CSE 640: Graph Mining and Management

Lecture 2 (Feb 7)

A. Erdem Sariyuce

Project Ideas

- 1) Surveys on certain hot topics
  - Surveys are key for advancing research
  - With a codebase for comparison
- 2) Specific topics that I'm doing research on
  - I have very specific tasks and execution plans
  - More on this later on
- 3) Repeatability experiments for some popular papers
  - And extensions
- ~~4) Data Cup (or ML) challenges~~
- 5) Any idea you may want to go for!
  - Consultation with instructor

- Promising projects will be continued after the semester! Funded positions available.

Proposal deadline: Feb 16th 1:30pm

- You should
  - determine your groups,
  - think about your topic,
  - discuss it with me,
  - and finalize the report and presentation by answering the questions below.
    - What is the problem?
    - Why do we care?
    - What is your execution plan?
- Start today!!

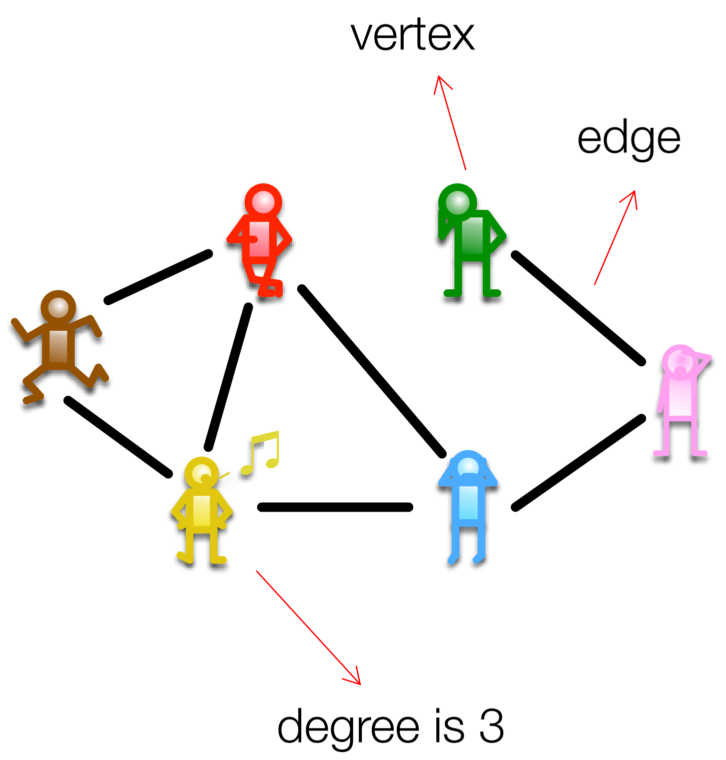
- By Feb 16th, Wed: Report is due 1:30pm, Presentation is in-class
- **I may ask for changes after reviewing your proposal**

Review of Graph Theory

Representing Networks

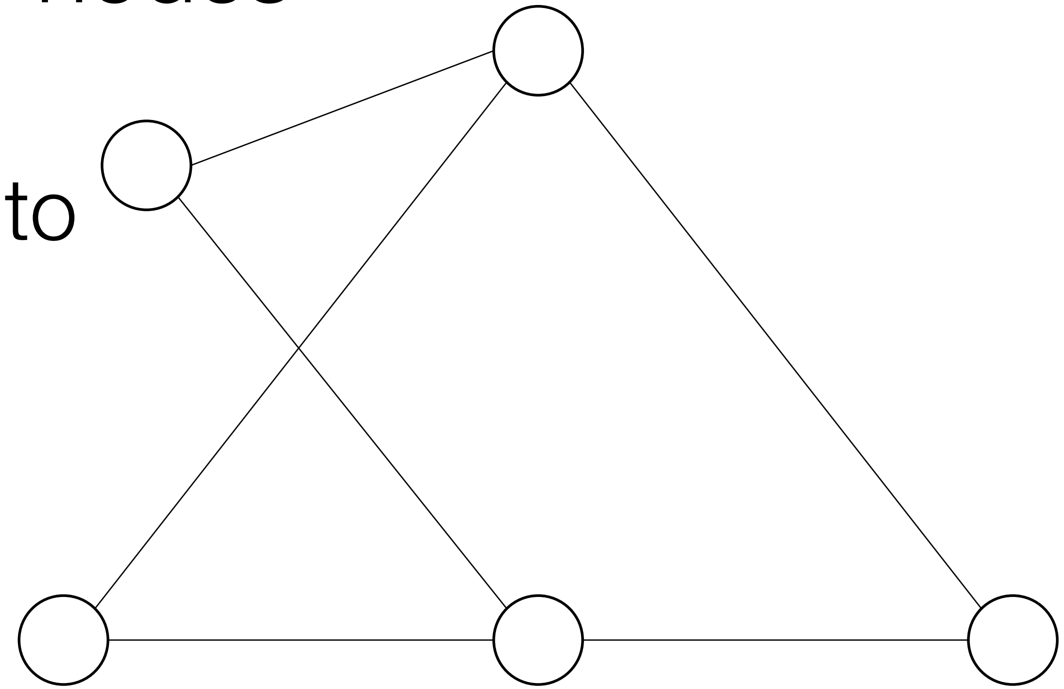
- Distinct entities: Nodes (or vertices)
- Connections: Links (or edges)

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



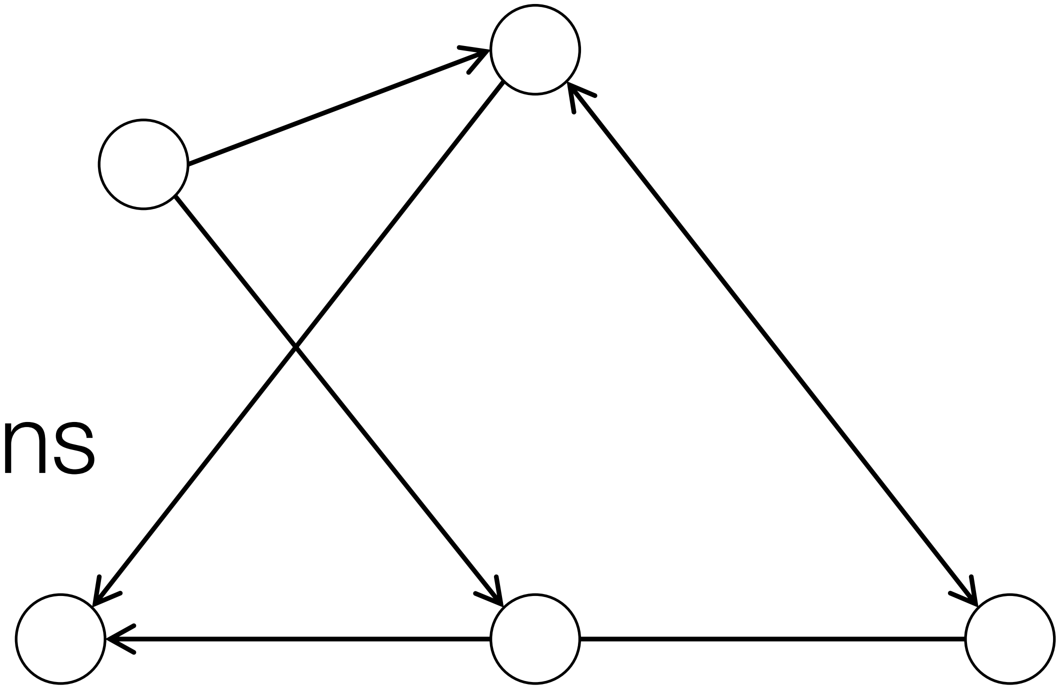
Simple graph is just nodes and edges

- Dots and lines
- Single edge between any pair of nodes
- Degree of a vertex
  - Number of edges it is connected to
- 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

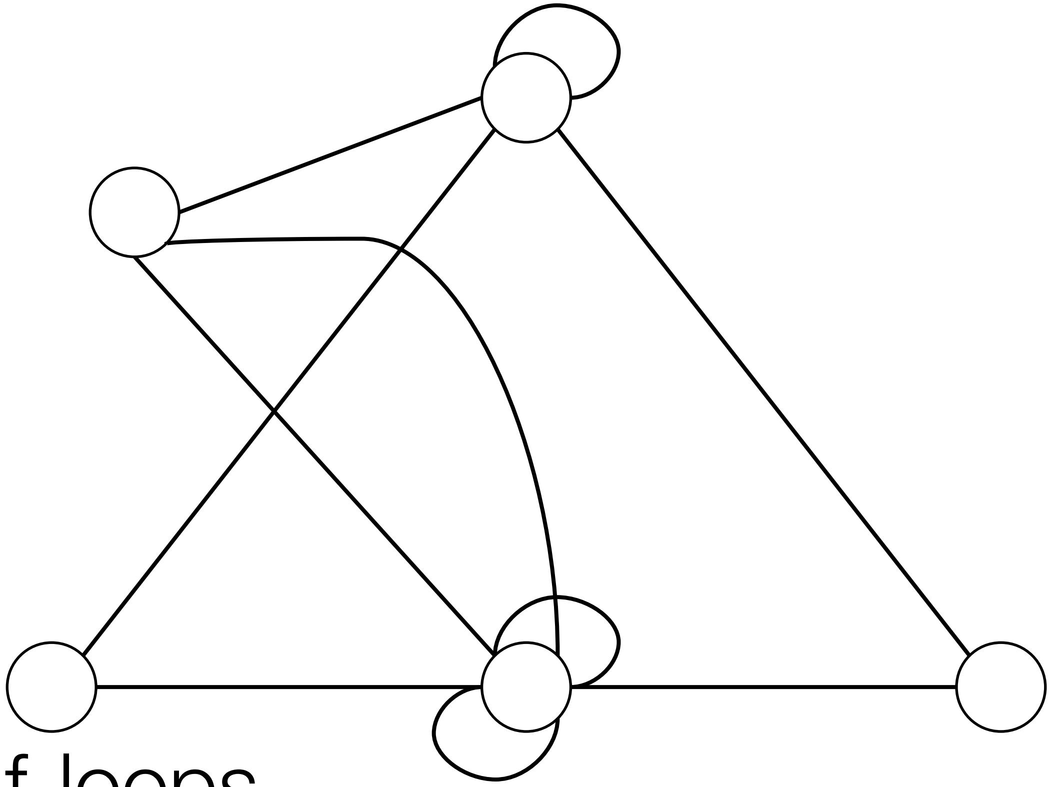


Multi-edges and self-loops

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

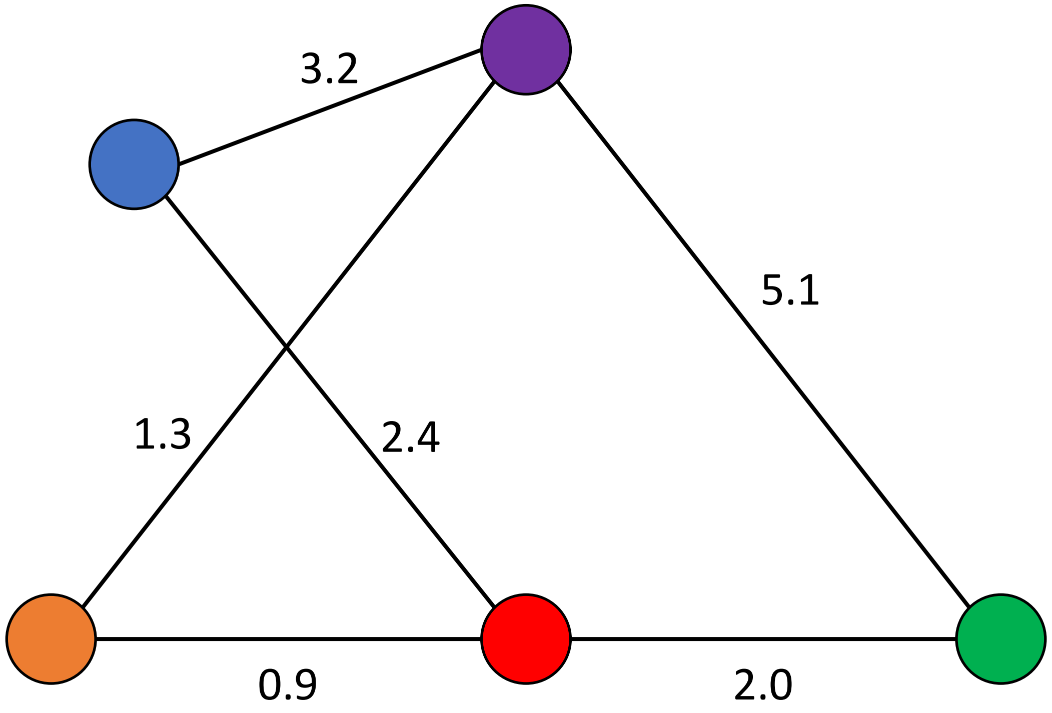
- Self-loops
  - A vertex is connected to itself

- Any combination is possible
  - E.g., directed multiedges with self-loops



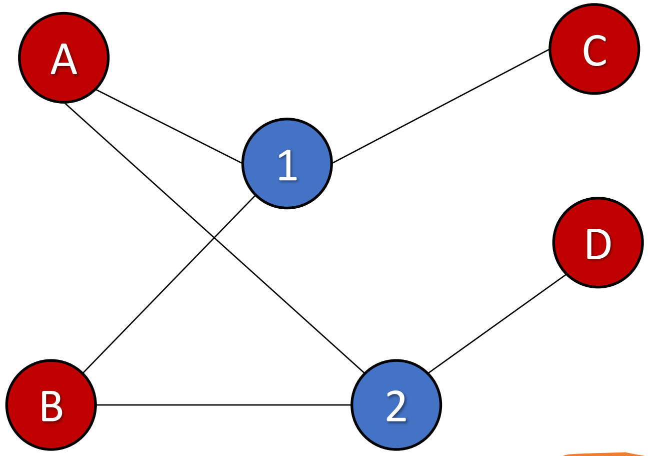
Attributes on vertices and edges

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

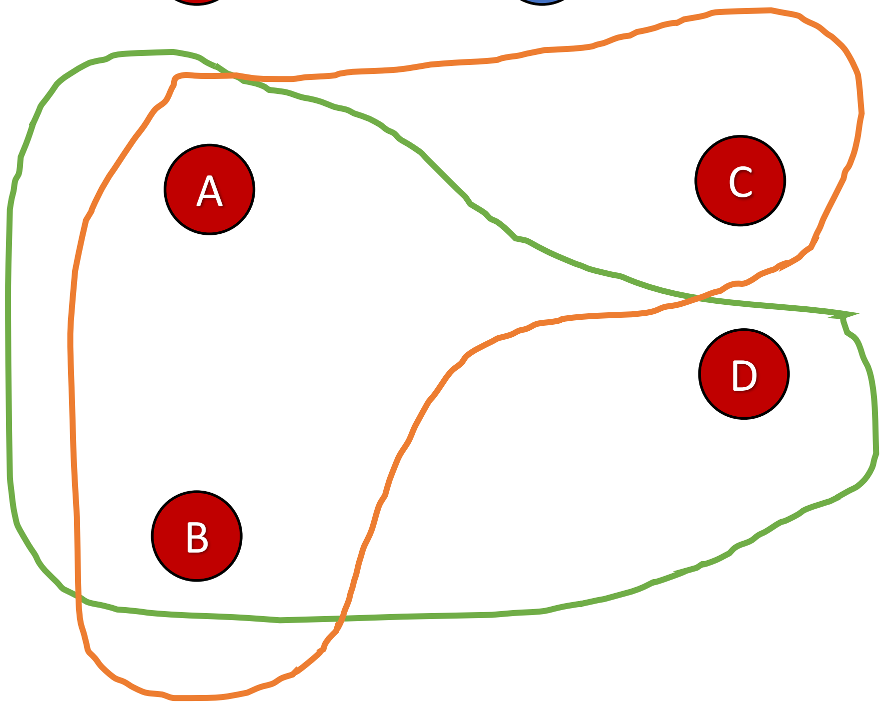


Hypergraphs (or bipartite graphs)

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

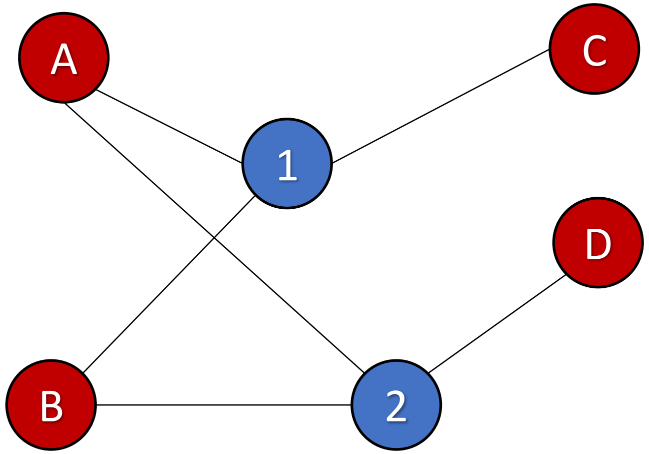


Network	Vertex	Group
Film actors	Actor	Cast of a film
Coauthorship	Author	Authors of an article
Boards of directors	Director	Board of a company
Social events	People	Participants at social event
Recommender system	People	Those who like a book, film, etc.
Keyword index	Keywords	Pages where words appear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

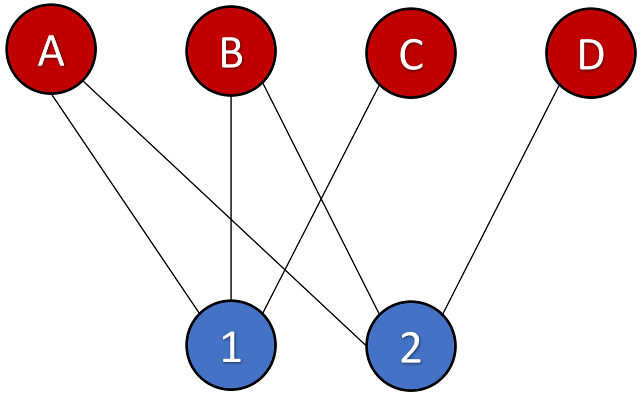


Hypergraphs (or bipartite graphs)

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

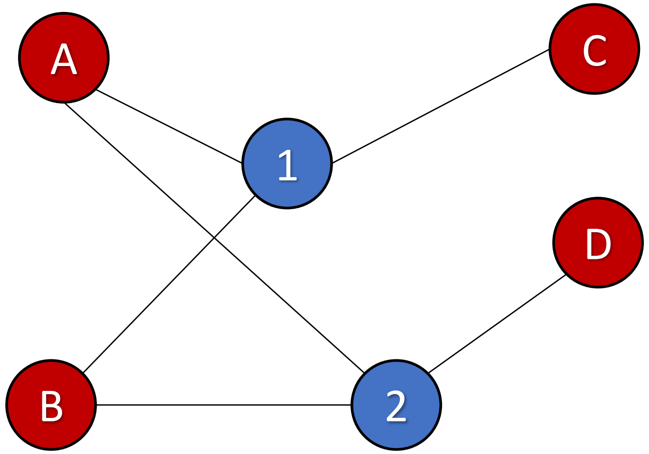


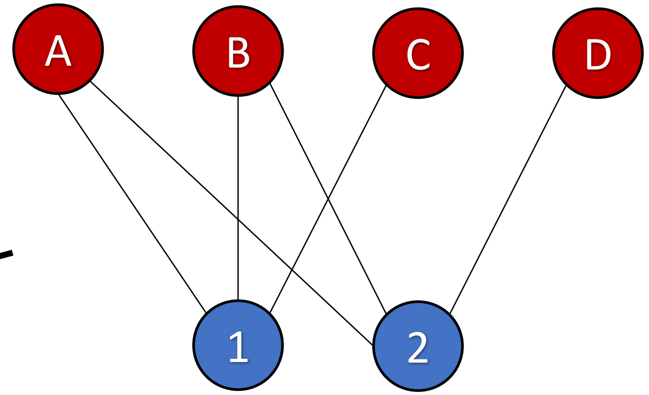
Network	Vertex	Group
Film actors	Actor	Cast of a film
Coauthorship	Author	Authors of an article
Boards of directors	Director	Board of a company
Social events	People	Participants at social event
Recommender system	People	Those who like a book, film, etc.
Keyword index	Keywords	Pages where words appear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

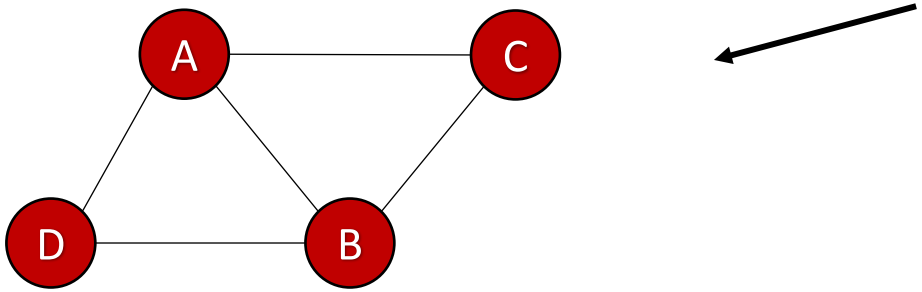


Hypergraphs (or bipartite graphs)

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

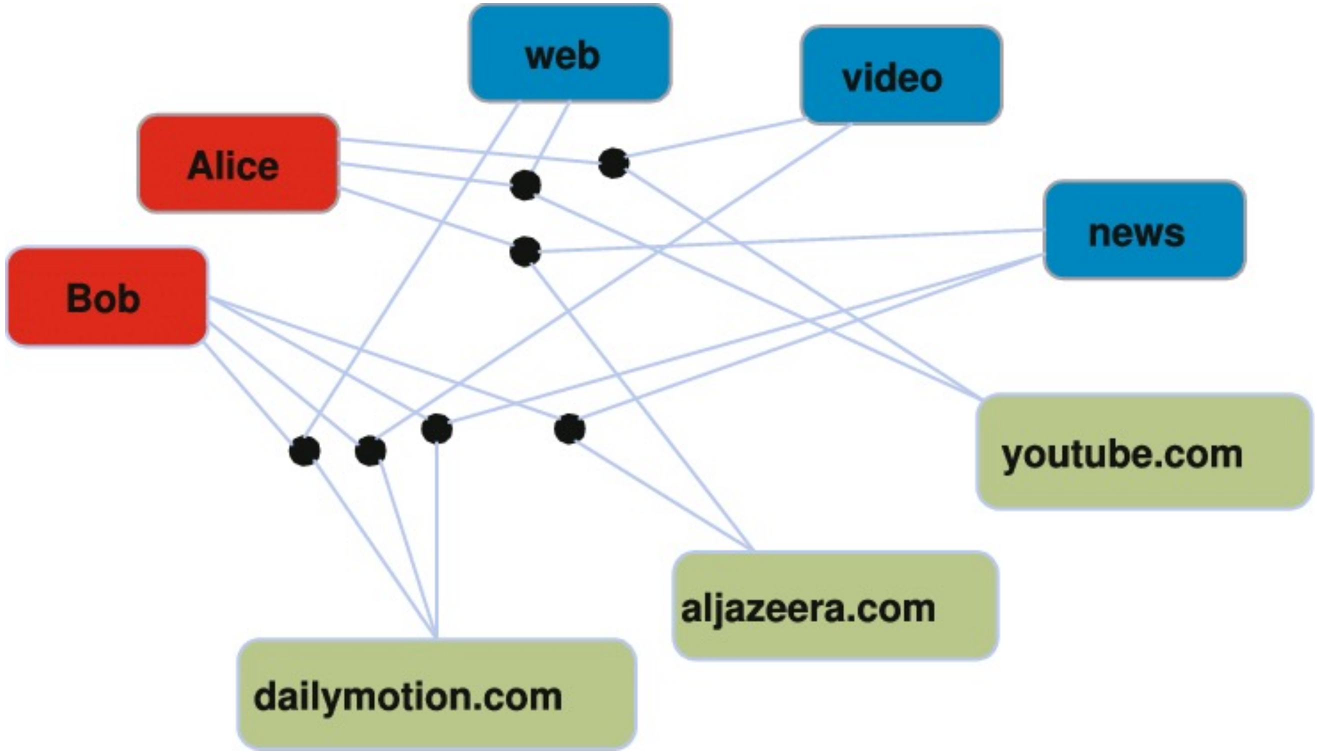






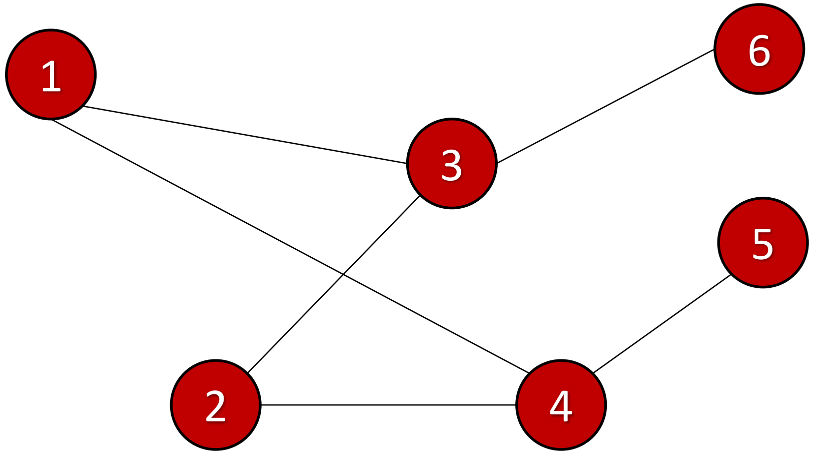
k-partite graphs

- 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

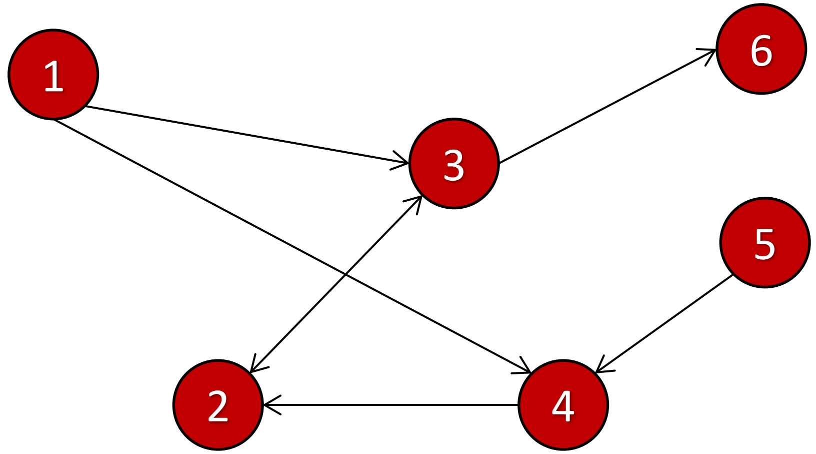
- Adjacency matrix
- For n vertices
  - Square matrix: n x n
  - Undirected -> binary, symmetric



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

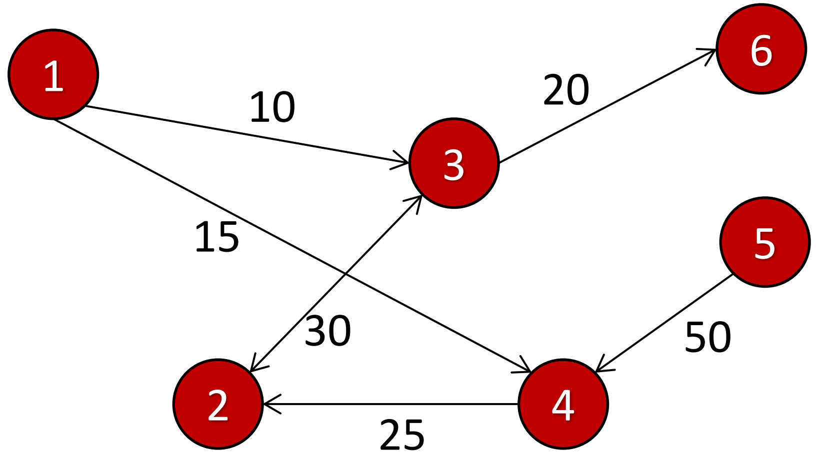
- Adjacency matrix
- For n vertices
  - Square matrix: n x n
  - Undirected -> binary, symmetric
  - Directed -> binary, not symmetric



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

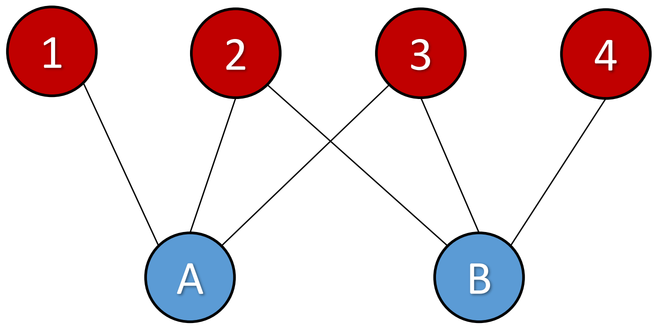
- Adjacency matrix
- For n vertices
  - Square matrix: n x n
  - Undirected -> binary, symmetric
  - Directed -> binary, not symmetric
  - Weighted -> not binary



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

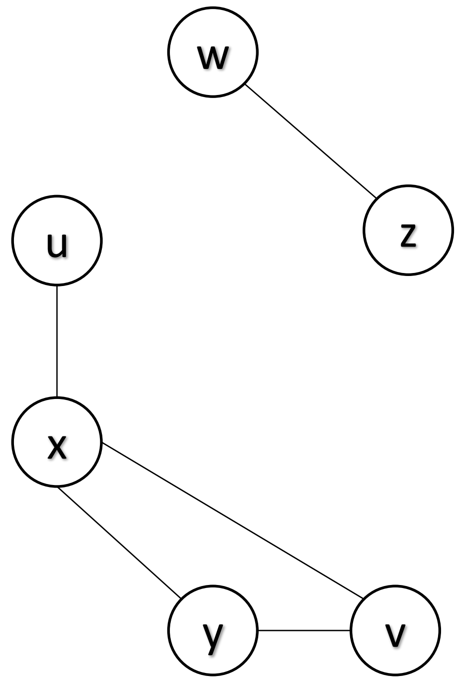


	A	B
1	1	0
2	1	1
3	1	1
4	0	1

- How to store matrices in memory?

Path and connectedness

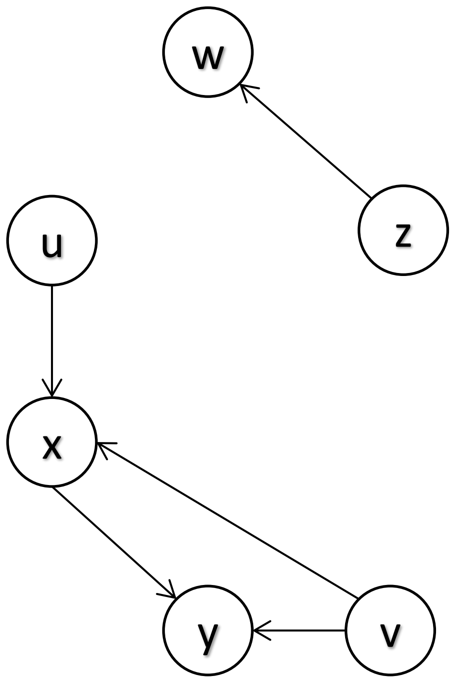
- Path: Sequence of vertices
  - Every consecutive vertex pair is connected
  - Length is the number of edges
- Two vertices are connected if there is a path between them
  - u and v are connected, u and w are not
  - Connected component
    - Maximal set of connected vertices
      - u,x,y,v and w,z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



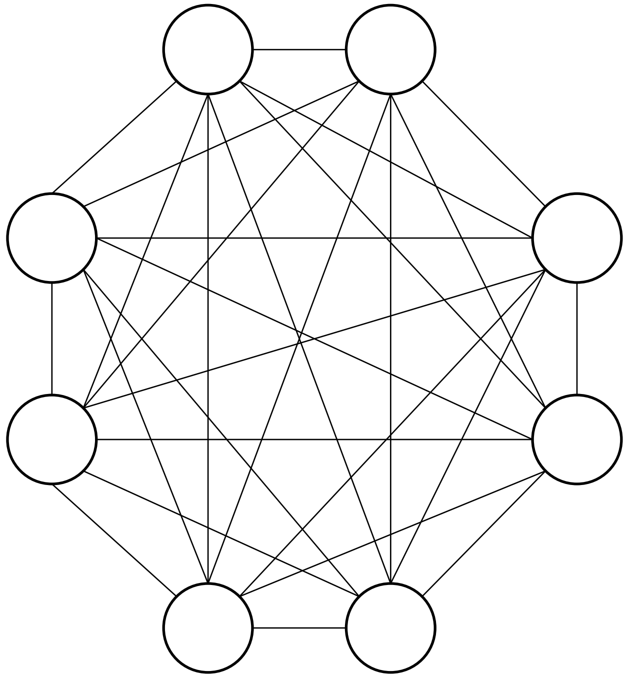
Special graphs

- Clique
- Cycle
- Tree
- DAG
- Planar

- Less complex algorithms
  - And easier analysis

Clique

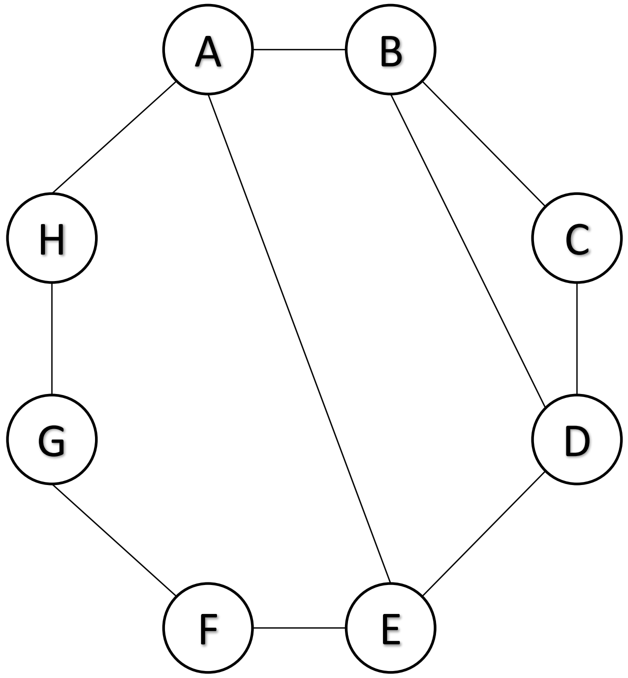
- Clique: Complete graph
  - All the vertices are directly connected to each other
  - k-clique has k vertices
    - k choose 2 edges
- Maximal-clique
  - Not a part of larger clique
- Maximum-clique
  - Largest clique in the graph
  - NP-Hard



Cycle

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

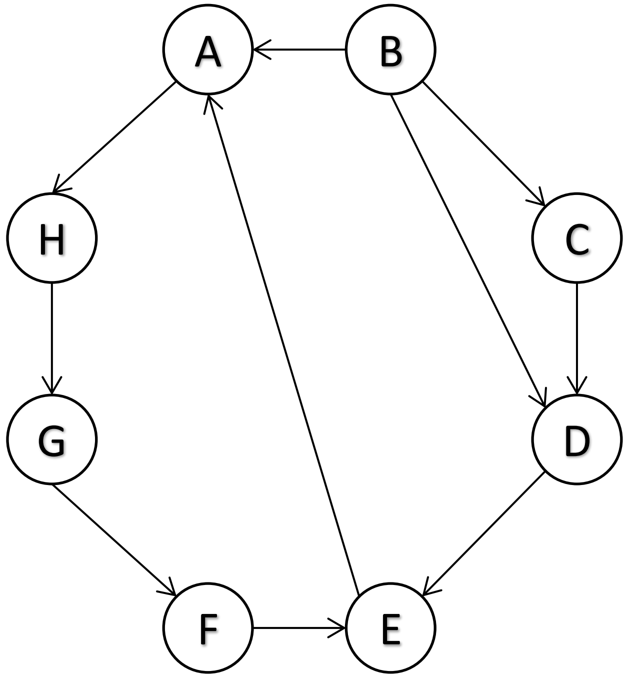
- ABCDEFGH: 8-cycle
  - ABDE: 4-cycle
  - BCD: 3-cycle



Cycle

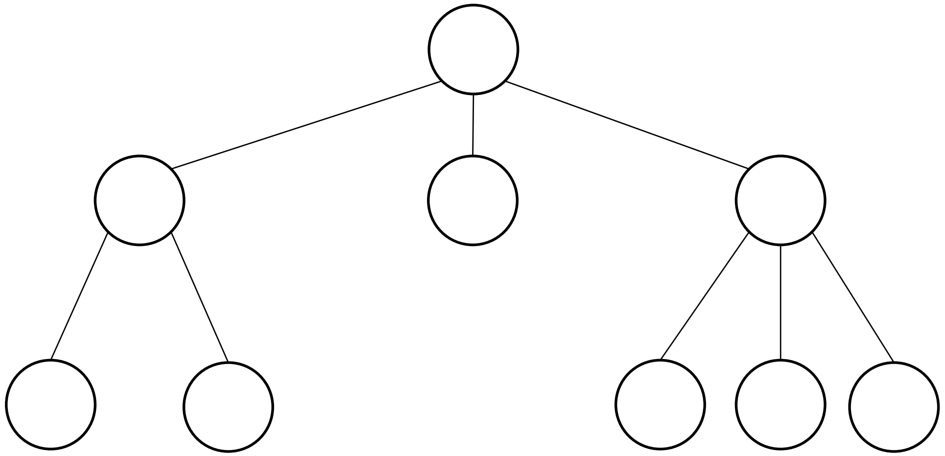
- Every vertex can reach itself via some path
  - k-cycle: Cycle with a path of k edges

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



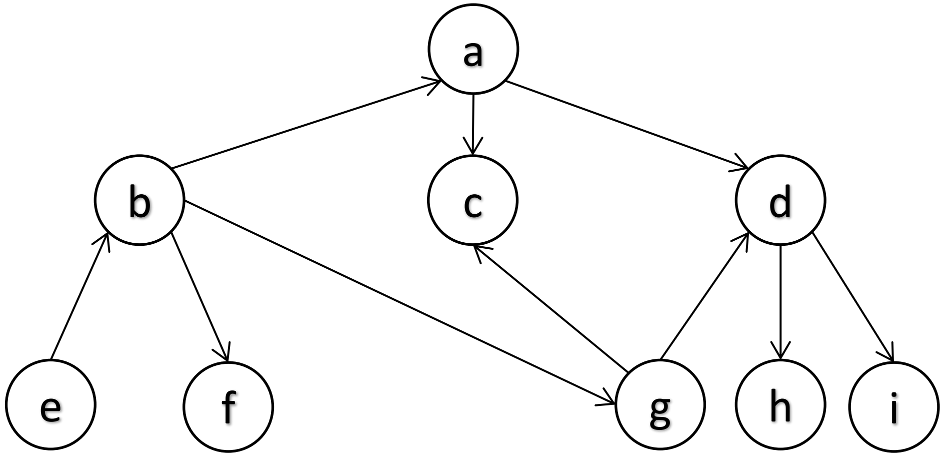
Tree

- Graph with no cycles
- n vertices, n-1 edges
  - Root
  - Parent
  - Children
- One parent for each vertex, except root



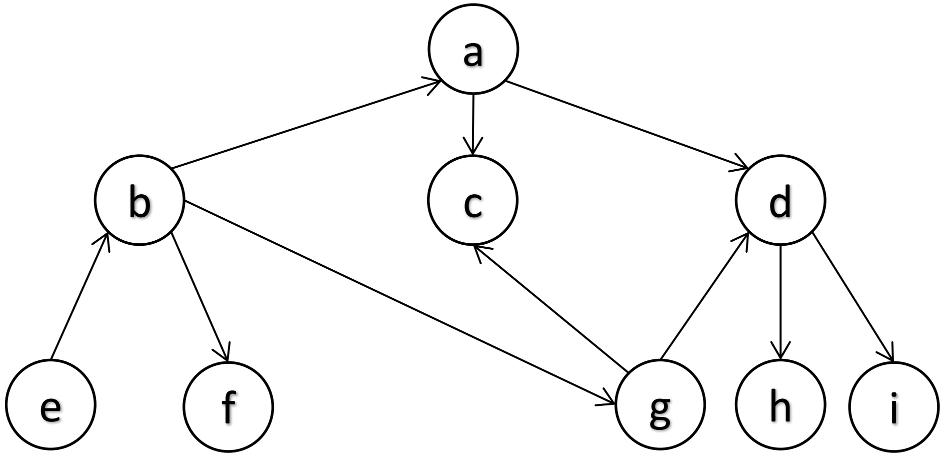
Directed Acyclic Graphs (DAG)

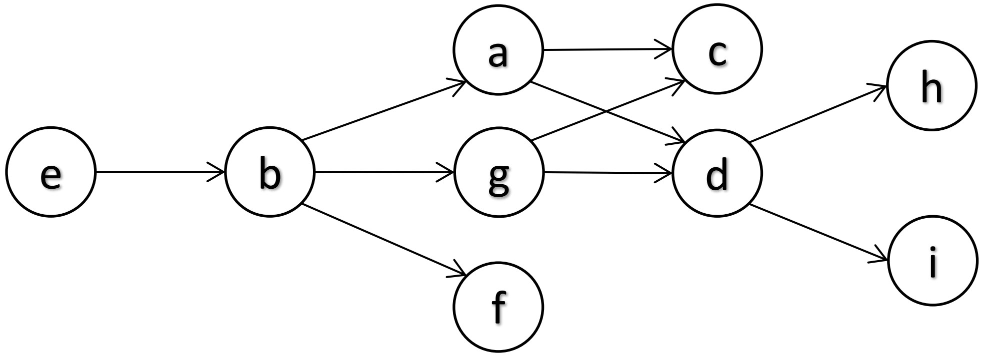
- Graph with no directed cycles
- n vertices
  - How many edges?
- Root?
  - Source and sink
- Citation networks!



Directed Acyclic Graphs (DAG)

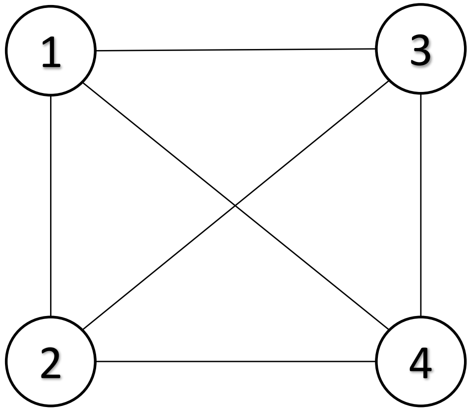
- Graph with no directed cycles
- n vertices
  - How many edges?
- Root?
  - Source and sink
- Citation networks!
  - e
  - b
  - a,g,f
  - c,d
  - h,i





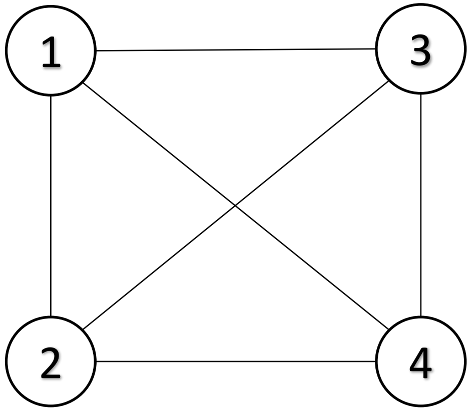
Planar graphs

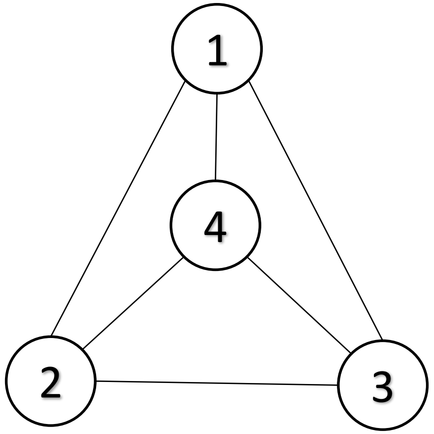
- Can be drawn on a plane with no crossing edges
- Cycles? Trees?
- Triangle?
- Four-clique?



Planar graphs

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





- Smallest non-planar clique?

- Road networks!