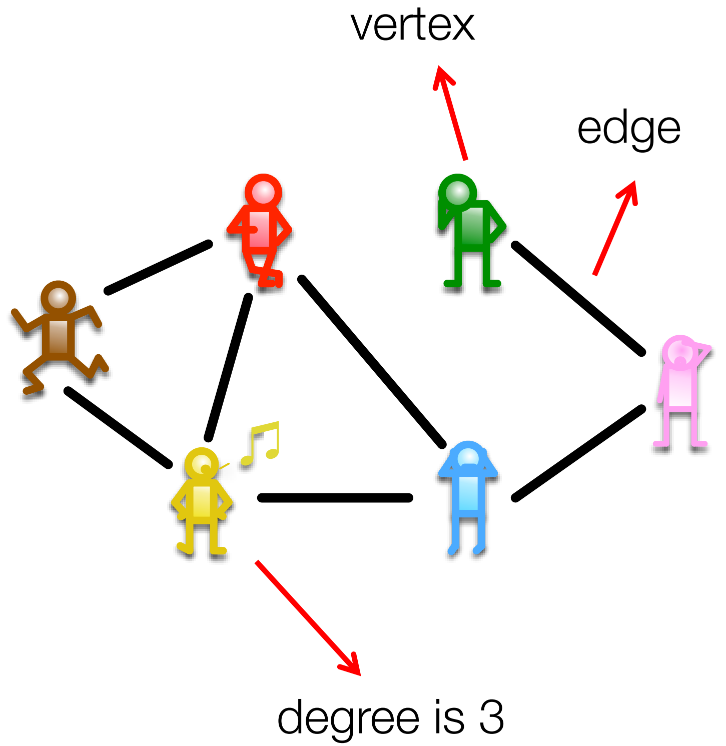
CSE 610: Special Topics in Social and Information Network Analysis

A. Erdem Sariyuce

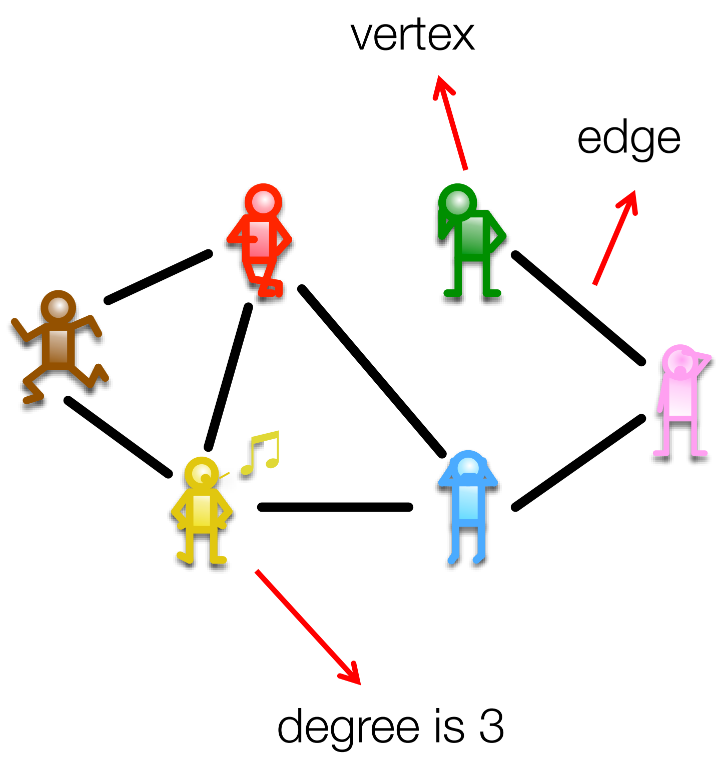
What is “Network Science”?

- Study of complex networks (complex means non-trivial)



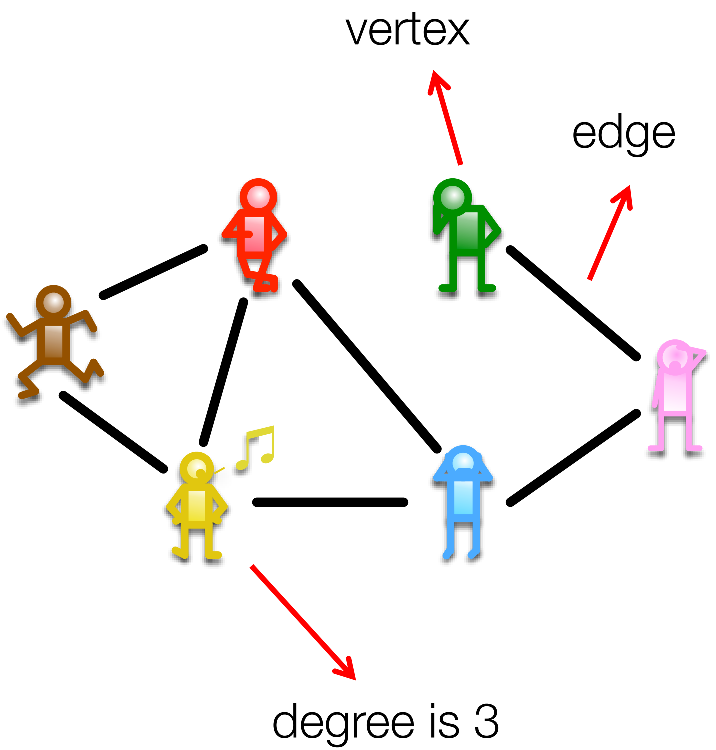
What is "Network Science"?

- Study of complex networks (complex means non-trivial)
  - Social networks
  - Information networks
  - Web networks
  - Telecommunication networks
  - Computer networks
  - Biological networks
  - Cognitive and semantic networks



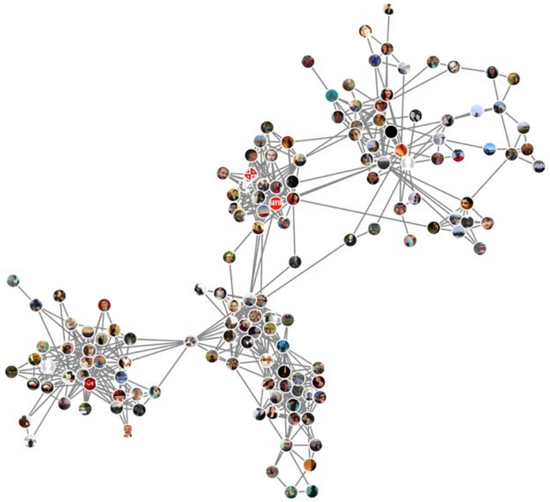
What is "Network Science"?

- Study of complex networks (complex means non-trivial)
  - Social networks
  - Information networks
  - Web networks
  - Telecommunication networks
  - Computer networks
  - Biological networks
  - Cognitive and semantic networks
- Distinct entities: Nodes (or vertices)
- Connections: Links (or edges)

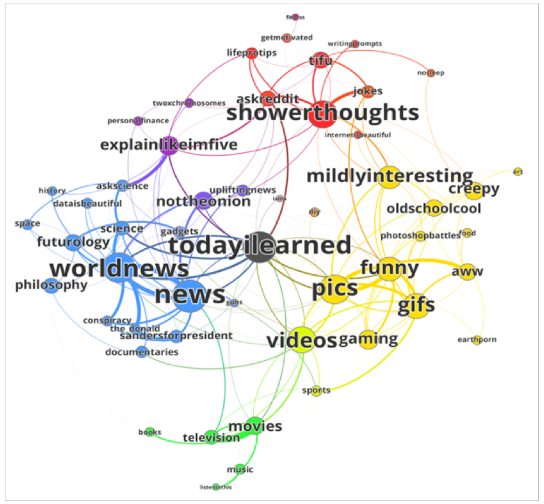


Graphs (networks) are everywhere

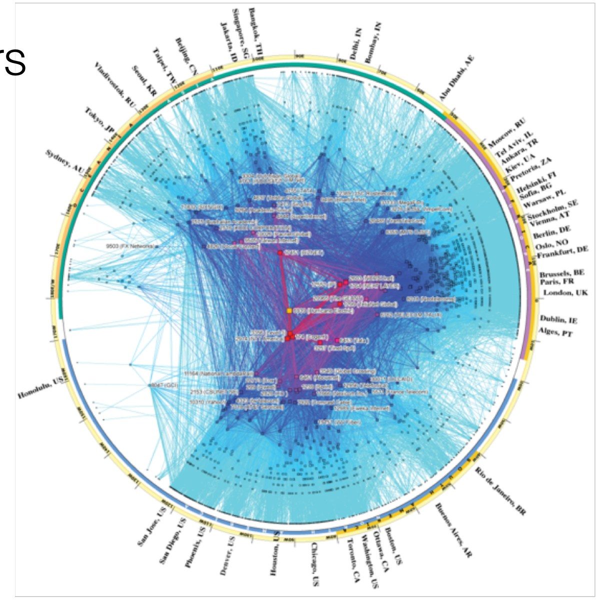
Social



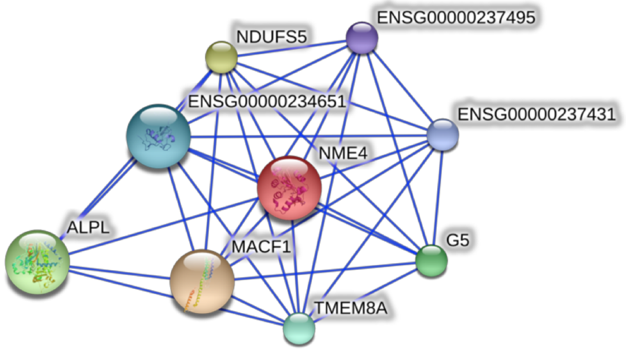
Information



Routers



Protein-interaction



Transformers of disciplines

- Graph theory
  - Mathematics
- Statistical mechanics
  - Physics
- Data mining
  - Computer science
- Inferential modeling
  - Statistics
- Social structure
  - Sociology

Questions we ask

- Real-world networks
  - What characteristics do we observe?

- Nodes and edges
  - Which ones are more important than others?
  - What does important mean in this context?

Questions we ask

- Given a person in a social network;
    - How do we determine her social circles?
    - How do we suggest new friends?
    - Can we infer her spouse? (there is a paper on that)

- Or given a webpage about booking flights,
    - How likely it will be accessed next week?
    - How can we make it appear on top in search results?

Questions we ask

- Graph algorithms
  - What is used in the state-of-the-art search engines?
  - What about recommendation systems?

- How does a network evolve over time?
  - Which nodes will get more edges?
  - Which edges will be removed or added?

Logistics

- Class hours: MW 3:30-4:50 @ Fronzcak 422

- Office: 323 Davis Hall
- Office hours: T 10-12

- erdem@buffalo.edu
- Website: http://sariyuce.com/F19.html
- Piazza page

This class is not hard

- No prerequisite needed
    - Background in graph theory, discrete math

- No textbook required. Will benefit from
    - Networks: An Introduction
        - By M. Newman
    - Networks, Crowds and Markets (has a webpage)
        - By D. Easley and J. Kleinberg

Lectures and papers

- Papers will be pointed for advanced topics
    - Community Detection, Temporal Networks …
- Great papers from the top venues!
    - Science, Nature
    - SIGKDD, WWW, WSDM, ICDM, SDM
    - VLDB, SIGMOD, ICDE

Grading

- Homeworks
  - 4 x 10%
- Midterm
  - 20% (In class)
- Project
  - 40%

Homeworks

- Combination of data analysis, algorithm design, and math
  - Analysis and discussions by charts, tables
  - Require light coding to automate things

- Due in one week
- Individually

- All homeworks should be typed!
  - Email it before the class on the due date

Project

- Proposal by 2nd week
  - Report (1p): 3%
  - Presentation (5m): 2%
- Progress by 8th week
  - Report (≤9p): 8%
  - Presentation (5m): 7%
- Final by last week
  - Report (≤9p): 10%
  - Presentation (15m): 10%

- Can do in pairs
  - Don't have to
- Ideas will be provided
  - Don't worry, I'll guide
- We have to meet every other week, at least
  - Office hours! Tue 10-12
- **We aim to publish papers!**

Academic integrity

- Don’t cheat in homeworks, please, really easy to detect!
- New university policy this year!
  - https://grad.buffalo.edu/succeed/current-students/policy-library.academics.html#grievanceandintegrity
- Zero tolerance: Failure in the course for first attempt
- Grads: Sanctions can even reach to RA/TA cancellation

Questions?

Project Ideas

- Repeatability experiments for some popular papers
  - And extensions
- Surveys on certain hot topics
  - With a codebase for comparison
- Any idea you may want to go for!
  - Consultation with instructor

Project Ideas

- Graph summarization by tree hierarchy
  - VLDB tutorial

- Finding dense regions in weighted networks
  - Driven by weighted motifs

- Speeding up fundamental graph computations
  - Triangle counting
  - Peeling with bucket structure

Project Ideas

- Collective similarity measurements in bipartite graphs
  - Beyond pairwise, related to distance metrics
- Conductance measurement
  - State-of-the-art dense subgraph discovery algorithms
- Core-periphery structure in non-traditional graphs
  - Bipartite graphs
  - Weighted graphs

Schedule

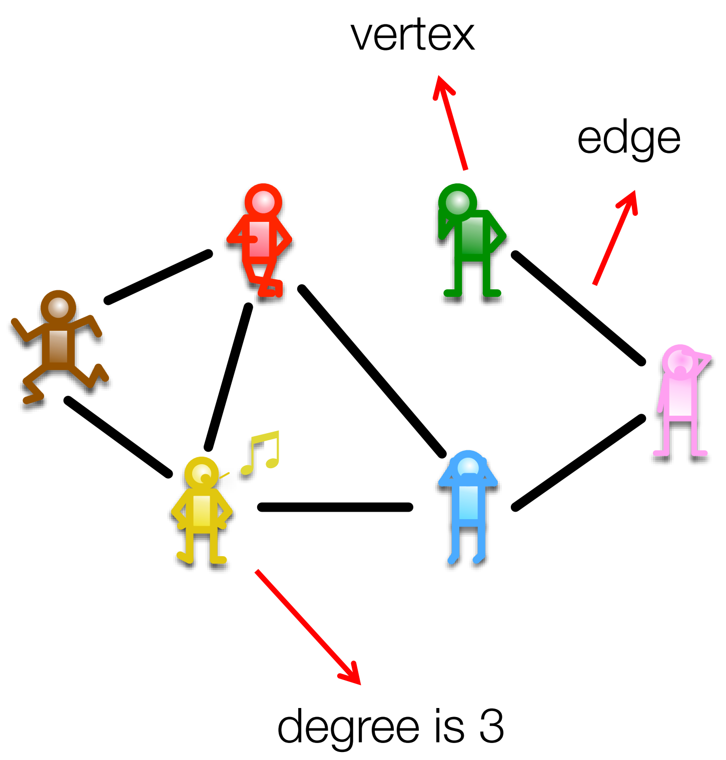
9/4: Introduction, Course Overview, Graph Theory	10/21: Network Motifs 10/23: **Progress Presentations** **(Progress Report due)**
9/9: Link analysis 9/11: PageRank, **Proposal Presentations** **(Prop. Rep. due)**	10/28: NO CLASS (Instructor on travel) 10/30: Non-traditional and heterogeneous networks
9/16: Graph Traversal **(HW 1 out)** 9/18: Maximum Flow	11/4: Temporal Graphs I 11/6: Temporal Graphs II **(HW 3 out)**
9/23: Shortest Path **(HW 1 due)** 9/25: Network Centrality	11/11: Streaming graph algorithms I 11/13: Streaming graph algorithms II **(HW 3 due)**
9/30: Community Detection I 10/2: Community Detection II	11/18: Machine Learning on Networks 11/20: Graph embeddings **(HW 4 out)**
10/7: Dense Subgraph Discovery **10/9: Midterm Exam** **(HW 2 out)**	11/25: Parallel Graph Analytics 11/27: Parallel Graph Frameworks I **(HW 4 due)**
10/14: Graph partitioning I 10/16: Graph partitioning II **(HW 2 due)**	12/2: Parallel Graph Frameworks II 12/4: **Final Presentations** **(Final Report due)**

Review of Graph Theory

Representing Networks

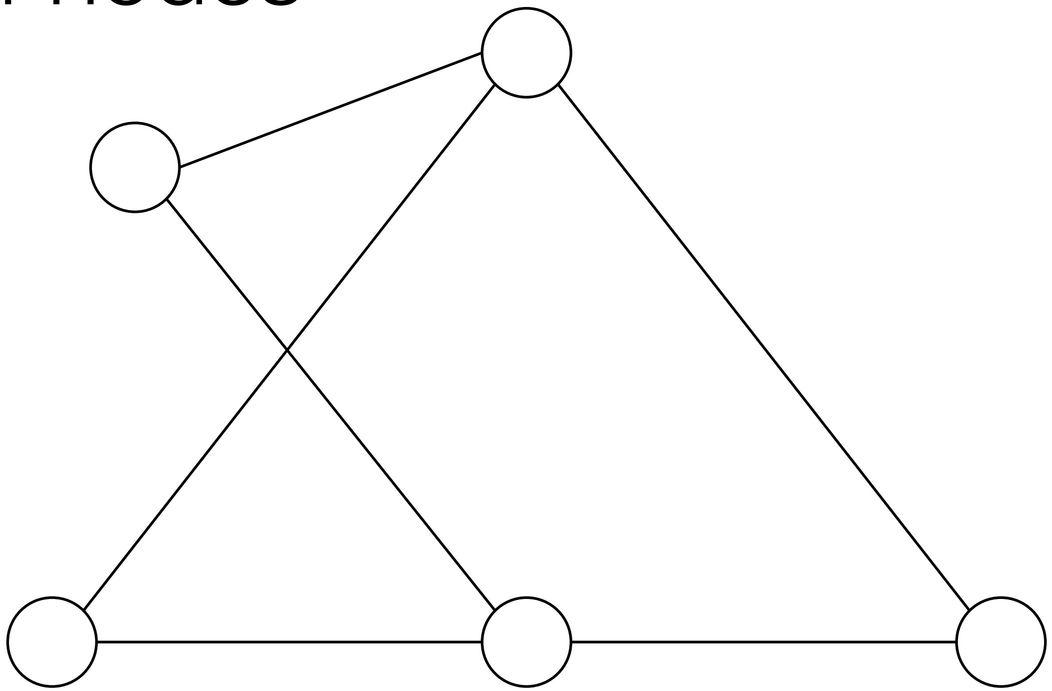
- Distinct entities: Nodes (or vertices)
- Connections: Links (or edges)

Network	Vertex	Edge
Internet	Computer or router	Cable or wireless data connection
World Wide Web	Web page	Hyperlink
Citation network	Article, patent, or legal case	Citation
Power grid	Generating station or substation	Transmission line
Friendship network	Person	Friendship
Metabolic network	Metabolite	Metabolic reaction
Neural network	Neuron	Synapse
Food web	Species	Predation



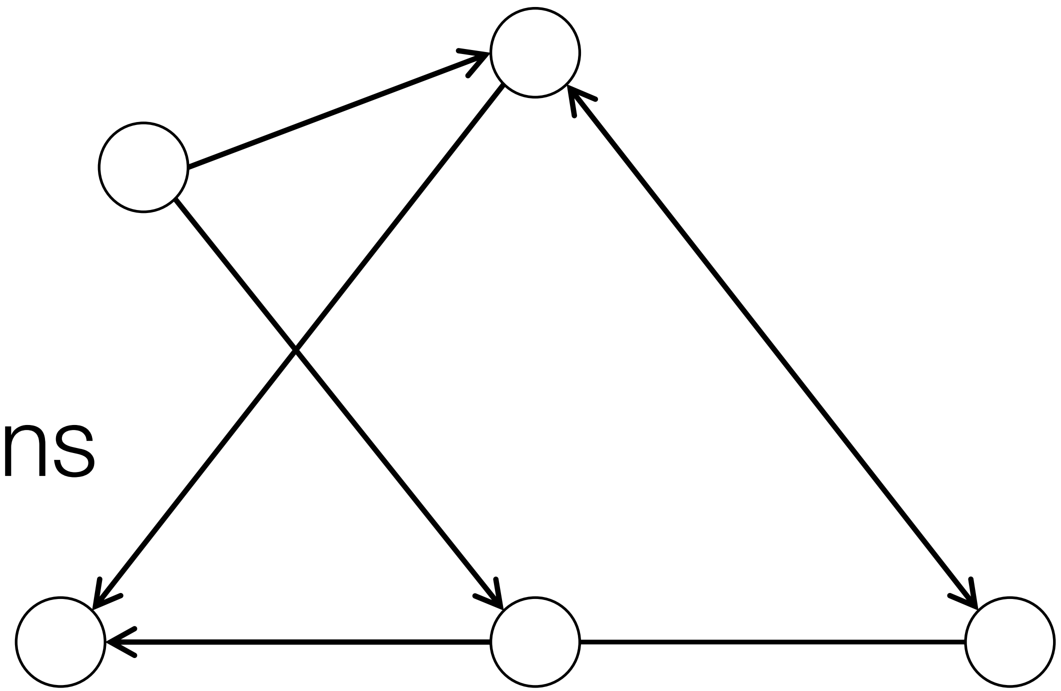
Simple graph is just nodes and edges

- Dots and lines
- Single edge between any pair of nodes
- Degree of a vertex
  - Number of edges it is connected
- No other information



Directed graph has directions on edges

- Edge is from a source vertex to target vertex
- In-degree of a vertex
    - Number of in-coming edges
- Out-degree of a vertex
    - Number of out-going edges
- An edge can be in both directions
    - Regarded as undirected

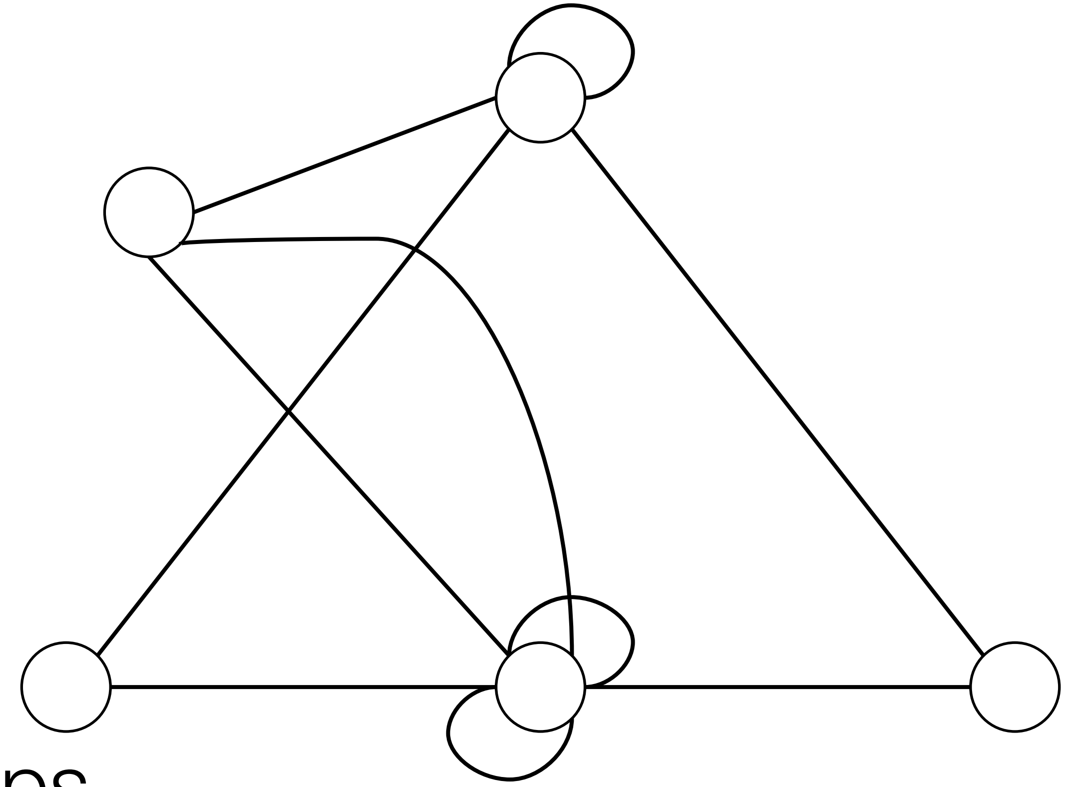


Multi-edges and self-loops

- Multiple edges between two vertices
  - Multi-edge
  - Example?

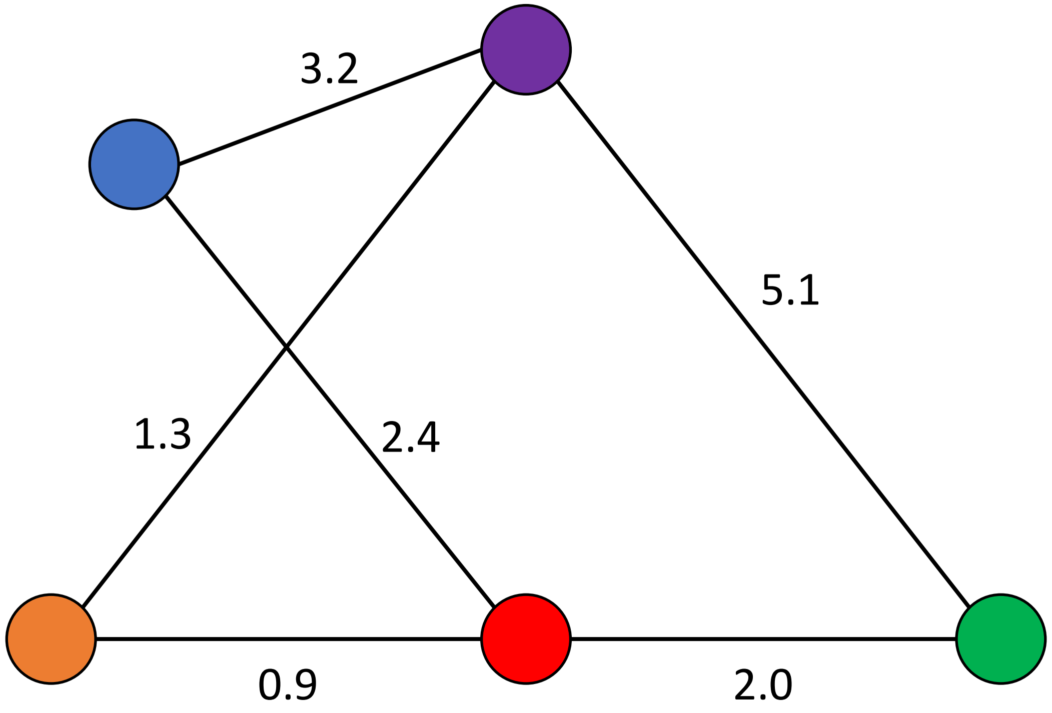
- Self-loops
  - A vertex is connected to itself

- Any combination is possible
  - Directed multiedges with self-loops



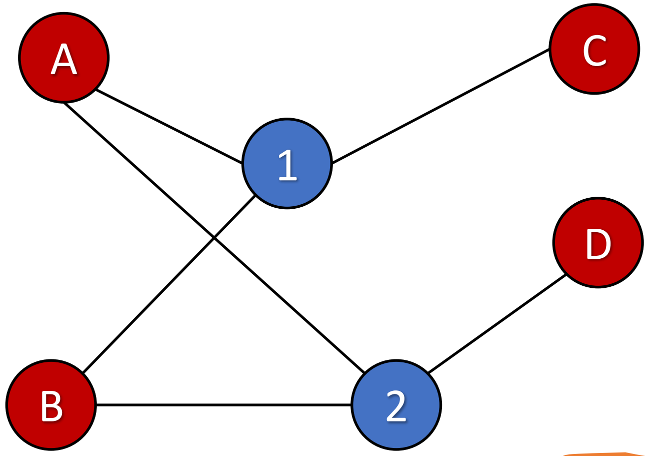
Attributes on vertices and edges

- Labels
  - Binary or multiple classes
  - Multiple labels
- Scalar values
  - Integer or fraction
- Can be combined with others
- Algorithms get more complex

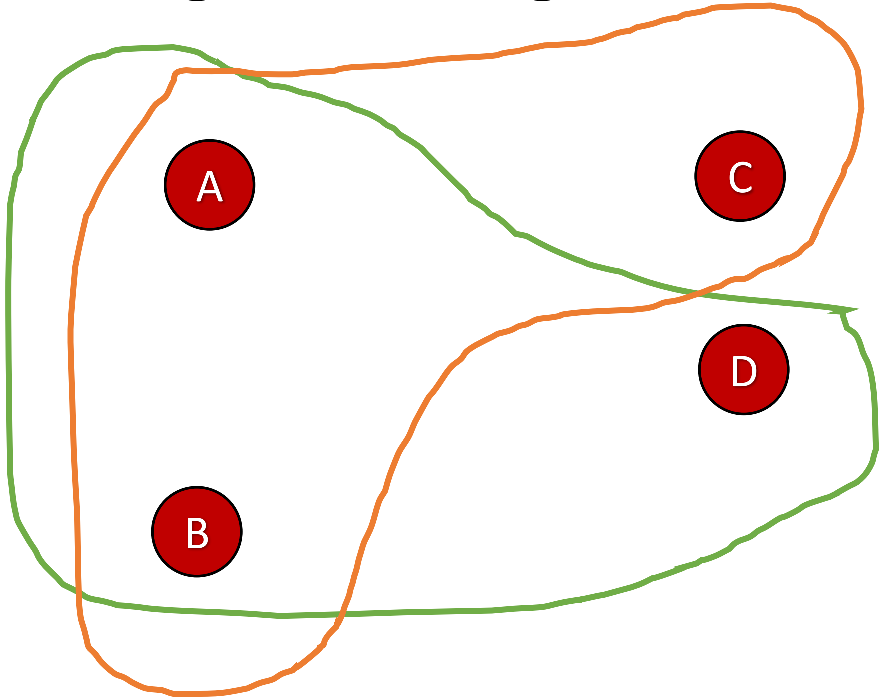


Hypergraphs (or bipartite graphs)

- Not all relations are pairwise
  - Edges with any number of vertices
  - Hyperedge
  - Group relations

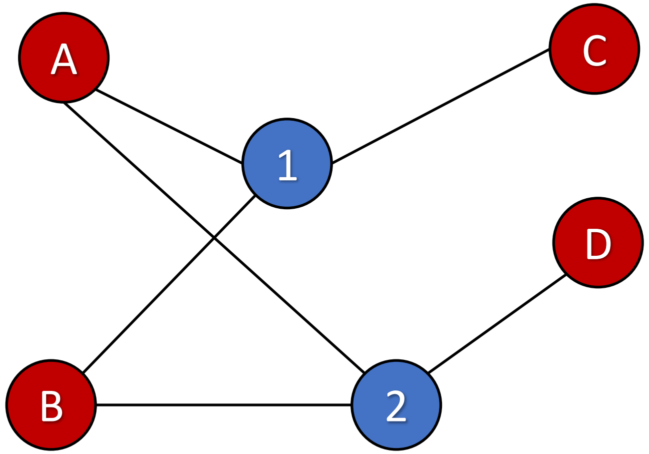


Network	Vertex	Group
Film actors	Actor	Cast of a film
Coauthorship	Author	Authors of an article
Boards of directors	Director	Board of a company
Social events	People	Participants at social event
Recommender system	People	Those who like a book, film, etc.
Keyword index	Keywords	Pages where words appear
Rail connections	Stations	Train routes
Metabolic reactions	Metabolites	Participants in a reaction

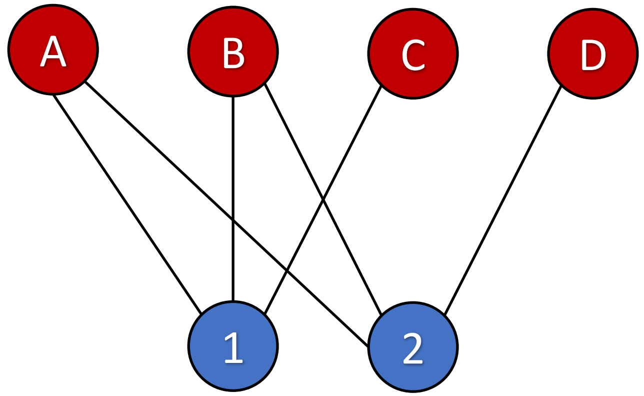


Hypergraphs (or bipartite graphs)

- Affiliation networks (two-mode)
- Bipartite graph
  - Two set of vertices
  - Edges only across sets

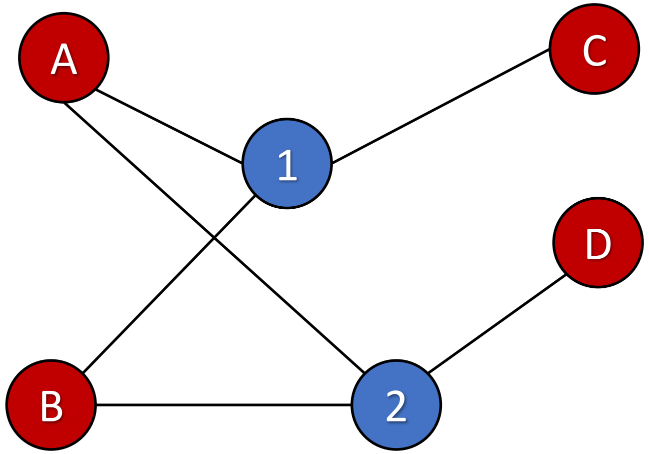


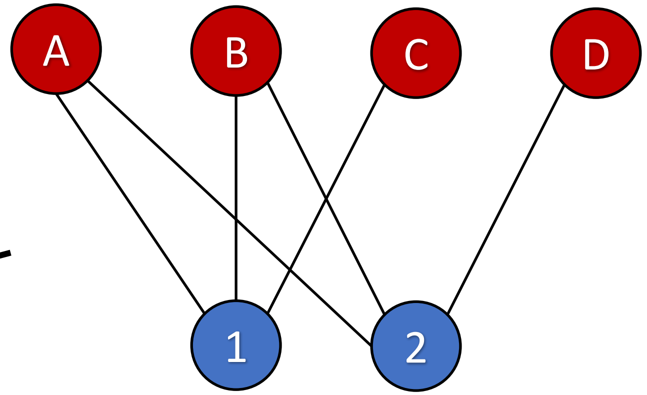
Network	Vertex	Group
Film actors	Actor	Cast of a film
Coauthorship	Author	Authors of an article
Boards of directors	Director	Board of a company
Social events	People	Participants at social event
Recommender system	People	Those who like a book, film, etc.
Keyword index	Keywords	Pages where words appear
Rail connections	Stations	Train routes
Metabolic reactions	Metabolites	Participants in a reaction

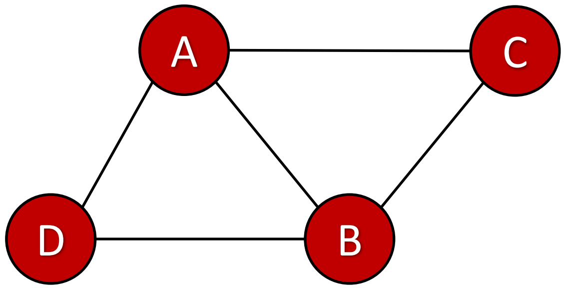


Hypergraphs (or bipartite graphs)

- Affiliation networks (two-mode)
- Bipartite graph
  - Two set of vertices
  - Edges only across sets
- Projection to one-mode
  - Sharing affiliations
  - Author-paper → Co-authorship

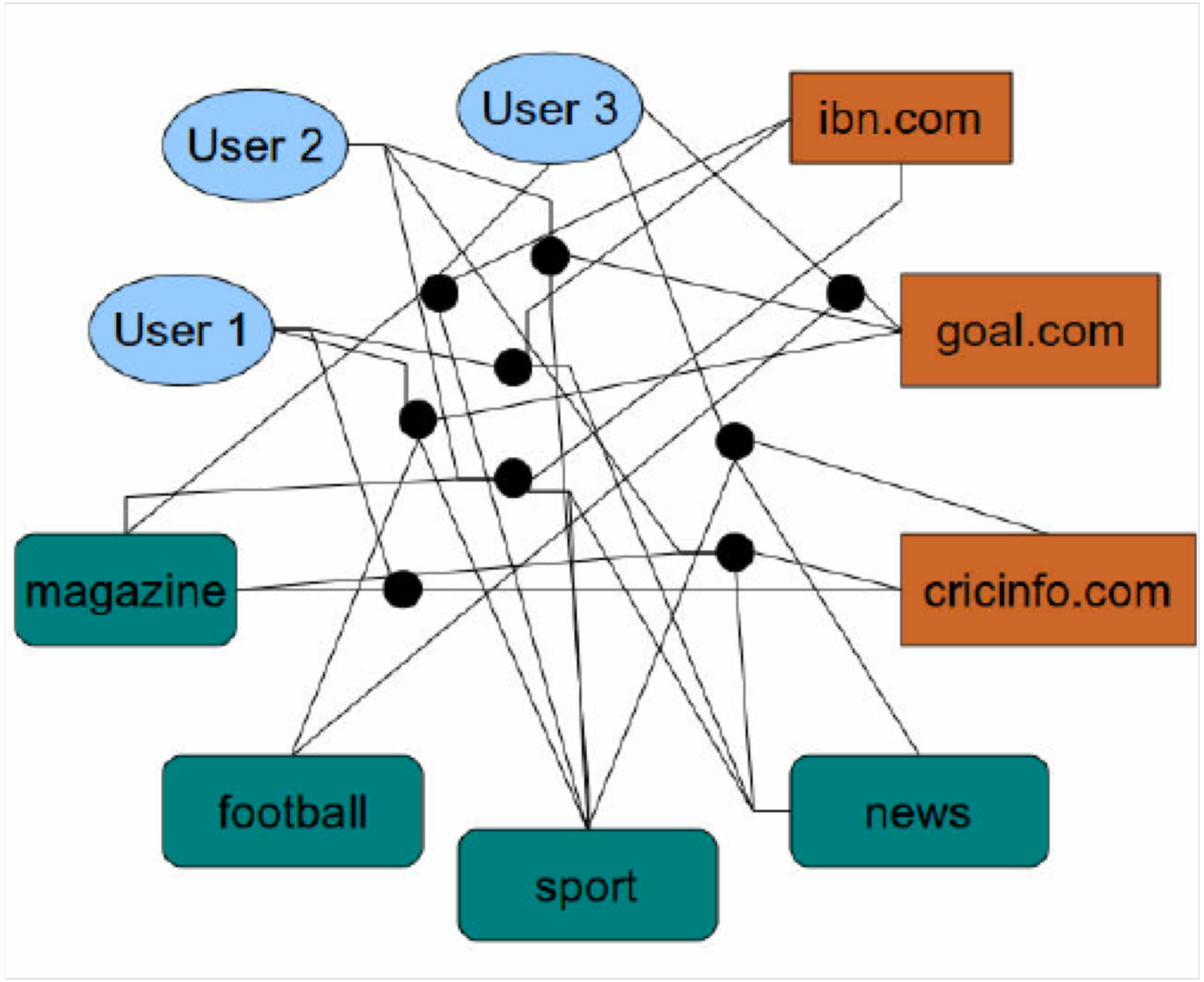






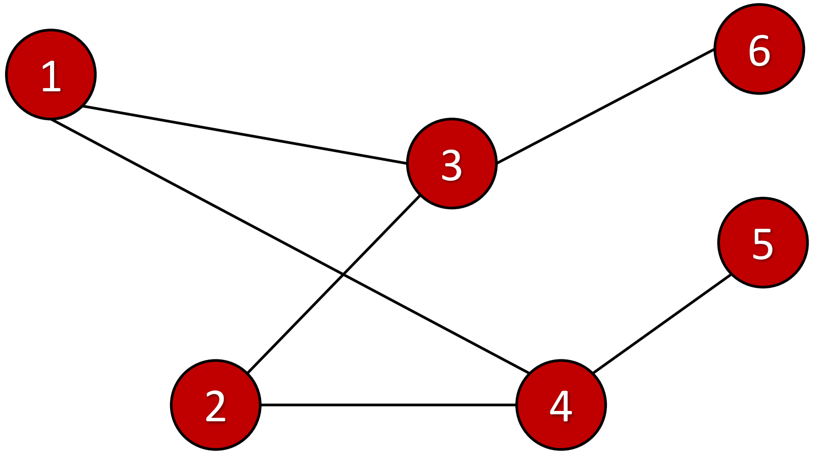
k-partite graphs

- k partitions
- No edge among nodes in the same partition
- Folksonomy data
  - Users put tags to posts
    - Users
    - Tags
    - Posts
  - 3-partite



Graph is a matrix

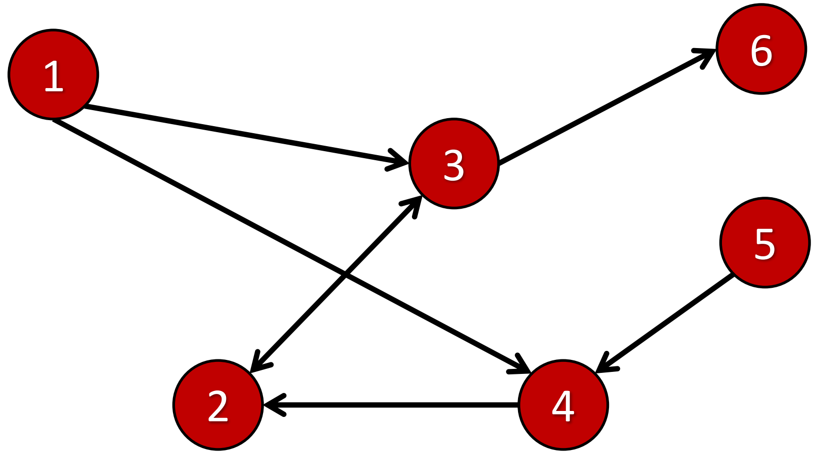
- Adjacency matrix
- For n vertices
  - Square matrix: n x n
  - Undirected -> binary, symmetric



0	0	1	1	0	0
0	0	1	1	0	0
1	1	0	0	0	1
1	1	0	0	1	0
0	0	0	1	0	0
0	0	1	0	0	0

Graph is a matrix

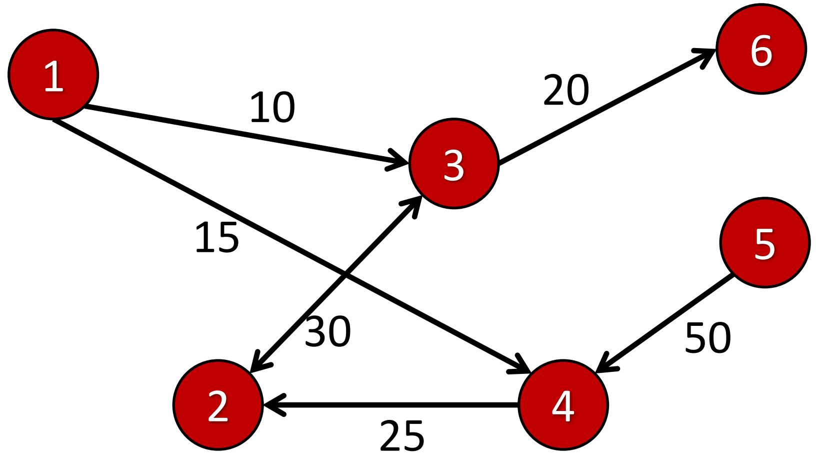
- Adjacency matrix
- For n vertices
  - Square matrix: n x n
  - Undirected -> binary, symmetric
  - Directed -> binary, not symmetric



0	0	1	1	0	0
0	0	1	0	0	0
0	1	0	0	0	1
0	1	0	0	0	0
0	0	0	1	0	0
0	0	0	0	0	0

Graph is a matrix

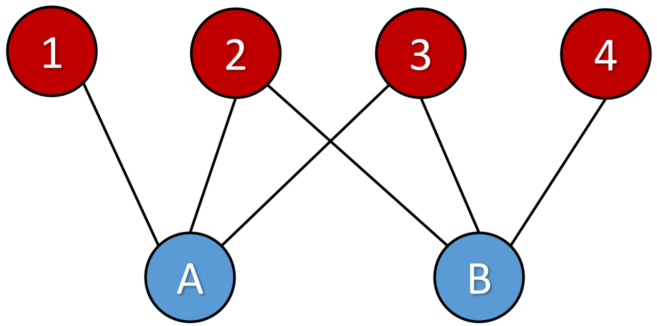
- Adjacency matrix
- For n vertices
  - Square matrix: n x n
  - Undirected -> binary, symmetric
  - Directed -> binary, not symmetric
  - Weighted -> not binary



0	0	10	15	0	0
0	0	30	0	0	0
0	30	0	0	0	20
0	25	0	0	0	0
0	0	0	50	0	0
0	0	0	0	0	0

Graph is a matrix

- Adjacency matrix
- For bipartite graph
  - n vertices and m hyperedges
  - Rectangle matrix: n x m
    - Rows are vertices
    - Columns are edges

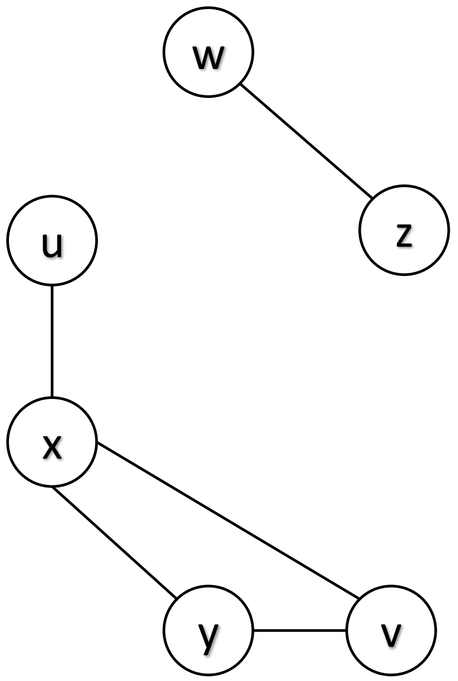


	A	B
1	1	0
2	1	1
3	1	1
4	0	1

- How to store matrices in memory?

Path and connectedness

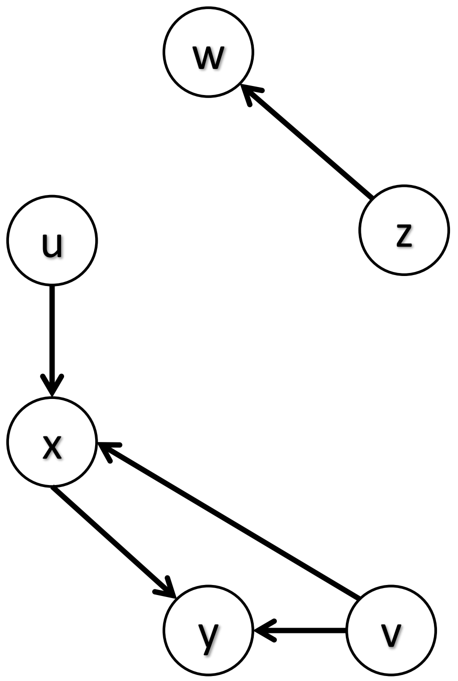
- Path: Sequence of vertices
  - Every consecutive vertex pair is connected
  - Length is the number of edges
- Two vertices are connected if there is a path between them
  - u and v are connected, u and w are not
  - Connected component
    - Maximal set of connected vertices
      - u,x,y,v and w,z
- Geodesic path: Shortest path
  - Path with smallest length: Least number of edges
  - For u,v : u-x-v, for u-w: infinity



Path and connectedness

- Can be directed as well
- Directed path
  - For every consecutive vertex pair u,v:
    - u → v edge should exist

  - No directed path from u to v



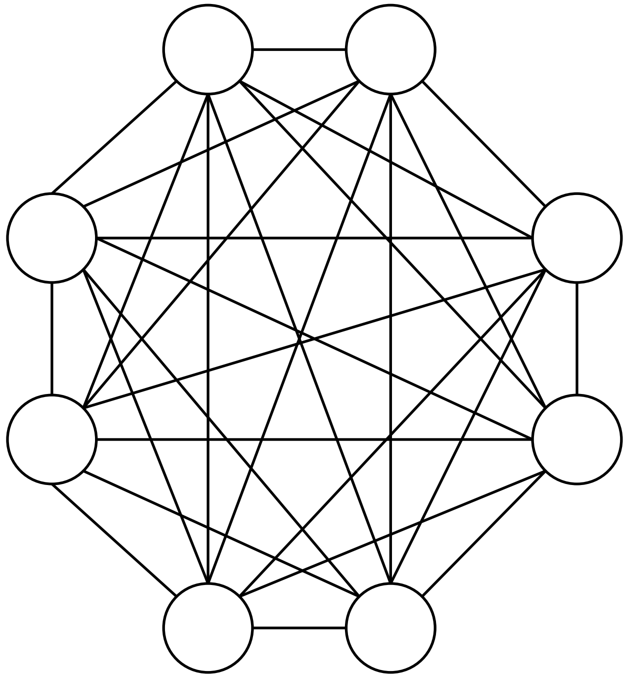
Special graphs

- Clique
- Cycle
- Tree
- DAG
- Planar

- Less complex algorithms
  - And easier analysis

Clique

- Clique: Complete graph
  - All the vertices are directly connected to each other
  - k-clique has k vertices
    - k choose 2 edges

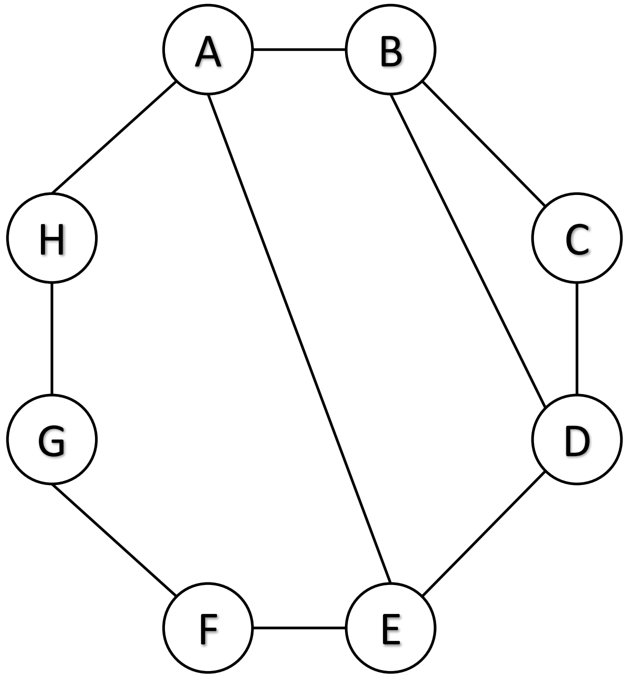


- Maximal-clique
  - Not a part of larger clique

- Maximum-clique
  - Largest clique in the graph
  - NP-Hard

Cycle

- Every vertex can reach itself via some path
    - k-cycle: Cycle with a path of k edges

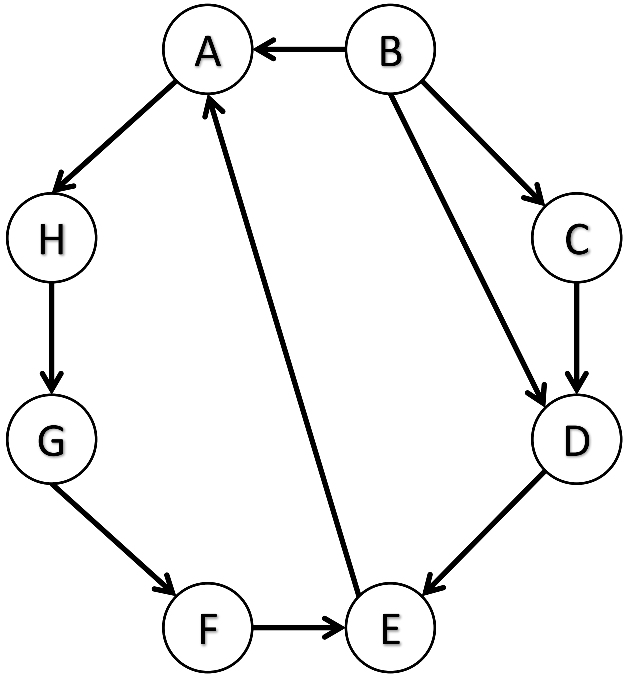


- ABCDEFGH: 8-cycle
    - ABDE: 4-cycle
    - BCD: 3-cycle

Cycle

- Every vertex can reach itself via some path
  - k-cycle: Cycle with a path of k edges

- Can be directed as well
  - Directed path
  - AHGFE is a cycle

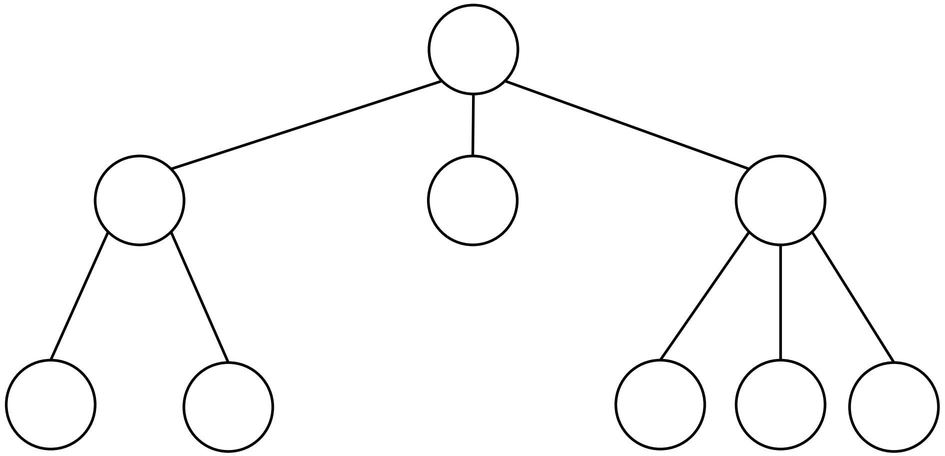


Tree

- Graph with no cycles

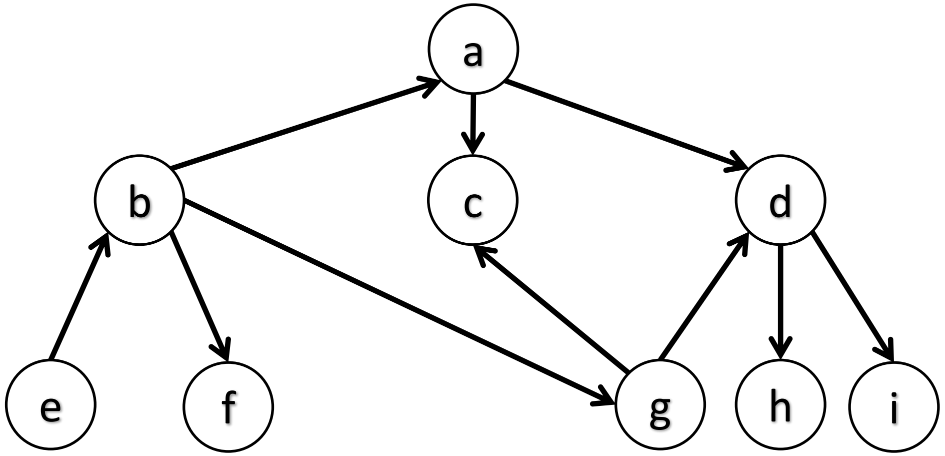
- n vertices, n-1 edges
  - Root
  - Parent
  - Children

- One parent for each vertex, except root



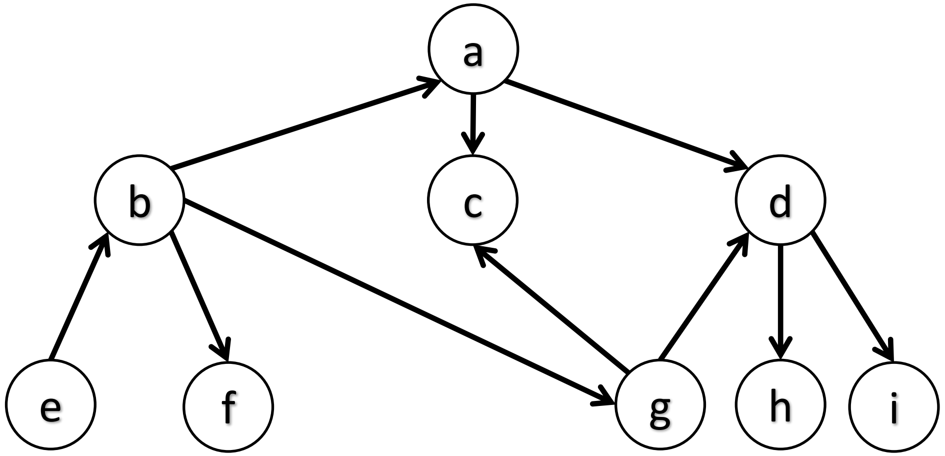
Directed Acyclic Graphs (DAG)

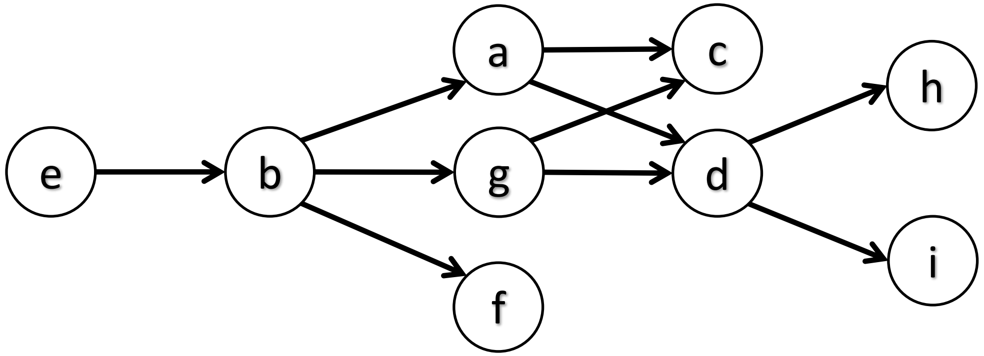
- Graph with no directed cycles
- n vertices
  - How many edges?
- Root?
  - Source and sink
- Citation networks!



Directed Acyclic Graphs (DAG)

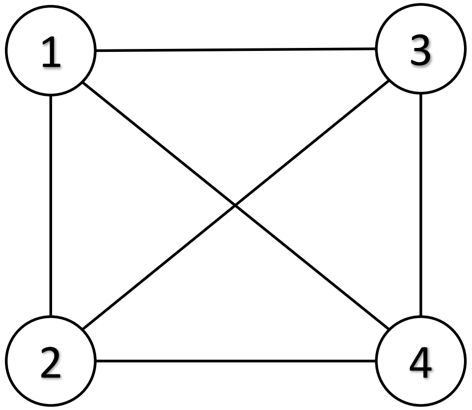
- Graph with no directed cycles
- n vertices
  - How many edges?
- Root?
  - Source and sink
- Citation networks!
  - e
  - b
  - a,g,f
  - c,d
  - h,i





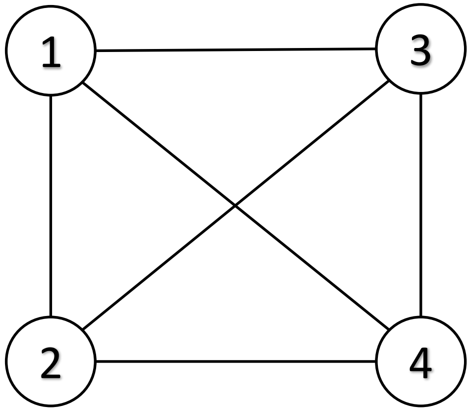
Planar graphs

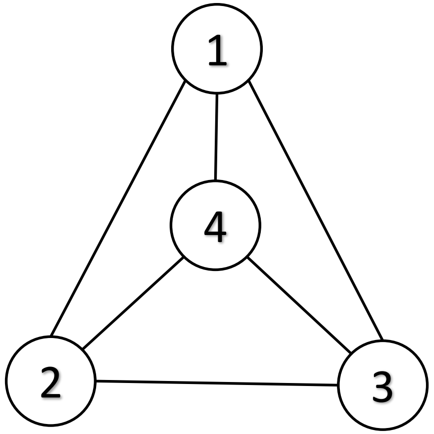
- Can be drawn on a plane with no crossing edges
- Cycles? Trees?
- Triangle?
- Four-clique?



Planar graphs

- Can be drawn on a plane with no crossing edges
- Cycles? Trees?
- Triangle?
- Four-clique?





- Smallest non-planar clique?

- Road networks!