Tema 5: Grafos y Árboles

5.1 Introducción a los grafos y árboles

Los **grafos** son estructuras discretas que constan de **vértices** (*vertice, vertex;* también llamados *nodos*) y **aristas** (*edges;* también llamados *trazos, bordes, líneas, conexiones, …*) que conectan estos vértices. Hay diferentes tipos de grafos, dependiendo de si las aristas tienen direcciones, si varias aristas pueden conectar el mismo par de vértices, o bien, si se permiten o no bucles. Los problemas en casi todas las disciplinas imaginables se pueden resolver utilizando modelos de grafos.

Los grafos son herramientas muy útiles que se aplican a una amplia variedad de áreas del conocimiento, como, por ejemplo:

- representar la competencia de diferentes especies de un ecosistema, determinar quién influye en quién en una organización, o bien, dar los resultados de los torneos de todos contra todos.
- contribuir a modelar las relaciones entre personas, la colaboración entre investigadores, las llamadas telefónicas entre los números de teléfono y los vínculos entre los sitios web. Incluso pueden, a su vez, modelar hojas de ruta y asignar trabajos a los empleados de una organización.
- determinar si es posible caminar por todas las calles de una ciudad sin tener que recorrer una calle dos veces, o bien, encontrar la cantidad de colores necesarios para colorear las regiones de un mapa, o también, para determinar si un circuito se puede implementar en una placa de circuito plana, distinguir entre dos compuestos químicos con la misma fórmula molecular pero con diferente estructura espacial, o bien, encontrar si dos computadoras están conectadas por un enlace de comunicaciones utilizando modelos grafos de redes de computadoras.

Los grafos con pesos asignados a sus aristas se usan para resolver problemas tales como, por ejemplo, encontrar el camino más corto entre dos ciudades en una red de transporte. También podemos utilizar grafos para programar exámenes y asignar canales a estaciones de televisión.

Un tipo particular de grafo es el **árbol**, y se llama así porque tales grafos se asemejan a árboles. Este tipo de grafos se utilizan para la toma de decisiones en condiciones de riesgo y para un gran número de áreas de conocimiento.

En este capítulo se introducen los conceptos básicos de la teoría de grafos y se presentan muchos modelos de aplicabilidad diferente. Para resolver la amplia variedad de problemas presentaremos algunos algoritmos de grafos diferentes.

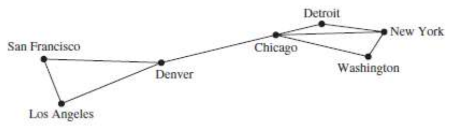


FIGURE 1 A Computer Network.

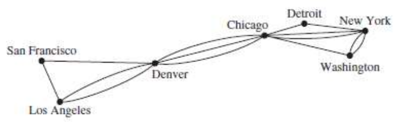


FIGURE 2 A Computer Network with Multiple Links between Data Centers.

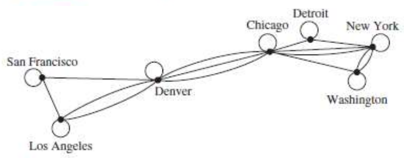


FIGURE 3 A Computer Network with Diagnostic Links.

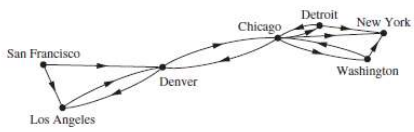


FIGURE 4 A Communications Network with One-Way Communications Links.

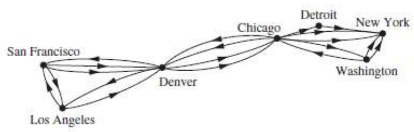
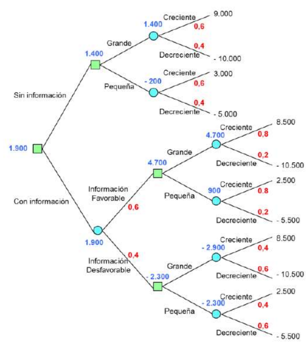


FIGURE 5 A Computer Network with Multiple One-Way Links.



Ej. Árbol de decisión

5.2 Terminología y algunos modelos de grafos

Definición principal:

Un grafo G = (V, E) consta de V, un conjunto no vacío de vértices (o nodos) y E, un conjunto de aristas (trazos, bordes o líneas). Cada arista tiene uno o dos vértices asociados, llamados puntos finales. Se dice que un borde conecta sus puntos finales.

En esta introducción a los grafos solo haremos referencia a los *grafos finitos*, que son aquellos que tienen un número finito de vértices. Dentro de éstos podemos clasificarlos de muchas maneras, una de ellas es a través del número de aristas, su direccionalidad y la existencia o no de bucles. A continuación, resumimos en una tabla esta primera clasificación, mostramos algunas definiciones y teoremas de la terminología básica empleada y, por último, algunos ejemplos de modelos de grafos utilizados.

Tipos de grafos:

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

Terminología básica:

Def.1) Dos vértices *u* y *v* en un grafo no dirigido G se denominan adyacentes (o vecinos) en G si *u* y *v* son puntos finales de una arista *e* de G. Dicha arista *e* se llama *incidente con* los vértices *u* y *v,* y *e* se dice que *conecta u* y *v.*

Def.2) El conjunto de todos los vecinos de un vértice v de G = (V, E), denotado por N (v), se llama vecindad de v. Si A es un subconjunto de V, denotamos por N (A) el conjunto de todos los vértices en G que son adyacentes a al menos un vértice en A. Entonces, N (A) = $\bigcup_{v\in A}$ N (v).

Def.3) El grado de un vértice en un grafo no dirigido es el número de aristas incidentes con él, excepto que un bucle en un vértice contribuye dos veces al grado de ese vértice. El grado del vértice v se denota mediante deg(v).

Th.1 – de *Handshaking:*
Sea *G = (V, E)* un grafo no dirigido con m aristas. Entonces $2m = SUM_{v\in V}\ deg\ (v)$.
(*Tenga en cuenta que esto se aplica incluso si hay múltiples aristas y bucles).*

Th.2
Un grafo no dirigido tiene un número par de vértices de grado impar.

Def.4) Cuando (*u*, *v*) es una arista del grafo G con aristas dirigidas, se dice que *u* es adyacente a *v* y *v* se dice que es adyacente a *u*. El vértice *u* se denomina vértice inicial de (*u, v*) y *v* se llama vértice terminal o final de (*u, v*). El vértice inicial y el vértice terminal de un bucle son iguales.

Def.5) En un grafo con aristas dirigidas, el grado de un vértice *v*, denotado por *deg−(v),* es el número de aristas con v como su vértice terminal. El grado de salida de v, denotado por *deg+ (v),* es el número de aristas con *v* como su vértice inicial.
(Tenga en cuenta que un bucle producido en un vértice contribuye con1 tanto al grado de entrada como al de salida de este vértice).

Th.3
Sea *G = (V, E)* una gráfica con aristas dirigidas, entonces

$$SUM_{v \in V}\ deg^{-}(v) = SUM_{v \in V}\ deg^{+}(v) = |\ E\ |$$

<u>Algunos modelos aplicados de grafos:</u>

A continuación, se detallan algunos ejemplos de grafos aplicados a diferentes áreas de conocimiento:

a. Redes sociales:

a.1) Grafos de conocidos y amistades
a.2) Grafos de influencia
a.3) Grafos colaboración

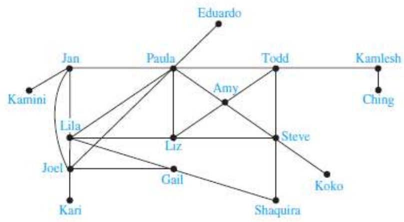


FIGURE 6 An Acquaintanceship Graph.

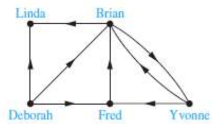


FIGURE 7 An Influence Graph.

b. Redes de comunicación:

b.1) Grafos de llamadas

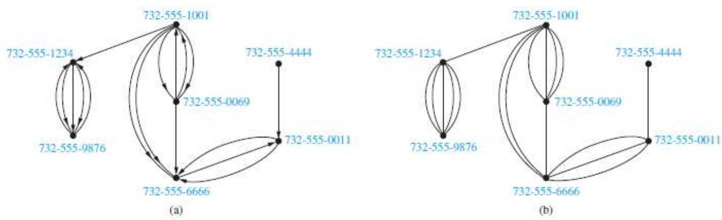


FIGURE 8 A Call Graph.

c. Redes de información:

c.1) Grafos de la web
c.2) Grafos de citas

d. Aplicaciones de diseño de software:

d.1) Grafos de dependencia de módulos
d.2) Grafos de precedencia y procesamiento concurrente

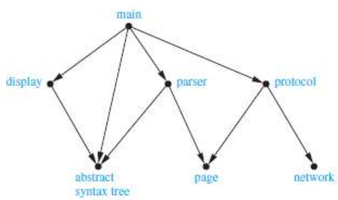


FIGURE 9 A Module Dependency Graph.

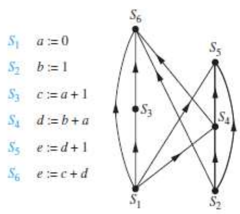


FIGURE 10 A Precedence Graph.

e. Redes de transporte:

e.1) Rutas de aerolíneas
e.2) Redes de carreteras

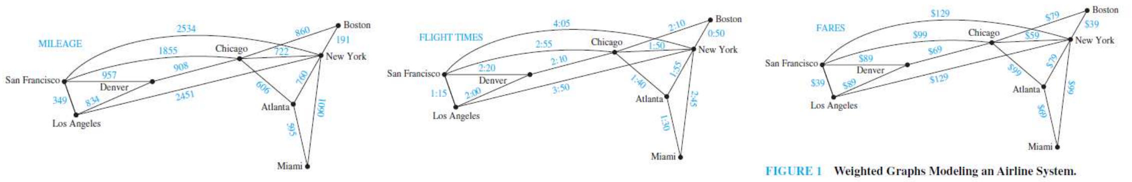


FIGURE 1 Weighted Graphs Modeling an Airline System.

f. Redes biológicas:

f.1) Grafos de superposición de nichos en ecología
f.2) Grafos de interacción de proteínas

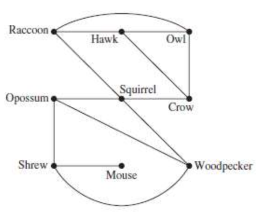


FIGURE 11 A Niche Overlap Graph.

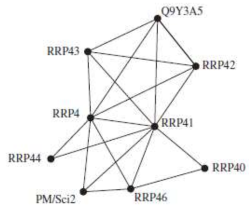


FIGURE 12 A Module of a Protein Interaction Graph.

g. Torneos:

g.1) Torneos Round-Robin
g.2) Torneos de eliminación simple

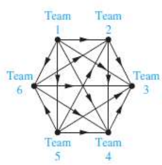


FIGURE 13 A Graph Model of a Round-Robin Tournament.

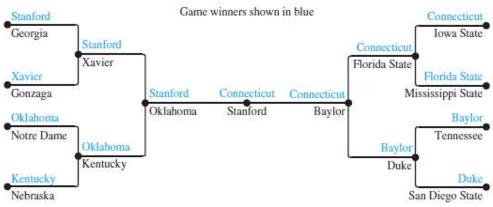


FIGURE 14 A Single-Elimination Tournament.

5.3 Representación y operativa de los grafos

Hay muchas formas útiles de representar los grafos. Como veremos en este capítulo, al trabajar con un grafo es útil poder elegir su representación más conveniente. En esta sección, mostraremos cómo hacerlo de formas diferentes, según la naturaleza de los mismos y la operativa requerida para trabajar con ellos.

En muchas ocasiones, dos grafos tienen exactamente la misma forma, esto es, que existe una correspondencia uno a uno entre sus conjuntos de vértices que conservan las aristas. En tal caso, decimos que los dos grafos son **isomorfos**. Determinar si dos grafos son isomorfos es un problema importante de la teoría de grafos como veremos a continuación.

<u>Representación de grafos:</u>

Una forma de representar un *grafo sin múltiples aristas* es **enumerar las aristas** de este grafo. Otra forma de representar este tipo específico de grafos es usar **listas de adyacencias**, que especifican los vértices adyacentes a cada vértice del grafo.

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. ◀

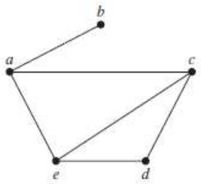


FIGURE 1 A Simple Graph.

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

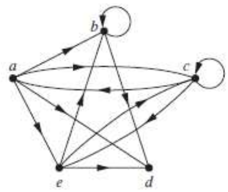
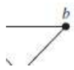


FIGURE 2 A Directed Graph.

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

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. ◀

La utilización de algoritmos en la representación de grafos mediante **listas de aristas** o **listas de adyacencia** puede resultar engorrosa si hay muchos bordes en el grafo. Para simplificar el cálculo, los grafos se pueden representar mediante **matrices**. Aquí se presentarán dos tipos de matrices que se utilizan comúnmente para representar grafos. Uno se basa en la *adyacencia de vértices* y el otro en la *incidencia de vértices y aristas*.

Matrices de adyacencia

Suponga que $G = (V, E)$ es un grafo simple donde $| V | = n$. Suponga que los vértices de G se enumeran arbitrariamente como $v_1, v_2, . . . , v_n$. La **matriz de adyacencia** A (o A_G) de G, con respecto a esta lista de vértices, es la matriz nxn cero-uno, de valor 1 en el término *(i, j)-ésimo* cuando v_i y v_j son adyacentes, y de valor 0 en el término *(i, j)* cuando no son adyacentes. En otras palabras, si su matriz de adyacencia es $A_{nxn} = [a_{ij}]$, entonces

a_{ij} = 1 si $\{v_i, v_j\}$ es una arista de G,
= 0 en caso contrario.

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

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

◀

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.

FIGURE 4 A Graph with the Given Adjacency Matrix.

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

EXAMPLE 5 Use an adjacency matrix to represent the pseudograph shown in Figure 5.

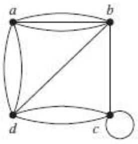


FIGURE 5 A Pseudograph.

Solution: The adjacency matrix using the ordering of vertices a, b, c, d is

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}.$$

◀

We used zero–one matrices in Chapter 9 to represent directed graphs. The matrix for a directed graph $G = (V, E)$ has a 1 in its (i, j)th position if there is an edge from v_i to v_j, where $v_1, v_2, \ldots, v_n$ is an arbitrary listing of the vertices of the directed graph. In other words, if $\mathbf{A} = [a_{ij}]$ is the adjacency matrix for the directed graph with respect to this listing of the vertices, then

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

The adjacency matrix for a directed graph does not have to be symmetric, because there may not be an edge from v_j to v_i when there is an edge from v_i to v_j.

Adjacency matrices can also be used to represent directed multigraphs. Again, such matrices are not zero–one matrices when there are multiple edges in the same direction connecting two vertices. In the adjacency matrix for a directed multigraph, a_{ij} equals the number of edges that are associated to (v_i, v_j).

Cuando un grafo simple contiene relativamente pocas aristas, es decir, cuando es escaso respecto al número de vértices, generalmente es preferible usar listas de adyacencia en lugar de una matriz de adyacencia para representar el grafo.

Ahora suponga que una gráfica simple es densa, es decir, suponga que contiene muchas aristas, como una gráfica que contiene más de la mitad de todas las aristas posibles. En este caso, generalmente es preferible usar una matriz de adyacencia para representar el grafo que usar listas de adyacencia.

Matrices de incidencia

Otra forma común de representar grafos es utilizar **matrices de incidencia**. Sea $G = (V, E)$ un grafo no dirigido con o sin bucles. Suponga que $v_1, v_2, \ldots, v_n$ son los vértices y $e_1, e_2, \ldots, e_m$ son las aristas de G. Entonces, la matriz de incidencia con respecto a este orden de V y E es la matriz $M_{n\times m} = [m_{ij}]$, donde

m_{ij} = 1 cuando la arista e_j es incidente con v_i,
= 0 en caso contrario.

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

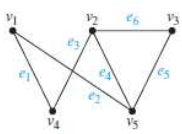


FIGURE 6 An Undirected Graph.

Solution: The incidence matrix is

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

◀

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.

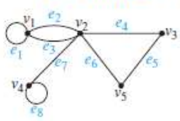


FIGURE 7 A Pseudograph.

Solution: The incidence matrix for this graph is

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

◀

A menudo necesitamos saber si es posible dibujar dos grafos de la misma manera. Es decir, ¿las gráficas tienen la misma estructura cuando ignoramos las identidades de sus vértices?.

Por ejemplo, en química, los grafos se utilizan para modelar compuestos químicos. Diferentes compuestos pueden tener la misma fórmula molecular pero pueden diferir en su forma espacial o estructura. Estos compuestos se pueden representar mediante grafos que no se pueden dibujar de la misma manera. Los grafos que representan compuestos previamente conocidos pueden usarse para determinar si un compuesto supuestamente nuevo se ha estudiado antes.

Existe una terminología útil para grafos con la misma estructura: los **grafos isomorfos**.

<u>Isomorfismo de grafos</u>

Las gráficas simples *G1 = (V1, E1)* y *G2 = (V2, E2)* son isomorfas si existe una función uno a uno *f* de *V1* en *V2* que cumple la siguiente propiedad: a y b son vértices adyacentes en *G1* si y solo si *f(a)* y *f(b)* son vértices adyacentes en *G2*, para todo *a* y *b* en *V1*. Tal función *f* se llama un isomorfismo. Dos grafos simples que no son isomorfos se denominan no isomorfos.

En otras palabras, cuando dos grafos simples son isomórficos, existe una correspondencia uno a uno entre los vértices de los dos grafos que conserva la relación de adyacencia. *El isomorfismo de grafos simples es una relación de equivalencia.*

EXAMPLE 8 Show that the graphs $G = (V, E)$ and $H = (W, F)$, displayed in Figure 8, are isomorphic.

Solution: The function f with $f(u_1) = v_1$, $f(u_2) = v_4$, $f(u_3) = v_3$, and $f(u_4) = v_2$ is a one-to-one correspondence between V and W. To see that this correspondence preserves adjacency, note that adjacent vertices 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. ◀

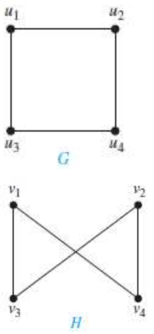


FIGURE 8 The Graphs G and H.

Una propiedad preservada por el isomorfismo de los grafos se denomina **invariante de grafos**.

Por ejemplo, los grafos simples isomorfos deben tener el mismo número de vértices, porque hay una correspondencia uno a uno entre los conjuntos de vértices de los grafos. Los grafos simples isomorfos también deben tener el mismo número de aristas, porque la correspondencia uno a uno entre vértices establece una correspondencia uno a uno entre aristas. Además, los grados de los vértices en grafos simples isomorfos deben ser los mismos. Es decir, un vértice *v* de grado *d* en *G* debe corresponder a un vértice *f(v)* de grado *d* en *H*, porque un vértice *w* en *G* es adyacente a *v* si y solo si *f(v)* y *f(w)* son adyacentes en *H*.

EXAMPLE 9 Show that the graphs displayed in Figure 9 are not isomorphic.

Extra Examples

Solution: Both G and H have five vertices and six edges. However, H has a vertex of degree one, namely, e, whereas G has no vertices of degree one. It follows that G and H are not isomorphic. ◀

The number of vertices, the number of edges, and the number of vertices of each degree are 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 mean that the two graphs are isomorphic. There are no useful sets of invariants currently known that can be used to determine whether simple graphs are isomorphic.

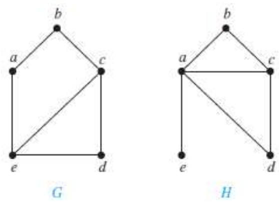


FIGURE 9 **The Graphs G and H.**

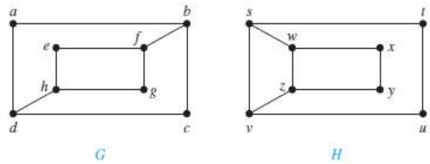


FIGURE 10 **The Graphs G and H.**

EXAMPLE 10 Determine whether the graphs shown in Figure 10 are isomorphic.

Solution: The graphs G and H both have eight 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 degree two in H, which is not true for a in G.

Another way to see that G and H are not isomorphic is to note that the subgraphs of G and H made up of vertices of degree three and the edges connecting them must be isomorphic if these two graphs are isomorphic (the reader should verify this). However, these subgraphs, shown in Figure 11, are not isomorphic. ◀

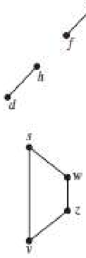


FIGURE 11 **The Subgraphs of G and H Made Up of Vertices of Degree Three and the Edges Connecting Them.**

Para mostrar que una función *f* del conjunto de vértices de un grafo *G* al conjunto de vértices de un grafo *H* es un isomorfismo, debemos demostrar que *f* conserva la presencia y ausencia de aristas. Una forma útil de hacer esto es utilizar matrices de adyacencia. En particular, para mostrar que *f* es un isomorfismo, podemos mostrar que la matriz de adyacencia de G es la misma que la matriz de adyacencia de *H*, cuando las filas y columnas están etiquetadas para corresponder a las imágenes bajo f de los vértices en *G* que son las etiquetas de estas filas y columnas en la matriz de adyacencia de *G*. Ilustramos con el siguiente ejemplo el procedimiento a seguir.

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.

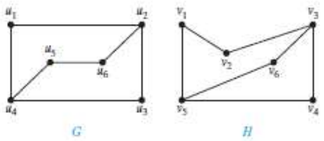


FIGURE 12 **Graphs G and H.**

Ahora definiremos una función f y luego determinaremos si es un isomorfismo:

(1) *Debido a que deg(u1) = 2 y debido a que u1 no es adyacente a ningún otro vértice de grado dos, la imagen de u1 debe ser v4 o v6, los únicos vértices de grado dos en H no adyacentes a un vértice de grado dos. Establecemos arbitrariamente f(u1) = v6. [Si encontráramos que esta elección no condujo a isomorfismo, intentaríamos f(u1) = v4].*

(2) *Como u2 es adyacente a u1, las posibles imágenes de u2 son v3 y v5. Establecemos arbitrariamente f(u2) = v3. Continuando de esta forma, si usamos la adyacencia de vértices y grados como guía, establecemos f(u3) = v4, f(u4) = v5, f(u5) = v1, y f(u6) = v2.*

(3) *Ahora tenemos una correspondencia uno a uno entre el conjunto de vértices de G y el conjunto de vértices de H, es decir, f(u1) = v6, f(u2) = v3, f(u3) = v4, f(u4) = v5, f(u5) = v1, f(u6) = v2.*

Para ver si f conserva las aristas, examinamos la matriz de adyacencia de G,

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

y la matriz de adyacencia de H con las filas y columnas etiquetadas por las imágenes de los vértices correspondientes en G,

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

Como $A_G = A_H$, se deduce que f conserva los bordes. Concluimos que f es un isomorfismo, por lo que G y H son isomorfos. Tenga en cuenta que si f resultara no ser un isomorfismo, no habríamos establecido que G y H no son isomorfos, porque otra correspondencia de los vértices en G y H puede ser un isomorfismo.

5.4 Algunos modelos de grafos y sus algoritmos de resolución:

Existen muchos algoritmos para resolver un buen número de problemas de diferente índole a partir de grafos. Aquí mostramos algunos de ellos:

(1) Problemas de “El camino más corto”: son grafos que se modelan asignando pesos a sus aristas y que simulan distancias, tiempos o tarifas, por ejemplo.

1.1) El “**algoritmo de Dijkstra”** es un buen método para resolver *“El camino más corto”* este tipo de problemas. Al final de este tema (anexo 1) mostramos un ejemplo donde se muestra cómo se aplica este algoritmo:

ALGORITHM 1 Dijkstra's Algorithm.

procedure *Dijkstra*(G: weighted connected simple graph, with all weights positive)
{G has vertices $a = v_0, v_1, \ldots, v_n = z$ and lengths $w(v_i, v_j)$ where $w(v_i, v_j) = \infty$ if $\{v_i, v_j\}$ is not an edge in G}
for $i := 1$ **to** n
  $L(v_i) := \infty$
$L(a) := 0$
$S := \emptyset$
{the labels are now initialized so that the label of a is 0 and all other labels are ∞, and S is the empty set}
while $z \notin S$
  $u :=$ a vertex not in S with $L(u)$ minimal
  $S := S \cup \{u\}$
  for all vertices v not in S
    if $L(u) + w(u, v) < L(v)$ **then** $L(v) := L(u) + w(u, v)$
    {this adds a vertex to S with minimal label and updates the labels of vertices not in S}
return $L(z)$ {$L(z)$ = length of a shortest path from a to z}

1.2) El problema de “**El vendedor ambulante**”: también emplea grafos con pesos que ponderan cada arista. Un vendedor ambulante desea visitar cada una de las n ciudades exactamente una vez y regresar a su punto de partida.
El problema del vendedor ambulante pide el circuito de peso total mínimo en un gráfico ponderado, completo y no dirigido que visita cada vértice exactamente una vez y vuelve a su punto de partida. Esto equivale a pedir un circuito de Hamilton con un peso total mínimo en el gráfico completo, porque cada vértice se visita exactamente una vez en el circuito. La forma más sencilla de resolver una instancia del problema del vendedor ambulante es examinar todos los circuitos de Hamilton posibles y seleccionar uno de longitud total mínima.

(2) Grafos planos: Un gráfico se llama plano si se puede dibujar en el plano sin ningún cruce de bordes (donde un cruce de bordes es la intersección de las líneas o arcos que los representan en un punto distinto de su punto final común). Este dibujo se denomina representación plana del gráfico, muy útil para diseñar redes de carreteras. Euler demostró que todas las representaciones planas de un gráfico dividen el plano en el mismo número de regiones. Lo logró al encontrar una relación entre el número de regiones, el número de vértices y el número de aristas de un gráfico plano. A partir de aquí se desarrollaron diversas variantes en esta dirección.

(3) Coloración de grafos. Una coloración de un gráfico simple es la asignación de un color a cada vértice del gráfico para que no se asigne el mismo color a dos vértices adyacentes. Esto ha sido estudiado durante más de 100 años y ha dado lugar a uno de los más famosos teoremas matemáticos: Teorema de los cuatro colores “*El número cromático de un gráfico plano no es mayor de cuatro*”.

Como referencias bibliográficas de manuales de diseño de algoritmos basados en teoría de grafos cabe destacar:

https://www.amazon.com/exec/obidos/ASIN/1848000693/thealgorithmrepo

5.5 Modelos y propiedades de árboles

Existe un tipo de grafo que modela sistemas a partir de ramificaciones según el criterio, camino, clave o peso que se determine seguir. A este tipo de grafo le llamamos **árbol.** A su vez, puede ser muy variados:

- árboles de búsqueda binaria,
- códigos de prefijo
- arboles de decisión (nos centraremos en esta introducción en este tipo de aplicaciones)
- árboles de juego

De todos ellos, nos centrarnos aquí en aquellos que modelan condiciones de riesgo o incertidumbre y que denominados **árboles de decisión**, porque se basan en vértices o nodos de decisión y nodos de evento, y que podríamos definir como:

Modelos en los que se toman secuencias de decisiones ante diferentes escenarios alternativos posibles y que cada uno de ellos comporta, en principio, un riesgo, consecuencia o probabilidad de que ocurra diferente. El árbol de decisiones es una *representación gráfica y esquemática, cronológica y secuencial, de un problema que busca un objetivo que se quiere alcanza*r. Se considerarán dos tipos de vértices que llamaremos nodos: *nodo de decisión* (■) y *nodo de evento* (o) y las aristas o ramas que les unen representan las diferentes alternativas que existen.

Normalmente, este tipo de problemas plantean la toma de decisión de diferente índole y pretende que a partir de una consecuencia o ponderación de cada potencial rama se optimice la función que se busca. Este tipo de planteamientos gráficos suelen conllevar el teorema Bayes, propio de la probabilidad condicionada.

En el vídeo siguiente se indica cómo se plantea y resuelve un problema a través de la construcción de un árbol de decisión:

https://www.youtube.com/watch?v=rgAY3fczx1M&ab_channel=AlvaroJesusNinaLaura

Para la toma de decisiones en condiciones de riesgos se puede utilizar el criterio del valor monetario esperado o valor esperado: VME o VE. Este valor esperado se utiliza para poder determinar la mejor decisión en función del objetivo del problema planteado y que usualmente suele ser de tipo valor máximo (beneficios, pe) o valor mínimo (costes, pe) deseado. Para ello, se deberá calcular ese VE de cada alternativa posible para así evaluar cuál será la decisión óptima.

En el siguiente vídeo, se recopilan las características más relevantes de dichos valores esperados y que se tienen en cuenta en este tipo de modelos de decisión:

https://www.youtube.com/watch?v=12RF_SZ5S0g&list=PLt_0UnwONqJnhyD3Conb1fMv6h3v2H_x6&ab_channel=AlvaroJesusNinaLaura

Algún otro ejemplo:

https://www.youtube.com/watch?v=cuhd9r_uXz4&list=PLt_0UnwONqJkaDPS49iwIHqfLDsVHmRRh&index=5&ab_channel=AlvaroJesusNinaLaura

5.6 Algunas aplicaciones de árboles para la toma de decisiones.

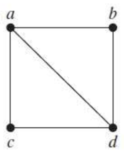
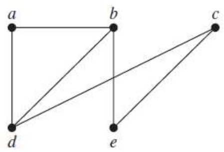
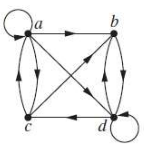
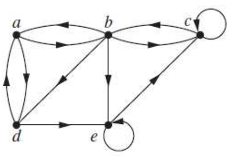
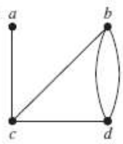
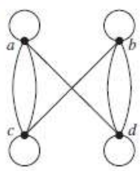
Se puede plantear la resolución de este tipo de problemas siguiendo los pasos del siguiente algoritmo:

- Paso1- Enumerar las diferentes alternativas de decisión
- Paso2 – Enumerar para cada una de las alternativas de decisión, los estados de la naturaleza asociada a la misma
- Paso3 – Desplegar el árbol de decisión
- Paso4 - Asignar las probabilidades de cada uno de los estados de la naturaleza. En este caso se trata de probabilidades a posteriori, por lo que debe utilizar el teorema de Bayes para calcular dichas probabilidades
- Paso5 - Indicar el valor resultante de cada una de las ramas del árbol
- Paso6 - Resolver el árbol de decisión de derecha a izquierda. Dado que la etapa final es probabilística debe aplicar el criterio de la esperanza matemática con el objetivo de determinar el valor resultante esperado de cada alternativa de decisión
- Paso7 – Repita el paso anterior en cada nivel del árbol correspondiente hasta llegar al nodo de decisión que planteaba el problema y vaya colocando el valor esperado (VE) en el nodo de decisión de dicho nivel según sean las condiciones u objetivo perseguido como solución óptima del problema planteado.

Para practicar con algunos ejemplos este algoritmo, se muestra el siguiente enlace, donde se desarrollan algunos ejemplos de diferente índole, siguiendo este algoritmo:

https://upcommons.upc.edu/bitstream/handle/2117/18004/garriga+garzon+problemas+teoria+decision.pdf?sequence=1

5.7 Ejercicios

#1	In Exercises 1–4 use an adjacency list to represent the given graph.	**#2**	In Exercises 13–15 represent the given graph using an adjacency matrix.
	1.  2.  3.  4.  **5.** Represent the graph in Exercise 1 with an adjacency matrix. **6.** Represent the graph in Exercise 2 with an adjacency matrix. **7.** Represent the graph in Exercise 3 with an adjacency matrix. **8.** Represent the graph in Exercise 4 with an adjacency matrix.		**13.**  **14.**  **15.**  In Exercises 16–18 draw an undirected graph represented by the given adjacency matrix. **16.** $\begin{bmatrix} 1 & 3 & 2 \\ 3 & 0 & 4 \\ 2 & 4 & 0 \end{bmatrix}$ **17.** $\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$ **18.** $\begin{bmatrix} 0 & 1 & 3 & 0 & 4 \\ 1 & 2 & 1 & 3 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ 0 & 3 & 0 & 0 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$
#4	Use an incidence matrix to represent the graphs in Exercises 1 and 2.	**#5**	Use an incidence matrix to represent the graphs in Exercises 13–15.

#4

Are the simple graphs with the following adjacency matrices isomorphic?

a) $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

b) $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

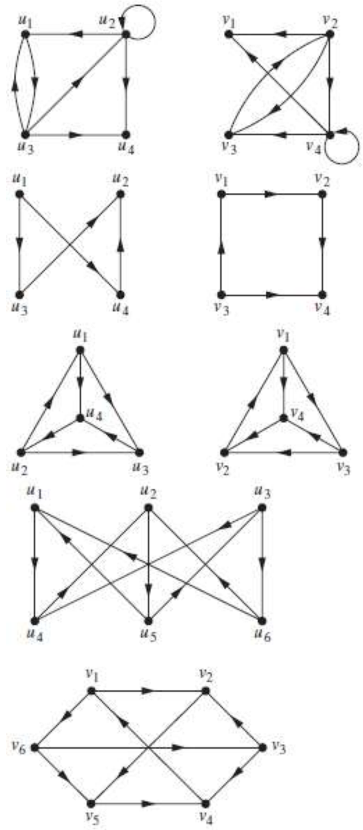
Determine whether the graphs without loops with these incidence matrices are isomorphic.

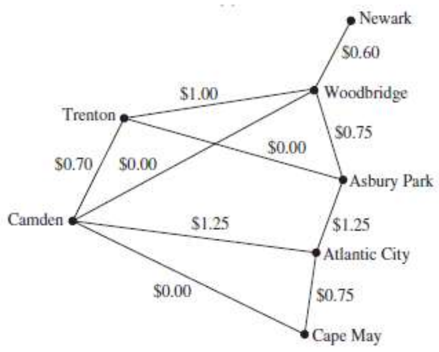
a) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

b) $\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$

#5

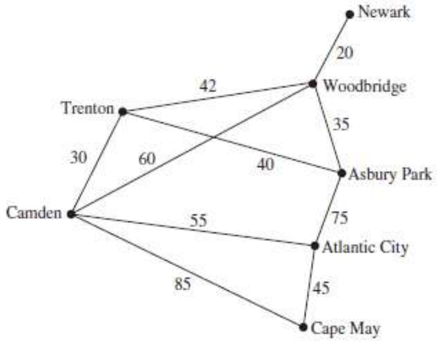
In Exercises 61–64 determine whether the given pair of directed graphs are isomorphic. (See Exercise 60.)



<table>
<tr>
<td>#6</td>
<td>The weighted graphs in the figures here show some major roads in New Jersey. Part (a) shows the distances between cities on these roads; part (b) shows the tolls.</td>
<td rowspan="2">


(b)

a) Find a shortest route in distance between Newark and Camden, and between Newark and Cape May, using these roads.

b) Find a least expensive route in terms of total tolls using the roads in the graph between the pairs of cities in part (a) of this exercise.
</td>
</tr>
<tr>
<td></td>
<td>


(a)
</td>
</tr>
</table>

#7 Se adjuntan, como documentación adicional, otros ejemplos para practicar técnicas de resolución a partir de árboles para la toma de decisiones:

http://www.dmae.upct.es/~mcruiz/Telem06/Teoria/arbol_decision.pdf

https://www.docsity.com/es/ejercicios-resueltos-arbol-de-decision/5327924/

https://www.plandemejora.com/ejercicios-resueltos-de-arbol-de-decisiones/

5.8 Referencias

Basado en los cap's. 10 y 11 de Rosen, K. "Discrete Mathematics and Its Applications", McGrawHill (7th edition, 2012)

Garriga Garzón, F. "Problemas resueltos de la Teoría de la Decisión", OmniaScience Publisher SL (2013)

Anexo 1

EXAMPLE 2 Use Dijkstra's algorithm to find the length of a shortest path between the vertices a and z in the weighted graph displayed in Figure 4(a).

Solution: The steps used by Dijkstra's algorithm to find a shortest path between a and z are shown in Figure 4. At each iteration of the algorithm the vertices of the set S_k are circled. A shortest path from a to each vertex containing only vertices in S_k is indicated for each iteration. The algorithm terminates when z is circled. We find that a shortest path from a to z is a, c, b, d, e, z, with length 13.

Remark: In performing Dijkstra's algorithm 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.

Next, we use an inductive argument to show that Dijkstra's algorithm produces the length of a shortest path between two vertices a and z in an undirected connected weighted graph. Take as the inductive hypothesis the following assertion: At the kth iteration

(i) the label of every vertex v in S is the length of a shortest path from a to this vertex, and
(ii) the label of every vertex not in S is the length of a shortest path from a to this vertex that contains only (besides the vertex itself) vertices in S.

When $k = 0$, before any iterations are carried out, $S = \emptyset$, so the length of a shortest path from a to a vertex other than a is ∞. Hence, the basis case is true.

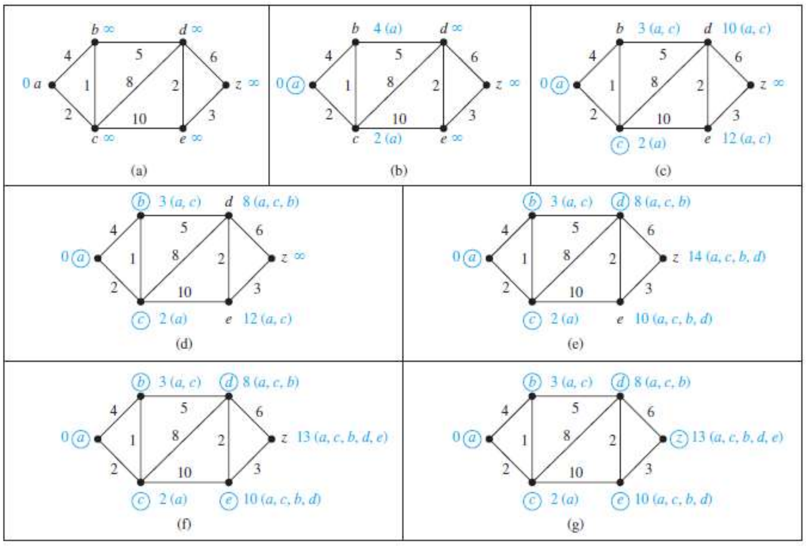


FIGURE 4 **Using Dijkstra's Algorithm to Find a Shortest Path from a to z.**