Mardi 15 août 2023 — 10 Graphs

10.1 Graphs and Graph Models

Definition 1. A **graph** $G=(V,E)$ consists of V, a nonempty set of vertices (or nodes) and E, a set of edges.
Each edge has either one or two vertices associated with it, called its endpoints. An edge is said to connect its endpoints.

A graph in which each edge connect two different vertices and where no two edges connect the same pair of vertices is called a **simple graph**.

Graphs that may have multiple edges connecting the same vertices are called **multigraphs**.
$\underbrace{\text{multiple edges}}_{\text{multiplicity } m}$

Graphs that may include loops, and possibly multiple edges connecting the same pair of vertices or a vertex to itself, are sometimes called **pseudographs**

Undirected

Definition 2. A **directed graph** (V,E) consist of a nonempty set of vertices V and a set of directed edges (or arcs) E.
Each directed edge is associated with an ordered pair of vertices (u,v), said to start at u and end at v.

When a directed graph has no loops and has no multiple directed edges, it is called a **simple directed graph**.

For some models we may need a graph where some edges are undirected, while others are directed. A graph with both directed and undirected edges is called a **mixed graph**.

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

(III.) Information Networks

Graphs can be used to model various networks that link particular types of information.

Example 5 | The Web Graph

The web can be modeled as a directed graph where each web page is represented by a vertex and where an edge starts at the webpage a and ends at the web page b if there is a link on a pointing to b. Because new web pages are created and others removed somewhere on the web almost every second, the web graph changes on an almost continual basis. Many people are studying the properties of the web graph to better understand the nature of the web.

Example 6 | Citation Graphs

Graphs can be used to represent citations in different types of documents, including academic papers, patents, and legal opinions. In such graphs, each document is represented by a vertex, and there is an edge from one document to a second document if the first document cites the second in its citation list.
In an academic paper, the citation list is the bibliography, or list of references; in a patent it's the list of previous patents that are cited; and in an legal opinion it's the list of previous opinion cited. A citation graph is a directed graph without loops or multiple edges.

(V.) Transportation Networks

We can use graphs to model many different types of transportation networks, including road, air, and rail networks, as well shipping networks.

Example 9 | Airline Routes

We can model airline networks by representing each airport by a vertex. In particular, we can model all the flights by a particular airline each day using a directed edge to represent each flight, going from the vertex representing the departure airport to the vertex representing to the destination airport. The resulting graph will generally be a directed multigraph, as there may be multiple flights from one airport to some other airport during the same day.

(VI.) Biological Networks

Many aspect of the biological sciences can be modeled using graphs.

Example 11 | Niche Overlap Graphs in Ecology

Graphs are used in many models involving the interaction of different species of animals. For instance, the competition between species in an ecosystem can be modeled using a **niche overlap graph**. Each species is represented by a vertex. An undirected edge connects two vertices if the two species represented by these vertices compete (that is, some of the food resources they use are the same). A niche overlap graph is a simple graph because no loops or multiple edges are needed in this model.

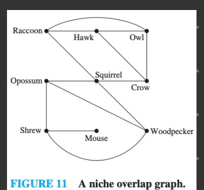


FIGURE 11 A niche overlap graph.

Example 12 | Protein Interaction Graphs

...

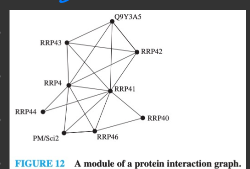


FIGURE 12 A module of a protein interaction graph.

(VII.) Semantic Networks

Graph models are used extensively in natural language understanding and in information retrieval. **Natural language understanding (NLU)** is the subject of enabling machines to disassemble and parse human speech. Its goal is to allow machines to understand and communicate as human do. **Information retrieval (IR)** is the subject of obtaining information from a collection of sources based on various types of searches. NLU is the enabling technology when we converse with automated customer service agents. In graph models for NLU and IR applications, vertices often represent words, phrases, or sentences, and edges represent connections relating to the meaning of these objects.

Example 13 | A **semantic relation relation** is a relationship between two or more words that is based on meaning of the words.

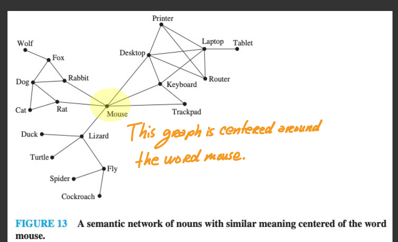


FIGURE 13 A semantic network of nouns with similar meaning centered of the word mouse.

The graph illustrates that there are two distinct meanings for mouse. When a NLU program encounters the word mouse in a sentence, it can see which words with similar meaning would fit the sentence to help determine whether mouse refers to an animal or computer hardware in that sentence.

Graphs Models

(I.) Social Networks

Graphs are extensively used to model social structures based on different kinds of relationships between people or groups of people. These social structures, and the graphs that represent them, are known as **social networks**. In these graph models, individuals or organizations are represented by vertices; relationships between individuals or organizations are represented by edges.
The study of social networks is an extremely active multidisciplinary area, and many different types of relationships between people have been studied using them. We will introduce some of the most commonly studied social networks here: Ex. 1-3

Example 1 | Acquaintanceship and Friendship Graphs

We can use a simple graph to represent whether two people know each other, that is, whether they are acquainted, or whether they are friends (either in the real world or in the virtual world via a social networking site such as Facebook). Each person in a particular group of people is represented by a vertex. An undirected edge is used to connect two people when these people know each other, when we are concerned only with acquaintanceship, or wheter they are friends. No multiple edges and usually no loops are used. (If we want to include the notion of self-knowledge, we would include loops). The acquaintanceship graph of all people in the world has more than six billion vertices and probably more than one trillion edges!

Example 2 | Influence Graphs

In studies of group behavior it is observed that certain people can influence the thinking of others. A directed graph called an **influence graph** can be used to model this behavior. Each person of the group is represented by a vertex. There is a directed edge from vertex a to b when the person represented by vertex a can influence the person b. This graph does not contain loops and it does not contain multiple directed edges.

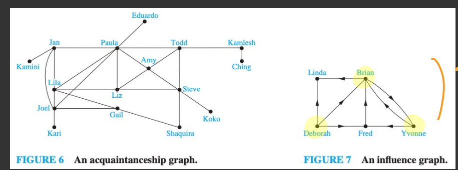


FIGURE 6 An acquaintanceship graph. FIGURE 7 An influence graph.

Deborah cannot be influenced.
Yvonne and Brian can influence each other.

(IV.) Software Design Applications

Graphs models are usefull tools in the design of software.

Example 7 | Module Dependency Graphs

One of the most important tasks in designing software is how to structure a program into different parts, or modules. Understanding how the different modules of a program interact is essential not only for program design, but also for testing and maintenance of the resulting software.
A **module dependency graph** provides a useful tool for understanding how different modules of a program interact. In a program dependency graph, each module to a second module if the second module depends on the first.

Example 10 | Road Networks

Graph can be used to model road networks. In such models, vertices represent intersections and edges represent roads. When all roads are two-way and there is at most one road connecting two intersections, we can use a simple undirected graph to model the road network. However, mixed graphs are often needed to model road networks that include both one-way and two-way roads, and also loop roads.

(VIII.) Tournaments

Graphs can also be used to model different kinds of tournaments.

Example 14 | Round-Robin Tournaments

A tournament where each team plays every other team exacty once and no ties are allowed is called a **round-robin tournament**.

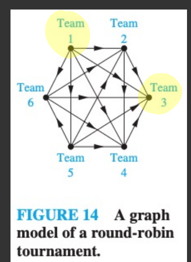


FIGURE 14 A graph model of a round-robin tournament.

(a,b) is an edge if team a beats team b.

Team 1 is undefeated
Team 3 is winless.

Example 15 | Single-Elimination Tournaments

A tournament where each contestant is eliminated after one loss is called a **single-elimination tournament**.

FIGURE 15 A single-elimination tournament.

Example 3 | Collaboration Graphs

A **collaboration graph** is used to model social networks where two people are related by working together in a particular way. Collaboration graphs are simple graphs, as edges in these graphs are undirected and there are no multiple edges or loops. Vertices represent people; two people are connected by an undirected edge when the people have collaborated. There are no loops nor multiple edges in these graphs. The **Hollywood graph** is a collaborator graph that represent actors by vertices and connects two actors with an edge if they have worked together on a movie or television show. The Hollywood graph is a huge graph with more than 2,9 millions vertices (as early of 2018)

In an **academic collaboration graph**, vertices represent people (members of a certain academic community), and edges link to people if they have jointly publish a paper. The collaboration graph for people who have published research papers in mathematics was found in 2004 to have more than 400 000 vertices and 675 000 edges, and these numbers have grown considerably since then. Collaboration graph are also used in sports, where two professional athletes are considered to have collaborated if they have ever played on the same team during a regular season of their sport.

(II.) Communication Networks

We can model different communications networks using vertices to represent devices and edges to represent the particular type of communications links of interest.

Example 4 | Call graphs

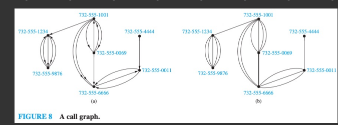


FIGURE 8 A call graph.

Graphs can be used to model telephone calls made in a network, such as a long-distance telephone network. In particular, a directed multigraph can be used to model calls where each telephone number is represented by a vertex and each telephone call is represented by a directed edge. The edge representing a call starts at the telephone number from which the call was made and ends at the telephone number to which the call was made.
We need directed edges because the direction in which the call is made matters. We need multiple directed edges because we want to repreent each call made from a particular telephone number to a second number. When we care only whether there has been a call connecting two telephone numbers, we use an undirected graph with an edge connecting telephone numbers when there has been a call between these numbers.

Example 8 | Precedence Graphs and Concurrent Processing

Computer programs can be executed more rapidly by executing certain statements concurently. It is important not to execute a statement that requires results of statements not yet executed. The dependence of statements on previous statements can be represented by a directed graph. Each statement is represented by a vertex, and there is an edge from one statement to a second statement if the second statement cannot be executed before the first statement. This resulting graph is called a **precedence graph**.

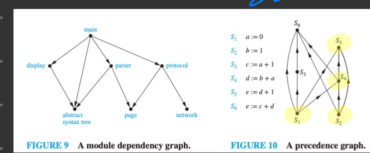


FIGURE 9 A module dependency graph. FIGURE 10 A precedence graph.

For instance, the graph shows that statement S_5 cannot be executed before S_1, S_2 and S_4 are executed.

10.2 Graph Terminology and Special Types of Graphs

Definition 1. Two vertices u and v in an undirected graph G are called **adjacent** (or **neighbors**) in G if u and v are endpoints of an edge e of G. Such edge e is called **incident with** the vertices u and v and e is said to **connect** u and v.

Definition 2. The set of all neighbors of vertex v of $G=(V,E)$, denoted by $N(v)$, is called the **neighborhood** of v. If A is a subset of V, $N(A)=\bigcup_{u\in A} N(v)$

Definition 3. The **degree** of a vertex in an undirected graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex. The degree of the vertex v is denoted by $\deg(v)$.

A vertex of deg zero is called **isolated**. It follows that an isolated vertex is not adjacent to any vertex.
A vertex is **pendant** iff it has deg one. Consequently, a pendant vertex is adjacent to exactly one other vertex.

Theorem 1. The Handshaking Theorem
Let $G=(V,E)$ be an undirected graph with m edges. Then $\sum_{v\in V} \deg(v) = 2m$
Note that this applies even if multiple edges and loops are present.

Theorem 2. An undirected graph has an even number of vertices of odd degrees.

Proof. Let V_1 and V_2 be the set of vertices of even deg and odd deg, respectively, in an undirected graph $G=(V,E)$ with m edges.
Then, $\underbrace{2m}_{\text{even}} = \sum_{v\in V} \deg(v) = \underbrace{\sum_{v\in V_1} \deg(v)}_{\text{even}} + \underbrace{\sum_{v\in V_2} \deg(v)}_{\text{must be an even number of term, because all the terms in this sum are odd.}}$
Thus, there are an even number of vertices of odd deg. ∎

Definition 4. When (u,v) is an edge of the graph G with directed edges, u is said to be **adjacent to** v, and v is said to be **adjacent from** u. $(\underbrace{u}_{u \text{ initial vertex}},\underbrace{v}_{v \text{ terminal or end vertex}})$

Definition 5. In a graph with directed edges the **in-degree** of a vertex v, denoted by $\deg^-(v)$, is the number of edges with terminal vertex.
The **out-degree** of v, denoted by $\deg^+(v)$, is the number of edges with initial vertex.

Theorem 3. Let $G=(V,E)$ be a graph with directed edges. The $\sum_{v\in V} \deg^-(v) = \sum_{v\in V}\deg^+(v) = \underbrace{|E|}_{=m}$

Bipartite Graphs

Definition 6. A simple graph G is called **bipartite** if its vertex V can be partitioned into two disjoint sets V_1 and V_2 such that every edge connects a vertex in V_1 and V_2

Theorem 4. A simple graph is bipartite iff it is possible to assign one of two different colors to each vertex of the graph so that no two adjacent vertices are assigned the same color.

Necessary and Sufficient Conditions for complete matchings.

Find an assignment of jobs to employees can be thought of as finding a matching in the graph model, where a **matching** M in a simple graph $G=(V,E)$ is a subset of E such that no two edges are incident with the same vertex.

Theorem 5. Hall's Marriage Theorem
Complete matching from V_1 to V_2 iff $|N(A)| \geq |A|$ for all subsets A of V_1

Removing edges: $G-\overset{\{a,b\}}{e} = (V, E-\{e\})$ $\longrightarrow$ Edge contractions: $\underbrace{G'}_{\text{not a subgraph of } G} = (V',E')$, where $V' = V - \overset{\{b,c\}}{\{u,v\}} \cup \overset{\{f\}}{\{w\}}$

Adding edges: $G+\overset{\{a,b\}}{e} = (V, E\cup\{e\})$

Removing vertices: $G-v=(V-\{v\}, E)$

$G' = (V-\{b,c\}\cup\{f\}, E)$

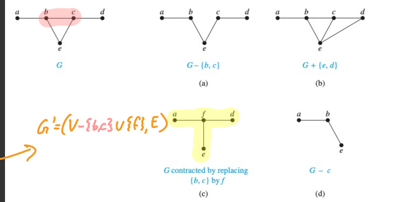


FIGURE 16 The graph G and four graphs resulting from different operations on G.

Example 1 | What are the degrees and what are the neighborhoods of the vertices in the graphs G and H?

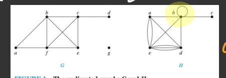


FIGURE 1 The undirected graphs G and H.

$\deg(b)=6$
$N(b)=\{a,b,c,d,e\}$

In G,
$\deg(a)=2$ $\qquad N(a)=\{b,f\}$
$\deg(b)=\deg(c)=\deg(f)=4$ $\qquad N(b)=\{a,c,e,f\}$
$\deg(d)=1$ d pendant $\qquad N(c)=\{b,d,e,f\}$
$\deg(e)=3$ $\qquad N(d)=\{d\}$
$\deg(g)=0$ g isolated $\qquad N(e)=\{b,c,f\}$
$N(f)=\emptyset$

Example 3 | How many edges are there in a graph with 10 vertices each of degree six?
Because the sum of the degrees of the vertices is $6\cdot 10=60$, it follows that $2m=60$ where m is the number of edges. Therefore, $m=30$.

Example 4 | Find the in-degree and out-degree of each vertex in the G with directed edges.

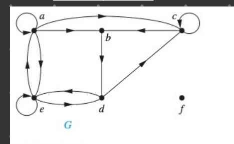


FIGURE 2 The directed graph G.

$\deg^-(a)=\deg^-(b)=\deg^-(d)=2$
$\deg^-(c)=\deg^-(e)=3$
$\deg^-(f)=0$

$\deg^+(a)=4$
$\deg^+(b)=5$
$\deg^+(c)=\deg^+(d)=2$
$\deg^+(e)=3$
$\deg^+(f)=0$

Some Special Simple Graphs

Example 5 | Complete Graphs
A **complete graph** on n vertices, denoted by K_n, is a simple graph that contains exactly one edge between each pair of distinct vertices.

A simple graph for which there is at least one pair of distinct vertex not connected by an edge is called **noncomplete**.

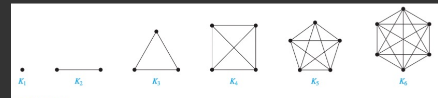


FIGURE 3 The graphs K_n for $1 \leq n \leq 6$.

EXAMPLE 6 **Cycles** A **cycle** C_n, $n \geq 3$, consists of n vertices $v_1, v_2, \dots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}$, and $\{v_n, v_1\}$. The cycles C_3, C_4, C_5, and C_6 are displayed in Figure 4.

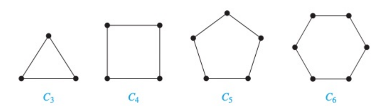


FIGURE 4 The cycles C_3, C_4, C_5, and C_6.

EXAMPLE 7 **Wheels** We obtain a **wheel** W_n when we add an additional vertex to a cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n, by new edges. The wheels W_3, W_4, W_5, and W_6 are displayed in Figure 5.

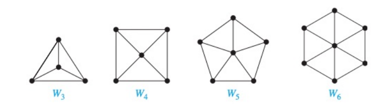


FIGURE 5 The wheels W_3, W_4, W_5, and W_6.

EXAMPLE 8 ***n*-Cubes** An ***n*-dimensional hypercube**, or ***n*-cube**, denoted by Q_n, is a graph that has vertices representing the 2^n bit strings of length n. Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position. We display Q_1, Q_2, and Q_3 in Figure 6.

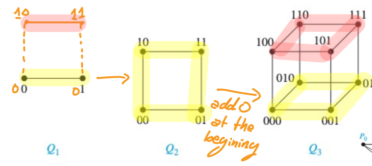


FIGURE 6 The n-cube Q_n, $n = 1, 2, 3$.

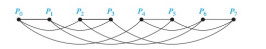


FIGURE 14 A hypercube network for eight processors.

Note that you can construct the $(n+1)$-cube Q_{n+1} from the n-cube Q_n by making two copies

Many computer have been built using a hypercube network, and many parallel algorithms have been devised to use a hypercube network.
It balances the number of direct connection for each processor and the number of intermediate connections required so that processors can communicate.
Also used for errors in gray code with Hamilton circuit.

Example 9 | C_6 is bipartite.

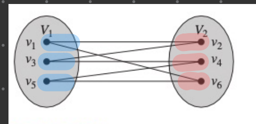


FIGURE 7 Showing that C_6 is bipartite.

Example 10 | K_3 is not bipartite

Example 11 |

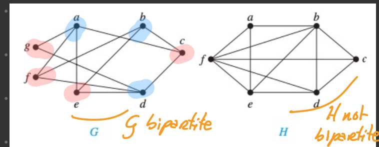


FIGURE 8 The undirected graphs G and H.

EXAMPLE 13 **Complete Bipartite Graphs** A **complete bipartite graph** $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets of m and n vertices, respectively with an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset. The complete bipartite graphs $K_{2,3}$, $K_{3,3}$, $K_{3,5}$, and $K_{2,6}$ are displayed in Figure 9.

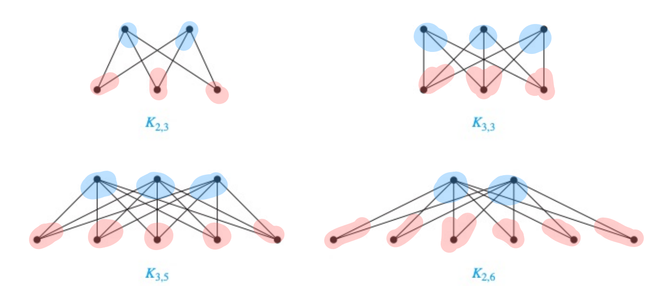


FIGURE 9 Some complete bipartite graphs.

EXAMPLE 16 **Local Area Networks** The various computers in a building, such as minicomputers and personal computers, as well as peripheral devices such as printers and plotters, can be connected using a *local area network*. Some of these networks are based on a *star topology*, where all devices are connected to a central control device. A local area network can be represented using a complete bipartite graph $K_{1,n}$, as shown in Figure 11(a). Messages are sent from device to device through the central control device.

Links

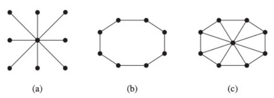


FIGURE 11 Star, ring, and hybrid topologies for local area networks.

Other local area networks are based on a *ring topology*, where each device is connected to exactly two others. Local area networks with a ring topology are modeled using n-cycles, C_n, as shown in Figure 11(b). Messages are sent from device to device around the cycle until the intended recipient of a message is reached.

Finally, some local area networks use a hybrid of these two topologies. Messages may be sent around the ring, or through a central device. This redundancy makes the network more reliable. Local area networks with this redundancy can be modeled using wheels W_n, as shown in Figure 11(c).

10.3 Representing Graphs and Graph Isomorphism

Carrying out graph algorithms using the representation of graphs by lists of edges, or by adjacency lists, can be cumbersome if there are many edges in the graphs. To simplify computation, graphs can be represented using matrices (I.) Adjacency matrices (II.) Incidence matrices

Adjacency lists

EXAMPLE 1 Use adjacency lists to describe the simple graph given in Figure 1.

Solution: Table 1 lists those vertices adjacent to each of the vertices of the graph.

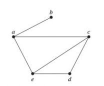


FIGURE 1 A simple graph.

TABLE 1 An Adjacency List for a Simple Graph.

Vertex	Adjacent Vertices
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

EXAMPLE 2 Represent the directed graph shown in Figure 2 by listing all the vertices that are the terminal vertices of edges starting at each vertex of the graph.

Solution: Table 2 represents the directed graph shown in Figure 2.

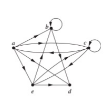


FIGURE 2 A directed graph.

TABLE 2 An Adjacency List for a Directed Graph.

Initial Vertex	Terminal Vertices
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

Usually preferable to use adj. list rather than adj. matrix to represent the graph, when it contains relatively few edges (sparse graphs)
Graphe creux

Adjacency Matrices

$A=[a_{ij}]$, with $a_{ij}=\begin{cases}1 & \text{, if } \{v_i, v_j\} \text{ is an edge of } G\\ 0 & \text{, Otherwise}\end{cases}$

EXAMPLE 3 Use an adjacency matrix to represent the graph shown in Figure 3.

Solution: We order the vertices as a, b, c, d. The matrix representing this graph is

$$\begin{bmatrix}0&1&1&1\\1&0&1&0\\1&1&0&0\\1&0&0&0\end{bmatrix}.$$

The adj. matrix of a simple graph is symmetric
each $a_{ii}=0$ for $i=1,2,\dots,n$ because a simple graph has no loops.

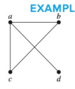


FIGURE 3 A simple graph.

EXAMPLE 4 Draw a graph with the adjacency matrix

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

with respect to the ordering of vertices a, b, c, d.

Solution: A graph with this adjacency matrix is shown in Figure 4.

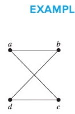


FIGURE 4

Note that an adjacency matrix of a graph is based on the ordering chosen for the vertices.
Hence, there may be as many a $n!$ different adjacency matrices for a graph with n vertices.

Incidence Matrices

$M=[m_{ij}]$, where $m_{ij}=\begin{cases}1 & \text{when edge } e_j \text{ is incident with } v_i\\ 0 & \text{, otherwise}\end{cases}$

EXAMPLE 6 Represent the graph shown in Figure 6 with an incidence matrix.

Solution: The incidence matrix is

$$\begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6\\ \hline v_1 & 1&1&0&0&0&0\\ v_2 & 0&0&1&1&0&1\\ v_3 & 0&0&0&0&1&1\\ v_4 & 1&0&1&0&0&0\\ v_5 & 0&1&0&1&1&0\end{array}$$

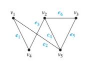


FIGURE 6 An undirected graph.

Incidence matrices can also be used to represent multiple edges and loops. Multiple edges are represented in the incidence matrix using columns with identical entries, because these edges are incident with the same pair of vertices. Loops are represented using a column with exactly one entry equal to 1, corresponding to the vertex that is incident with this loop.

EXAMPLE 7 Represent the pseudograph shown in Figure 7 using an incidence matrix.

Solution: The incidence matrix for this graph is

$$\begin{array}{c|cccccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8\\ \hline v_1 & 1&1&1&0&0&0&0&0\\ v_2 & 0&1&1&1&0&1&1&0\\ v_3 & 0&0&0&1&1&0&0&0\\ v_4 & 0&0&0&0&0&0&1&1\\ v_5 & 0&0&0&0&1&1&0&0\end{array}$$

Pseudograph

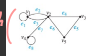


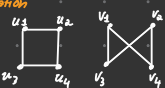
FIGURE 7

Isomorphism of Graphs

We often need to know whether it is possible to draw two graphs in the same way.
That is, do the graphs have the same structure when we ignore the identities of their vertices?

Definition 1. The simple graphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$ are **isomorphic** if there exists a one-to-one correspondance function f from V_1 to V_2 with the property that a and b are adjacent in G_1 iff $f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1. Such function f is called **isomorphism**. Two simple graphs that are non isomorphic are called **nonisomorphic**.
Isomorphism of simple graphs in an equivalence relation

Example 8 | Show that $G=(V,E)$ and $H=(W,F)$ are isomorphic.



The function f with:
$$\begin{cases} f(u_1)=v_1\\ f(u_2)=v_4\\ f(u_3)=v_3\\ f(u_4)=v_2\end{cases}$$
is a one-to-one correspondance between V and W.

To see that this correspondence preserves adjacency, note that adjacent in G are u_1 and u_2, u_1 and u_3, u_2 and u_4, and u_3 and u_4, and each of the pairs $f(u_1)=v_1$ and $f(u_2)=v_4$, $f(u_1)=v_1$ and $f(u_3)=v_3$, $f(u_2)=v_4$ and $f(u_4)=v_2$, and $f(u_3)=v_3$ and $f(u_4)=v_2$ consists of two adjacent vertices in H.

best algo known is $O((\log n)^3)$

It is often difficult to determine whether two simple graphs are isomorphic. There are $n!$ possible one-to-one correspondense between the vertex sets of two simple graphs with n vertices.
A property preserved by isomorphism of graphs is called a **graph invariant**. (I.) Same number of vertices
(II.) Same number of edges
(III.) The degrees of the vertices must be the same (and adjacent)
} All invariants under isomorphism

If any of these quantities differ in two simple graphs, these graphs cannot be isomorphic.
However, when these invariants are the same, it does not necessarily means that the two graphs are isomorphic

EXAMPLE 11 Determine whether the graphs G and H displayed in Figure 12 are isomorphic.

Solution: Both G and H have six vertices and seven edges. Both have four vertices of degree two and two vertices of degree three. It is also easy to see that the subgraphs of G and H consisting of all vertices of degree two and the edges connecting them are isomorphic (as the reader should verify). Because G and H agree with respect to these invariants, it is reasonable to try to find an isomorphism f.

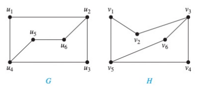


FIGURE 12 Graphs G and H.

We now will define a function f and then determine whether it is an isomorphism. Because $\deg(u_1) = 2$ and because u_1 is not adjacent to any other vertex of degree two, the image of u_1 must be either v_4 or v_6, the only vertices of degree two in H not adjacent to a vertex of degree two. We arbitrarily set $f(u_1) = v_6$. [If we found that this choice did not lead to isomorphism, we would then try $f(u_1) = v_4$.] Because u_2 is adjacent to u_1, the possible images of u_2 are v_3 and v_5. We arbitrarily set $f(u_2) = v_3$. Continuing in this way, using adjacency of vertices and degrees as a guide, we set $f(u_3) = v_4$, $f(u_4) = v_5$, $f(u_5) = v_1$, and $f(u_6) = v_2$. We now have a one-to-one correspondence between the vertex set of G and the vertex set of H, namely, $f(u_1) = v_6$, $f(u_2) = v_3$, $f(u_3) = v_4$, $f(u_4) = v_5$, $f(u_5) = v_1$, $f(u_6) = v_2$. To see whether f preserves edges, we examine the adjacency matrix of G,

$$\mathbf{A}_G = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{array} \begin{array}{c} \begin{array}{cccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array},$$

and the adjacency matrix of H with the rows and columns labeled by the images of the corresponding vertices in G,

$$\mathbf{A}_H = \begin{array}{c} \\ v_6 \\ v_3 \\ v_4 \\ v_5 \\ v_1 \\ v_2 \end{array} \begin{array}{c} \begin{array}{cccccc} v_6 & v_3 & v_4 & v_5 & v_1 & v_2 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}.$$

Because $\mathbf{A}_G = \mathbf{A}_H$, it follows that f preserves edges. We conclude that f is an isomorphism, so G and H are isomorphic. Note that if f turned out not to be an isomorphism, we would *not* have established that G and H are not isomorphic, because another correspondence of the vertices in G and H may be an isomorphism.

Example 10 | Determine whether G and H are isomorphic.

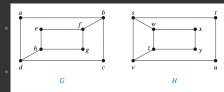


FIGURE 10 The graphs G and H.

Both G and H have 8 vertices and 10 edges.
They also both have four vertices of degree two,
and four of degree three
Because these invariants all agree, it is still conceivable that these graphs are isomorphic.

However, G and H are not isomorphic. To see this, note that because $\deg(a)=2$ in G, a must correspond to either $t, u, x,$ or y in H, because these are the vertices of degree two in H. However, each of these four vertices in H is adjacent to another vertex of deg two in H, which is not true in G.

10.4 Connectivity

Definition 1. Let n be a nonnegative integer and G an undirected graph. A **path** (walk) of length n from u to v in G is a sequence of n edges $e_1, \dots, e_n$ of G for which there exist a sequence $x_0 = u, x_1, \dots, x_{n-1}, x_n = v$ of vertices such that e_i has, for $i = 1, \dots, n$, the endpoints x_{i-1} and x_i.
When the graph is simple, we denote this path by its vertex sequence $x_0, x_1, \dots, x_n$. The path is a **circuit** if it begins and ends at the same vertex, that is, if $u=v$, and has length greater than zero. The path or circuit is said to pass through the vertices $x_1, x_2, \dots, x_{n-1}$ or traverses the edges $e_1, e_2, \dots, e_n$.
Closed walk
A path or circuit is **simple** if it does not contain the same edge more than once.
Simple path ≡ trail   path = trail with no repeated vertices

Definition 2. Let $n \in \mathbb{N}$ and G a directed graph. A path of length n from u to v in G is a sequence of edges $e_1, e_2, \dots, e_n$ of G such that e_1 is associated with (x_0, x_1), e_2 with (x_1, x_2), and so on, with e_n with (x_{n-1}, x_n), where $x_0 = u$ and $x_n = v$. When there are no multiple edges in the directed graph, this path is denoted by its vertex sequence $x_0, x_1, \dots x_n$. A path of length greater than zero that begins and ends at the same vertex is called a **circuit** or **cycle**. A path or circuit is called **simple** ...

Definition 3. A undirected graph is called **connected** if there is a path between every pair of distinct vertices of the graph.
An undirected graph is not connected is called **disconnected**. We say that we disconnect a graph when we remove vertices or edges, or both, to produce a disconnected subgraph.

Theorem 1. There is a simple path between every pair of distinct vertices of a connected undirected graph.

Connected Components

A **connected components** of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G. That is, a connected component of a graph G is a maximal connected subgraph of G. A graph G that is not connected has two or more connected components that are disjoint and have G as their union.

Example 2 | Path in Acquaintanceship Graph

Many social scientist have conjectured that almost every pair of people in the world are linked by a small chain of people, perhaps containing just **five or fewer people**. This would mean that almost every pair of vertices in the acquaintanceship graph containing all people in the world is linked by a path of length not exceeding four.

Example 3 | Path in Collaboration Graphs

In academic collaboration graph of people who have written papers in mathematics, the **Erdős number** of a person m is the length of the shortest path between m and the extremely prolific mathematician Paul Erdős. Idem in the Hollywood graph with the Bacon number for Kevin Bacon.

TABLE 1 The Number of Mathematicians with a Given Erdős Number (as of early 2006).

Erdős Number	*Number of People*
0	1
1	504
2	6,593
3	33,605
4	83,642
5	87,760
6	40,014
7	11,591
8	3,146
9	819
10	244
11	68
12	23
13	5

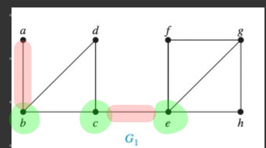
TABLE 2 The Number of Actors with a Given Bacon Number (as of August 2017).

Bacon Number	*Number of People*
0	1
1	3,452
2	401,636
3	1,496,104
4	390,878
5	4,388
6	631
7	131
8	9
9	1

How Connected is a Graph?

We would also like to understand how reliable a network is. For instance, will it still be possible for all computers to communicate after a router or communication link fails? To answer this and similar questions, we now develop some new concepts. Sometimes the removal from a graph of a vertex and all incident edges produces a subgraph with more connected components. Such vertices are called **cut vertices (or articulation points)**. The removal of a cut vertex from a connected graph produces a subgraph that is not connected. Analogously, an edge whose removal produces a graph with more connected components than in the original graph is called a **cut edge** or **bridge**.
Note that in a graph representing a computer network, a cut vertex and a cut edge represent an essential router and an essential link that cannot fail for all computers to be able to communicate.

Example 7 | Find the cut vertices and cut edges in the graph G_1.



The cut vertices are b, c and e.) $\{b,c,e\} = V'$ seperating set. $V' \subseteq V$
The removal of one of these vertices disconnects the graph.
The cut edges are $\{a,b\}$ and $\{c,e\}$.) $\{\{a,b\},\{c,e\}\} = E'$ cut edge
Removing either one of these edges disconnects G_1. $E' \subseteq E$

Applications of vertex and edge connectivity

We can model a data network using vertices to represent routers and edges to represent links between them (fiber optic). We can model a highway network, using vertices to represent highway intersections and edges to represent sections of roads running between intersections.

Vertex Connectivity

Not all graphs have cut vertices. For example, the complete graph K_n, where $n \geq 3$, has no cut vertices. When you remove a vertex from K_n and all edges incident to it, the resulting subgraph is the complete graph K_{n-1}, a connected graph. Connected graphs without cut vertices are called **nonseparable graphs**, and can be thought of as more connected than those with a cut vertex. We define the **vertex connectivity** of a noncomplete graph G, denoted by $K(G)$, as the minimum of vertices in a vertex cut. When G is a complete graph, $K(K_n) = n-1$, the number of vertices needed to be removed to produce a graph with a single vertex
We have $0 \leq K(G) \leq n-1$ if G has n vertices;
$K(G) = 0$, iff G is disconnected or $G = K_1$,
and $K(G) = n-1$ iff G is complete (K_n).
The larger $K(G)$ is, the more connected we consider G to be.

We say that a graph is K-connected (or K-vertex-connected) if $K(G) \geq k$. A graph is 2-connected or biconnected, if it is nonseparable and has at least three vertices.

Connectedness in Directed Graphs

Definition 4. A directed graph is **strongly connected** if there is a path from a to b and b to a whenever a and b are vertices in the graph.
Definition 5. A directed graph is **weakly connected** if there is path between every two vertices in the underlying undirected graph.
Any strongly connected directed graph is also weakly connected.

Example 10 | Are the directed graphs G and H strong connected?

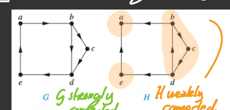


FIGURE 5 The directed graphs G and H.

Example 11 | H has three strongly connected components

Example 8 | Find the vertex connectivity for each of the graphs below.

Example 9 | Find edge connectivity of each of the graphs.

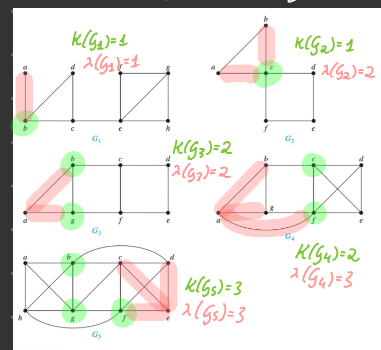


FIGURE 4 Some connected graphs.

An inequality for vertex connectivity and edge connectivity

$$K(G) \leq \lambda(G) \leq \min_{v \in V} \deg(v)$$

To see this observe that deleting all the edges that have a fixed vertex of minimum degree disconnects G, and deleting all the egdes that have a fixed vertex minimum degree as an endpoint disconnects G.

Paths and Isomorphism

There are several ways that paths and circuits can help determine whether two graphs are isomorphic. For example, the existance of a simple circuit of a particular length is a useful invariant that can be used to show that two graphs are not isomorphic.

Example 13 | Determine whether the graphs G and H are isomorphic.

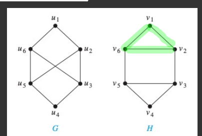


FIGURE 6 The graphs G and H.

Both G and H have six vertices and eight vertices. Each has four vertices of degree three, and two vertices of degree two.
So, the three invariants - number of vertices, number of edges, and degrees of vertices - all agree for the two graphs.
However, H has a simple circuit of length three, namely v_1, v_2, v_6, v_1, whereas G has no simple circuit of length three.
Because the existance of a simple circuit of length three is an isomorphic invariant, G and H are not isomorphic ☒

Example 14 | Determine whether the graphs G and H are isomorphic.

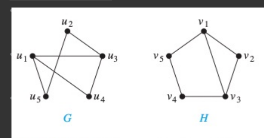


FIGURE 7 The graphs G and H.

We can also use paths to find mappings that are potential isomorphisms.
The paths u_1, u_4, u_3, u_2, u_5 in G and v_3, v_2, v_1, v_5, v_4 in H both go through every vertex in the graph
deg: 3, 4, 3, 2, 2   3, 2, 3, 2, 2
By following these paths through the graphs, we define the mapping f with $f(u_1) = v_3$
⋮
$f(u_5) = v_4$.

Edge Connectivity

We can also measure the connectivity of a connected graph $G = (V, E)$ in terms of the minimum number of edges that we can remove to disconnect it.
The **edge connectivity** of a graph G, denoted by $\lambda(G)$, is the minimum number of edges in an edge cut of G.
Note that $\lambda(G) = 0$ if G is not connected. We also specify that $\lambda(G) = 0$ if G is a graph consisting of a single vertex. It follows that if G is a graph with n vertices, then $0 \leq \lambda(G) \leq n-1$.
$\lambda(G) = n-1$ iff $G = K_n \Rightarrow 0 \leq \lambda(G) \leq n-2$ when G is not a complete graph.

Counting Paths Between Vertices

Theorem 2. Let G be a graph with adjacency matrix A. The number of different paths of length r from v_i to v_j equals the (i,j)th entry of A^r.

$$\text{Number of path of length } r = [a_{ij}]^r$$

Th 2. can be used to find the length of the shortest path betwee two vertices of a graph, and it can also be used to determine whether a graph is connected.

Example 15 | How many path of length four are there from a to d in the simple graph G

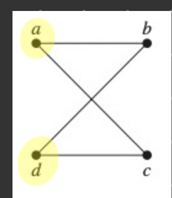


FIGURE 8 graph G.

The adjacency matrix of G (ordering the vertices as a, b, c, d) is

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

Hence, the number of paths of length from a to d is $(1,4)$th entry of A^4.

$$A^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

By inspection of the graph, we see that
abab d; acab d;
abac d; acac d;
abdb d; acdb d;
abdc d; acdc d;

10.5 Euler and Hamilton Paths.

Definition 1. An **Euler circuit** in graph G is a simple circuit containing every edge of G.
An **Euler path** in G is a simple path containing every edge of G.

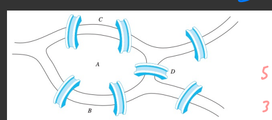


FIGURE 1 The seven bridges of Königsberg.

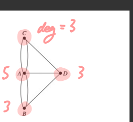


FIGURE 2 Multigraph model of the town of Königsberg.

Example 1 | Which of the undirected graphs have an Euler circuit? Of those that do not, which have an Euler path?

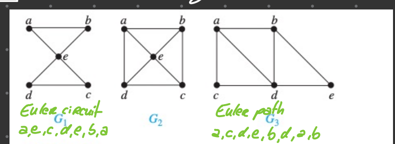


FIGURE 3 The undirected graphs G_1, G_2, and G_3.

Example 2 |

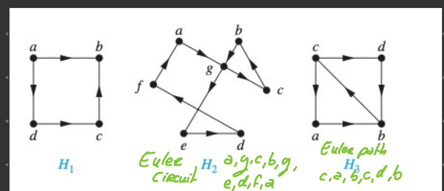


FIGURE 4 The directed graphs H_1, H_2, and H_3.

Necessary and Sufficient Conditions for Euler Circuits and Paths.

Theorem 1. A connected multigraph with at least two vertices has an Euler circuit iff each of its vertices has even degree.

Theorem 2. A connected multigraph has an Euler path but not an Euler circuit iff it has exactly two vertices of odd degree.

ALGORITHM 1 Constructing Euler Circuits.

procedure *Euler*(G: connected multigraph with all vertices of even degree)
circuit := a circuit in G beginning at an arbitrarily chosen vertex with edges successively added to form a path that returns to this vertex
$H := G$ with the edges of this circuit removed
while H has edges
  subcircuit := a circuit in H beginning at a vertex in H that also is an endpoint of an edge of *circuit*
  $H := H$ with edges of *subcircuit* and all isolated vertices removed
  circuit := *circuit* with *subcircuit* inserted at the appropriate vertex
return *circuit* {*circuit* is an Euler circuit}

Worst case complexity is $O(m)$, where m is the number of edges

Example 3 | Many puzzles ask you to draw a picture in a continuous motion without lifting a pencil so that no part of the picture is retraced.
Can Mohammed's scimitars be drawn in this way, where the drawing begins and ends at the same point?

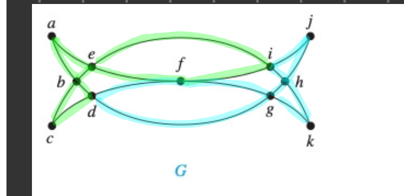


FIGURE 6 Mohammed's scimitars.

We can solve this problem because the graph G has an Euler circuit. It has such a circuit because all its vertices have even degree.
We will use Algo. 1 to construct an Euler circuit.
First we form the circuit a,b,d,c,b,e,i,f,e,a.
We obtain the subgraph H by deleting the edges in this circuit and all vertices that become isolated when these edges are removed.
The we form the circuit d,g,h,j,i,h,k,g,f,d in H. After forming this circuit we have used all edges in G.
Splicing this new circuit into the first circuit at the appropriate place produces the Euler circuit a,b,d,g,h,j,i,h,k,g,f,d,c,b,e,i,f,e,a
This circuit gives a way to draw the scimitars without lifting the pencil or retracing part of the picture.
We can also use Fleury's algorithm (elegant but inefficient)

Example 4 | Which graphs have an Euler path?

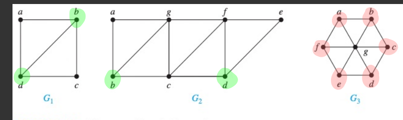


FIGURE 7 Three undirected graphs.

G_1 contains exactly two vertices of odd degree, namely, b and d. Hence, it has an Euler path that must have b and d as its endpoints. One such Euler path is d,a,b,c,d,b.
G_2 idem
G_3 has no Euler path because it has six vertices of odd degree.

Exo 16. A directed multigraph having no isolated vertices has an Euler circuit iff the graph is weakly connected and the in-degree and out-degree of each vertex are equal.
Exo 17. Idem for Euler path the graph is weakly connected and the in-degree and out-degree of each vertex are equal for all but two vertices, one that has in-degree one larger than its out-degree and the other is the opposite.

Hamilton Paths and Circuits

Definition 2. A simple path in a graph G that passes through every vertex exactly once is called a Hamilton path.
/simple circuit /Hamilton circuit.

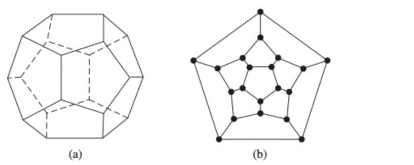


FIGURE 8 Hamilton's "A Voyage Round the World" puzzle.

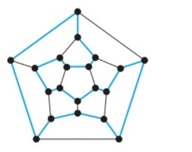


FIGURE 9 A solution to the "A Voyage Round the World" puzzle.

CONDITIONS FOR THE EXISTENCE OF HAMILTON CIRCUITS Is there a simple way to determine whether a graph has a Hamilton circuit or path? At first, it might seem that there should be an easy way to determine this, because there is a simple way to answer the similar question of whether a graph has an Euler circuit. Surprisingly, there are no known simple necessary and sufficient criteria for the existence of Hamilton circuits. However, many theorems are known that give sufficient conditions for the existence of Hamilton circuits. Also, certain properties can be used to show that a graph has no Hamilton circuit. For instance, a graph with a vertex of degree one cannot have a Hamilton circuit, because in a Hamilton circuit, each vertex is incident with two edges in the circuit. Moreover, if a vertex in the graph has degree two, then both edges that are incident with this vertex must be part of any Hamilton circuit. Also, note that when a Hamilton circuit is being constructed and this circuit has passed through a vertex, then all remaining edges incident with this vertex, other than the two used in the circuit, can be removed from consideration. Furthermore, a Hamilton circuit cannot contain a smaller circuit within it.

Example 5 | Which of the simple graphs have a Hamilton circuit or, if not, a Hamilton path?

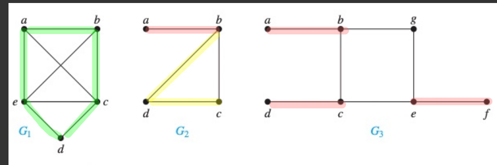


FIGURE 10 Three simple graphs.

G_1 has a Hamilton circuit: a,b,c,d,e,a
There is no Hamilton circuit in G_2 (this can be seen by noting that any circuit containing every vertex must contain the edge {a,b} twice), but G_2 does have a Hamilton path, namely a,b,c,d.
G_3 has neither a Hamilton circuit nor a Hamilton path, because any path containing all vertices must contain one of the edges {a,b}, {e,f}, and {c,d} more than once.

Example 6 | Show that neither graph has a Hamilton circuit.

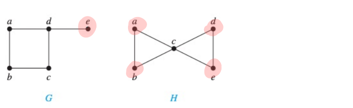


FIGURE 11 Two graphs that do not have a Hamilton circuit.

There is no Hamilton circuit in G because it has a vertex of degree one, namely e.
Now consider H. Because the degrees of the vertices a, b, d, and e are all two, every edge incident with these vertices must be part of any Hamilton circuit.
It is now easy to see that no Hamilton circuit can exist in H, for any Hamilton circuit would have to contain four incident with c, which is impossible.

Example 7 | Show that K_n has a Hamilton circuit whenever $n \geq 3$.
This possible because there are edges in K_n between any two vertices.

Note that the more edges a graph has, the more likely it is to have a Hamilton circuit. Furthermore, adding edges (but not vertices) to a graph with a Hamilton circuit produces a graph with same Hamilton circuit.
So as we add edges to a graph, especially when we make sure to add edges to each vertex, we make it increasingly likely that a Hamilton circuit exist in this graph.

Theorem 3. Dirac's Theorem
If G is a simple graph with n vertices with $n \geq 3$ such that the degree of every vertex in G is at least $\frac{n}{2}$, then G has a Hamilton circuit.

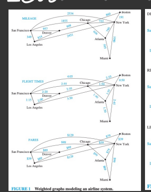
Theorem 4. Ore's Theorem (Corollary)
If G is a simple graph with n vertices with $n \geq 3$ such that $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices u and v in G, then G has a Hamilton circuit.

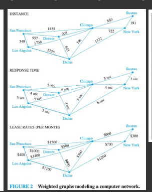
These theorems do not provide necessary conditions for the existence of Hamilton circuit.
For example, the graph C_5 has a Hamilton circuit but does not satisfy the hypotheses of either Ore's Th. or Dirac's Th.

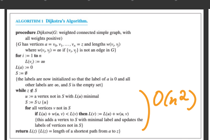
$n = 5 \Rightarrow \frac{n}{2} = 2{,}5$ $\forall v \in V,\ \deg(v) = 2 < 2{,}5$
$\forall u, v \in V$, where u, v nonadjacent, $\deg(u) + \deg(v) = 2 + 2 = 4 < 5$

The best algorithms known for finding a Hamilton circuit in a graph or determining that no such circuit exists have exponential worst-case time complexity.

10.6 Shortest-Path Problems







$O(n^2)$

$L_k(a,v) = \min\{L_{k-1}(a,v);\ L_{k-1}(a,u) + w(u,v)\}$

In performing Dijkstra's algo. it is sometimes more convenient to keep track of labels of vertices in each step using a table instead of redrawing the graph for each step.

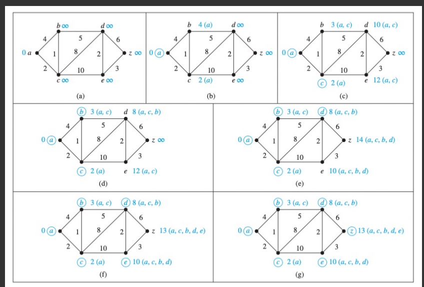


FIGURE 4 Using Dijkstra's algorithm to find a shortest path from a to z.

10.7 Planar Graphs

Consider the problem of joining three houses to each of three separate utilities, as shown in Figure 1. Is it possible to join these houses and utilities so that none of the connections cross? This problem can be modeled using the complete bipartite graph $K_{3,3}$. The original question can be rephrased as: Can $K_{3,3}$ be drawn in the plane so that no two of its edges cross?

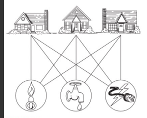
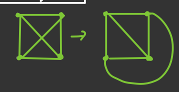


FIGURE 1 Three houses and three utilities.

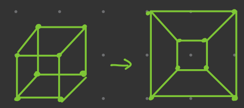
Definition 1. A graph is called planar if it can be drawn in the plane without any edges crossing. Such a drawing is called a planar representation of the graph.

A graph may be planar even if it is usually drawn with crossings, because it may be possible to draw it in a different way without crossings.

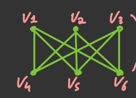
Example 1 | Is K_4 planar?



Example 2 | Is Q_3 planar?

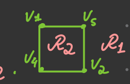


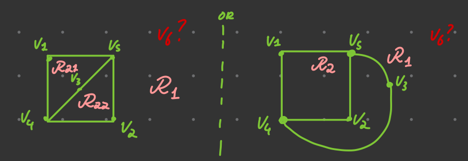
Example 3 | Is $K_{3,3}$ planar?



No...

In any planar representation of $K_{3,3}$, v_1 and v_2 must be connected to both v_4 and v_5. These four edges form a closed curve that splits the plane into two regions, R_1 and R_2. The vertex v_3 is in either R_1 or R_2. When v_3 is in R_2, the inside of the closed curve, the edges between v_3 and v_4 and between v_3 and v_5 separate R_2 into two subregions, R_{21} and R_{22}. Note there is no way to place the final vertex v_6.





Theorem 1. Euler's Formula

Let G be a connected planar graph with e edges and v vertices. Let R be the number of regions in a planar representation of G. Then $R = e - v + 2$

Example 4 | Suppose that a connected planar simple graph has 20 vertices, each of degree 3. Into how many regions does a representation of this planar graph split the plane.

This graph has 20 vertices, each of degree 3, so $v = 20$. Because the sum of the degrees of the vertices, $3v = 3 \cdot 20 = 60$, is equal to twice the number of edges, $2e$, we have $2e = 60$, or $e = 30$. Consequently, from Euler's formula, the number of regions is $R = e - v + 2 = 30 - 20 + 2 = 12$

Corollary 1. If G is a connected planar simple graph with e edges and v vertices, where $v \geq 3$, then $e \leq 3v - 6$

$e \leq 3v - 6$ satisfied does not imply that a graph is planar. E.g. for $K_{3,3}$, $e = 9 \leq 12 = 3 \cdot 6 - 6$

Example 5 | Show that K_5 is nonplanar using corollary 1.

The graph K_5 has five vertices and 10 edges. However, the inequality $e \leq 3v - 6$ is not satisfied for this graph because $e = 10$ and $3v - 6 = 9$. Therefore, K_5 is not planar. ∎

Example 6 | Use Corollary 3 to show that $K_{3,3}$ is nonplanar

Because $K_{3,3}$ has no circuits of length three, Corollary 3 can be used. $K_{3,3}$ has six vertices and nine edges. Because $e = 9$ and $2v - 4 = 8$, $e > 2v - 4$. Therefore $K_{3,3}$ is nonplanar ∎

Corollary 2. If G is a connected planar simple graph, the G has a vertex of degree not exceeding five.

Corollary 3. If a connected planar simple graph has e edges and v vertices with $v \geq 3$ and no circuits of length three, then $e \leq 2v - 4$.

Kuratowski's Theorem

Theorem 2. A graph is nonplanar if and only if it contains a subgraph homeomorphic to $K_{3,3}$ or K_5.

Obtained from the same graph by a sequence of elementary subdivisions.

Removed

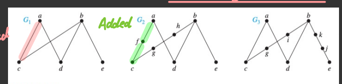


FIGURE 12 Homeomorphic graphs.

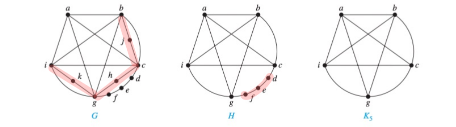


FIGURE 13 The undirected graph G, a subgraph H homeomorphic to K_5, and K_5.

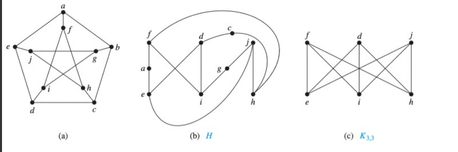


FIGURE 14 (a) The Petersen graph, (b) a subgraph H homeomorphic to $K_{3,3}$, and (c) $K_{3,3}$.

10.8 Graph Coloring

Definition 1. A coloring of a simple graph is the assignment of color to each vertex of the graph so that no two adjacent vertices are assigned the same color, denoted by $\chi(G)$

Definition 2. The chromatic number of a graph is the least number of colors needed for a coloring of this graph.

Theorem 1. The Four Color Theorem

The chromatic number of a planar graph is no greater than four.

Example 1 | What are the chromatic numbers of the graphs G and H?

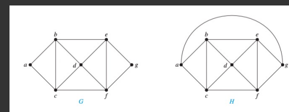


FIGURE 3 The simple graphs G and H.

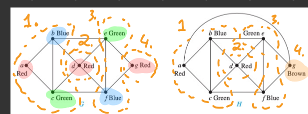


FIGURE 4 Colorings of the graphs G and H.

Hence, $\chi(G) = 3$, $\chi(H) = 4$

Example 2 | What is the chromatic number of K_n?

$\chi(K_n) = n$. Recall that K_n is not planar when $n \geq 5$.

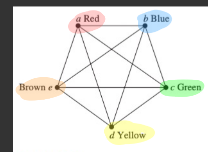


FIGURE 5 A coloring of K_5.

$\chi(K_5) = 5$

Example 3 | What is the chromatic number of the complete bipartite graph $K_{m,n}$, where m and n are positive integers?

Because $K_{m,n}$ is a bipartite graph, $\chi(K_{m,n}) = 2$.

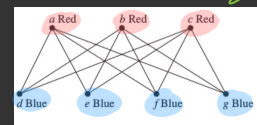


FIGURE 6 A coloring of $K_{3,4}$.

$\chi(K_{3,4}) = 2$

Example 4 | What is the chromatic number of the graph C_n, where $n \geq 3$?

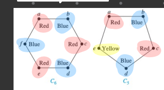


FIGURE 7 Colorings of C_5 and C_6.

$$\chi(C_n) = \begin{cases} 2, & n \text{ even and } n \geq 4 \\ 3, & n \text{ odd and } n \geq 3 \end{cases}$$