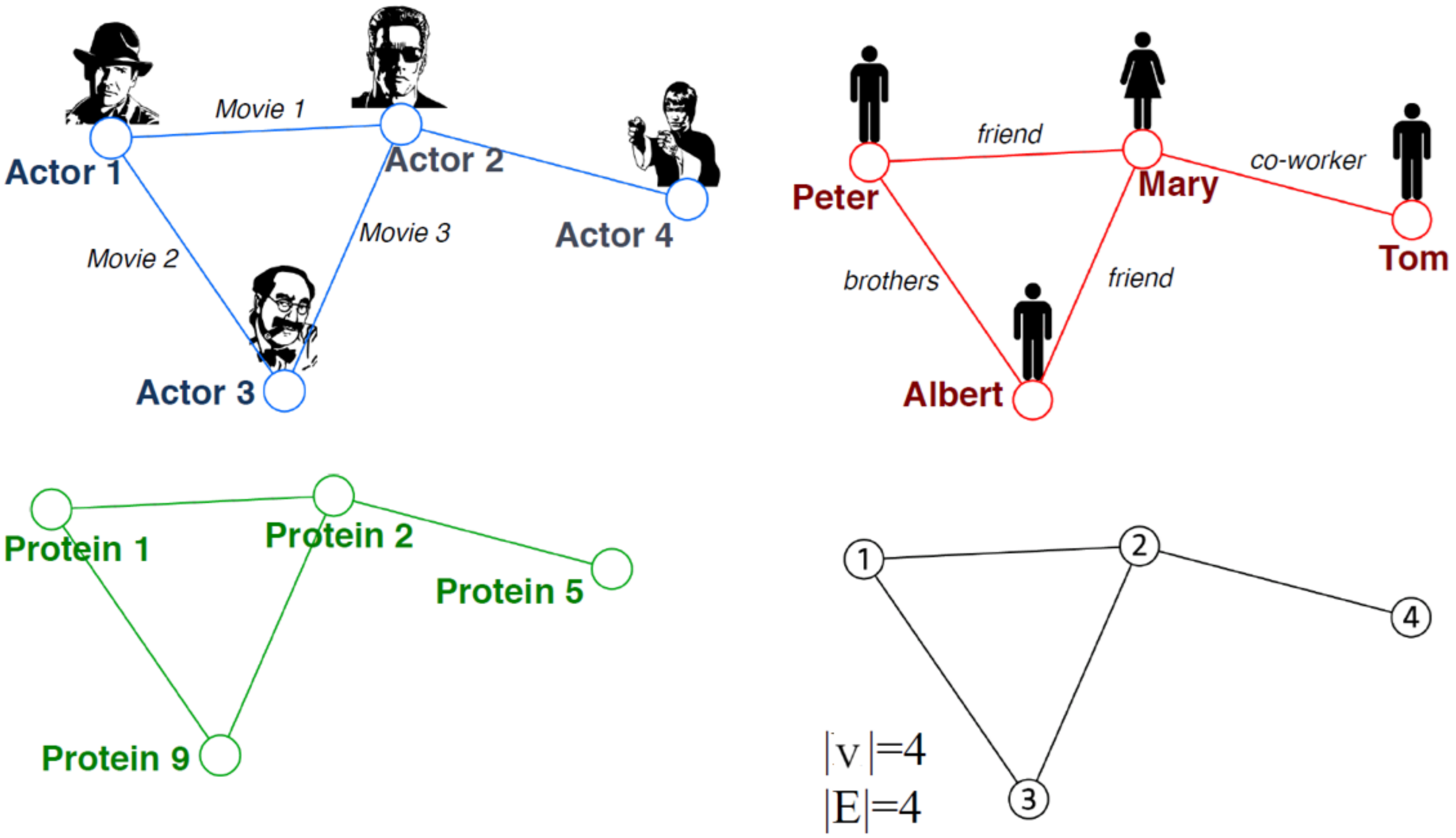
- Objects: nodes, vertices V where number of nodes is N
- Interactions: links, edges E
- System: network, graph G(V,E)

Networks or Graphs?

- Network often refers to real systems
  - Web, Social network, Metabolic network
  - Language: Network, node, link
- Graph is a mathematical representation of a network
  - Web graph, Social graph, Knowledge Graph
  - Language: Graph, vertex, edge

We will try to make this distinction whenever it is appropriate, but in most cases we will use the two terms interchangeably

Networks: Common Language



How do you define a network?

- How to build a graph:
  - What are nodes?
  - What are edges?
- Choice of the proper network representation of a given domain/problem determines our ability to use networks successfully:
  - In some cases there is a unique, unambiguous representation
  - In other cases, the representation is by no means unique
  - The way you assign links will determine the nature of the question you can study

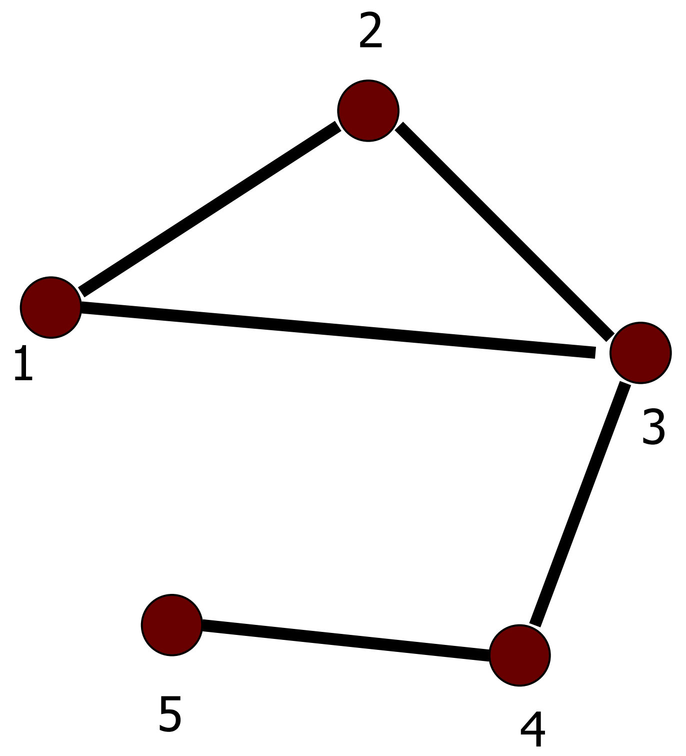
02

Types of Graphs

Undirected Graph

- Graph G=(V,E)
  - V = set of vertices
  - E = set of edges

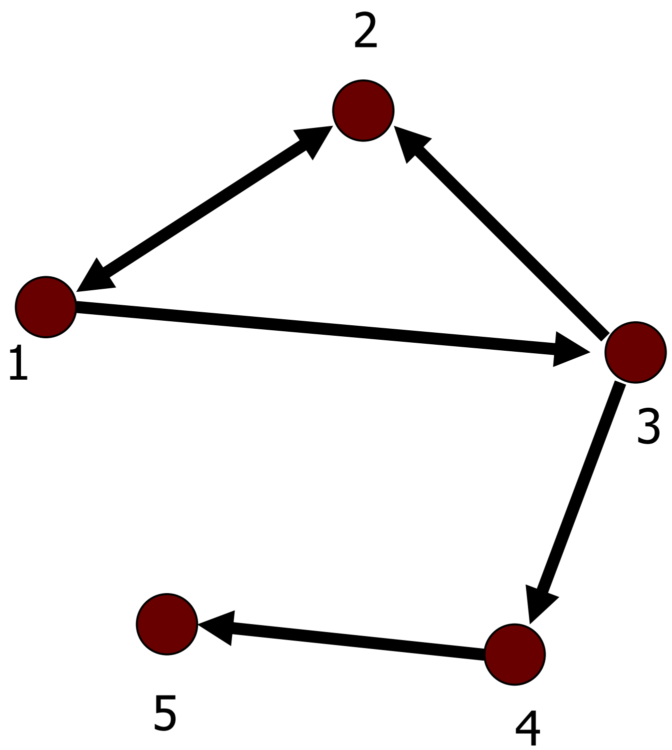
undirected graph
V = {1, 2, 3, 4, 5}
E={(1,2),(1,3),(2,3),(3,4),(4,5)}



Directed Graph

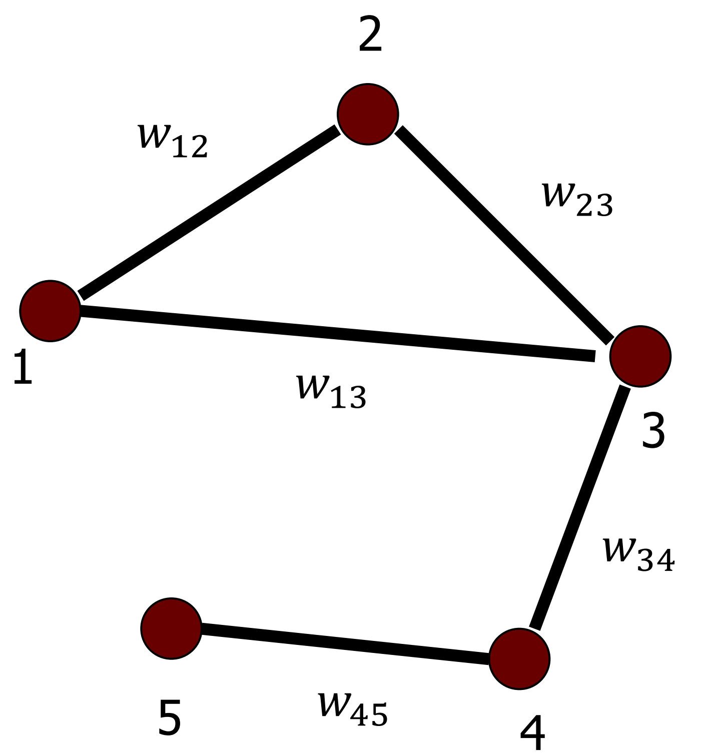
- Graph G=(V,E)
  - V = set of vertices
  - E = set of edges

directed graph
V = {1, 2, 3, 4, 5}
E={‹1,2›, ‹2,1› ‹1,3›, ‹3,2›, ‹3,4›, ‹4,5›}



Weighted Graph

- Graph G=(V,E)
  - V = set of vertices
  - E = set of edges and their weights

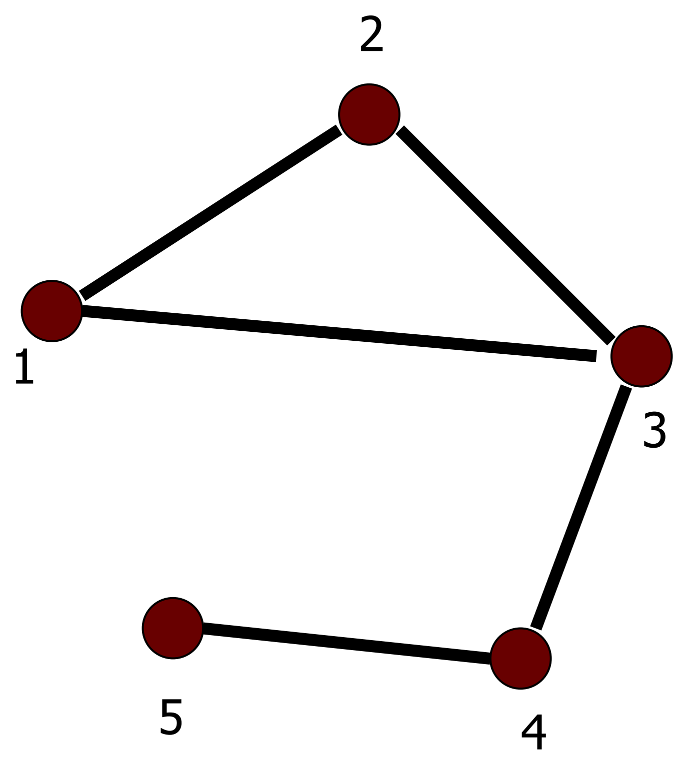


Weights can be either distances or similarities

weighted graph
V = {1, 2, 3, 4, 5}
E={(1,2,w_{12}),(1,3, w_{12}),(2,3, w_{12}),(3,4, w_{12}),(4,5, w_{12})}

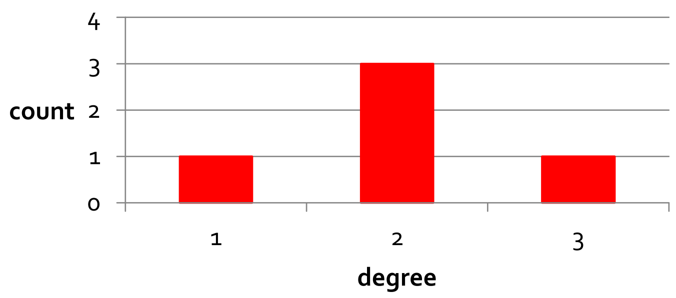
Undirected graph

- Neighborhood N(i) of node i
  - Set of nodes adjacent to i
- degree d(i) of node i
  - Size of N(i)
  - number of edges incident on i

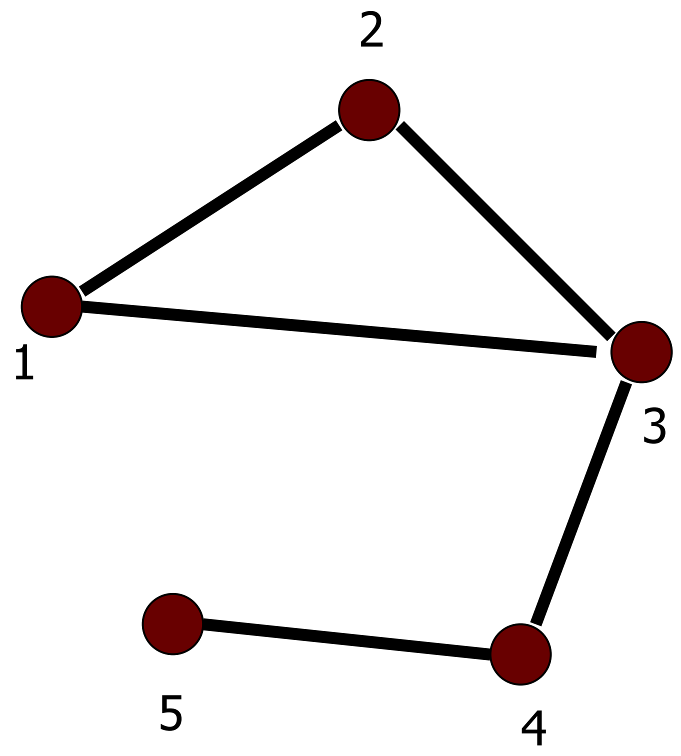


Undirected graph

- degree sequence
  - [d(1),d(2),d(3),d(4),d(5)]
  - [2,2,3,2,1]
- degree histogram
  - [(1:1),(2:3),(3,1)]

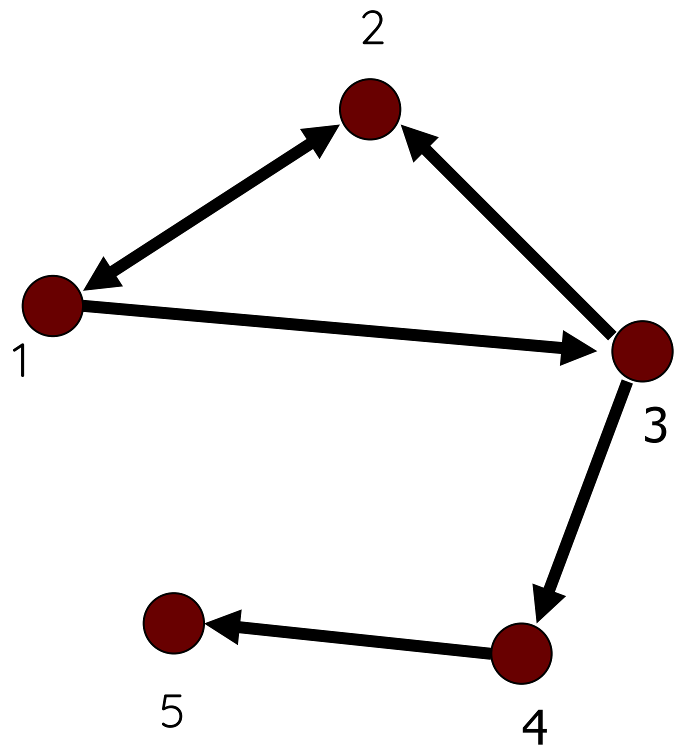


- degree distribution
  - [(1:0.2),(2:0.6),(3,0.2)]



Directed Graph

- in-degree $d_{in}(i)$ of node i
  - number of edges incoming to node i
- out-degree $d_{out}(i)$ of node i
  - number of edges leaving node i

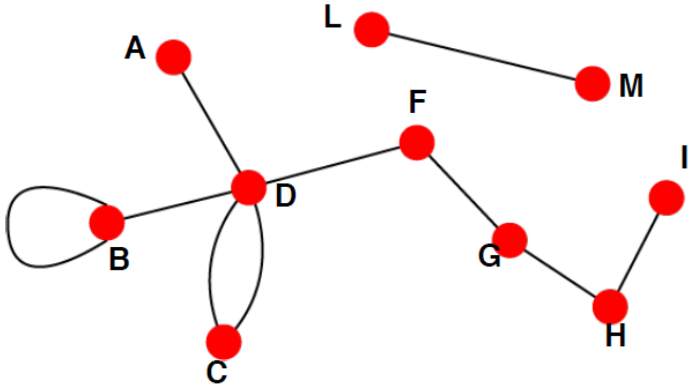


- in-degree sequence
  - [1,2,1,1,1]
- out-degree sequence
  - [2,1,2,1,0]
- in-degree histogram
  - [(1:4),(2:1)]
- out-degree histogram
  - [(0:1),(1:2),(2:2)]

Directed vs Undirected Graphs

Undirected

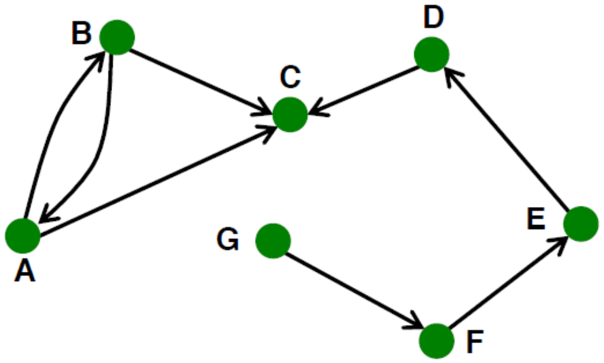
- **Links:** undirected (symmetrical, reciprocal)



- **Examples:**
  - Collaborations
  - Friendship on Facebook

Directed

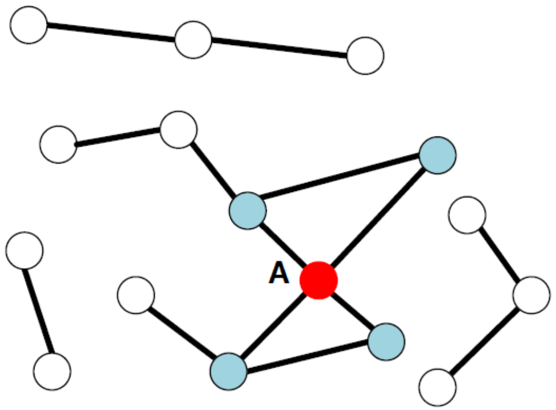
- **Links:** directed (arcs)



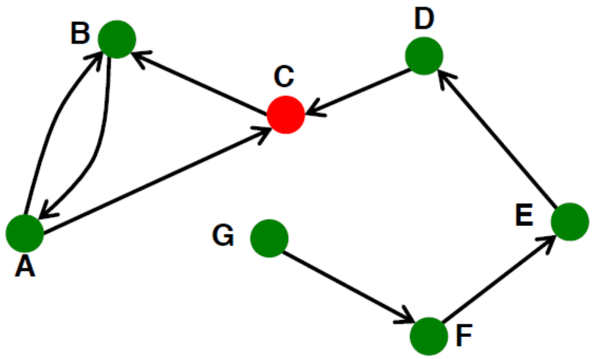
- **Examples:**
  - Phone calls
  - Following on Twitter

Node Degrees

Undirected



Directed



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

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

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

$$k_A = 4$$

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

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

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

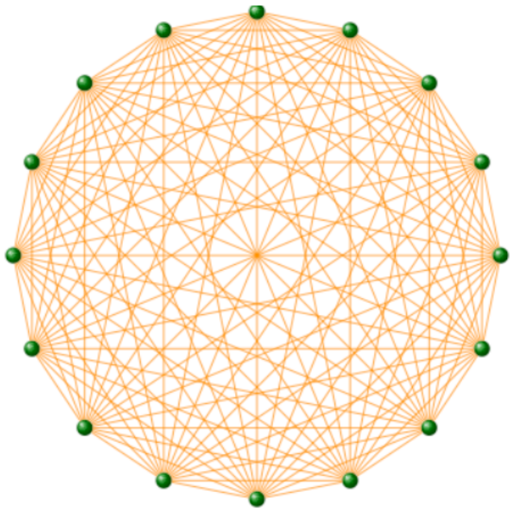
$$k_C^{in} = 2 \qquad k_C^{out} = 1 \qquad k_C = 3$$

$$\bar{k} = \frac{E}{N} \qquad \overline{k^{in}} = \overline{k^{out}}$$

Complete Graph

The **maximum number of edges** in an undirected graph on N nodes is

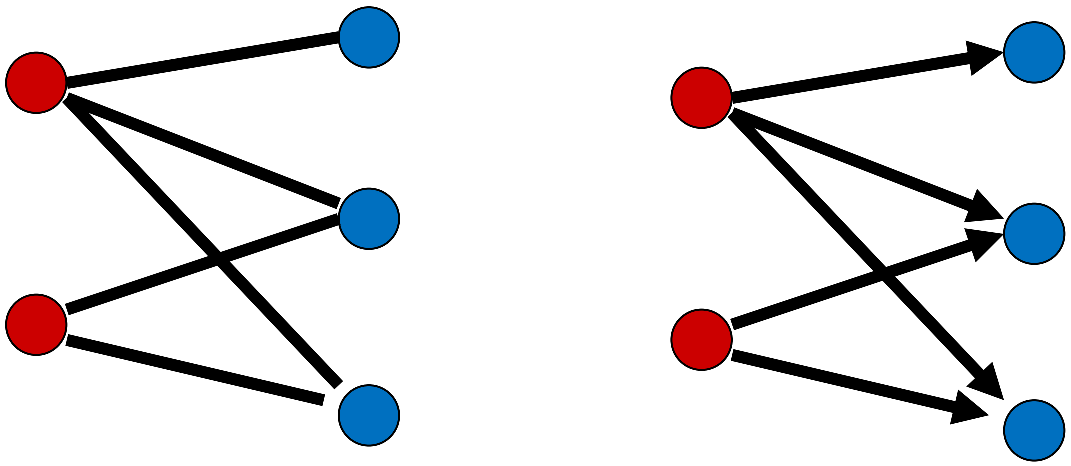
$$E_{\max} = \binom{N}{2} = \frac{N(N-1)}{2}$$



An undirected graph with the number of edges $E = E_{max}$ is called a **complete graph**, and its average degree is $N\text{-}1$

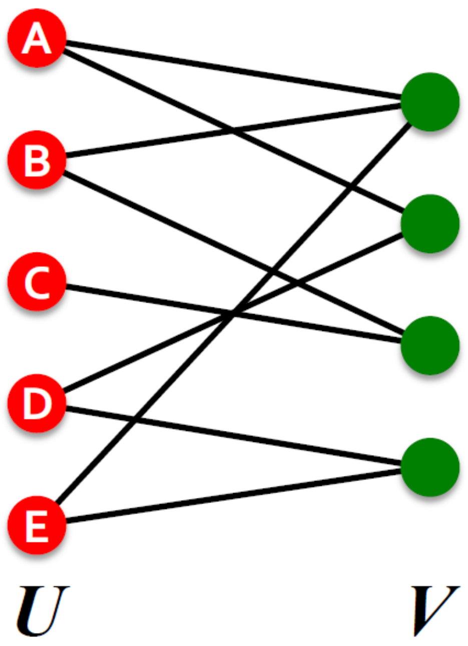
Bipartite graphs

- Graphs where the set of nodes V can be partitioned into two sets L and R, such that there are edges only between nodes in L and R, and there is no edge within L or R

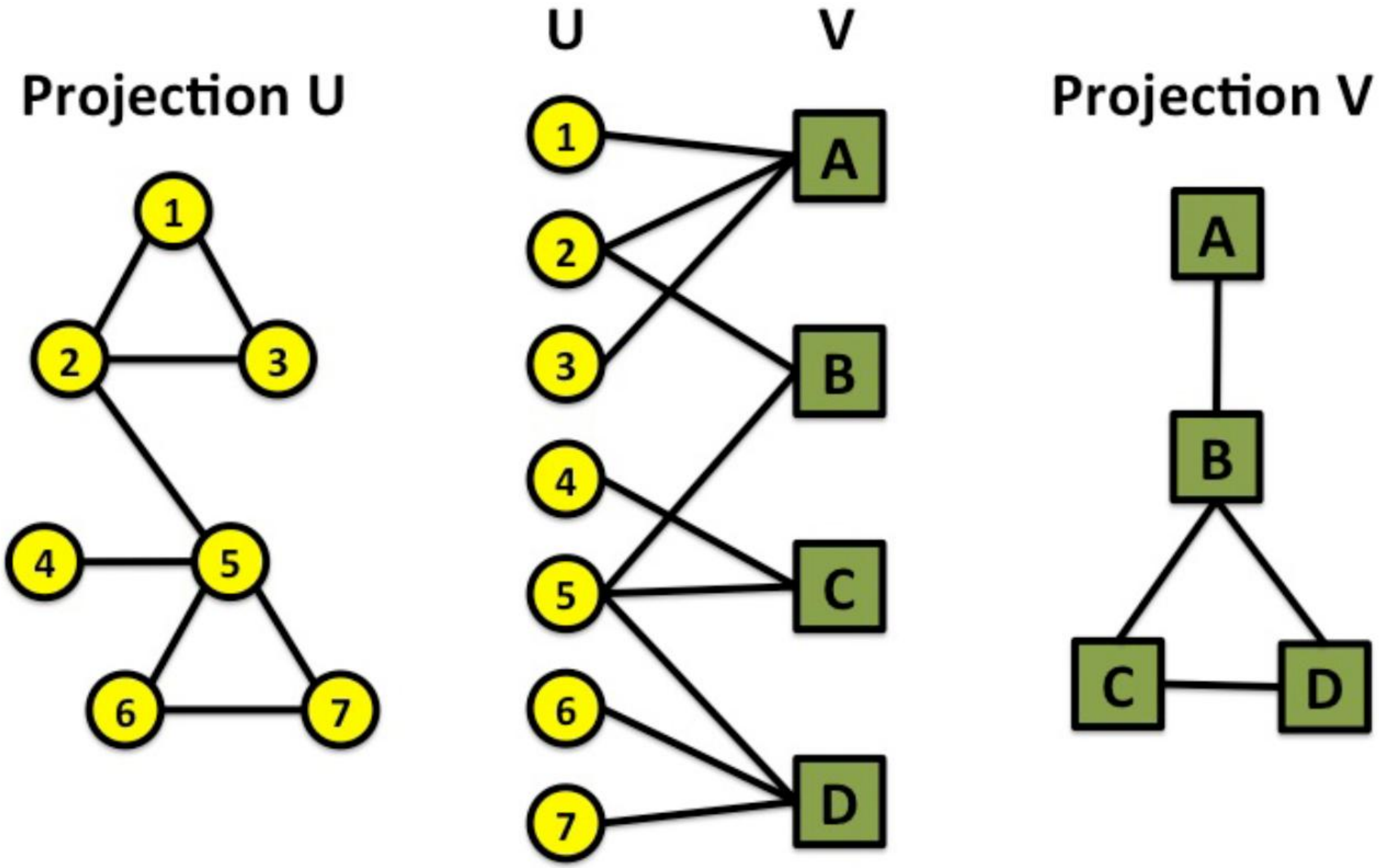


Bipartite Graph

- **Bipartite graph** is a graph whose nodes can be divided into two disjoint sets U and V such that every link connects a node in U to one in V; that is, U and V are **independent sets**
- **Examples:**
  - Authors-to-Papers (they authored)
  - Actors-to-Movies (they appeared in)
  - Users-to-Movies (they rated)
  - Recipes-to-Ingredients (they contain)
- **"Folded" networks:**
  - Author collaboration networks
  - Movie co-rating networks



Folded/Projected Bipartite Graph



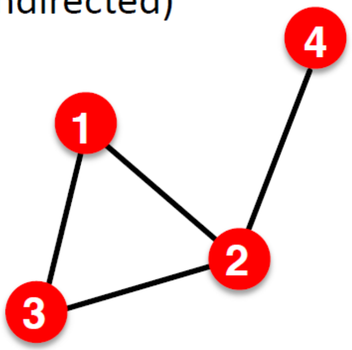
Edge Attributes

Possible options:

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

More Types of Graphs

- **Unweighted** (undirected)



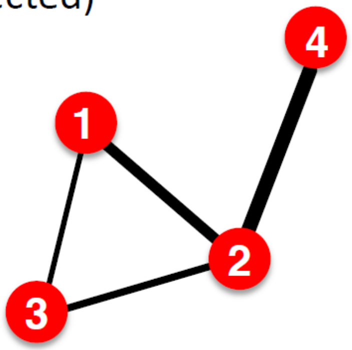
$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

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

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

Examples: Friendship, Hyperlink

- **Weighted** (undirected)



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

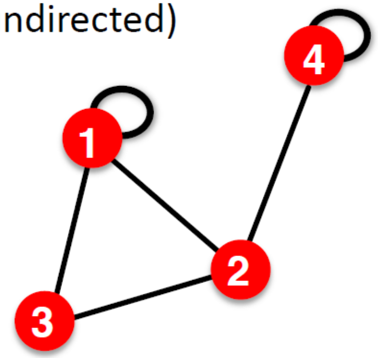
$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

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

Examples: Collaboration, Internet, Roads

More Types of Graphs

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



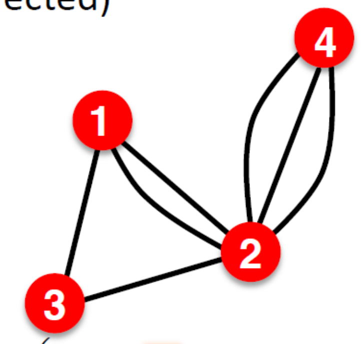
$$A_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

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

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

Examples: Proteins, Hyperlinks

- **Multigraph**
  (undirected)



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

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

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

Examples: Communication, Collaboration

03

Graph Traversals

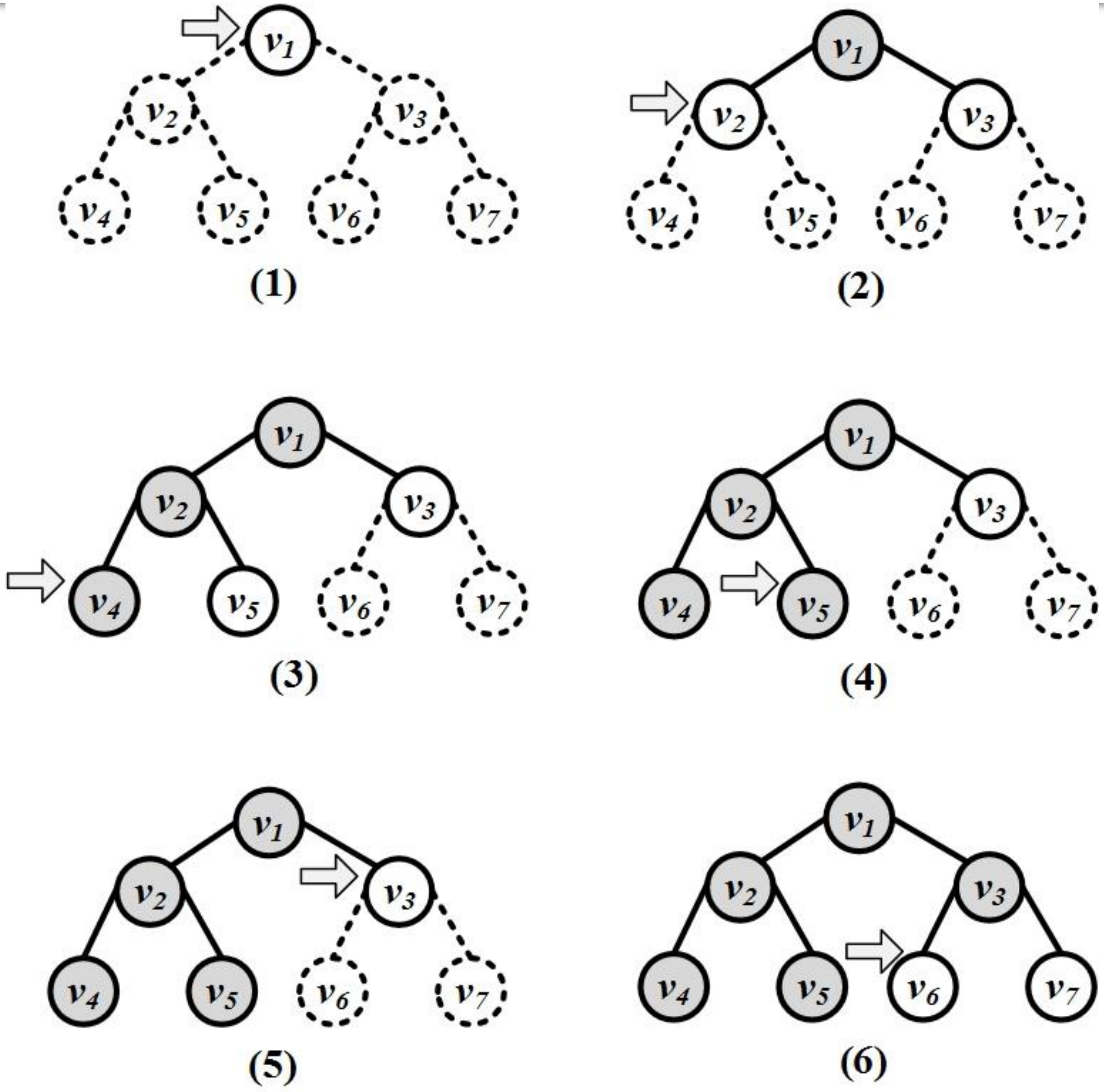
Graph Traversals

- A traversal is a procedure for visiting (going through) all the nodes in a graph:
  - Depth First Search (DFS)
  - Breadth First Search (BFS)

Depth First Search Traversal

- Depth-First Search (DFS) starts from a node i, selects one of its neighbors j from N(i) and performs Depth-First Search on j before visiting other neighbors in N(i).
  - The algorithm can be implemented using a *stack structure*

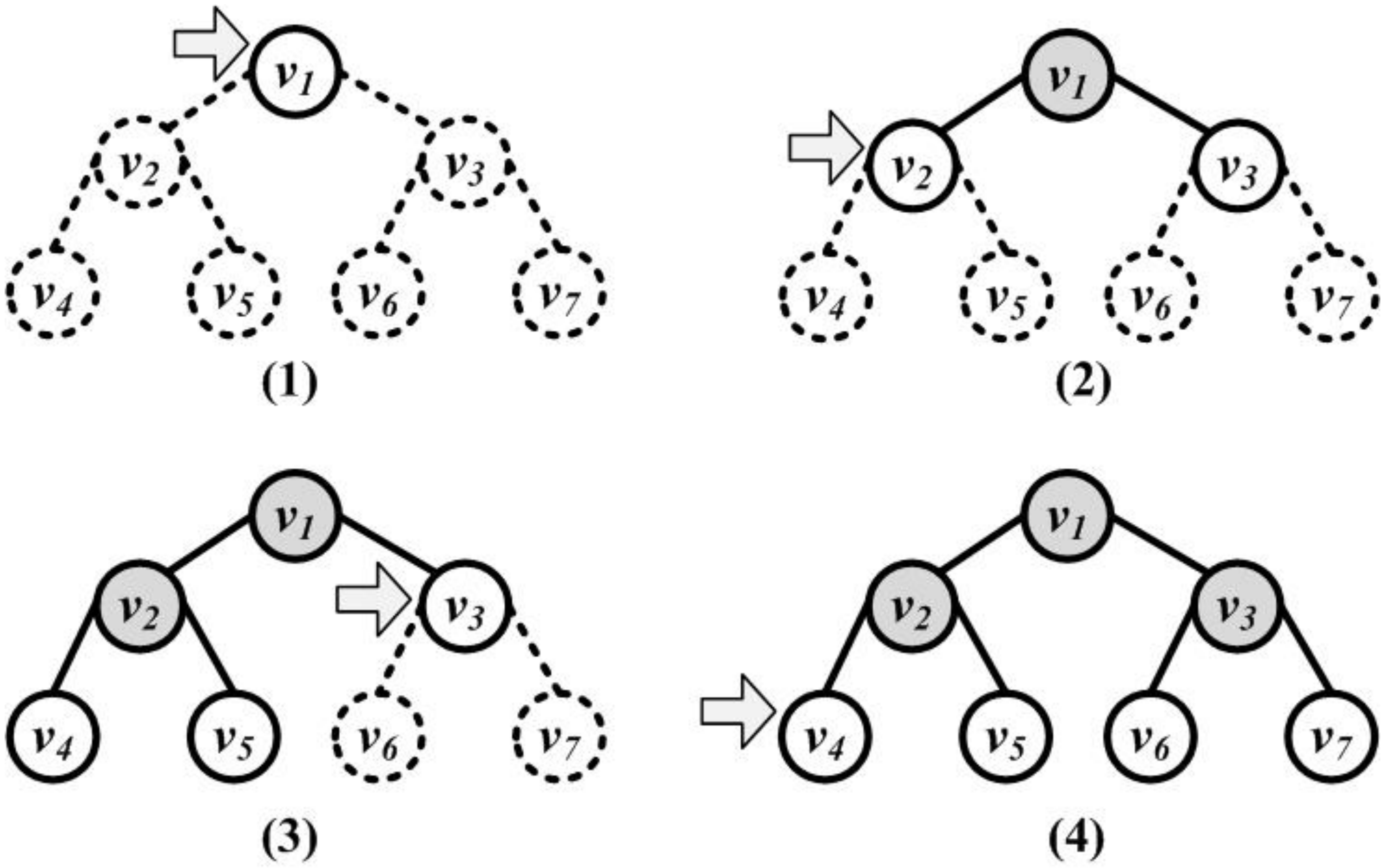
Example of DFS



Breadth First Search Traversal

- Breadth-First-Search (BFS) starts from a node, visits all its immediate neighbors first, and then moves to the second level by traversing their neighbors.
  - The algorithm can be implemented using a *queue structure*

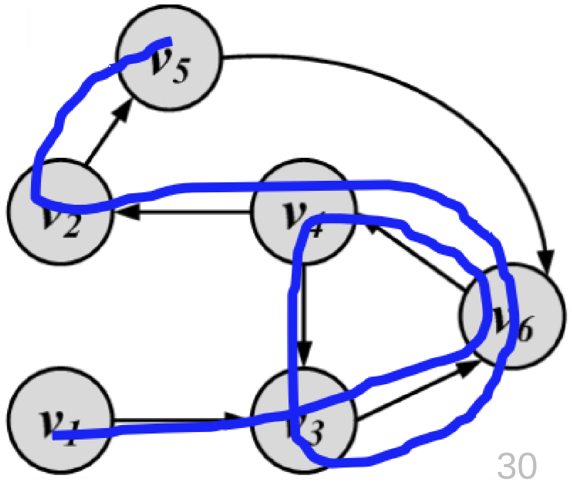
Example of BFS



Walk

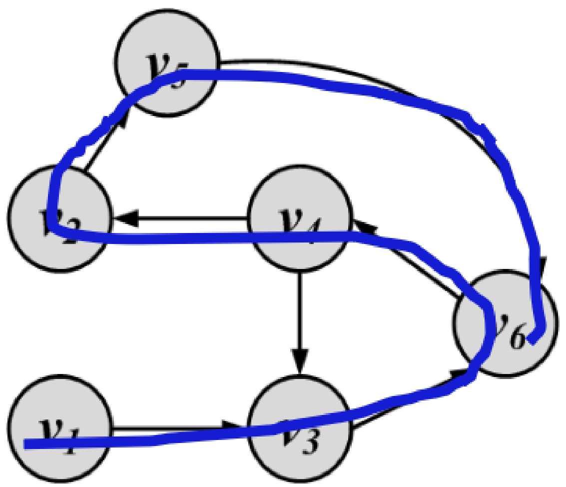
- A walk is a sequence of incident edges visited one after another
  - Open walk: A walk does not end where it starts
  - Closed walk: A walk returns to where it starts
- Representing a walk:
  - A sequence of edges: $e1, e2, \ldots, en$
  - A sequence of nodes: $v1, v2, \ldots, vn$
- Length of walk: the number of visited edges

Length of walk= 8



Trail

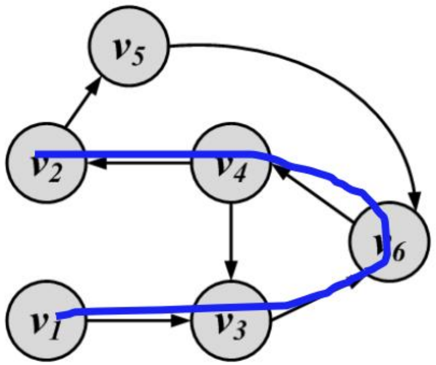
- A trail is a walk where no edge is visited more than once and all walk edges are distinct



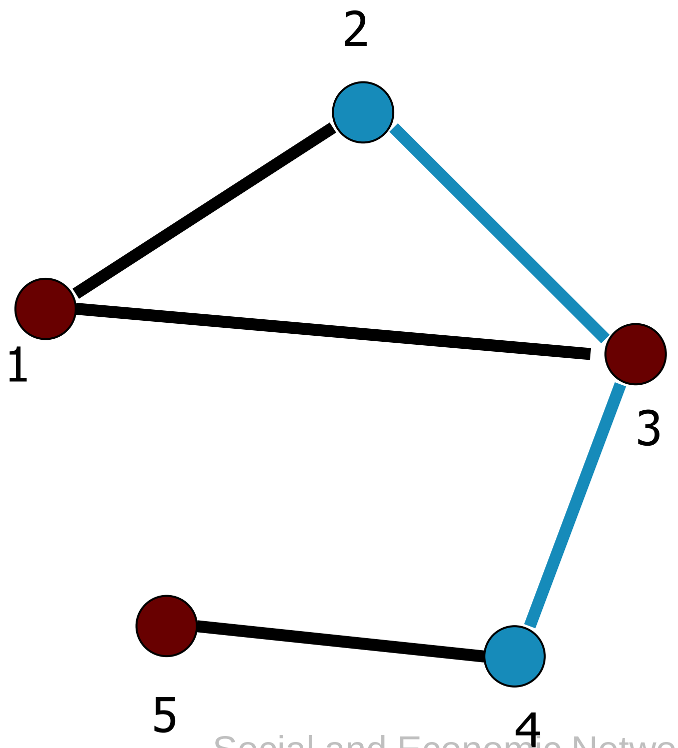
- A closed trail (one that ends where it starts) is called a tour or circuit

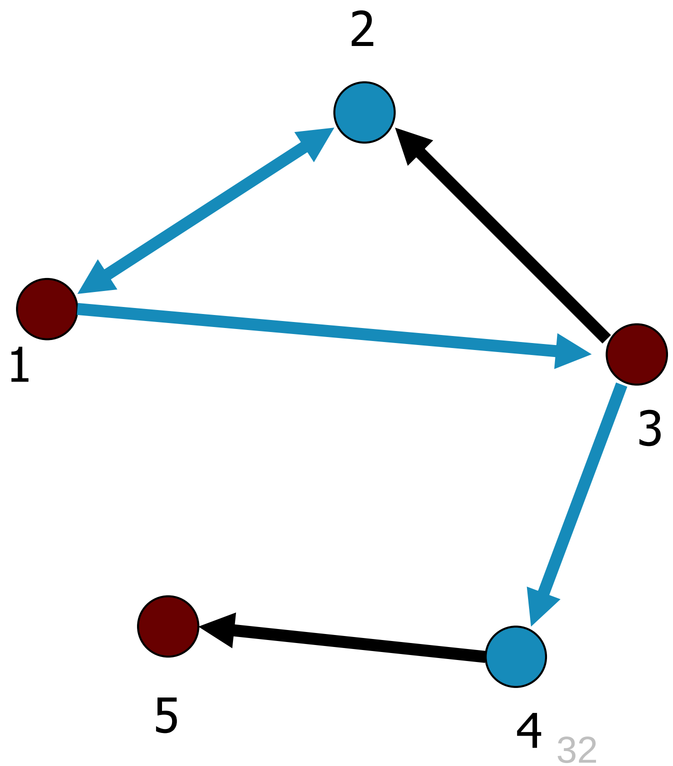
Paths

- A walk where nodes and edges are distinct is called a path.
- Path from node i to node j: a sequence of edges (directed or undirected from node i to node j)
  - **path length**: number of edges on the path nodes i and j are connected
  - **cycle**: a path that starts and ends at the same node. A closed path!



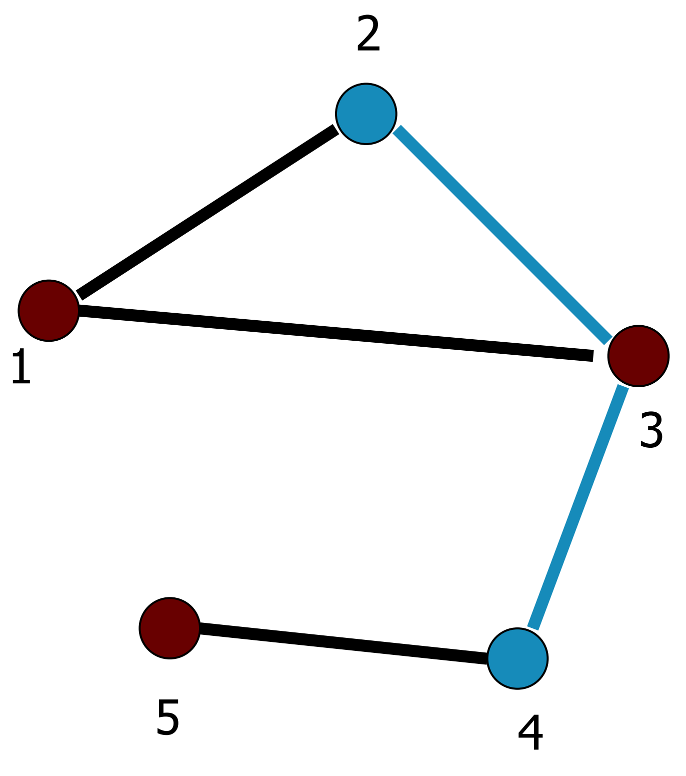
Length of path= 4

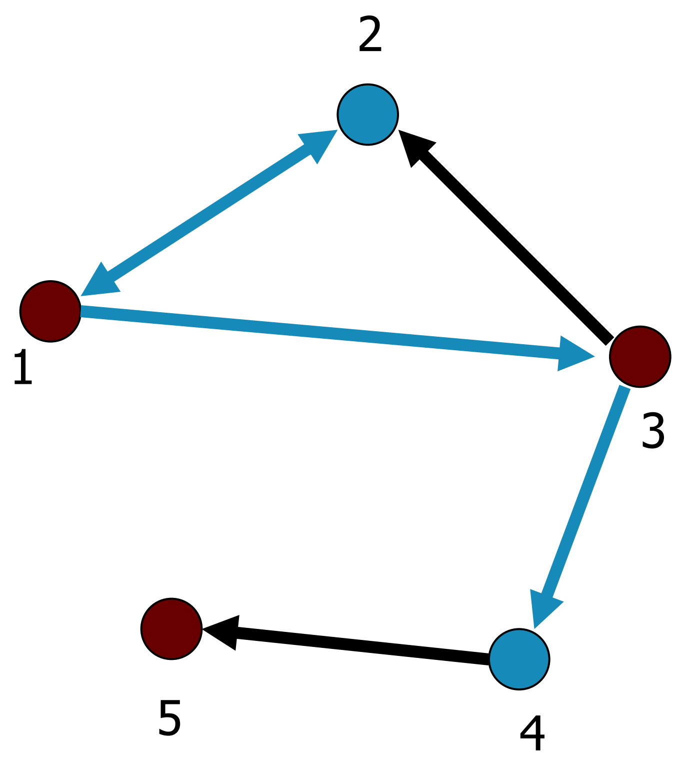




Shortest Paths

- Shortest Path from node i to node j
  - also known as BFS path, or geodesic path



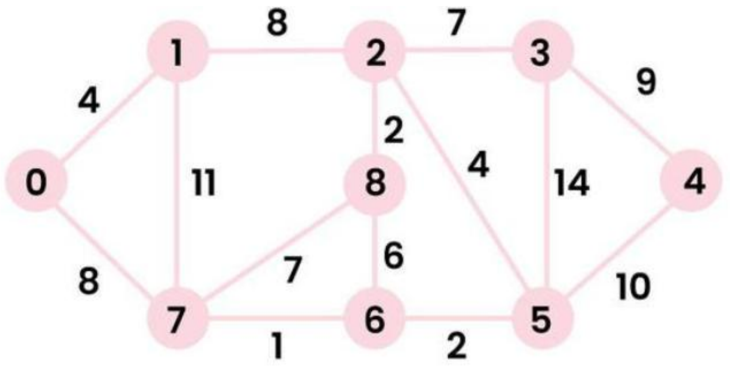


Shortest paths on weighted graphs

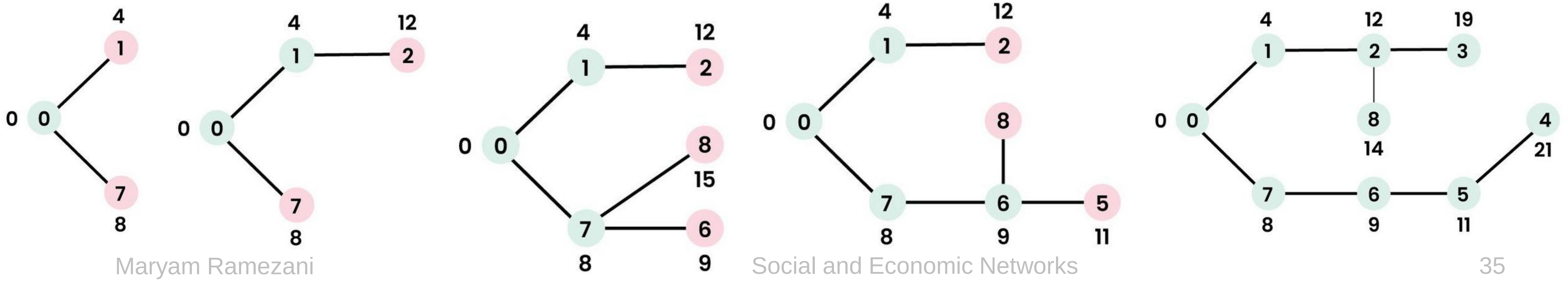
- Shortest paths on weighted graphs are harder to construct
  - There are several well known algorithms for finding single-source, or all-pairs shortest paths
  - For example: Dijkstra's Algorithm

Dijkstra's Algorithm

- To understand the Dijkstra's Algorithm lets take a graph and find the shortest path from source to all nodes. Consider below graph and src = 0.

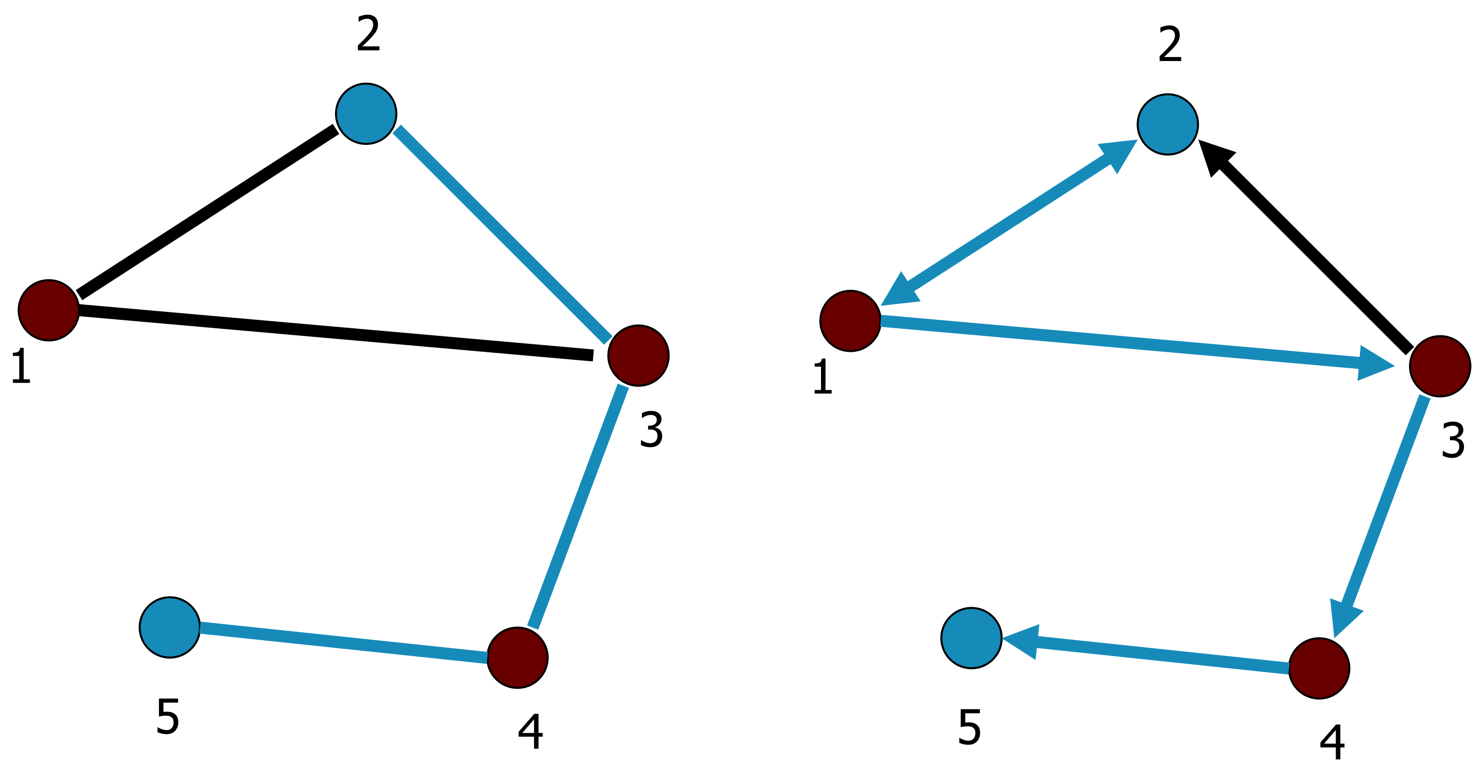


sptSet={0, INF, INF, INF, INF, INF, INF, INF}



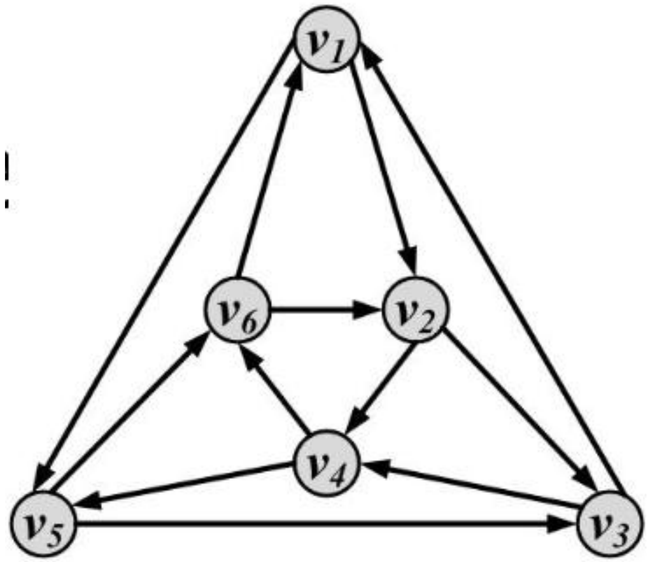
Diameter

- The longest shortest path in the graph



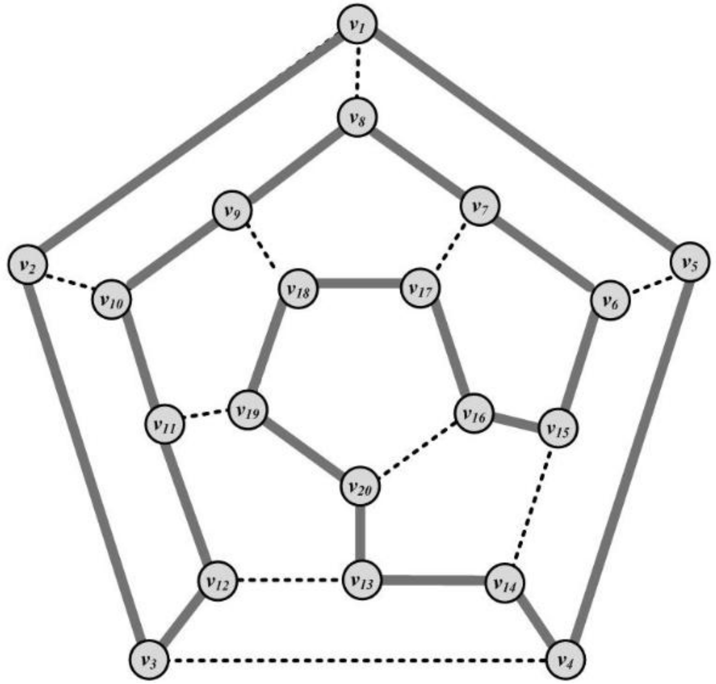
Eulerian Tour

- All edges are traversed only once
  - Konigsberg bridges



Hamiltonian Cycle

- A cycle that visits all nodes

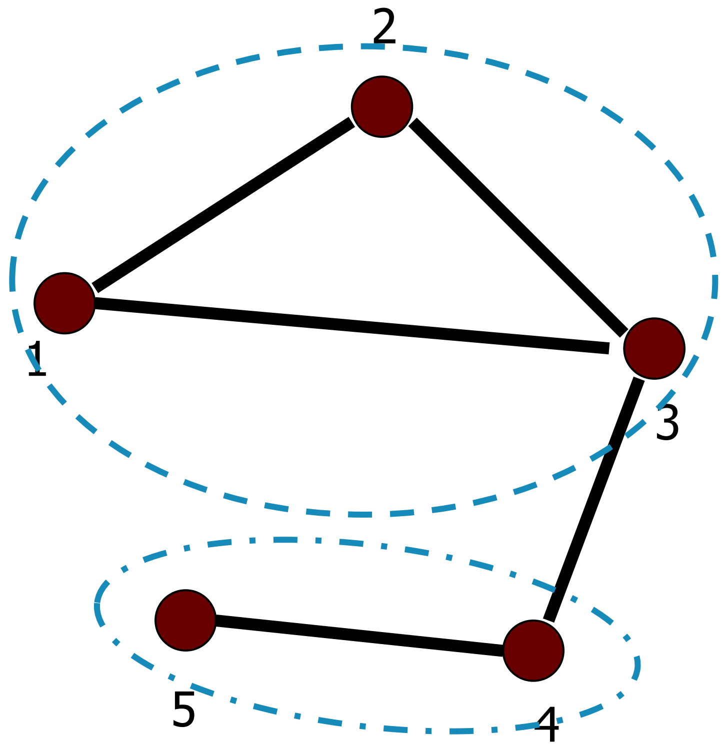


04

Graph Connectivity

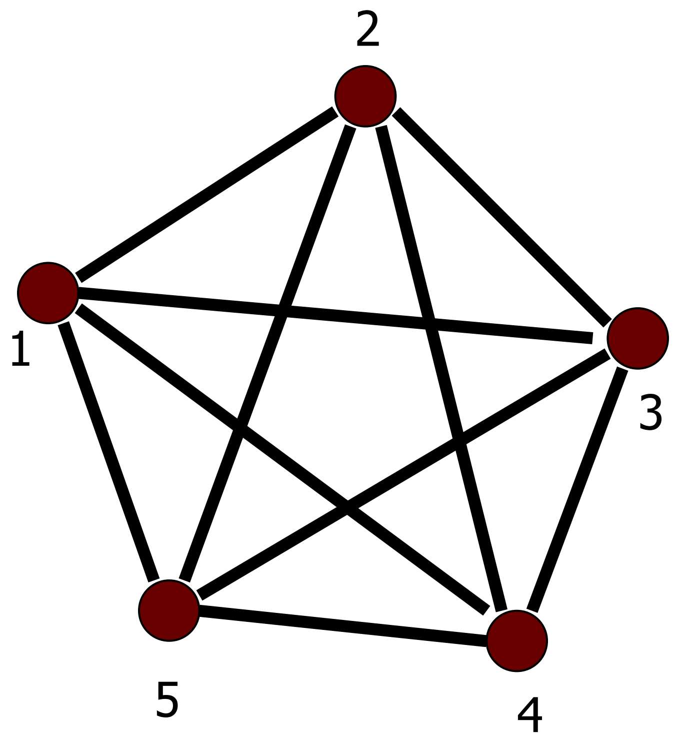
Undirected graph

- **Connected graph**: a graph where there every pair of nodes is connected
- **Disconnected graph**: a graph that is not connected
- **Connected Components**: subsets of vertices that are connected



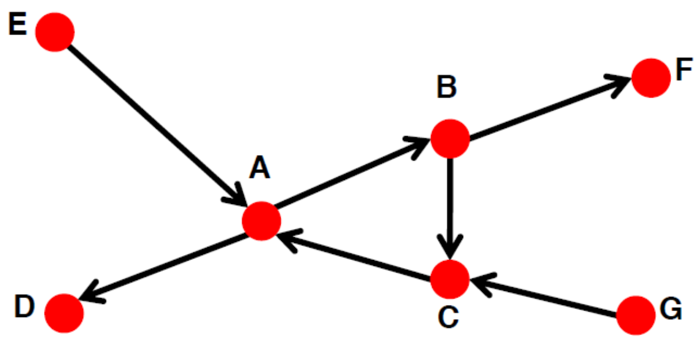
Fully Connected Graph

- Clique K_n
- A graph that has all possible n(n-1)/2 edges

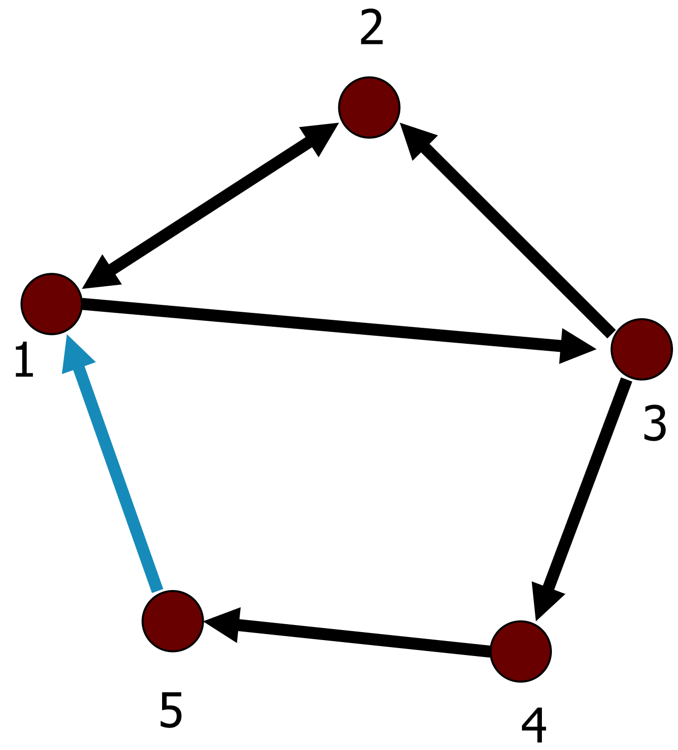


Connectivity of Directed Graph

- **Strongly connected graph**: there exists a path from every i to every j. has a path from each node to every other node and vice versa (e.g., A-B path and B-A path)
- **Weakly connected graph**: If edges are made to be undirected the graph is connected

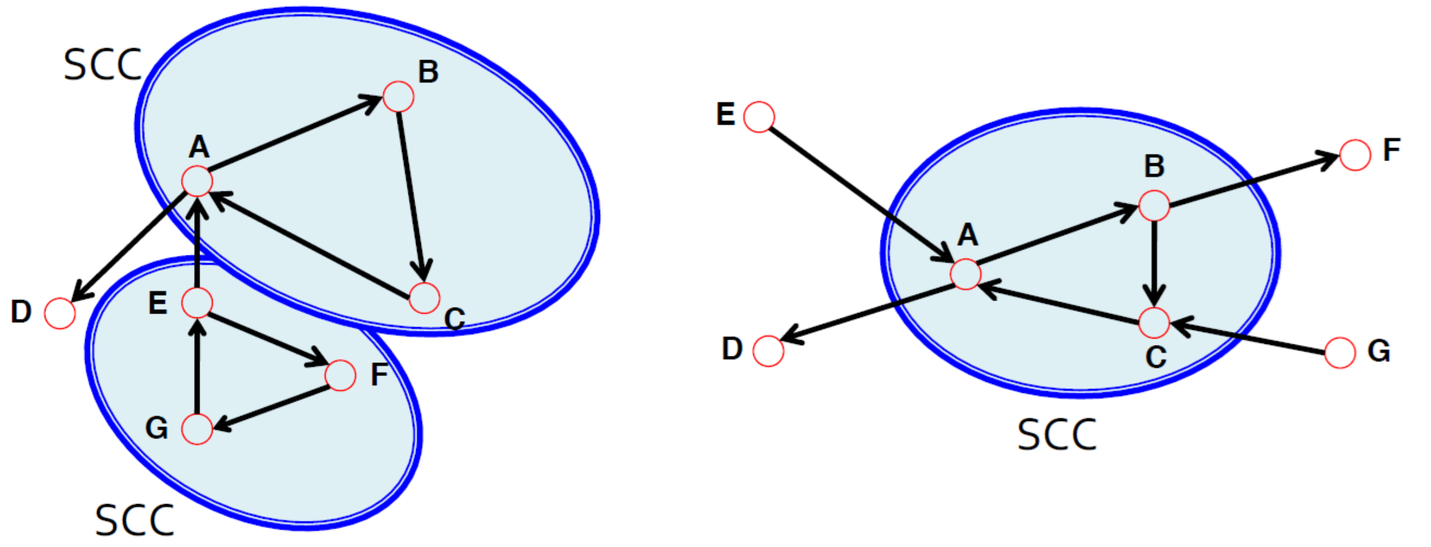


Graph on the left is connected but not strongly connected (e.g., there is no way to get from F to G by following the edge directions).



Connectivity of Directed Graph

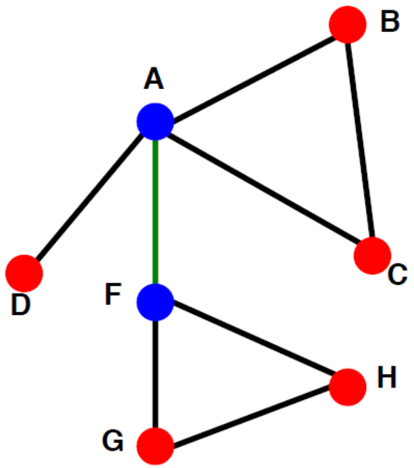
- **Strongly connected components (SCCs) can be identified, but not every node is part of a nontrivial strongly connected component.**

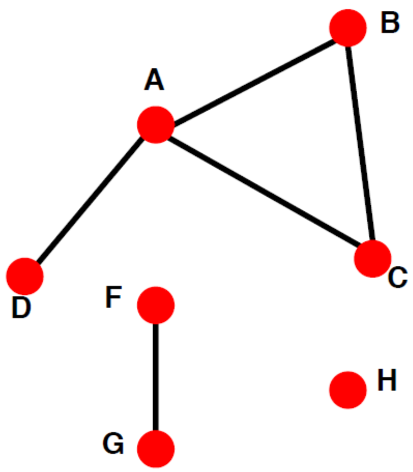


- In-component: nodes that can reach the SCC,
- Out-component: nodes that can be reached from the SCC.

Connectivity of Undirected Graphs

- **Connected (undirected) graph:**
  - Any two vertices can be joined by a path
- A disconnected graph is made up by two or more connected components





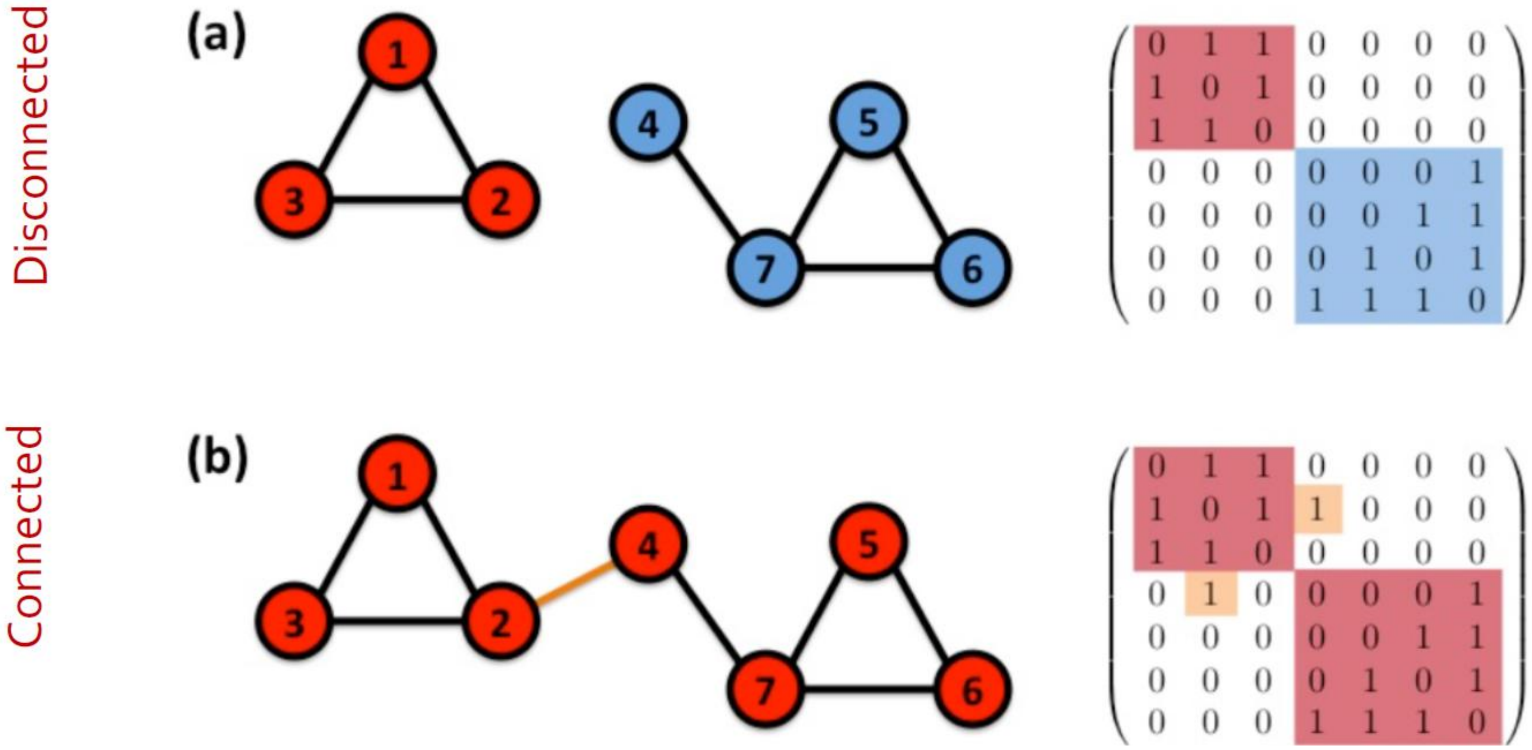
Largest Component:
Giant Component

Isolated node (node H)

- **Bridge edge:** If we erase the **edge**, the graph becomes disconnected
- **Articulation node:** If we erase the **node**, the graph becomes disconnected

Connectivity Example

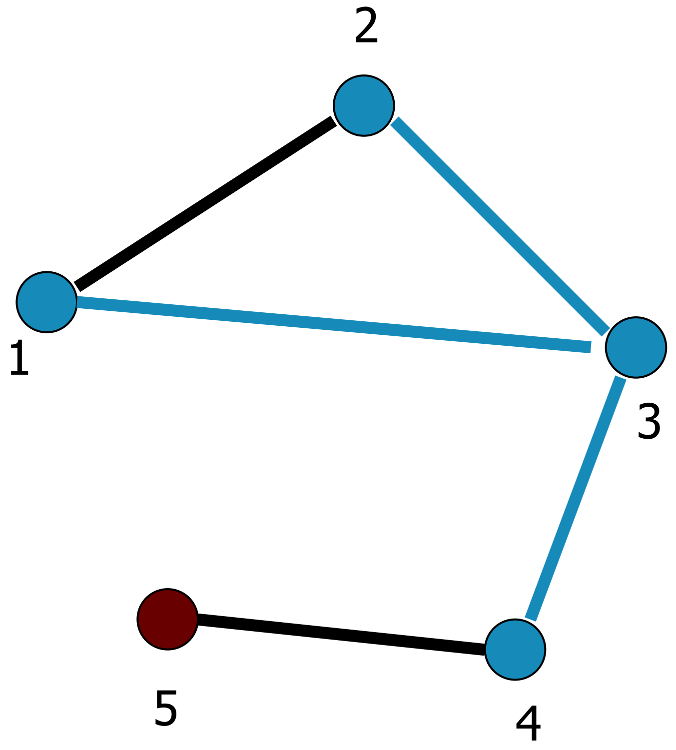
- The adjacency matrix of a network with several components can be written in a block- diagonal form, so that nonzero elements are confined to squares, with all other elements being zero:



Subgraphs

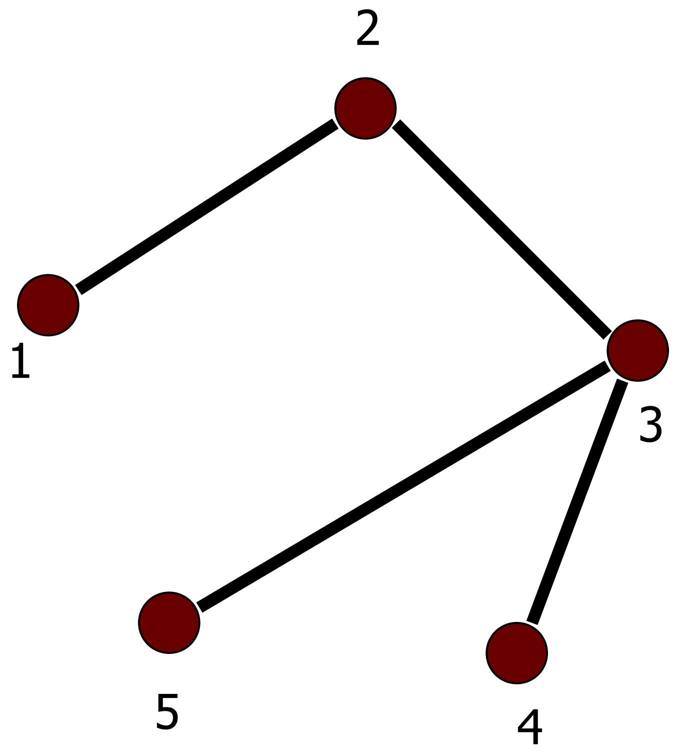
- Subgraph: Given V' $\subseteq$ V, and E' $\subseteq$ E, the graph G'=(V',E') is a subgraph of G.

- Induced subgraph: Given V' $\subseteq$ V, let E' $\subseteq$ E is the set of all edges between the nodes in V'. The graph G'=(V',E'), is an induced subgraph of G



Trees

- Connected Undirected graphs without cycles



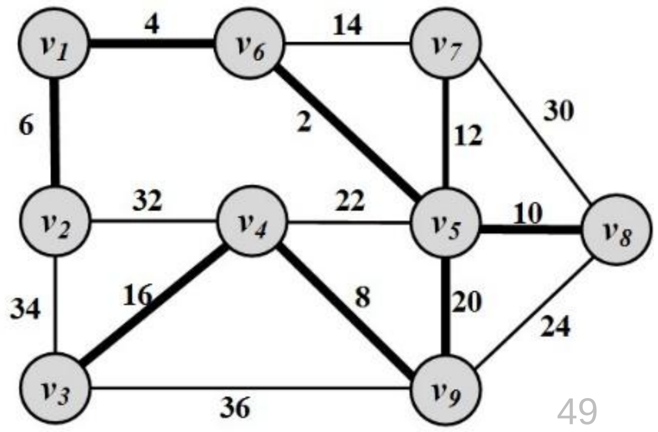
Trees Properties

- Edges and Vertices Relationship: A tree with n vertices has exactly n-1 edges.
- Unique Path: There is a unique path between any two vertices in a tree.
- All Edges Are Bridges: In a tree, every edge is a bridge; removing any edge will disconnect the graph.
- At Least Two Leaves: Every tree with at least two vertices has at least two vertices of degree one, known as leaves.

Spanning Tree

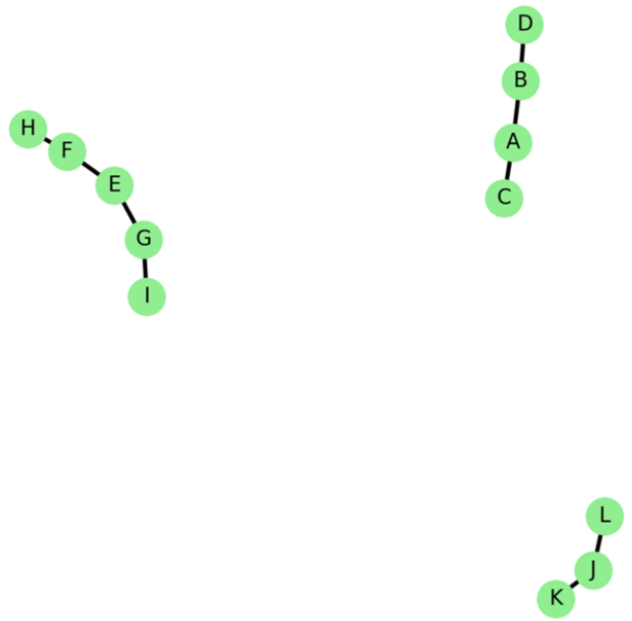
- For any connected graph, the spanning tree is a subgraph and a tree that includes all the nodes of the graph
- There may exist multiple spanning trees for a graph.
- For a weighted graph and one of its spanning tree, the weight of that spanning tree is the summation of the edge weights in the tree.
- Among the many spanning trees found for a weighted graph, the one with the minimum weight

  is called the minimum spanning tree (MST)



Forest

- A simple, undirected graph with no cycles. It consists of a collection of disjoint trees, where each connected component is a tree.

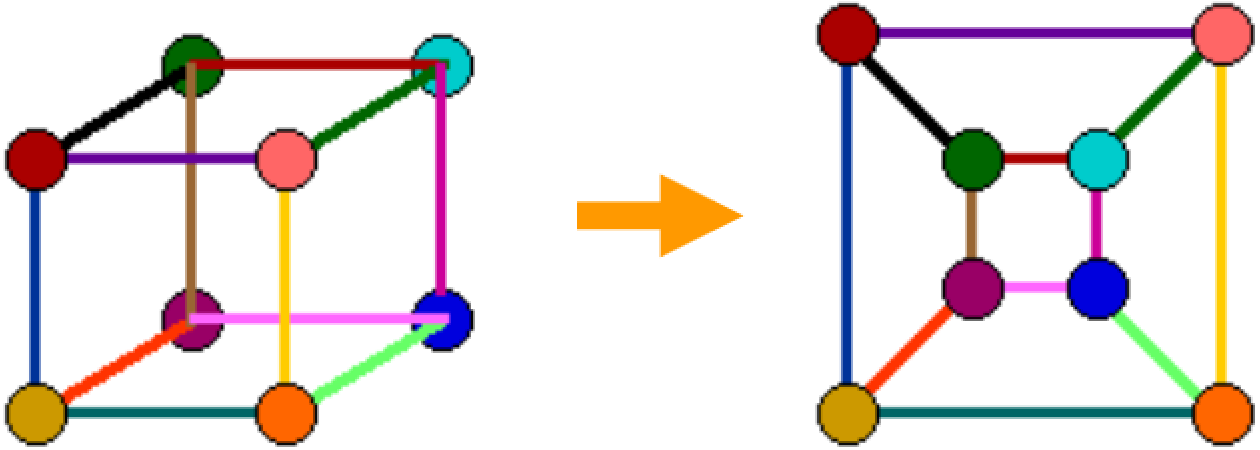


Forest Properties

- Number of Edges: A forest with n vertices and k connected components has exactly n - k edges. This is because each tree with m vertices contains m - 1 edges; thus, the total number of edges in the forest is the sum of the edges in all its trees.
- Acyclic Nature: Forests contain no cycles; consequently, every connected subgraph within a forest is also acyclic.
- Connected Components: Each connected component in a forest is a tree; therefore, a forest can be viewed as a collection of separate trees.

Planar Graphs

- A graph is planar if it can be drawn on a plane without any edges crossing.



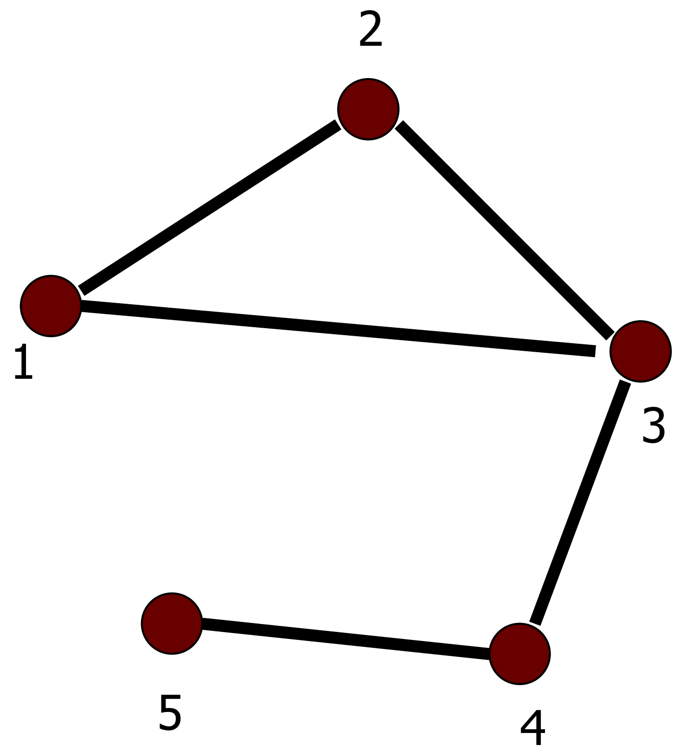
04

Graph Representation

Graph Representation

- Adjacency Matrix
  - symmetric matrix for undirected graphs

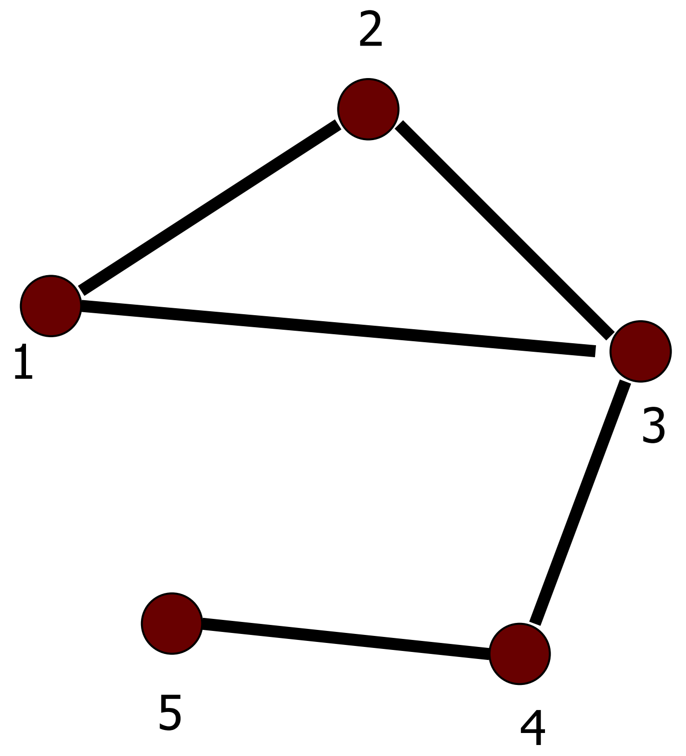
$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$



Graph Representation

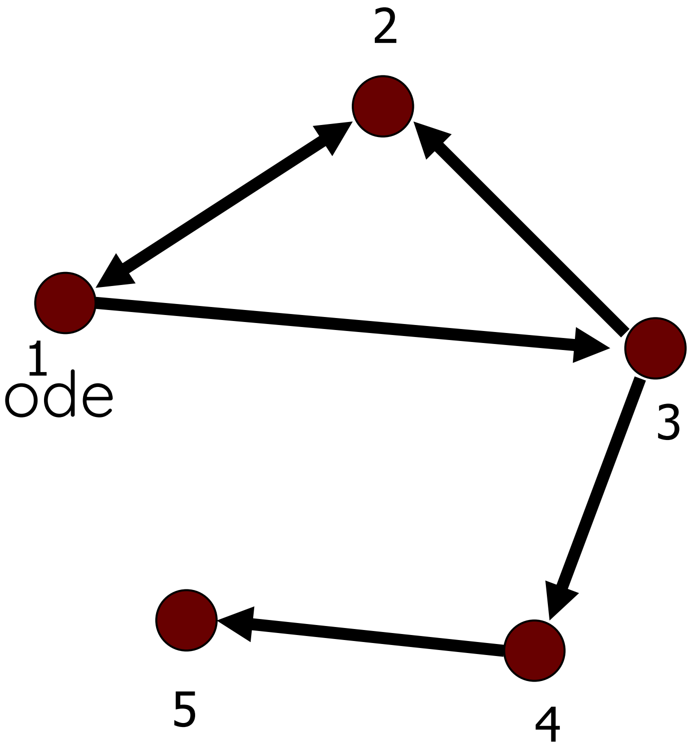
- Adjacency List
  - For each node keep a list with neighboring nodes

1: [2, 3]
2: [1, 3]
3: [1, 2, 4]
4: [3, 5]
5: [4]



Graph Representation

- Adjacency List
  - For each node keep a list of the nodes it points to
  - Easier to work with if network is
    - Large
    - Sparse
  - Allows us to quickly retrieve all neighbors of a given node

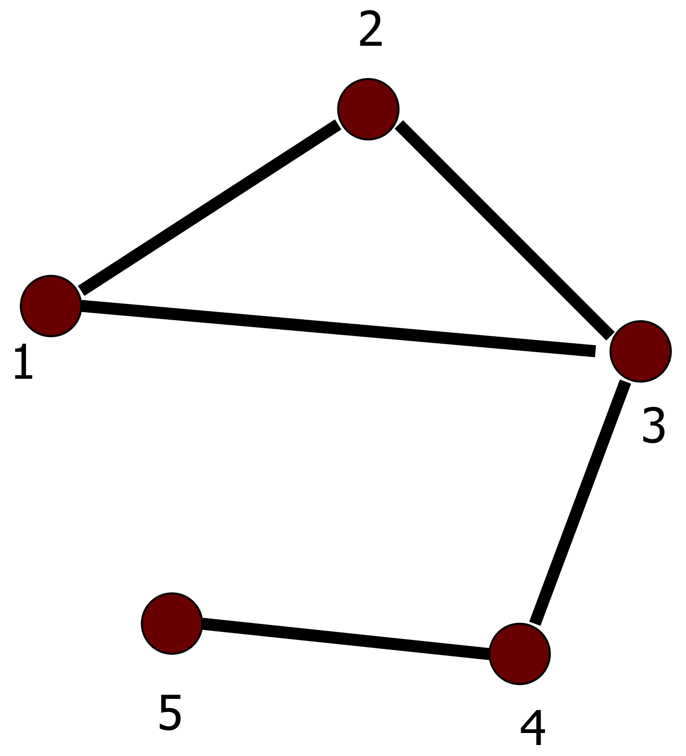


1: [2, 3]
2: [1]
3: [2, 4]
4: [5]
5: [null]

Graph Representation

- List of Edges
  - Keep a list of all the edges in the graph

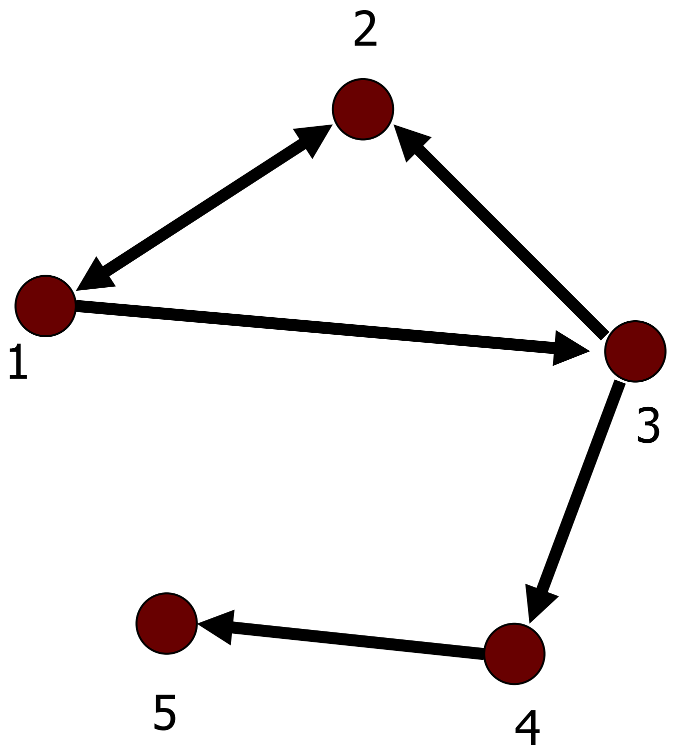
(1,2)
(2,3)
(1,3)
(3,4)
(4,5)



Graph Representation

- List of Edges
  - Keep a list of all the directed edges in the graph

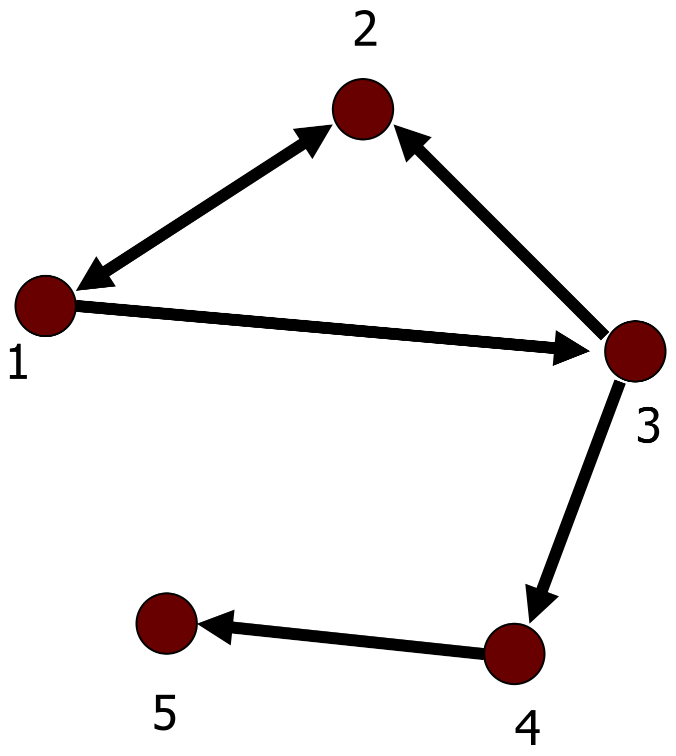
(1,2)
(2,1)
(1,3)
(3,2)
(3,4)
(4,5)



Graph Representation

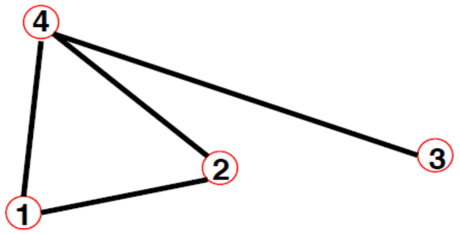
- Adjacency Matrix
  - unsymmetric matrix for undirected graphs

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



Adjacency Matrix

Undirected



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

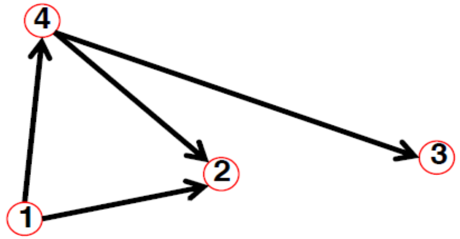
$$A_{ij} = A_{ji}$$
$$A_{ii} = 0$$

$$k_i = \sum_{j=1}^{N} A_{ij}$$

$$k_j = \sum_{i=1}^{N} A_{ij}$$

$$L = \frac{1}{2}\sum_{i=1}^{N} k_i = \frac{1}{2}\sum_{ij}^{N} A_{ij}$$

Directed



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

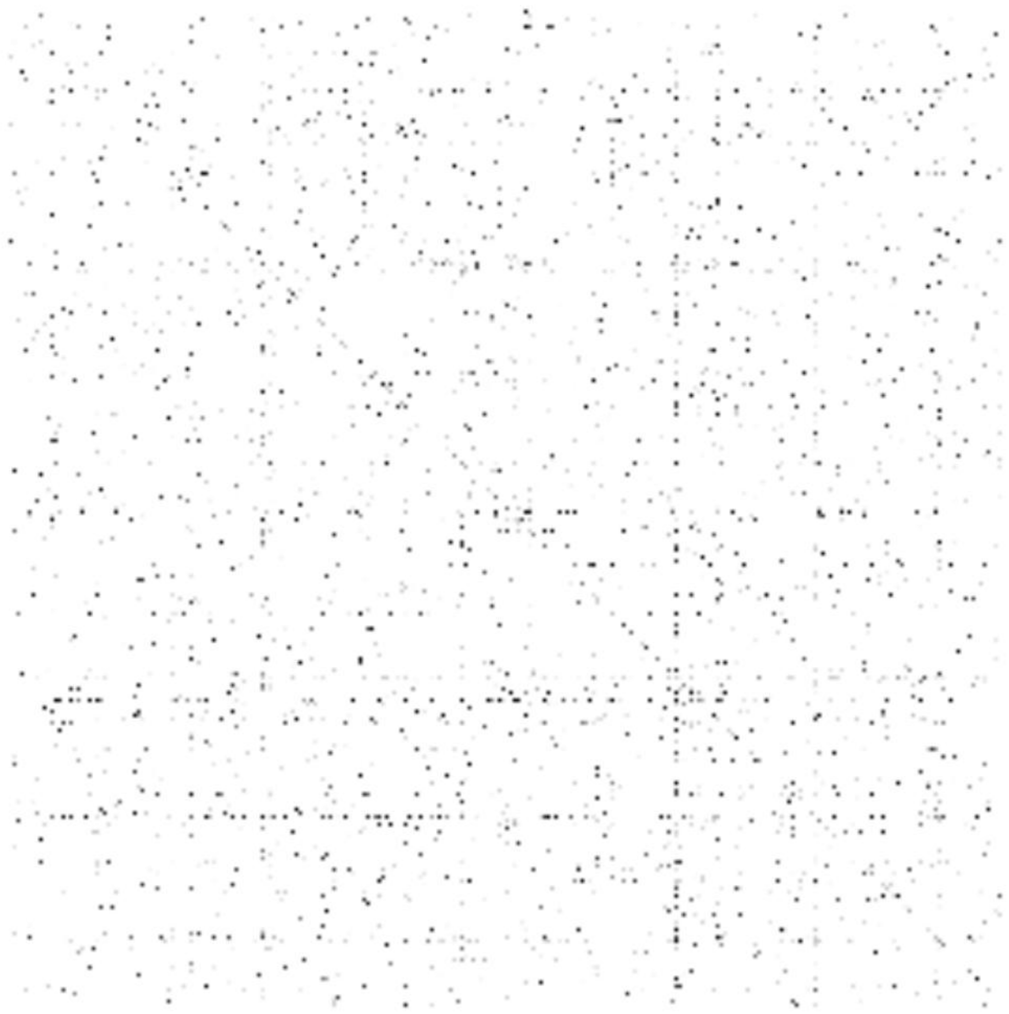
$$A_{ij} \neq A_{ji}$$
$$A_{ii} = 0$$

$$k_i^{out} = \sum_{j=1}^{N} A_{ij}$$

$$k_j^{in} = \sum_{i=1}^{N} A_{ij}$$

$$L = \sum_{i=1}^{N} k_i^{in} = \sum_{j=1}^{N} k_j^{out} = \sum_{i,j}^{N} A_{ij}$$

Adjacency Matrices are Sparse



Networks are Sparse Graphs

Most real-world networks are sparse

$$E \ll E_{max} \quad (\text{or } \bar{k} \ll N-1)$$

WWW (Stanford-Berkeley):	$N=319{,}717$	$\langle k\rangle=9.65$
Social networks (LinkedIn):	$N=6{,}946{,}668$	$\langle k\rangle=8.87$
Communication (MSN IM):	$N=242{,}720{,}596$	$\langle k\rangle=11.1$
Coauthorships (DBLP):	$N=317{,}080$	$\langle k\rangle=6.62$
Internet (AS-Skitter):	$N=1{,}719{,}037$	$\langle k\rangle=14.91$
Roads (California):	$N=1{,}957{,}027$	$\langle k\rangle=2.82$
Proteins (S. Cerevisiae):	$N=1{,}870$	$\langle k\rangle=2.39$

(Source: *Leskovec et al., Internet Mathematics, 2009*)

Consequence: Adjacency matrix is filled with zeros!

(**Density of the matrix (E/N^2):** WWW$=1.51\times10^{-5}$, MSN IM $= 2.27\times10^{-8}$)

Network Representations

Email network >> directed multigraph with self-edges

Facebook friendships >> undirected, unweighted

Citation networks >> unweighted, directed, acyclic

Collaboration networks >> undirected multigraph or weighted graph

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

Protein Interactions >> undirected, unweighted with self-interactions

Any Question?