Discrete Structures

Graphs

Text book: Kenneth H. Rosen, Discrete Mathematics and Its Applications

Section: 10.1

Graphs

Chapter 10

Chapter Summary

- Graphs and Graph Models
- Graph Terminology and Special Types of Graphs
- Representing Graphs and Graph Isomorphism
- Connectivity

Graphs and Graph Models

Section 10.1

Section Summary

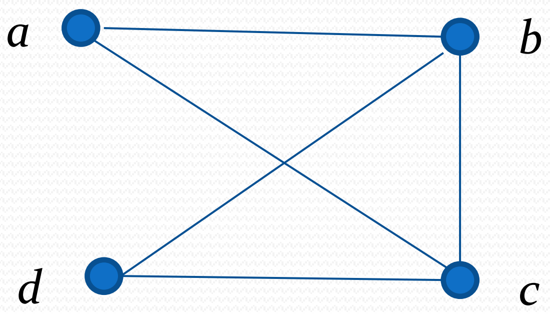
- Introduction to Graphs
- Graph Taxonomy
- Graph Models

Graphs

Definition: A *graph* $G = (V, E)$ consists of a nonempty set V of *vertices* (or *nodes*) and a set E of *edges*. Each edge has either one or two vertices associated with it, called its *endpoints*. An edge is said to *connect* its endpoints.

Example:

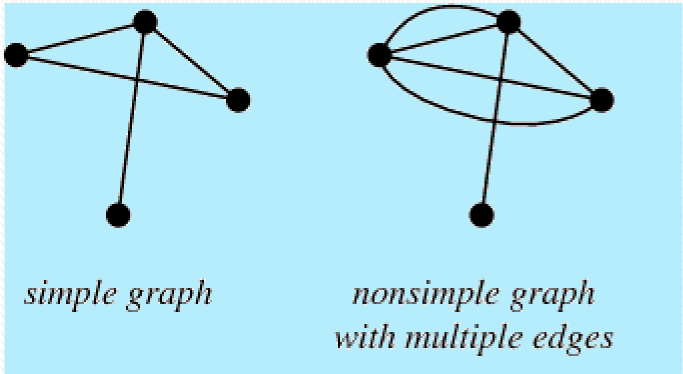
This is a graph with four vertices and five edges.



Remarks:

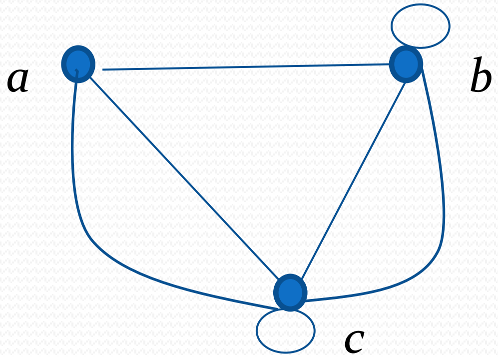
- We have a lot of freedom when we draw a picture of a graph. All that matters is the connections made by the edges, not the particular geometry depicted. For example, the lengths of edges, whether edges cross, how vertices are depicted, and so on, do not matter
- A graph with an infinite vertex set is called ***an infinite graph***. A graph with a finite vertex set is called a ***finite graph***. We restrict our attention to finite graphs.

Some Terminology

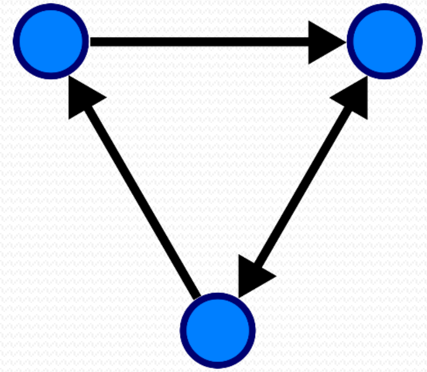


- In a *simple graph* each edge connects two different vertices and no two edges connect the same pair of vertices.
- *Multigraphs* may have multiple edges connecting the same two vertices. When m different edges connect the vertices u and v, we say that $\{u,v\}$ is an edge of *multiplicity* m.
- An edge that connects a vertex to itself is called a ***loop***.
- A *pseudograph* may include loops, as well as multiple edges connecting the same pair of vertices.

Example:
This pseudograph has both multiple edges and loops.

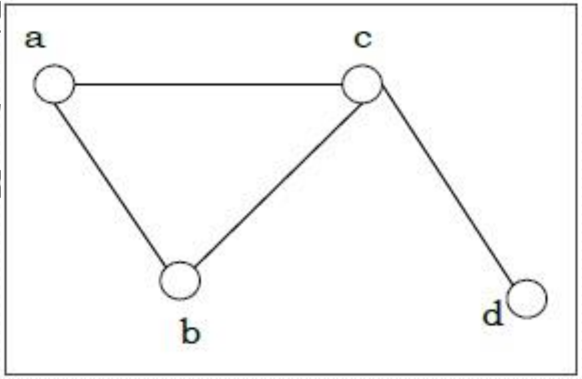


Remark: There is no standard terminology for graph theory. So, it is crucial that you understand the terminology being used whenever you read material about graphs.



Directed Graphs

Definition: An ***directed graph*** (or *digraph*) $G = (V, E)$ consists of a nonempty set V of *verti*
and a set E of *directed edges* (or *arcs*). E
associated with an ordered pair of verti
directed edge associated with the order
said to *start at u* and *end at v*.



Remark:

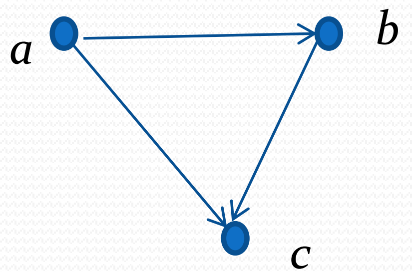
- Graphs where the end points of an edge are not ordered are said to be ***undirected graphs*.**

Some Terminology (*continued*)

- A ***simple directed graph*** has no loops and no multiple edges.

Example:

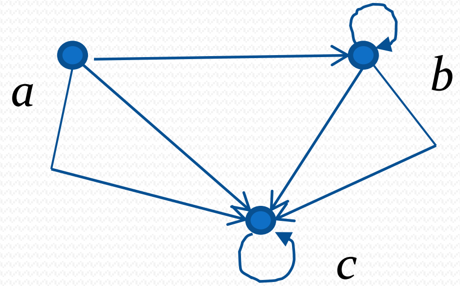
This is a simple directed graph with three vertices and three edges.



- A ***directed multigraph*** may have multiple directed edges. When there are m directed edges from the vertex u to the vertex v, we say that (u,v) is an edge of *multiplicity* m.

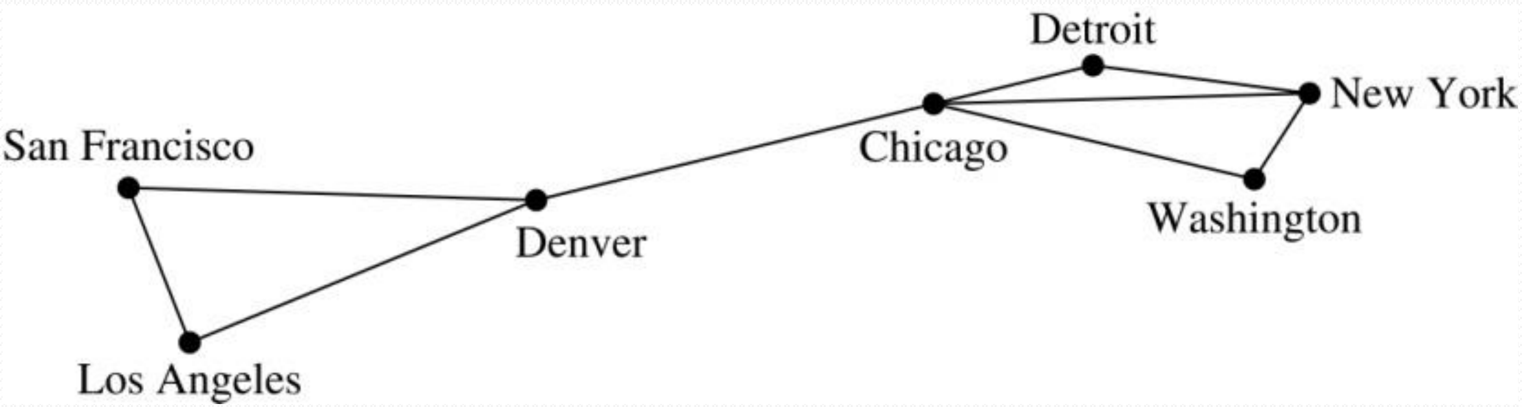
Example:

In this directed multigraph the multiplicity of (a,b) is 1 and the multiplicity of (b,c) is 2.



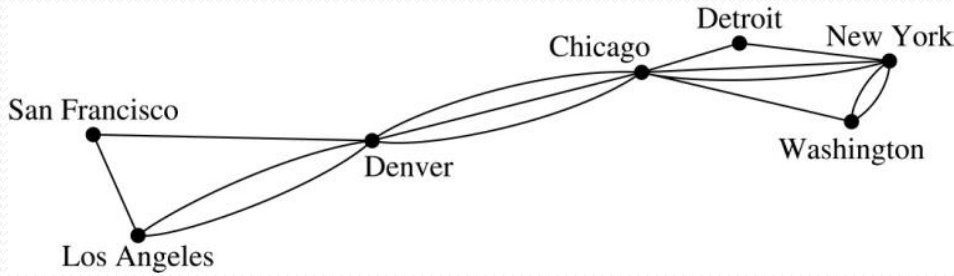
Graph Models: Computer Networks

- When we build a graph model, we use the appropriate type of graph to capture the important features of the application.
- We illustrate this process using graph models of different types of computer networks. In all these graph models, the vertices represent data centers and the edges represent communication links.
- To model a computer network where we are only concerned whether two data centers are connected by a communications link, we use a simple graph. This is the appropriate type of graph when we only care whether two data centers are directly linked (and not how many links there may be) and all communications links work in both directions.

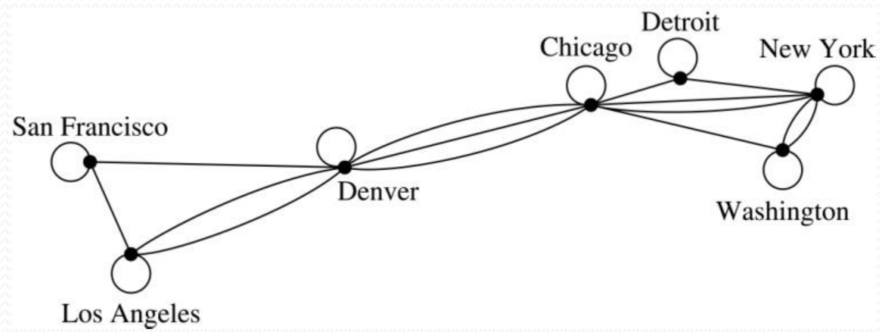


Graph Models: Computer Networks (*continued*)

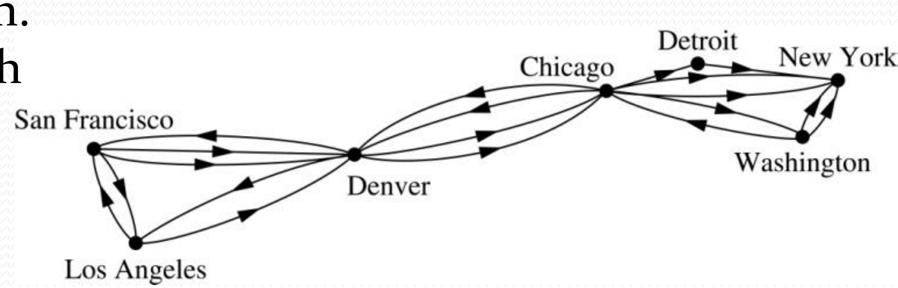
- To model a computer network where we care about the number of links between data centers, we use a multigraph.



- To model a computer network with diagnostic links at data centers, we use a pseudograph, as loops are needed.



- To model a network with multiple one-way links, we use a directed multigraph. Note that we could use a directed graph without multiple edges if we only care whether there is at least one link from a data center to another data center.



Graph Terminology: Summary

- To understand the structure of a graph and to build a graph model, we ask these questions:
  - Are the edges of the graph undirected or directed (or both)?
  - If the edges are undirected, are multiple edges present that connect the same pair of vertices? If the edges are directed, are multiple directed edges present?
  - Are loops present?

TABLE 1 Graph Terminology.

Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

Other Applications of Graphs

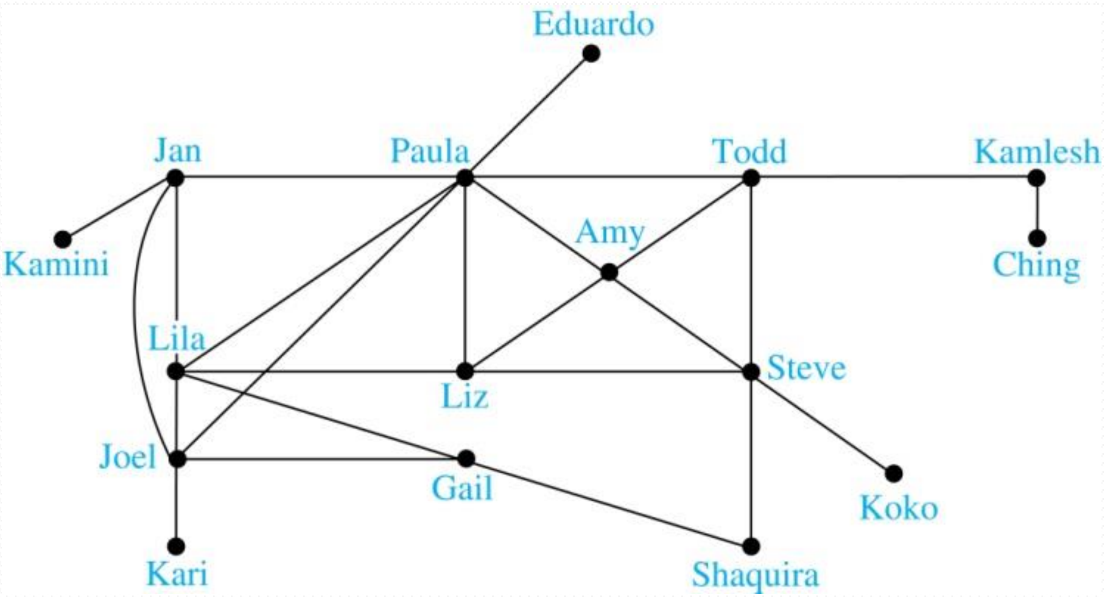
- We will illustrate how graph theory can be used in models of:
  - Social networks
  - Communications networks
  - Information networks
  - Software design
  - Transportation networks
  - Biological networks
- It's a challenge to find a subject to which graph theory has not yet been applied. Can you find an area without applications of graph theory?

Graph Models: Social Networks

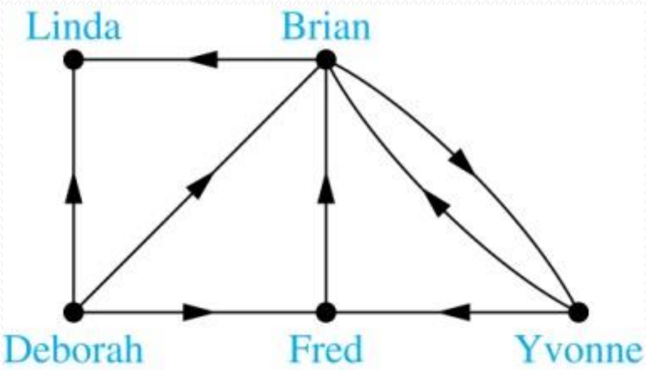
- Graphs can be used to model social structures based on different kinds of relationships between people or groups.
- In a *social network*, vertices represent individuals or organizations and edges represent relationships between them.
- Useful graph models of social networks include:
  - *friendship graphs* - undirected graphs where two people are connected if they are friends (in the real world, on Facebook, or in a particular virtual world, and so on.)
  - *collaboration graphs* - undirected graphs where two people are connected if they collaborate in a specific way
  - *influence graphs* - directed graphs where there is an edge from one person to another if the first person can influence the second person

Graph Models: Social Networks (*continued*)

Example: A friendship graph where two people are connected if they are Facebook friends.



Example: An influence graph



Examples of Collaboration Graphs

- The *Hollywood graph* models the collaboration of actors in films.
    - We represent actors by vertices and we connect two vertices if the actors they represent have appeared in the same movie.
- An *academic collaboration graph* models the collaboration of researchers who have jointly written a paper in a particular subject.
    - We represent researchers in a particular academic discipline using vertices.
    - We connect the vertices representing two researchers in this discipline if they are coauthors of a paper.

Applications to Information Networks

- Graphs can be used to model different types of networks that link different types of information.
- In a *web graph*, web pages are represented by vertices and links are represented by directed edges.
  - A web graph models the web at a particular time.
  - The web graph is used by search engines
- In a *citation network*:
  - Research papers in a particular discipline are represented by vertices.
  - When a paper cites a second paper as a reference, there is an edge from the vertex representing this paper to the vertex representing the second paper.

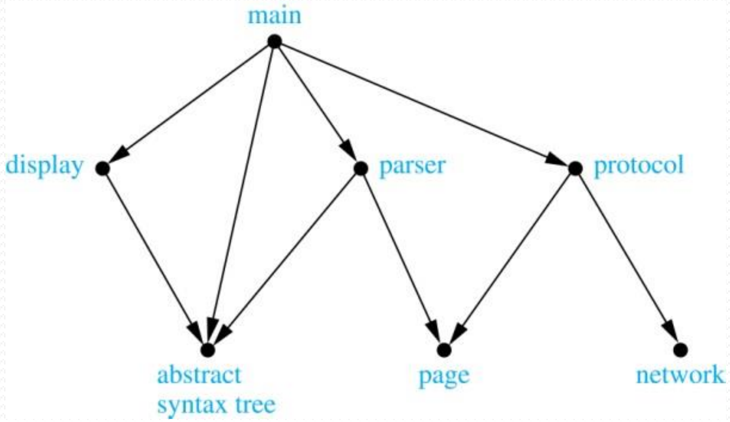
Transportation Graphs

- Graph models are extensively used in the study of transportation networks.
- *Airline networks* can be modeled using directed multigraphs where
  - airports are represented by vertices
  - each flight is represented by a directed edge from the vertex representing the departure airport to the vertex representing the destination airport
- Road networks can be modeled using graphs where
  - vertices represent intersections and edges represent roads.
  - undirected edges represent two-way roads and directed edges represent one-way roads.

Software Design Applications

- Graph models are extensively used in software design. We will introduce two such models here; one representing the dependency between the modules of a software application and the other representing restrictions in the execution of statements in computer programs.
- When a top-down approach is used to design software, the system is divided into modules, each performing a specific task.
- We use a *module dependency graph* to represent the dependency between these modules. These dependencies need to be understood before coding can be done.
  - In a module dependency graph vertices represent software modules and there is an edge from one module to another if the second module depends on the first.

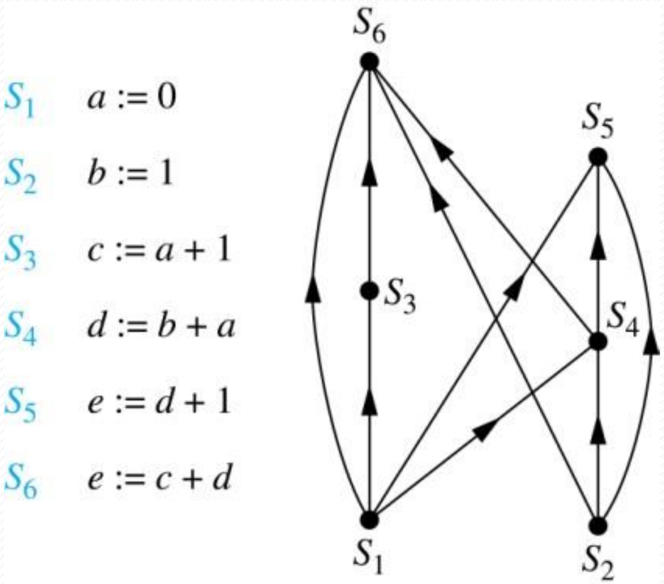
Example: The dependencies between the seven modules in the design of a web browser are represented by this module dependency graph.



Software Design Applications (*continued*)

- We can use a directed graph called a *precedence graph* to represent which statements must have already been executed before we execute each statement.
  - Vertices represent statements in a computer program
  - There is a directed edge from a vertex to a second vertex if the second vertex cannot be executed before the first

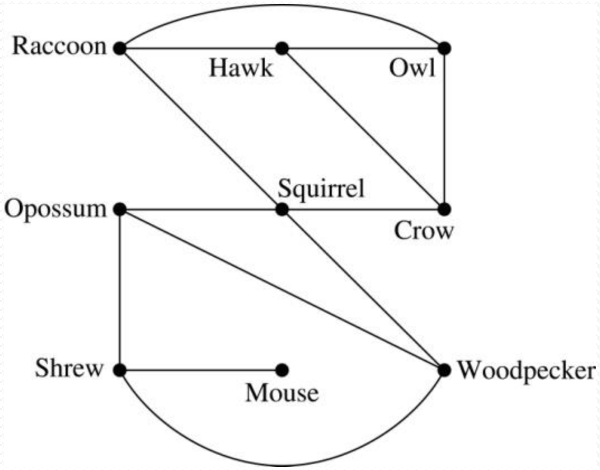
Example: This precedence graph shows which statements must already have been executed before we can execute each of the six statements in the program.



Biological Applications

- Graph models are used extensively in many areas of the biological science. We will describe two such models, one to ecology and the other to molecular biology.
- *Niche overlap graphs* model competition between species in an ecosystem
  - Vertices represent species and an edge connects two vertices when they represent species who compete for food resources.

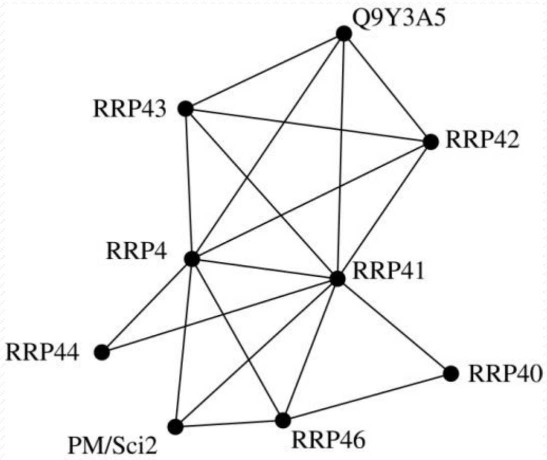
Example: This is the niche overlap graph for a forest ecosystem with nine species.

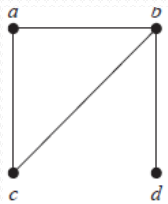


Biological Applications (*continued*)

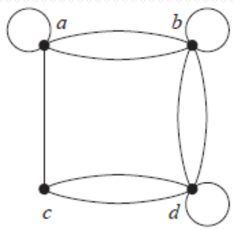
- We can model the interaction of proteins in a cell using a *protein interaction network.*
- In a *protein interaction graph*, vertices represent proteins and vertices are connected by an edge if the proteins they represent interact.
- Protein interaction graphs can be huge and can contain more than 100,000 vertices, each representing a different protein, and more than 1,000,000 edges, each representing an interaction between proteins
- Protein interaction graphs are often split into smaller graphs, called *modules*, which represent the interactions between proteins involved in a particular function.

Example: This is a module of the protein interaction graph of proteins that degrade RNA in human cells.

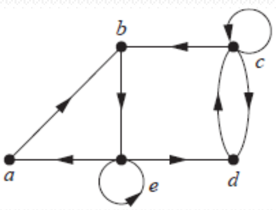




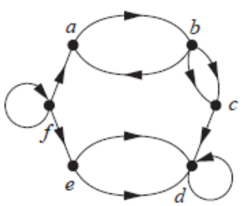
a simple graph; the edges are undirected, and there are no parallel edges or loops



a pseudograph; the edges are undirected, but there are loops and parallel edges



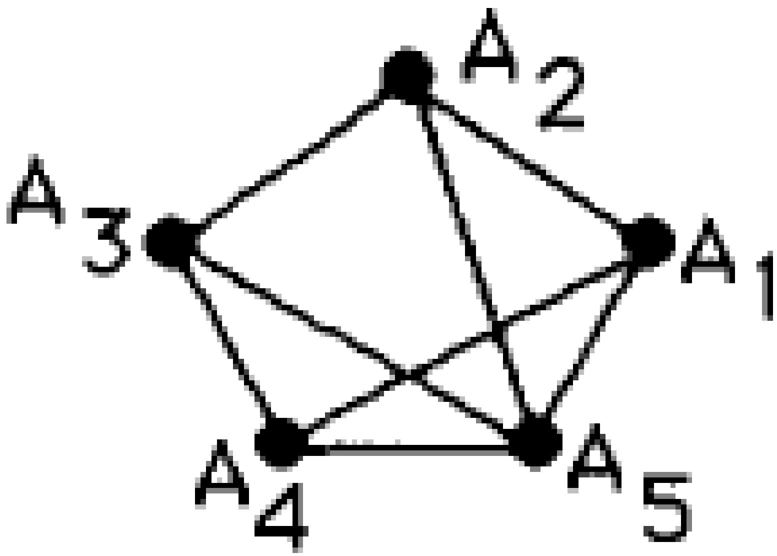
This is a directed graph; the edges are directed, but there are no parallel edges.



a directed multigraph; the edges are directed, and there is a set of parallel edges.

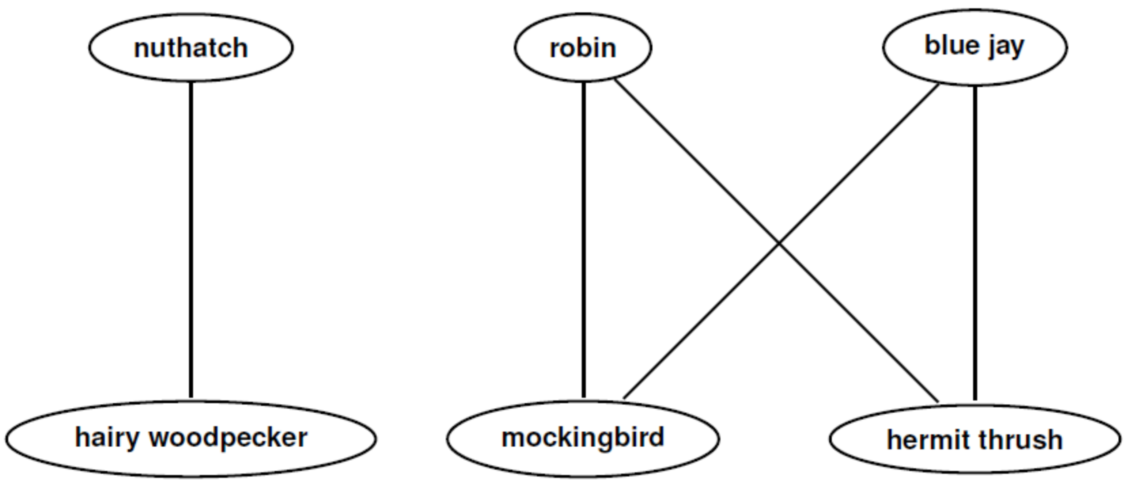
13. The **intersection graph** of a collection of sets A_1, $A_2, \ldots, A_n$ is the graph that has a vertex for each of these sets and has an edge connecting the vertices representing two sets if these sets have a nonempty intersection. Construct the intersection graph of these collections of sets.

$A_1 = \{0, 2, 4, 6, 8\}$, $A_2 = \{0, 1, 2, 3, 4\}$, $A_3 = \{1, 3, 5, 7, 9\}$, $A_4 = \{5, 6, 7, 8, 9\}$, $A_5 = \{0, 1, 8, 9\}$



15. Construct a niche overlap graph for six species of birds, where the hermit thrush competes with the robin and with the blue jay, the robin also competes with the mockingbird, the mockingbird also competes with the blue jay, and the nuthatch competes with the hairy woodpecker.

We draw a picture of the graph in question, which is a simple graph. Two vertices are joined by an edge if we are told that the species compete (such as robin and mockingbird) but there is no edge between pairs of species that are not given as competitors (such as robin and blue jay).



19. Construct an influence graph for the board members of a company if the President can influence the Director of Research and Development, the Director of Marketing, and the Director of Operations; the Director of Research and Development can influence the Director of Operations; the Director of Marketing can influence the Director of Operations; and no one can influence, or be influenced by, the Chief Financial Officer.

We draw a picture of the graph in question, which is a directed graph. We draw an edge from u to v if we are told that u can influence v. For instance the Chief Financial Officer is an isolated vertex since she is influenced by no one and influences no one.