

8

Teoría de grafos

CAPÍTULO

8.1 INTRODUCCIÓN, ESTRUCTURA DE DATOS

Grafos, grafos dirigidos, árboles y árboles binarios se utilizan en muchas áreas de las matemáticas y de la computación. Dichos temas se cubrirán en este capítulo y en los dos próximos. No obstante, con el fin de comprender cómo se almacenan estos objetos en la memoria y entender sus algoritmos, es necesario conocer ciertas estructuras de datos. Aquí se supondrá que el lector comprende los arreglos lineales y bidimensionales; por tanto, a continuación sólo se estudiarán listas ligadas y apuntadores, así como pilas y colas.

Listas ligadas y apuntadores

Las listas ligadas y los apuntadores se presentarán por medio de un ejemplo. Suponga que una empresa de corredería mantiene un archivo en el que cada registro contiene el nombre de un cliente y el de un vendedor; por ejemplo, el archivo contiene los datos siguientes:

Cliente	Adams	Brown	Clark	Drew	Evans	Farmer	Geller	Hiller	Infeld
Vendedor	Smith	Ray	Ray	Jones	Smith	Jones	Ray	Smith	Ray

Hay dos operaciones básicas que a veces es necesario efectuar con los datos:

Operación A: dado el nombre de un cliente, encontrar su vendedor.

Operación B: dado el nombre de un vendedor, encontrar la lista de sus clientes.

Enseguida se analizan varias formas para almacenar los datos en una computadora, así como la facilidad con que es posible realizar las operaciones *A* y *B* con los datos.

Resulta evidente que el archivo se puede almacenar en la computadora por medio de un arreglo con dos renglones (o columnas) de nueve nombres. Puesto que los nombres de los clientes están en orden alfabético, es fácil efectuar la operación *A*. No obstante, para efectuar la operación *B* es necesario buscar a través de todo el arreglo.

Es fácil almacenar los datos en la memoria si usa un arreglo bidimensional en el que, por ejemplo, los renglones correspondan a una lista en orden alfabético de los nombres de los clientes y las columnas correspondan a una lista en orden alfabético de los nombres de los vendedores, y en cuya matriz haya un 1 que indica el vendedor de un cliente y haya ceros en el resto de la matriz. La desventaja más importante de esta representación es que se desperdicia bastante memoria debido a que en la matriz puede haber muchos ceros. Por ejemplo, si la firma tiene 1 000 clientes y 20 vendedores, podría ser necesario contar con 20 000 localizaciones de memoria para los datos, aunque sólo 1 000 de ellas serían útiles.

A continuación se analiza una forma para almacenar los datos en la memoria, en la cual se utilizan listas ligadas y apuntadores. Una *lista ligada* es una colección lineal de elementos de datos, denominados *nodos*, en la que el orden lineal se proporciona por medio de un campo de apuntadores. La figura 8-1 es un esquema de una lista ligada con seis nodos. Cada nodo está dividido en dos partes: la primera contiene la información del elemento (por ejemplo, NAME, ADDRESS, ...) y la segunda parte, denominada *campo liga* (*link field*) o *campo apuntador al siguiente elemento*

(*nextpointer field*), contiene la dirección del siguiente nodo en la lista. Este campo apuntador se indica con una flecha trazada de un nodo al siguiente nodo en la lista. En la figura 8-1 también hay un apuntador variable, denominado START, que proporciona la dirección del primer nodo en la lista. Además, el campo apuntador del último nodo contiene una dirección no válida, denominada *apuntador nulo*, que indica el fin de la lista.

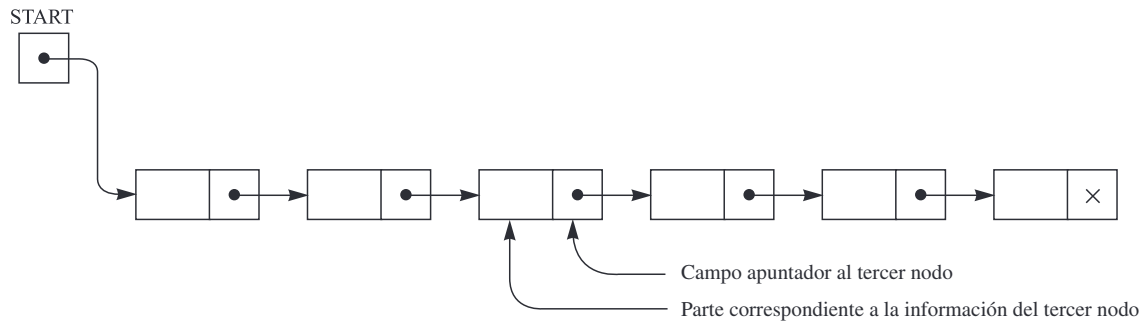


Figura 8-1 Lista ligada con seis nodos

Una forma primordial para almacenar los datos originales representados en la figura 8-2 utiliza listas ligadas. Observe que hay arreglos por separado (en orden alfabético) para los nombres de clientes y de los vendedores. También hay un arreglo apuntador SLSM paralelo a CUSTOMER que proporciona la ubicación del vendedor de un cliente, de modo que la operación *A* puede efectuarse muy rápida y fácilmente. Además, la lista de clientes de cada vendedor es una lista ligada, como ya se analizó. En efecto, hay un arreglo apuntador START paralelo a SALESMAN que indica al primer cliente de un vendedor, y hay un arreglo NEXT que indica la ubicación del siguiente cliente en la lista del vendedor (o contiene un 0 para indicar el fin de la lista). Este proceso se indica mediante las flechas en la figura 8-2 para el vendedor Ray.

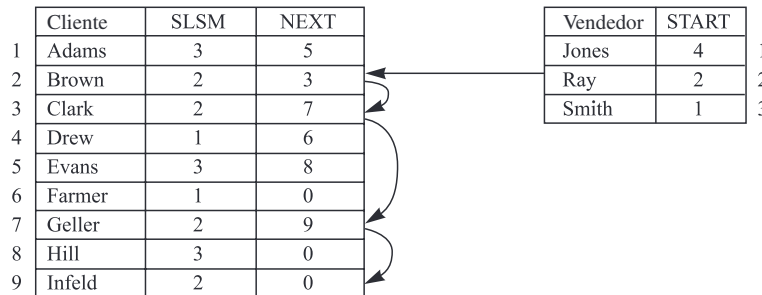


Figura 8-2

Ahora la operación *B* puede efectuarse fácil y rápidamente; no es necesario buscar en la lista de todos los clientes para obtener la lista de clientes de un vendedor dado. En la figura 8-3 se proporciona un algoritmo así (que está escrito en pseudocódigo).

Pilas, colas y colas prioritarias

Además de los arreglos y las listas ligadas hay otras estructuras de datos que aparecen en los algoritmos de grafos. Estas estructuras, pilas, colas y colas prioritarias se describen brevemente a continuación.

- a) **Pila:** también denominada sistema *último en entrar, primero en salir* (LIFO: *last-in, first-out*), es una lista lineal tal que las inserciones y las eliminaciones pueden llevarse a cabo sólo en un extremo, denominado “parte superior” de la lista. Esta estructura es semejante en su operación a una pila de platos montada en un sistema de resorte, como se muestra en la figura 8-4a). Observe que los nuevos platos se insertan sólo en la parte superior de la pila y que los platos pueden retirarse sólo de la parte superior de la pila.

Algoritmo 8.1 Se lee el nombre de un vendedor y se imprime la lista de sus clientes.

Paso 1. Leer XXX.

Paso 2. Encontrar K tal que $\text{SALESMAN}[K] = \text{XXX}$. [Usar búsqueda binaria].

Paso 3. Sea $\text{PTR} := \text{START}[K]$. [Inicializa el apuntador PTR].

Paso 4. Repetir while $\text{PTR} \neq \text{NULL}$.

a) Print $\text{CUSTOMER}[\text{PTR}]$.

b) Set $\text{PTR} := \text{NEXT}[\text{PTR}]$. [Actualiza PTR].

[Fin del ciclo].

Paso 5. Salir.

Figura 8-3

- b) **Cola:** también denominada sistema *primero en entrar, primero en salir* (FIFO: *first-in first-out*), es una lista lineal tal que las eliminaciones pueden llevarse a cabo sólo en un extremo de la lista, denominado “frente” de la lista, y las inserciones pueden llevarse a cabo sólo en el otro extremo de la lista, denominado “parte trasera” de la lista. La estructura opera de forma bastante parecida a una cola de personas que esperan en una parada de autobús, como se muestra en la figura 8-4b). Es decir, la primera persona en la cola es la primera que aborda el autobús, y una persona recién llegada se coloca al final de la cola.
- c) **Cola prioritaria:** sea S un conjunto de elementos en el que pueden insertarse periódicamente nuevos elementos, aunque siempre se elimina el mayor elemento actual (el elemento con la “prioridad más alta”). Entonces S se denomina *cola prioritaria*. Las reglas “mujeres y niños primero” y “la edad antes que la belleza” son ejemplos de cola prioritaria. Las pilas y las colas normales son tipos especiales de cola prioritaria. En efecto, el elemento con la prioridad más alta en una pila es el último elemento insertado, pero el elemento con la prioridad más alta en una cola es el primer elemento insertado.



Figura 8-4

8.2 GRAFOS Y MULTIGRAFOS

Un grafo G consta de dos partes:

- i) Un conjunto $V = V(G)$ cuyos elementos se denominan *vértices*, *puntos* o *nodos* de G .
- ii) Un conjunto $E = E(G)$ de pares no ordenados de vértices distintos denominados *aristas* de G .

Cuando se desea recalcar las dos partes de un grafo G , grafo se denota $G(V, E)$.

Los vértices u y v son *adyacentes* o *vecinos* si hay una arista $e = \{u, v\}$. En este caso, u y v se denominan *extremos* de e , y se dice que e *conecta* o *une* u y v , o también que la arista e es *incidente* (o que incide) en cada uno de sus extremos u y v . Los grafos se representan mediante diagramas en el plano de forma natural. Específicamente, cada vértice v en V se representa por un punto (o un círculo pequeño), y cada arista $e = \{v_1, v_2\}$ se representa por una curva que une sus puntos extremos v_1 y v_2 . Por ejemplo, la figura 8-5a) representa el grafo $G(V, E)$, donde:

- i) V consta de los vértices A, B, C, D .
- ii) E consta de las aristas $e_1 = \{A, B\}$, $e_2 = \{B, C\}$, $e_3 = \{C, D\}$, $e_4 = \{A, C\}$, $e_5 = \{B, D\}$.

De hecho, un grafo suele denotarse al trazar su diagrama en lugar de enumerar explícitamente sus vértices y aristas.

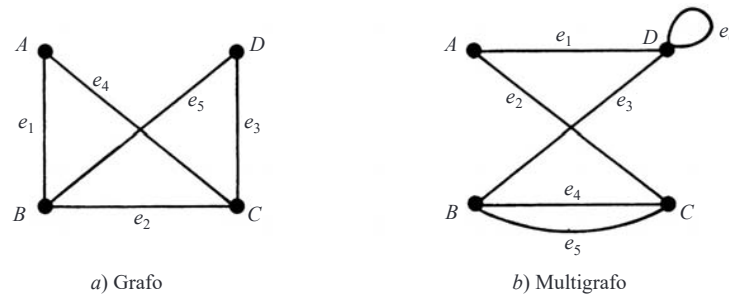


Figura 8-5

Multigrafos

Considere el diagrama en la figura 8-5b). Las aristas e_4 y e_5 se denominan *aristas múltiples* puesto que unen los mismos puntos extremos, y la arista e_6 se denomina *lazo* porque sus extremos tienen el mismo vértice. Este diagrama se denomina *multigrafo*; la definición formal de grafo no permite aristas múltiples ni lazos. Por tanto, un grafo se define como un multigrafo sin aristas múltiples ni lazos.

Observación: En algunos textos el término *grafo* se usa para incluir multigrafos y el término *grafo simple* para indicar un grafo sin aristas múltiples ni lazos.

Grado de un vértice

El *grado* de un vértice v en un grafo G , se escribe $\text{grd}(v)$, es igual al número de aristas en G que contienen a v ; es decir, que inciden sobre v . Puesto que cada arista se cuenta dos veces al contar los grados de los vértices de G , se tiene el siguiente resultado sencillo pero importante.

Teorema 8.1: La suma de los grados de los vértices de un grafo G es igual al doble del número de aristas en G .

Considere, por ejemplo, el grafo de la figura 8-5a). Se tiene

$$\text{grd}(A) = 2, \quad \text{grd}(B) = 3, \quad \text{grd}(C) = 3, \quad \text{grd}(D) = 2.$$

La suma de los grados es igual a 10 que, como era de esperar, es el doble del número de aristas. Un vértice es *par* o *impar* si su grado es un número par o impar. Por tanto, A y D son vértices pares, mientras B y C son vértices impares.

El teorema 8.1 también se cumple para multigrafos en las que un lazo se cuenta dos veces para el grado de ese vértice. Por ejemplo, en la figura 8-5b) se tiene $\text{grd}(D) = 4$, puesto que la arista e_6 se cuenta dos veces; por tanto, D es un vértice par.

Un vértice de grado cero se denomina vértice *aislado*.

Grafos finitos, grafos triviales

Un multigrafo se dice que es *finito* si tiene un número finito de vértices y de aristas. Observe como una consecuencia que un grafo con un número finito de vértices y aristas tiene que ser finito. Un grafo con un SÓLO vértice sin ninguna arista, un punto, se llama *grafo trivial*. A menos que se especifique otra cosa, en este libro los multigrafos son finitos.

8.3 SUBGRAFOS, GRAFOS ISOMORFOS Y HOMEOMORFOS

En esta sección se analizan relaciones importantes entre grafos.

Subgrafos

Considere un grafo $G = G(V, E)$. Un grafo $H = H(V', E')$ se denomina *subgrafo* de G si los vértices y las aristas de H están contenidas en los vértices y en las aristas de G ; es decir, si $V' \subseteq V$ y $E' \subseteq E$. En particular:

- i) Un subgrafo $H(V', E')$ de $G(V, E)$ se denomina subgrafo *inducido* por sus vértices V' si su conjunto de aristas E' contiene todas las aristas en G cuyos puntos extremos pertenecen a los vértices en H .
- ii) Si v es un vértice en G , entonces $G - v$ es el subgrafo de G obtenida al eliminar v de G y al eliminar todas las aristas en G que contienen a v .
- iii) Si e es una arista en G , entonces $G - e$ es el subgrafo de G obtenido al eliminar la arista e de G . Grafos isomorfos

Grafos isomorfos

Se dice que los grafos $G(V, E)$ y $G^*(V^*, E^*)$ son *isomorfos* si existe una correspondencia uno a uno $f; V \rightarrow V^*$ tal que $\{u, v\}$ es una arista de G si y sólo si $\{f(u), f(v)\}$ es una arista de G^* . Normalmente no se establece ninguna diferencia entre grafos isomorfos (aun cuando sus diagramas puedan “parecer diferentes”). En la figura 8-6 se proporcionan 10 grafos representados como letras; puede observar que A y R son grafos isomorfos, también lo son F y T , K y X son grafos isomorfos y M, S, V y Z también lo son.

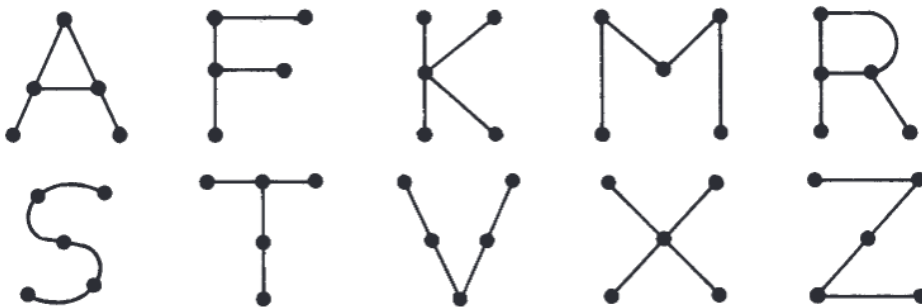


Figura 8-6

Grafos homeomorfos

Dado cualquier grafo G , es posible obtener un nuevo grafo al dividir una arista de G con vértices adicionales. Dos grafos G y G^* son *homeomorfos*, si es posible obtenerlos a partir del mismo grafo o grafos isomorfos al aplicar este método. Los grafos a) y b) en la figura 8-7 no son isomorfos, aunque son homeomorfos puesto que pueden obtenerse a partir del grafo c) al agregar vértices apropiados.

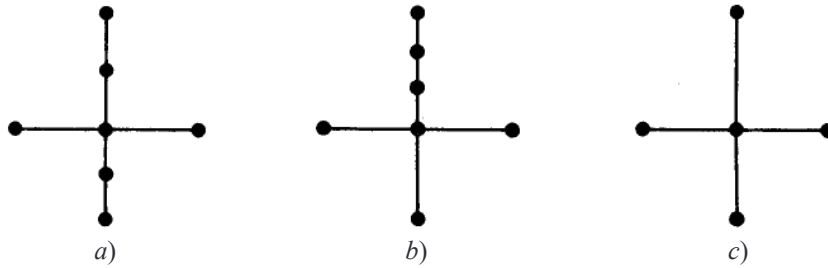


Figura 8-7

8.4 CAMINOS Y CONECTIVIDAD

Un camino en un multigrafo G consta de una secuencia alternada de vértices y aristas de la forma

$$v_0, e_1, v_1, e_2, v_2, \dots, e_{n-1}, v_{n-1}, e_n, v_n$$

donde cada arista e_i contiene a los vértices v_{i-1} y v_i (que aparecen a los lados de e_i en la secuencia). El número n de aristas se denomina *longitud* del camino. Cuando no hay ambigüedad, un camino se denota por su secuencia de vértices (v_0, v_1, \dots, v_n) . Se dice que el camino es *cerrado* si $v_0 = v_n$. En caso contrario, se dice que el camino es de v_0 a v_n o *entre* v_0 y v_n , o que *une* v_0 y v_n .

Un *camino simple* es un camino en el que todos los vértices son distintos. (Un camino en que todas las aristas son diferentes se denomina *recorrido*.) Un *ciclo* es un camino cerrado de longitud 3 o más donde todos los vértices son distintos excepto $v_0 = v_n$. Un ciclo de longitud k se denomina *k-ciclo*.

EJEMPLO 8.1 Considere el grafo G en la figura 8-8a). Considere las siguientes secuencias:

$$\begin{aligned} \alpha &= (P_4, P_1, P_2, P_5, P_1, P_2, P_3, P_6), & \beta &= (P_4, P_1, P_5, P_2, P_6), \\ \gamma &= (P_4, P_1, P_5, P_2, P_3, P_5, P_6), & \delta &= (P_4, P_1, P_5, P_3, P_6). \end{aligned}$$

La secuencia α es un camino de P_4 a P_6 ; pero no es un recorrido porque la arista $\{P_1, P_2\}$ se usa dos veces. La secuencia β no es un camino porque no hay arista $\{P_2, P_6\}$. La secuencia γ es un recorrido porque ninguna arista se usa dos veces; pero no es un camino simple porque el vértice P_5 se usa dos veces. La secuencia δ es un camino simple de P_4 a P_6 ; pero no es el camino más corto (con respecto a la longitud) de P_4 a P_6 . El camino más corto de P_4 a P_6 es el camino simple (P_4, P_5, P_6) , cuya longitud es 2.

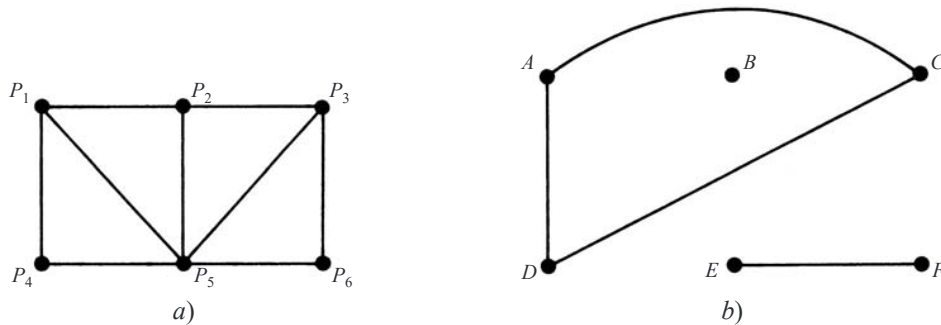


Figura 8-8

Al eliminar aristas innecesarias, no es difícil ver que cualquier camino desde un vértice u hasta un vértice v puede sustituirse por un camino simple de u a v . Este resultado se plantea formalmente a continuación.

Teorema 8.2: Hay un camino de un vértice u a un vértice v si y sólo si existe un camino simple de u a v .

Conectividad, componentes conexos

Un grafo G es *conexo* si existe un camino entre dos de sus vértices. El grafo en la figura 8-8a) es conexo, pero el grafo en la figura 8-8b) no es conexo ya que, por ejemplo, entre los vértices D y E no hay ningún camino.

Suponga que G es un grafo. Un subgrafo conexo H de G se denomina *componente conexo* de G si H no está contenido en ningún subgrafo conexo más grande de G . Resulta intuitivamente claro que cualquier grafo G puede partirse en sus componentes conexos. Por ejemplo, el grafo G en la figura 8-8b) tiene tres componentes conexos, los subgrafos inducidos por los conjuntos de vértices $\{A, C, D\}$, $\{E, F\}$ y $\{B\}$.

El vértice B en la figura 8-8b) se denomina *vértice aislado* porque B no pertenece a ninguna arista o, en otras palabras, $\text{grd}(B) = 0$. En consecuencia, como se observó, B mismo forma un componente conexo del grafo.

Observación: En términos formales, en el supuesto de que cualquier vértice u esté unido consigo mismo, la relación “ u está unido con v ” es una relación de equivalencia sobre el conjunto de vértices de un grafo G y las clases de equivalencia de la relación constituyen los componentes conexos de G .

Distancia y diámetro

Considere un grafo conexo G . La *distancia* entre los vértices u y v en G , que se escribe $d(u, v)$, es la longitud de la ruta más corta entre u y v . El *diámetro* de G , lo cual se escribe $\text{diám}(G)$, es la distancia máxima entre dos puntos cualesquiera en G . Por ejemplo, en la figura 8-9a), $d(A, F) = 2$ y $\text{diám}(G) = 3$, mientras que en la figura 8-9b), $d(A, F) = 3$ y $\text{diám}(G) = 4$.

Puntos de corte y puentes

Sea G un grafo conexo. Un vértice v en G se denomina *punto de corte* si $G - v$ es desconexo. (Recuerde que $G - v$ es el grafo obtenido a partir de G al eliminar v y todas las aristas que contienen a v .) Una arista e de G se denomina *puente* si $G - e$ es desconexo. (Recuerde que $G - e$ es el grafo obtenido a partir de G al eliminar la arista e .) En la figura 8-9a), el vértice D es un punto de corte y no hay puentes. En la figura 8-9b), la arista $\{D, F\}$ es un puente. (Sus puntos extremos D y F son necesariamente puntos de corte.)

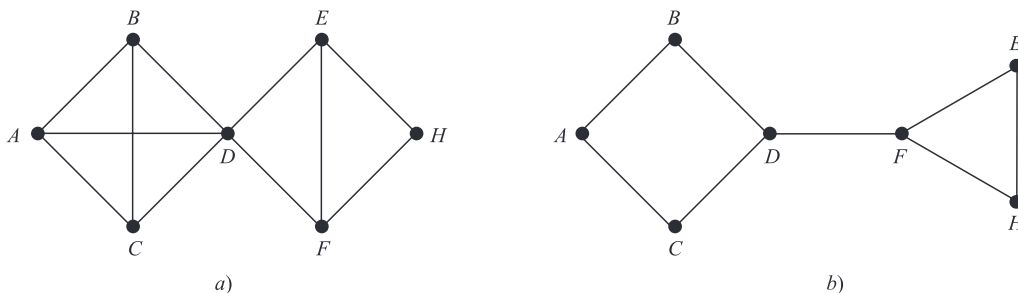


Figura 8-9

8.5 RECORRIDOS Y GRAFOS EULERIANOS, LOS PUENTES DE KÖNIGSBERG

En el siglo XVIII el oriental pueblo prusiano de Königsberg incluía dos islas y siete puentes, como se muestra en la figura 8-10a). Pregunta: si una persona empieza en cualquier punto y termina en cualquier punto, ¿es posible que recorra el pueblo de modo que cruce los siete puentes sin cruzar ninguno dos veces? Los ciudadanos de Königsberg escribieron al célebre matemático suizo L. Euler sobre esta cuestión. Euler demostró en 1736 que tal recorrido es imposible; sustituyó las islas y las dos orillas del río por puntos y los puentes por curvas, con lo que obtuvo la figura 8-10b).

Observe que la figura 8-10b) es un multigrafo. Se dice que un multigrafo es *recorrible* si “la curva puede trazarse sin interrupciones y sin que pase dos veces por cualquiera de las aristas”; es decir, si existe un camino que incluya todos los vértices y use cada arista exactamente una vez. Tal camino debe ser un recorrido (puesto que ninguna arista se usa dos veces), y se denomina *recorrido atravesable* o *recorrible*. Resulta evidente que un multigrafo recorrible debe ser finito y conexo.

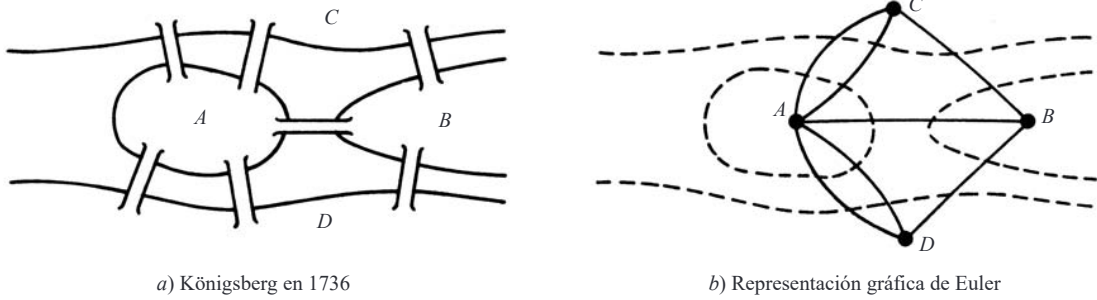


Figura 8-10

A continuación se mostrará cómo Euler probó que el multigrafo en la figura 8-10b) no es recorrible y, por tanto, que el recorrido a pie de Königsberg es imposible. Primero recuerde que un vértice es par o impar si su grado es un número par o impar. Suponga que un multigrafo es recorrible y que un recorrido accesible no empieza o termina en un vértice P . Se afirma que P es un vértice par. Ya que siempre que el recorrido accesible entra a P por una arista, siempre debe haber una arista no usada previamente por el cual el recorrido puede abandonar P . En consecuencia, las aristas del recorrido incidente con P deben aparecer por pares, de modo que P es un vértice par. Por consiguiente, si un vértice Q es impar, entonces el recorrido accesible debe empezar o terminar en Q . En consecuencia, un multigrafo con más de dos vértices impares no puede ser recorrible. Observe que el multigrafo correspondiente al problema de los puentes de Königsberg tiene cuatro vértices impares. Por tanto, no es posible recorrer Königsberg de modo que cada puente se cruce exactamente una vez.

Euler realmente demostró lo contrario del planteamiento anterior, que está contenido en los siguientes teorema y corolario. (El teorema se demuestra en el problema 8.9.). Un grafo G se denomina grafo *euleriano* si existe un recorrido accesible cerrado, denominado *recorrido euleriano*.

Teorema 8.3 (de Euler): Un grafo conexo finito es euleriano si y sólo si cualquier vértice tiene grado par.

Corolario 8.4: Cualquier grafo conexo finito con dos vértices impares es recorrible. Un recorrido accesible puede empezar en cualquier vértice impar y terminar en el otro vértice impar.

Grafos hamiltonianos

En el análisis anterior sobre los grafos Eulerianos se recalcaron las aristas recorridas; aquí la atención se centra en la visita de vértices. Un *circuito hamiltoniano* en un grafo G , así denominado en honor del matemático irlandés del siglo XIX William Hamilton (1803-1865) es un camino cerrado que visita todos los vértices en G exactamente una vez. (Este camino cerrado debe ser un ciclo.) Si G admite un circuito hamiltoniano, entonces G se denomina *grafo hamiltoniano*. Observe que un circuito de Euler recorre cada arista exactamente una vez, aunque puede repetir vértices, mientras que un circuito hamiltoniano visita cada vértice exactamente una vez aunque puede repetir aristas. En la figura 8-11 se proporciona un ejemplo de uno que es hamiltoniano pero no euleriano y viceversa.

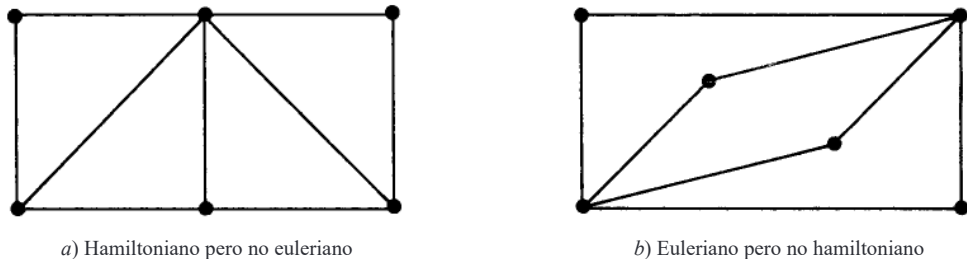


Figura 8-11

Aunque resulta evidente que sólo los grafos conexos pueden ser hamiltonianos, no hay ningún criterio simple para decidir si un grafo es o no hamiltoniano, como sí lo hay para el caso de los grafos eulerianos. Se cuenta con la siguiente condición suficiente, que se debe a G. A. Dirac.

Teorema 8.5: Sea G un grafo conexo con n vértices. Entonces G es hamiltoniano si $n \geq 3$ y $n \leq \text{grd}(v)$ para cada vértice v en G .

8.6 GRAFOS ETIQUETADOS Y PONDERADOS

Un grafo G se denomina *grafo etiquetado* si sus aristas y/o vértices son datos asignados de un tipo o del otro. En particular, G se denomina *grafo ponderado* si a cada arista e de G se asigna un número no negativo $w(e)$ denominado *peso* o *longitud* de v . En la figura 8-12 se muestra un grafo ponderado, en el que el peso de cada arista se proporciona en forma evidente. El *peso* (o la *longitud*) de un camino en tal grafo ponderado G se define como la suma de los pesos de las aristas en el camino. Un problema importante en teoría de grafos es encontrar *el camino más corto*; es decir, un camino de peso (longitud) mínimo(a), entre dos vértices arbitrarios dados. La longitud de un camino más corto entre P y Q en la figura 8-12 es 14; un camino es

$$(P, A_1, A_2, A_5, A_3, A_6, Q)$$

El lector puede tratar de encontrar otro camino más corto.

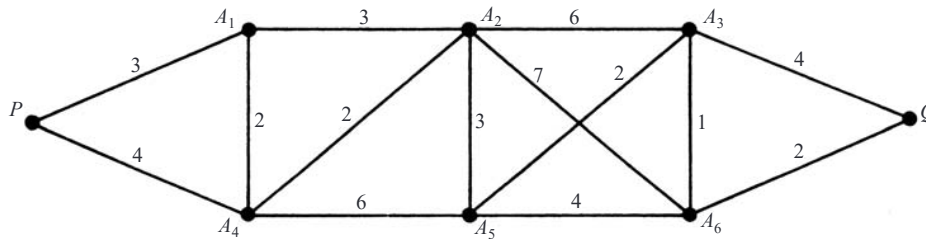


Figura 8-12

8.7 GRAFOS COMPLETOS, REGULARES Y BIPARTIDOS

Hay muchos tipos distintos de grafos. En esta sección se consideran tres: grafos completos, regulares y bipartidos.

Grafos completos

Un grafo G es *completo* si cualquier vértice en G está unido a todos los demás vértices en G . Por tanto, un grafo completo G debe ser conexo. El grafo completo con n vértices se denomina K_n . En la figura 8-13 se muestran los grafos K_1 a K_6 .

Grafos regulares

Un grafo G es *regular de grado k* o *k -regular* si sus vértices tienen grado k , si todos los vértices tienen el mismo grado.

Los grafos regulares conexos de grados 0, 1 o 2 se describen con facilidad. El grafo conexo 0-regular es el grafo trivial con un vértice y sin ninguna arista. El grafo conexo 1-regular es el grafo con dos vértices y una arista que los une. El grafo conexo 2-regular con n vértices es el grafo que consta de un solo n -ciclo. Vea la figura 8-14.

Los grafos 3-regular deben tener un número par de vértices, ya que la suma de los grados de los vértices es un número par (teorema 8.1). En la figura 8-15 se muestran dos grafos 3-regular conexos con seis vértices. En general,

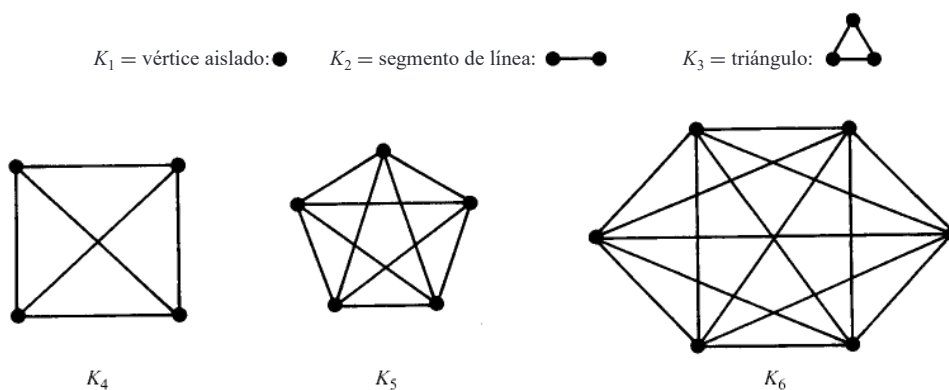


Figura 8-13

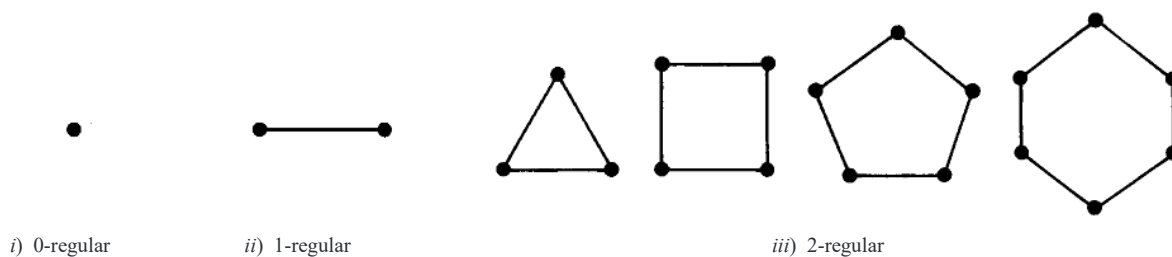


Figura 8-14

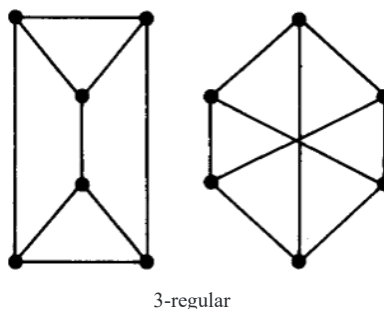


Figura 8-15

los grafos regulares pueden ser bastante complicados. Por ejemplo, hay 19 grafos 3-regular con 10 vértices. Observe que la gráfica completa con n vértices K_n es regular de grado $n - 1$.

Grafos bipartidos

Un grafo G es *bipartido* si sus vértices V pueden partirse en dos subconjuntos M y N tales que cada arista de G une un vértice de M con un vértice de N . Por un grafo bipartido completo se entiende que cada vértice de M está unido a cada vértice de N ; este grafo se denota por $K_{m,n}$, donde m es el número de vértices en M y n es el número de vértices en N y, por razones de estandarización, se supone $m \leq n$. En la figura 8-16 se muestran los grafos $K_{2,3}$, $K_{3,3}$ y $K_{2,4}$. Resulta evidente que el grafo $K_{m,n}$ tiene mn aristas.

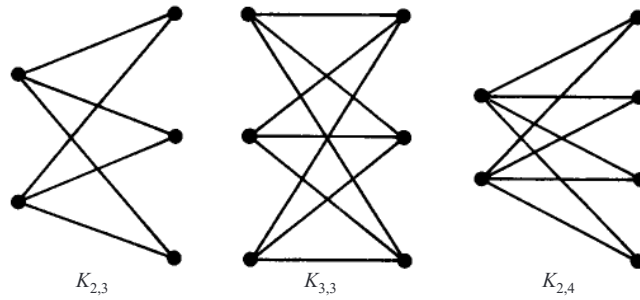


Figura 8-16

8.8 ÁRBOLES

Un grafo T se denomina *árbol* si T es conexo y T no tiene ciclos. En la figura 8-17 se muestran ejemplos de árboles. Un *bosque* G es un grafo sin ciclos; por tanto, los componentes conexos de un bosque G son árboles. Un grafo sin ciclos es *libre de ciclos*. El árbol que consta de un solo vértice sin aristas se denomina *árbol degenerado*.

Considere un árbol T . Resulta evidente que sólo hay un camino simple entre dos vértices de T ; en caso contrario, los dos caminos formarían un ciclo. También:

- a) Suponga que en T no hay ninguna arista $\{u, v\}$ y que a T se agrega la arista $e = \{u, v\}$. Entonces el camino simple de u a v en T y e forma un ciclo; por tanto, T ya no es un árbol.
- b) Por otra parte, suponga que en T hay una arista $e = \{u, v\}$, y que de T se elimina e . Entonces T ya no es conexo (puesto que no puede haber ningún camino de u a v); así, T ya no es un árbol.

El siguiente teorema (demostrado en el problema 8.14) es válido cuando los grafos son finitos.

Teorema 8.6: Sea G un grafo con $n > 1$ vértices. Entonces las siguientes afirmaciones son equivalentes:

- i) G es un árbol.
- ii) G es libre de ciclos y tiene $n - 1$ aristas.
- iii) G es conexo y tiene $n - 1$ aristas.

Este teorema también indica que un árbol finito con n vértices debe tener $n - 1$ aristas. Por ejemplo, el árbol en la figura 8-17a) tiene 9 vértices y 8 aristas, y el árbol en la figura 8-17b) tiene 13 vértices y 12 aristas.

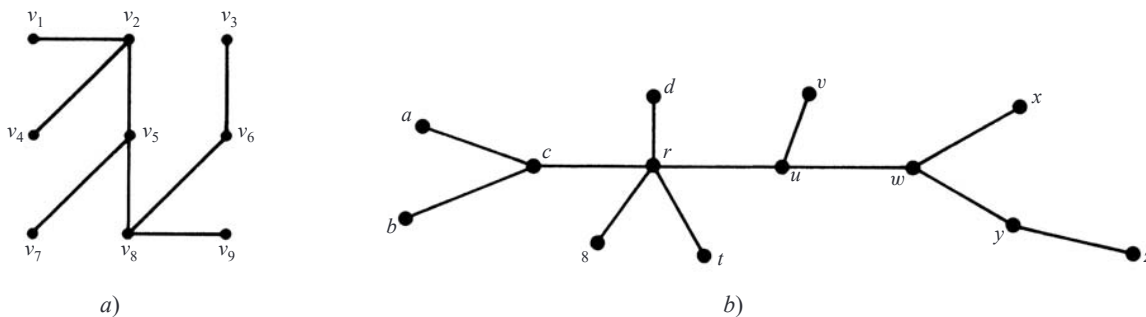


Figura 8-17

Árboles de expansión

Un subgrafo T de un grafo conexo G se denomina *árbol de expansión* de G si T es un árbol y T incluye a todos los vértices de G . En la figura 8-18 se muestra un grafo conexo G y árboles de expansión T_1, T_2 y T_3 de G .

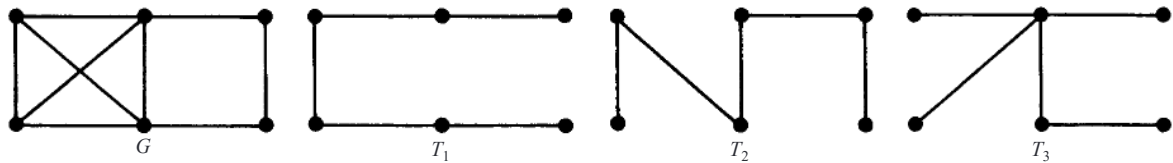


Figura 8-18

Árboles de expansión mínima

Suponga que G es un grafo ponderado conexo. Es decir, a cada arista de G se asigna un número no negativo denominado *peso* de la arista. Entonces a cualquier árbol de expansión T de G se asigna un peso total que resulta de sumar los pesos de las aristas en T . Un *árbol de expansión mínima* de G es un árbol de expansión cuyo peso total es el más pequeño posible.

Los algoritmos 8.2 y 8.3, que aparecen en la figura 8-19, permiten encontrar un árbol de expansión mínima T de un grafo ponderado conexo G , donde G tiene n vértices. (En cuyo caso T debe tener $n - 1$ aristas.)

Algoritmo 8.2: La entrada es un grafo ponderado conexo G con n vértices.

Paso 1. Las aristas de G se disponen en orden decreciente de peso.

Paso 2. Se procede secuencialmente para eliminar cada arista que no haga inconexo al grafo, hasta que queden $n - 1$ aristas.

Paso 3. Salir.

Algoritmo 8.3 (de Kruskal): La entrada es un grafo ponderado conexo G con n vértices.

Paso 1. Las aristas de G se disponen en orden creciente de peso.

Paso 2. Se empieza sólo con los vértices de G y en forma secuencial se agrega cada arista que no origine un ciclo hasta que se hayan agregado $n - 1$ aristas.

Paso 3. Salir.

Figura 8-19

El peso de un árbol de expansión mínima es único, aunque el árbol de expansión mínima en sí no lo es. Cuando dos o más aristas tienen el mismo peso pueden ocurrir distintos árboles de expansión mínima. En este caso, la disposición de las aristas en el paso 1 de los algoritmos 8.2 u 8.3 no es única, y así resultan árboles de expansión mínima distintos, como se ilustra en el siguiente ejemplo.

EJEMPLO 8.2 Encontrar un árbol de expansión mínima del grafo ponderado Q en la figura 8-20a). Observe que Q tiene seis vértices, de modo que un árbol de expansión mínima tiene cinco aristas.

a) Aquí se aplica el algoritmo 8.2.

Primero se ordenan las aristas en orden decreciente de peso y luego en forma consecutiva se eliminan las aristas sin hacer inconexo a Q hasta que queden cinco aristas. Así se obtienen los datos siguientes:

Aristas	BC	AF	AC	BE	CE	BF	AE	DF	BD
Peso	8	7	7	7	6	5	4	4	3
Eliminar	Sí	Sí	Sí	No	No	Sí			

Así, el árbol de expansión mínima de Q que se obtiene contiene las aristas

$$BE, CE, AE, DF, BD$$

El peso del árbol de expansión es 24 y se muestra en la figura 8-20b).

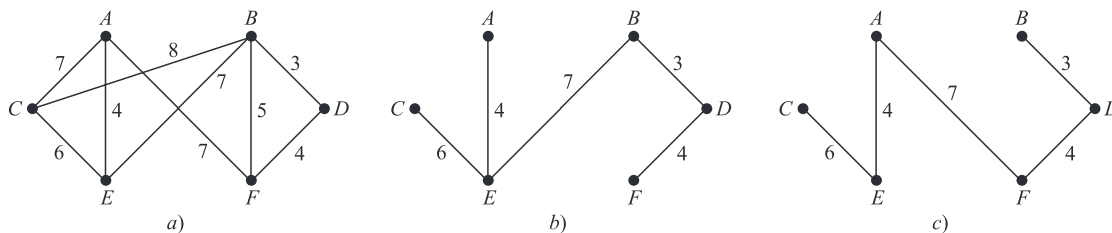


Figura 8-20

b) Aquí se aplica el algoritmo 8.3.

Primero se ordenan las aristas en orden creciente de peso y enseguida se agregan las aristas sin formar ningún ciclo hasta que se incluyen cinco aristas. Así se obtienen los datos siguientes:

Aristas	BD	AE	DF	BF	CE	AC	AF	BE	BC
Peso	3	4	4	5	6	7	7	7	8
¿Agregar?	Sí	Sí	Sí	No	Sí	No	Sí		

Así, el árbol de expansión mínima de Q contiene las aristas

$$BD, AE, DF, CE, AF$$

El árbol de expansión se muestra en la figura 8-20c). Observe que este árbol de expansión no es el mismo que se obtuvo al usar el algoritmo 8.2 y que, como era de esperar, su peso también es 24.

Observación: Los algoritmos anteriores se ejecutan fácilmente cuando el grafo G es relativamente pequeño, como en la figura 8-20a). Suponga que G tiene docenas de vértices y centenas de aristas que, por ejemplo, se proporcionan mediante una lista de pares de vértices. Entonces decidir si G es conexo no es evidente; puede ser necesario algún tipo de algoritmo de búsqueda en profundidad en grafos (DFS: Deep-first search) o de búsqueda en anchura (BFS: Breadth-first search) en grafos. En secciones ulteriores y en el siguiente capítulo se analizan formas para representar grafos G en la memoria y se abordarán varios algoritmos para grafos.

8.9 GRAFOS PLANOS

Un grafo o un multigrafo es *plano* cuando puede trazarse en el plano de modo que sus aristas no se crucen. Aunque grafo completo K_4 con cuatro vértices suele representarse con aristas cruzadas como en la figura 8-21a), también puede trazarse de modo que sus aristas no se crucen, como en la figura 8-21b); por tanto, K_4 es plano. Los árboles constituyen una clase importante de grafos planos. En esta sección se presentan estos grafos importantes.

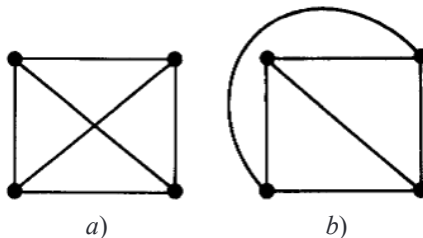


Figura 8-21

Mapas y regiones

Una representación plana particular de un multigrafo plano finito se denomina *mapa*. Se dice que el mapa es *conexo* si el multigrafo subyacente es conexo. Un mapa dado divide el plano en varias regiones. Por ejemplo, el mapa en la figura 8-22 con seis vértices y nueve aristas divide el plano en cinco regiones. Observe que cuatro de las regiones están acotadas y que la quinta región, fuera del diagrama, no está acotada. Así, no hay pérdida de generalidad al contar el número de regiones, si se supone que el mapa está contenido en algún gran rectángulo, en lugar de estarlo en todo el plano.

Observe que la frontera de cada región de un mapa consta de aristas. Algunas veces las aristas forman un ciclo, pero algunas veces no es así. Por ejemplo, en la figura 8-22 las fronteras de todas las regiones son ciclos excepto para r_3 . No obstante, si se realiza un movimiento en el sentido contrario al movimiento de las manecillas del reloj alrededor de r_3 empezando, por ejemplo, en el vértice C , entonces se obtiene el camino cerrado

$$(C, D, E, F, E, C)$$

donde la arista $\{E, F\}$ ocurre dos veces. Por el *grado* de una región r , que se escribe $\text{grd}(r)$, se entiende la longitud del ciclo o camino cerrado que rodea r . Observe que cada arista delimita dos regiones o está contenida en una región y ocurre dos veces en cualquier recorrido a lo largo de la frontera de la región. Por tanto, se tiene un teorema para regiones que es semejante al teorema 8.1 para vértices.

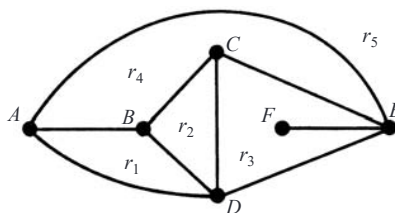


Figura 8-22

Teorema 8.7: La suma de los grados de las regiones de un mapa es igual al doble del número de aristas.

Los grados de las regiones en la figura 8-22 son:

$$\text{grd}(r_1) = 3, \quad \text{grd}(r_2) = 3, \quad \text{grd}(r_3) = 5, \quad \text{grd}(r_4) = 4, \quad \text{grd}(r_5) = 3$$

La suma de los grados es 18 y, como era de esperar, es el doble del número de aristas.

Por conveniencia en la notación, los vértices de un mapa se representan como puntos o círculos pequeños, o se supondrá que cualquier intersección de líneas o curvas en el plano es un vértice.

Fórmula de Euler

Euler proporcionó una fórmula que relaciona el número V de vértices, el número E de aristas y el número R de regiones de cualquier mapa conexo. Específicamente:

Teorema 8.8 (de Euler): $V - E + R = 2$.

(La demostración del teorema 8.8 se proporciona en el problema 8.18.)

Observe que en la figura 8-22, $V = 6$, $E = 9$ y $R = 5$, y, como era de esperar por la fórmula de Euler,

$$V - E + R = 6 - 9 + 5 = 2$$

Se recalca que el grafo subyacente de un mapa debe ser conexo para que se cumpla la fórmula de Euler.

Sea G un multigrafo plano conexo con tres o más vértices, de modo que G no es K_1 ni K_2 . Sea M una representación plana de G . No resulta difícil ver que 1) una región de M puede tener grado 1 sólo si su frontera es un lazo, y 2) una región de M puede tener grado 2 sólo si su frontera consta de dos aristas múltiples. En consecuencia, si G es un grafo, no un multigrafo, entonces toda región de M debe tener grado 3 o mayor. Este comentario y la fórmula de Euler se usan para demostrar el siguiente resultado sobre grafos planos.

Teorema 8.9: Sea G un grafo plano conexo con p vértices y q aristas, donde $p \geq 3$. Entonces $q \geq 3p - 6$.

Observe que el teorema no se cumple para K_1 , donde $p = 1$ y $q = 0$, y no es verdadero para K_2 donde $p = 2$ y $q = 1$.

Demostración: Sea r el número de regiones en una representación plana de G . Por la fórmula de Euler, $p - q + r = 2$.

Luego, la suma de los grados de las regiones es igual a $2q$ por el teorema 8.7. Pero cada región tiene 3 grados o más; por tanto, $2q \geq 3r$. Así, $r \geq 2q/3$. Al sustituir esto en la fórmula de Euler se obtiene

$$2 = p - q + r \leq p - q + \frac{2q}{3} \quad \text{o} \quad 2 \leq p - \frac{q}{3}$$

Al multiplicar la desigualdad por 3 se obtiene $6 \leq 3p - q$, con lo cual se llega al resultado. □

Grafos no planos, teorema de Kuratowski

Se proporcionan dos ejemplos de grafos no planos. Primero considere el *grafo de servicios*; es decir, a tres casas A_1, A_2, A_3 deben conectarse las tomas de agua, gas y electricidad B_1, B_2, B_3 como se muestra en la figura 8-23a). Observe que se trata del grafo $K_{3,3}$ y que tiene $p = 6$ vértices y $q = 9$ aristas. Suponga que el grafo es plano. Por la fórmula de Euler, una representación plana tiene $r = 5$ regiones. Observe que no hay tres vértices que estén unidos entre sí; por tanto, el grado de cada región debe ser 4 o mayor y así la suma de los grados de las regiones debe ser 20 o mayor. Por el teorema 8.7, el grafo debe tener 10 o más aristas. Esto contradice que el grafo tiene $q = 9$ aristas. Por tanto, el grafo de servicios $K_{3,3}$ no es plano.

Considere el *grafo estrella* en la figura 8-23b). Es el grafo completo K_5 sobre $p = 5$ vértices y tiene $q = 10$ aristas. Si el grafo es plano, entonces por el teorema 8.9,

$$10 = q \leq 3p - 6 = 15 - 6 = 9$$

lo cual es imposible. Por tanto, K_5 no es plano.

Durante muchos años los matemáticos intentaron caracterizar los grafos planos y los grafos no planos. Este problema fue resuelto finalmente en 1930 por el matemático polaco K. Kuratowski. La demostración de este resultado, que se plantea a continuación, rebasa el alcance de este texto.

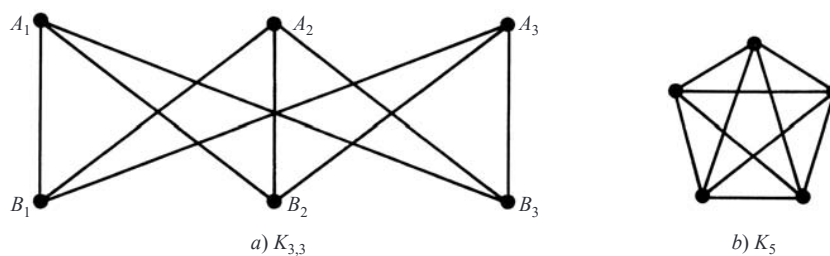


Figura 8-23

Teorema 8.10 (de Kuratowski): Un grafo es no plano si y sólo si contiene una subgrafo homeomorfo a $K_{3,3}$ o a K_5 .

8.10 COLOREADOS DE GRAFOS

Considere un grafo G . Un *coloreado de vértices*, o simplemente *coloreado de G* , es una asignación de colores a los vértices de G de modo que vértices adyacentes tengan diferentes colores. Se dice que G es n -coloreable si existe un coloreado de G en el que se usan n colores. (Puesto que el término “color” se usa como sustantivo, se intentará evitar

su uso como verbo al decir, por ejemplo, “pintura” G en lugar de “color” G cuando se asignen colores a los vértices de G .) El número mínimo de colores necesarios para pintar a G se denomina *número cromático* de G y se denota por $\chi(G)$.

En la figura 8-24 se proporciona un algoritmo propuesto por Welch y Powell para el coloreado de un grafo G . Se recalca que este algoritmo no siempre produce un coloreado mínimo de G .

Algoritmo 8.4 (de Welch y Powell): La entrada es un grafo G .

Paso 1. Los vértices de G se ordenan en orden decreciente de grado.

Paso 2. El primer color C_1 se asigna al primer vértice y después, en orden secuencial, C_1 se asigna a cada vértice que no sea adyacente al vértice previo al que se asignó C_1 .

Paso 3. El paso 2 se repite con un segundo color C_2 y la subsecuencia de vértices no coloreados.

Paso 4. El paso 3 se repite con un tercer color C_3 , y luego con un cuarto color C_4 , hasta que todos los vértices estén coloreados.

Paso 5. Salir.

Figura 8-24

EJEMPLO 8.3

a) Considere el grafo G en la figura 8-25. Se aplica el algoritmo 8.4, de Welch y Powell, para obtener un coloreado de G . Cuando los vértices se escriben en orden decreciente de grado se obtiene la siguiente secuencia:

$$A_5, A_3, A_7, A_1, A_2, A_4, A_6, A_8$$

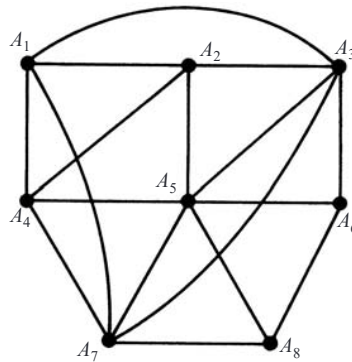


Figura 8-25

El primer color se asigna a los vértices A_5 y A_1 . El segundo color se asigna a los vértices A_3 , A_4 y A_8 . El tercer color se asigna a los vértices A_7 , A_2 y A_6 . A todos los vértices se ha asignado un color, de modo que G es 3-coloreable. Observe que G no es 2-coloreable puesto que a los vértices A_1 , A_2 y A_3 , que están unidos entre sí, deben asignarse colores diferentes. En consecuencia, $\chi(G) = 3$.

b) Considere el grafo completo K_n con n vértices. Puesto que cada vértice es adyacente a cualquier otro vértice, K_n requiere n colores en cualquier coloreado. Por tanto, $\chi(K_n) = n$.

No hay ninguna forma sencilla para determinar realmente si un grafo arbitrario es n -coloreable. Sin embargo, el siguiente teorema (que se demuestra en el problema 8.19) proporciona una caracterización simple de grafos 2-coloreables.

Teorema 8.11: Las siguientes afirmaciones son equivalentes para un grafo G :

- i) G es 2-coloreable.
- ii) G es bipartido.
- iii) Todo ciclo de G tiene longitud par.

No hay límite sobre el número de colores que pueden requerirse para colorear un grafo arbitrario puesto que, por ejemplo, el grafo completo K_n requiere n colores. No obstante, si el estudio se restringe a grafos planos, sin importar el número de vértices, bastan cinco colores. Específicamente, en el problema 8.20 se demuestra el siguiente:

Teorema 8.12: Cualquier grafo plano es 5-coloreable.

En realidad, desde el año de 1850 los matemáticos han conjeturado que los grafos planos son 4-coloreables, puesto que todo grafo plano conocido es 4-coloreable. En 1976 Kenneth Appel y Wolfgang Haken demostraron finalmente que esta conjetura es cierta. Es decir:

Teorema de los cuatro colores (Appel y Haken): Cualquier grafo plano es 4-coloreable.

Este teorema se analiza en la siguiente subsección.

Mapas duales y el teorema de los cuatro colores

Considere un mapa M ; por ejemplo, el mapa M en la figura 8-26a). En otras palabras, M es una representación plana de un multigrafo plano. Dos regiones de M son *adyacentes* si tienen una arista en común. Así, las regiones r_2 y r_3 en la figura 8-26a) son adyacentes, pero las regiones r_3 y r_5 no lo son. Por un *coloreado* de M se entiende la asignación de un color a cada región de M , de modo que regiones adyacentes tengan colores distintos. Un mapa es *n-coloreable* si existe un coloreado de M en el que se usen n colores. Por tanto, el mapa en la figura 8-26a) es 3-coloreable, ya que a las regiones pueden asignarse los siguientes colores:

$$r_1 \text{ rojo, } r_2 \text{ blanco, } r_3 \text{ rojo, } r_4 \text{ blanco, } r_5 \text{ rojo, } r_6 \text{ azul}$$

Observe la semejanza entre este análisis sobre coloreado de mapas y el análisis previo sobre coloreado de grafos. De hecho, al usar el concepto de mapa dual definido a continuación, puede demostrarse que el coloreado de un mapa es equivalente al coloreado de vértices de un grafo plano.

Considere un mapa M . En cada región de M se escoge un punto, y si dos regiones tienen una arista en común, entonces se unen los puntos correspondientes con una curva que pasa por la arista común. Estas curvas pueden trazarse de modo que no se crucen. Así se obtiene un nuevo mapa M^* denominado *dual* de M , tal que cada vértice de M^* corresponde exactamente a una región de M . En la figura 8-26b) se muestra el dual del mapa de la figura 8-26a). Puede demostrarse que cada región de M^* contiene exactamente un vértice de M y que cada arista de M^* corta exactamente una arista de M y viceversa. Por tanto, M es el dual del mapa M^* .

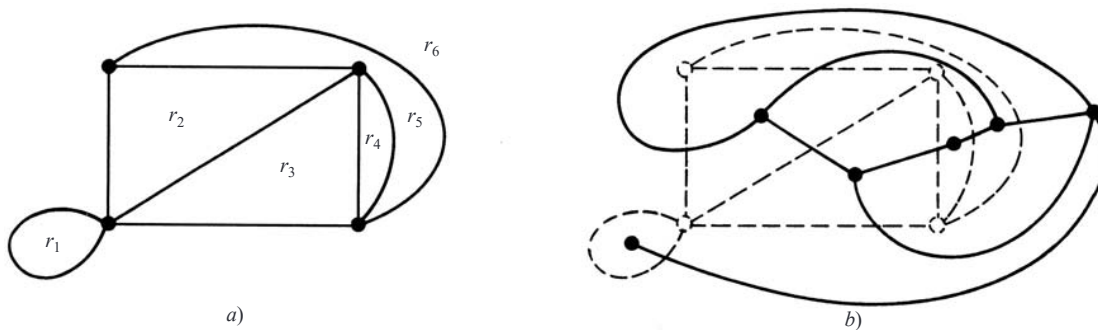


Figura 8-26

Observe que cualquier coloreado de las regiones de un mapa M corresponde a un coloreado de los vértices del mapa dual M^* . Por tanto, M es n -coloreable si y sólo si el grafo plano del mapa dual M^* de vértices es n -coloreable. Así, el teorema anterior puede volver a plantearse como sigue:

Teorema de los cuatro colores (de Appel y Haken): Si las regiones de cualquier mapa M se colorean de modo que regiones adyacentes tengan colores distintos, entonces no se requieren más de cuatro colores.

Para demostrar el teorema anterior se usan computadoras; puesto que Appel y Haken demostraron por primera vez que si el teorema de los cuatro colores es falso, entonces debe haber un contraejemplo entre aproximadamente 2 000 tipos distintos de grafos planos. Entonces demostraron, usando una computadora, que ninguno de estos tipos de grafos posee tal contraejemplo. El análisis de cada tipo de grafo diferente parece estar más allá del alcance del ser humano sin el uso de una computadora. Por tanto la demostración, a diferencia de la mayor parte de las demostraciones en matemáticas, depende de la tecnología; es decir, depende del desarrollo de computadoras de alta velocidad.

8.11 REPRESENTACIÓN DE GRAFOS EN LA MEMORIA DE LA COMPUTADORA

Hay dos formas normales para mantener un grafo G en la memoria de una computadora. Una forma, denominada *representación secuencial* de G , es por medio de su matriz de adyacencia A . La otra forma, denominada *representación enlazada* o *estructura de adyacencia* de G , usa listas ligadas de vecinos. Las matrices se usan cuando el grafo G es denso, y las listas ligadas suelen usarse cuando G es disperso. (Se dice que un grafo G con m vértices y n aristas es *denso* cuando $m = O(n^2)$ y *disperso*, cuando $m = O(n)$ o inclusive $O(n \log n)$.)

Sin importar la forma en que se mantenga un grafo G en la memoria, el grafo G normalmente se introduce en la computadora mediante su definición formal; es decir, como una colección de vértices y una colección de pares de vértices (aristas).

Matriz de adyacencia

Suponga que G es un grafo con m vértices, y suponga que los vértices se han ordenado; por ejemplo, v_1, v_2, \dots, v_m . Entonces la *matriz de adyacencia* $A = [a_{ij}]$ del grafo G es la matriz de $m \times m$ definida por

$$a_{ij} = \begin{cases} 1 & \text{si } v_i \text{ es adyacente a } v_j \\ 0 & \text{en otro caso} \end{cases}$$

La figura 8-27b) contiene la matriz de adyacencia del grafo G en la figura 8-27a), donde el orden de los vértices es A, B, C, D, E . Observe que cada arista $\{v_i, v_j\}$ de G está representado dos veces, por $a_{ij} = 1$ y $a_{ji} = 1$. Así, en particular, la matriz de adyacencia es simétrica.

La matriz de adyacencia A de un grafo G depende del orden de los vértices de G ; es decir, un orden diferente de los vértices produce una matriz de adyacencia diferente. Sin embargo, dos matrices de adyacencia arbitrarias están estrechamente relacionadas en el sentido de que una puede obtenerse a partir de la otra al intercambiar simplemente renglones y columnas. Por otra parte, la matriz de adyacencia no depende del orden en que las aristas (pares de vértices) se introducen en la computadora.

Hay variantes de la representación anterior. Si G es un multigrafo, entonces usualmente se deja que a_{ij} denote el número de aristas $\{v_i, v_j\}$. Además, si G es un multigrafo ponderado, entonces puede dejarse que a_{ij} denote el peso de la arista $\{v_i, v_j\}$.

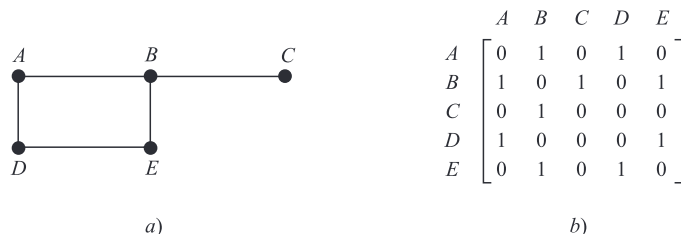


Figura 8-27

Representación enlazada de un grafo G

Sea G un grafo con m vértices. La representación de G en la memoria por medio de su matriz de adyacencia A presenta varias desventajas fundamentales. En primer lugar, puede ser difícil insertar o eliminar vértices en G . La razón es que puede ser necesario modificar el tamaño de A y reordenar los vértices, de modo que en la matriz A puede haber muchos, muchos cambios. Además, suponga que el número de aristas es $O(m)$ o inclusive $O(m \log m)$; es decir, suponga que G es disperso. Entonces la matriz A contiene muchos ceros; por tanto, se desperdicia bastante espacio de la memoria. En consecuencia, cuando G es disperso, G suele representarse en la memoria por medio de algún tipo de *representación enlazada*, también denominada *estructura de adyacencia*, que se describe a continuación mediante un ejemplo.

Considere el grafo G en la figura 8-28a). Observe que G puede definirse en forma equivalente por la tabla en la figura 8-28b), que muestra cada vértice en G seguido por su *lista de adyacencia*; es decir, su lista de vértices adyacentes (*vecinos*). Aquí el símbolo \emptyset denota una lista vacía. Esta tabla también se representa en forma más breve como

$$G = [A:B, D; \quad B:A, C, D; \quad C:B; \quad D:A, B; \quad E:\emptyset]$$

donde dos puntos “:” separan un vértice de su lista de vecinos; y un punto y coma “;” separa las distintas listas.

Observación: Cada arista de un grafo G se representa dos veces en una estructura de adyacencia; es decir, cualquier arista, por ejemplo $\{A, B\}$, se representa por B en la lista de adyacencia de A , y también por A en la lista de adyacencia de B . El grafo G en la figura 8-28a) tiene cuatro aristas, de modo que en las listas de adyacencia debe haber 8 vértices. Por otra parte, cada vértice en una lista de adyacencia corresponde a una arista única en el grafo G .

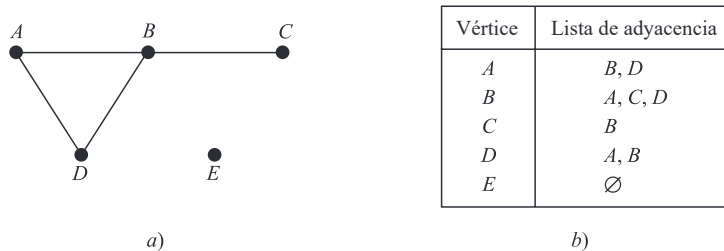


Figura 8-28

La *representación enlazada* de un grafo G , que mantiene a G en la memoria al usar sus listas de adyacencia, normalmente contiene dos archivos (o conjuntos de registros), uno denominado Vertex File y el otro denominado Edge File, como sigue.

a) **Vertex File:** este archivo contiene la lista de vértices del grafo G que suelen mantenerse por medio de un arreglo o una lista ligada. Cada registro de este archivo tiene la forma

VERTEX	NEXT-V	PTR	
--------	--------	-----	--

Aquí VERTEX es el nombre del vértice, NEXT-V apunta hacia el siguiente vértice en la lista de vértices en el Vertex File cuando los vértices se mantienen por medio de una lista ligada, y PTR apunta al primer elemento en la lista de adyacencia del vértice que aparece en el Edge File. El área sombreada indica que puede haber otra información en el registro correspondiente al vértice.

b) **Edge File:** este archivo contiene las aristas del grafo G . Específicamente, el Edge File contiene todas las listas de adyacencia de G , donde cada lista se mantiene en la memoria por medio de una lista ligada. Cada registro del Edge File corresponde a un vértice en una lista de adyacencia y, entonces, indirectamente, a una arista en G . El registro suele tener la forma

EDGE	ADJ	NEXT	
------	-----	------	--

Aquí:

- 1) EDGE es el nombre de la arista (en caso de tener una).
- 2) ADJ apunta a la ubicación del vértice en el Vertex File.
- 3) NEXT apunta a la ubicación del siguiente vértice en la lista de adyacencia.

Se recalca que cada arista está representada dos veces en el Edge File, pero cada registro del archivo corresponde a una arista única. El área sombreada indica que puede haber otra información en el registro correspondiente a la arista.

La figura 8-29 muestra cómo el grafo G en la figura 8-28a) puede aparecer en la memoria. Aquí los vértices de G se mantienen en la memoria por medio de una lista ligada que usa la variable START para apuntar hacia el primer vértice. (Una forma alterna para la lista de vértices es usar un arreglo lineal, y así NEXT-V no sería necesario.) Observe que el campo EDGE no es necesario aquí porque las aristas carecen de nombre. La figura 8-29 también muestra, con las flechas, la lista de adyacencia $[D, C, A]$ del vértice B .

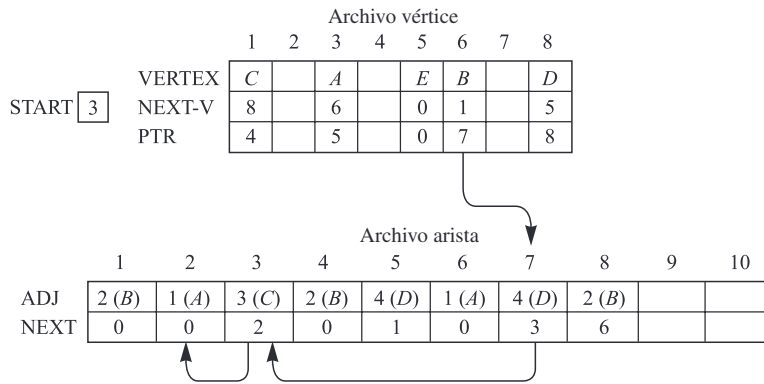


Figura 8-29

8.12 ALGORITMOS DE GRÁFICAS

En esa sección se analizan dos importantes algoritmos de grafos que examinan de manera sistemática los vértices y las aristas de un grafo G . Uno se denomina *búsqueda en profundidad* (DFS: depth-first search) y el otro, *búsqueda en anchura* (BFS: breadth-first search). Otros algoritmos de grafos se analizarán en el siguiente capítulo en relación con grafos dirigidos. Cualquier algoritmo de grafos particular puede depender de la forma en que G se mantiene en la memoria. Aquí se supone que G se mantiene en la memoria por medio de su lista de adyacencia. El grafo de prueba G con su estructura de adyacencia se muestra en la figura 8-30, donde se supone que los vértices están ordenados alfabéticamente.

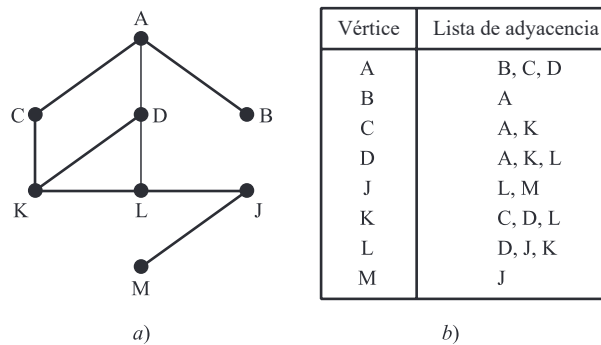


Figura 8-30

Durante la ejecución de los algoritmos, cada vértice (nodo) N de G se encuentra en uno de tres estados, denominados *status* de N , como sigue:

STATUS = 1: (Estado Ready) El estado inicial del vértice N .

STATUS = 2: (Estado Waiting) El vértice N está en una lista (de espera), en espera de ser procesado.

STATUS = 3: (Estado Processed) El vértice N ha sido procesado.

La lista de espera para la búsqueda en profundidad (DFS) será una STACK —modificada— (que se escribe horizontalmente con la parte superior de STACK a la izquierda) mientras que la lista de espera para la búsqueda en anchura (BFS) será una QUEUE.

Búsqueda en profundidad

La idea general detrás de una búsqueda en profundidad que empieza en un vértice inicial A es la siguiente: primero se procesa el vértice inicial A . Luego se procesa cada vértice N a lo largo de un camino P que empieza en A ; es decir, se procesa un vecino de A , luego un vecino de A y así sucesivamente. Después de llegar a un “punto muerto”; es decir, a un vértice sin vecino no procesado, se retrocede en el camino P hasta que es posible continuar a lo largo de otro camino P' . Y así en lo sucesivo. El retroceso se logra usando una STACK para mantener los vértices iniciales de posibles caminos futuros. También se requiere un campo STATUS que indique el estado actual de cualquier vértice, de modo que ningún vértice sea procesado más de una vez.

El algoritmo de la búsqueda en profundidad (DFS) se muestra en la figura 8-31. El algoritmo procesa sólo aquellos vértices que están unidos al vértice inicial A ; es decir, el componente conexo que incluye a A . Suponga que se desea procesar todos los vértices del grado G . Entonces es necesario modificar el algoritmo de modo que empiece de nuevo con otro vértice (que se denomina B) que aún se encuentre en el estado ready (STATE = 1). Este vértice B se obtiene al recorrer la lista de vértices.

Observación: La estructura STACK en el algoritmo anterior no es técnicamente una pila puesto que, en el paso 5b), se permite la eliminación de un vértice J que después se inserta enfrente de la pila. (Aunque se trata del mismo vértice J , suele representar una arista diferente en la estructura de adyacencia.) Si J no se elimina en el paso 5b), entonces se obtiene una forma alterna de la búsqueda en profundidad.

Algoritmo 8.5 (Búsqueda en profundidad): Este algoritmo ejecuta una búsqueda en profundidad sobre un grafo G ; la búsqueda empieza con un vértice inicial A .

Paso 1. Todos los vértices se inicializan en el estado ready (STATUS = 1).

Paso 2. El vértice inicial se introduce sobre STACK y se cambia el estado de A al estado waiting (STATUS = 2).

Paso 3. Se repiten los pasos 4 y 5 hasta que STACK esté vacía.

Paso 4. Se saca el vértice superior N de STACK. Se procesa N y se hace STATUS (N) = 3, el estado processed.

Paso 5. Se analiza cada vecino J de N .

- a) Si STATUS (J) = 1 (estado ready), J se introduce en STACK y se restablece STATUS (J) = 2 (estado waiting).
- b) Si STATUS (J) = 2 (estado waiting), el J previo se elimina de STACK y el J actual se introduce en STACK.
- c) Si STATUS (J) = 3 (estado processed), se ignora el vértice J .

[Fin del ciclo del paso 3].

Paso 6. Salir.

Figura 8-31

EJEMPLO 8.4 Suponga que el algoritmo 8.5 de búsqueda en profundidad en la figura 8-31 se aplica al grafo en la figura 8-30. Los vértices se procesan en el siguiente orden:

A, D, L, K, C, J, M, B

En la figura 8-32a) se muestra la secuencia de los vértices que están en proceso y la secuencia de las listas de espera en STACK. (Observe que después que se procesa el vértice A , sus vecinos B, C y D se añaden a STACK en el orden primero B , luego C y por último D ; por tanto, D está en la parte superior de STACK y D es el siguiente vértice que será procesado.) Cada vértice, excluyendo a A , proviene de una lista de adyacencia y entonces corresponde a una arista del grafo. Estas aristas constituyen un árbol de expansión de G que se muestra en la figura 8-32b). Los números indican el orden en que las aristas se agregan al árbol de expansión, y las líneas discontinuas indican retroceso.

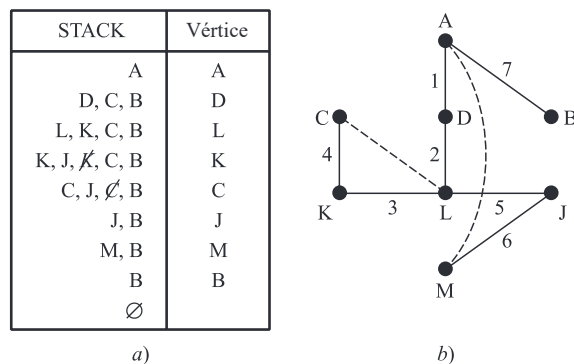


Figura 8-32

Búsqueda en anchura

La idea general detrás de una búsqueda en anchura que empieza en un vértice inicial A es la siguiente: primero se procesa el vértice inicial A . Luego se procesan todos los vecinos de A . Enseguida se procesan todos los vecinos de los vecinos de A . Lo natural es seguir la pista de los vecinos de un vértice, y es necesario garantizar que ningún vértice sea procesado dos veces. Esto se logra mediante el uso de una QUEUE para mantener los vértices que están en espera de ser procesados, y por un campo STATUS que indica el estado actual de un vértice.

El algoritmo de búsqueda en anchura (BFS) se muestra en la figura 8-33. De nuevo, el algoritmo sólo procesa los vértices que están unidos al vértice inicial A ; es decir, el componente conexo incluyendo a A . Suponga que desea procesar todos los vértices en el grafo G . Entonces es necesario modificar el algoritmo de modo que empiece de nuevo con otro vértice (que se denomina B) que aún se encuentre en el estado ready (STATUS = 1). Este vértice B puede obtenerse recorriendo la lista de vértices.

EJEMPLO 8.5 Suponga que el algoritmo 8.6 de búsqueda en anchura (BFS) en la figura 8-33 se aplica al grafo en la figura 8-30. Los vértices se procesan en el siguiente orden:

A, B, C, D, K, L, J, M

En la figura 8-34a) se muestra la secuencia de las listas de espera en QUEUE y la secuencia de los vértices que están siendo procesados. (Observe que después de procesar el vértice A , sus vecinos B, C y D se añaden a QUEUE en el orden primero B , luego C y por último D ; por tanto, B está al frente de la QUEUE y así B es el siguiente vértice que será procesado.) De nuevo, cada vértice, excluyendo a A , proviene de una lista de adyacencia y, por tanto, corresponde a una arista del grafo. Estas aristas forman un árbol de expansión de G que se muestra en la figura 8-34b). Una vez más, los números indican el orden en que las aristas se agregan al árbol de expansión. Observe que este árbol de expansión es diferente al de la figura 8-32b), que proviene de una búsqueda en profundidad.

Algoritmo 8.6 (Búsqueda en anchura): Al empezar en un vértice inicial A , este algoritmo ejecuta una búsqueda en anchura sobre un grafo G .

Paso 1. Todos los vértices se inicializan en el estado ready (STATUS = 1).

Paso 2. El vértice inicial A se coloca en QUEUE y el estado de A se cambia al estado waiting (STATUS = 2).

Paso 3. Se repiten los pasos 4 y 5 hasta que QUEUE esté vacía.

Paso 4. Se elimina el vértice frontal N en QUEUE. Se procesa N y se hace STATUS (N) = 3, el estado processed.

Paso 5. Se analiza cada vecino J de N .

a) Si STATUS (J) = 1 (estado ready), J se agrega a la parte trasera de QUEUE y se restablece STATUS (J) = 2 (estado waiting).

b) Si STATUS (J) = 2 (estado waiting) o STATUS(J) = 3 (estado processed), se ignora el vértice J .
[Fin del ciclo del paso 3].

Paso 6. Salir.

Figura 8-33

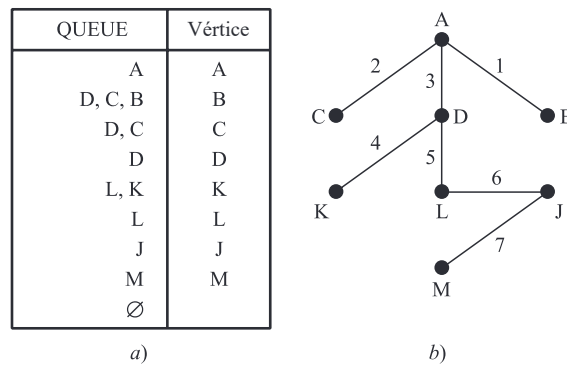


Figura 8-34

8.13 EL PROBLEMA DEL AGENTE VIAJERO

Sea G un grafo ponderado completo. (Los vértices de G se consideran ciudades y las aristas ponderadas de G las distancias entre las ciudades.) El “problema del agente viajero” busca encontrar un circuito hamiltoniano de peso mínimo para G .

Primero se observa el siguiente teorema, demostrado en el problema 8.33:

Teorema 8.13: El grafo completo K_n con $n \geq 3$ vértices tiene $H = (n - 1)!/2$ circuitos hamiltonianos (donde no se distingue entre un circuito y su opuesto).

Considere el grafo ponderado completo G en la figura 8-35a). Tiene cuatro vértices, A, B, C, D . Por el teorema 8.13, tiene $H = 3!/2 = 3$ circuitos hamiltonianos. En el supuesto de que los circuitos empiezan en el vértice A , a continuación se muestran los tres circuitos y sus pesos:

$$\begin{aligned}
 |ABCD A| &= 3 + 5 + 6 + 7 = 21 \\
 |ACDB A| &= 2 + 6 + 9 + 3 = 20 \\
 |ACBD A| &= 2 + 5 + 9 + 7 = 23
 \end{aligned}$$

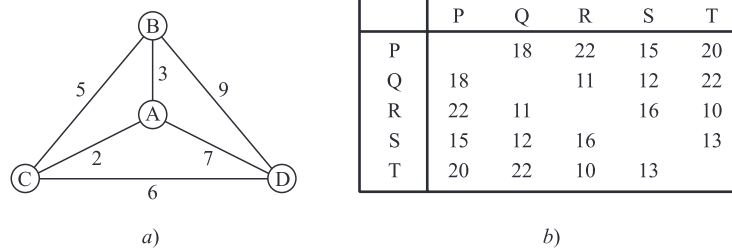


Figura 8-35

Por tanto, $ACDBA$ con peso 20 es el circuito hamiltoniano de peso mínimo.

El “problema del agente viajero” se resolvió para el grafo completo ponderado G en la figura 8-35a) al enumerar y determinar los pesos de sus tres posibles circuitos hamiltonianos. No obstante, para un grafo con muchos vértices, hacer lo anterior puede ser impráctico o incluso imposible. Por ejemplo, un grafo completo con 15 vértices tiene más de 40 millones de circuitos hamiltonianos. En consecuencia, para circuitos con muchos vértices, se requiere una estrategia para resolver o encontrar una solución aproximada al problema del agente viajero. A continuación se analiza uno de los algoritmos más simples.

Algoritmo del vecino más próximo

Este algoritmo, que empieza en un vértice dado, escoge la arista con peso mínimo hacia el siguiente vértice posible; es decir, al vértice “más próximo”. Esta estrategia continúa en cada vértice sucesivo hasta que se completa un circuito hamiltoniano.

EJEMPLO 8.6 Sea G el grafo ponderado de la tabla en la figura 8-35b). Es decir, G tiene los vértices P, Q, \dots, T , y la distancia de P a Q es 18; la de P a R es 22 y así hasta que la distancia de T a S es 13. El algoritmo del vecino más próximo se aplica a G en a) P , b) Q .

- a) Al empezar en P , el primer renglón de la tabla muestra que el vértice más próximo a P es S con distancia 15. El cuarto renglón muestra que el vértice más próximo a S es Q con distancia 12. El vértice más próximo a Q es R con distancia 11. Desde R , no hay ninguna opción más que dirigirse a T con distancia 10. Por último, desde T , no hay ninguna opción más que regresar a P con distancia 20. En consecuencia, el algoritmo del vecino más próximo empezando en P produce el siguiente circuito hamiltoniano ponderado:

$$|PSQRTTP| = 15 + 12 + 11 + 10 + 20 = 68$$

- b) Al empezar en Q , el vértice más próximo es R con distancia 11; desde R , el vértice más próximo es T con distancia 10; y desde T el vértice más próximo es S con distancia 13. Desde S es necesario ir hasta P con distancia 15 y, por último, desde P es necesario regresar a Q con distancia 18. En consecuencia, el algoritmo del vecino más próximo empezando en Q produce el siguiente circuito hamiltoniano ponderado:

$$|QRTSPQ| = 11 + 10 + 13 + 15 + 18 = 67$$

La idea detrás del algoritmo del vecino más próximo es minimizar el peso total al minimizar el peso en cada paso. Aunque esto puede parecer razonable, el ejemplo 8.6 muestra que no es posible obtener ningún circuito hamiltoniano de peso mínimo; es decir, puede no ser ambos 68 y 67. Sólo mediante la comprobación de todos los $H = (n - 1)!/2 = 12$ circuitos hamiltonianos de G es realmente posible saber cuál es el de peso mínimo. De hecho, el algoritmo del vecino más próximo empezando en A en la figura 8-35a) produce el circuito $ACBDA$ que tiene el peso máximo. Sin embargo, el algoritmo del vecino más próximo suele proporcionar un circuito hamiltoniano relativamente próximo al de peso mínimo.

PROBLEMAS RESUELTOS

TERMINOLOGÍA DE GRAFOS

8.1 Considere el grafo G en la figura 8-36a).

- a) Describir G formalmente; es decir, encontrar el conjunto $V(G)$ de vértices de G y el conjunto $E(G)$ de aristas de G .
- b) Encontrar el grado de cada vértice y comprobar el teorema 8.1 para este grafo.
- a) Hay cinco vértices, de modo que $V(G) = \{A, B, C, D, E\}$. Hay siete pares $\{x, y\}$ de vértices donde el vértice x está unido al vértice y ; así

$$E(G) = [\{A, B\}, \{A, C\}, \{A, D\}, \{B, C\}, \{B, E\}, \{C, D\}, \{C, E\}]$$

- b) El grado de un vértice es igual al número de aristas a las que pertenece; por ejemplo, $\text{grd}(A) = 3$ puesto que A pertenece a las tres aristas $\{A, B\}, \{A, C\}, \{A, D\}$. En forma semejante,

$$\text{grd}(B) = 3, \quad \text{grd}(C) = 4, \quad \text{grd}(D) = 2, \quad \text{grd}(E) = 2$$

La suma de los grados es $3 + 3 + 4 + 2 + 2 = 14$, que es igual a dos veces el número de aristas.

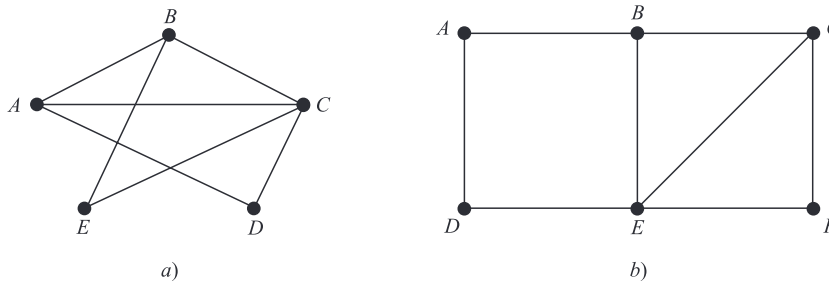


Figura 8-36

8.2 Considerar el grafo G en la figura 8-36b). Encontrar:

- a) todos los caminos simples de A a F ;
- b) todos los recorridos de A a F ;
- c) $d(A, F)$, la distancia de A a F ;
- d) $\text{diám}(G)$, el diámetro de G ;
- e) todos los ciclos que incluyen al vértice A ;
- f) todos los ciclos en G .

- a) Un camino simple de A a F es una en la cual ningún vértice, y por tanto ninguna arista, se repite. Hay siete rutas así, cuatro que empiezan con las aristas $\{A, B\}$ y tres que empiezan con las aristas $\{A, D\}$:

$$(A, B, C, F), \quad (A, B, C, E, F), \quad (A, B, E, F), \quad (A, B, E, C, F), \\ (A, D, E, F), \quad (A, D, E, B, C, F), \quad (A, D, E, C, F).$$

- b) Un recorrido de A a F es un camino tal que ninguna arista se repite. Hay nueve Recorridos así, los siete caminos simples de a) junto con

$$(A, D, E, B, C, E, F) \quad \text{y} \quad (A, D, E, C, B, E, F).$$

- c) Hay un camino; por ejemplo, (A, B, C, F) , de A a F de longitud 3 y ningún otro camino más corto de A a F ; por tanto, $d(A, F) = 3$.

- d) La distancia entre dos vértices cualesquiera no es mayor que 3, y la distancia de A a F es 3; por tanto, $\text{diám}(G) = 3$.

- e) Un ciclo es un camino cerrado en la que no se repite ningún vértice (excepto el primero y el último). Hay tres ciclos que incluyen el vértice A :

$$(A, B, E, D, A), \quad (A, B, C, E, D, A), \quad (A, B, C, F, E, D, A).$$

- f) En G hay seis ciclos; los tres en e) y

$$(B, C, E, B), \quad (C, F, E, C), \quad (B, C, F, E, B).$$

8.3 Considerar los multigrafos en la figura 8-37.

- a) ¿Cuáles son conexos? Si un grafo no es conexo, encuentre sus componentes conexos.
 - b) ¿Cuáles son libres de ciclos (sin ciclos)?
 - c) ¿Cuáles son libres de lazos (sin lazos)?
 - d) ¿Cuáles son grafos (simples)?
- a) Sólo 1) y 3) son conexos, 2) es inconexo; sus componentes conexos son $\{A, D, E\}$ y $\{B, C\}$; 4) es inconexo; sus componentes conexos son $\{A, B, E\}$ y $\{C, D\}$.
- b) Sólo 1) y 4) son libres de ciclos. 2) tiene el ciclo (A, D, E, A) , y 3) tiene el ciclo (A, B, E, A) .
- c) Sólo 4) tiene un lazo, que es $\{B, B\}$.
- d) Sólo 1) y 2) son grafos. El multigrafo 3) tiene las aristas múltiples $\{A, E\}$ y $\{A, E\}$; y 4) tiene tanto las aristas múltiples $\{C, D\}$ y $\{C, D\}$ como un lazo $\{B, B\}$.

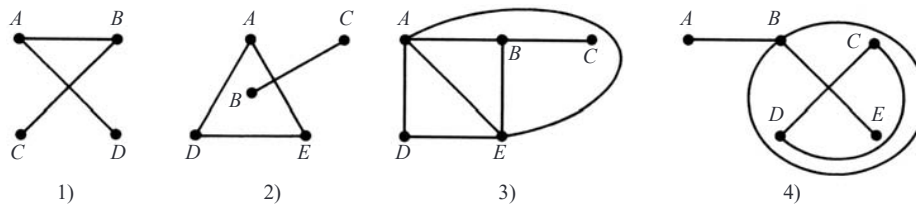


Figura 8-37

8.4 Sea G el grafo en la figura 8-38a). Encontrar:

- a) todos los caminos simples de A a C ;
 - b) todos los ciclos;
 - c) el subgrafo H generado por $V' = \{B, C, X, Y\}$;
 - d) $G - Y$;
 - e) todos los puntos de corte;
 - f) todos los puentes.
- a) Hay dos caminos simples de A a C : (A, X, Y, C) y (A, X, B, Y, C) .
- b) Hay sólo un ciclo: (B, X, Y, B) .
- c) Como se muestra en la figura 8-38b), H consta de los vértices V' y el conjunto E' de todas las aristas cuyos puntos extremos pertenecen a V' ; es decir, $E' = \{[B, X], [X, Y], [B, Y], [C, Y]\}$.
- d) Se eliminan el vértice Y de G y todas las aristas que contienen a Y para obtener el grafo $G - Y$ en la figura 8-38c). (Observe que Y es un punto de corte dado que $G - Y$ es inconexo.)
- e) Los vértices A, X y Y son puntos de corte.
- f) Una arista e es un puente si $G - e$ es inconexo. Así, hay tres puentes: $\{A, Z\}$, $\{A, X\}$ y $\{C, Y\}$.

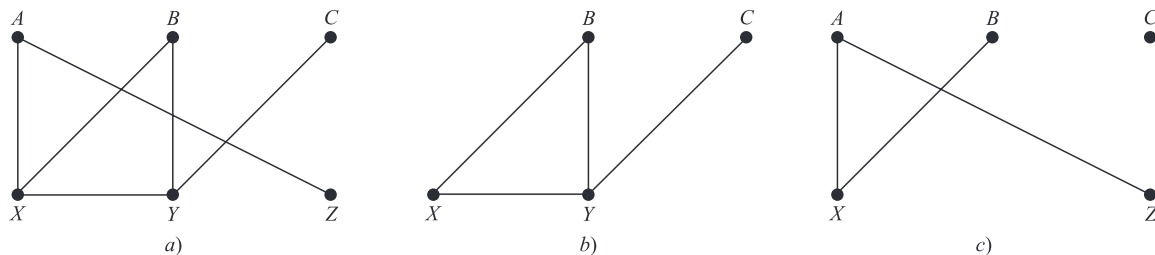


Figura 8-38

8.5 Considerar el grafo G en la figura 8-36b). Encontrar el subgrafo que resulta de eliminar cada vértice. ¿ G tiene puntos de corte?

Cuando se elimina un vértice de G , también se eliminan todas las aristas que contienen al vértice. Los seis grafos obtenidos al eliminar cada uno de los vértices de G se muestran en la figura 8-39. Todos los seis grafos son conexos; así, ningún vértice es un punto de corte.

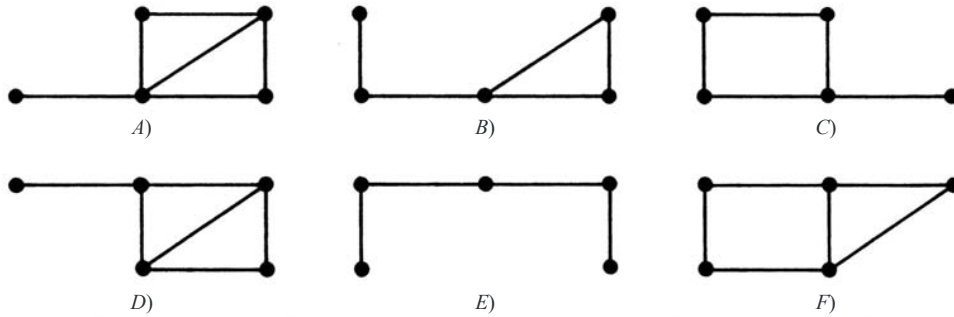


Figura 8-39

8.6 Demostrar que los seis grafos obtenidos en el problema 8.5 son distintos; es decir, que ningún par de ellos son isomorfos. Demostrar también que $B)$ y $C)$ son homeomorfos.

Los grados de los cinco vértices de cualquier grafo no pueden parearse con los grados de ningún otro grafo, excepto $B)$ y $C)$. Así, ninguno de los grafos es isomorfo, excepto quizá $B)$ y $C)$.

No obstante, si se elimina el vértice de grado 3 en $B)$ y $C)$, se obtienen subgrafos distintos. Por tanto, $B)$ y $C)$ tampoco son isomorfos; en consecuencia, todos los seis grafos son distintos. Sin embargo, $B)$ y $C)$ son homeomorfos, puesto que es posible obtenerlos a partir de grafos isomorfos al agregar vértices idóneos.

GRÁFICAS RECORRIBLES, CIRCUITOS EULERIANOS Y HAMILTONIANOS

8.7 Considerar cada grafo en la figura 8-40. ¿Cuáles son recorribles; es decir, que tienen caminos eulerianos?

¿Cuáles son eulerianos; es decir, que tienen un circuito euleriano? Los que no sean eulerianos, explicar por qué.

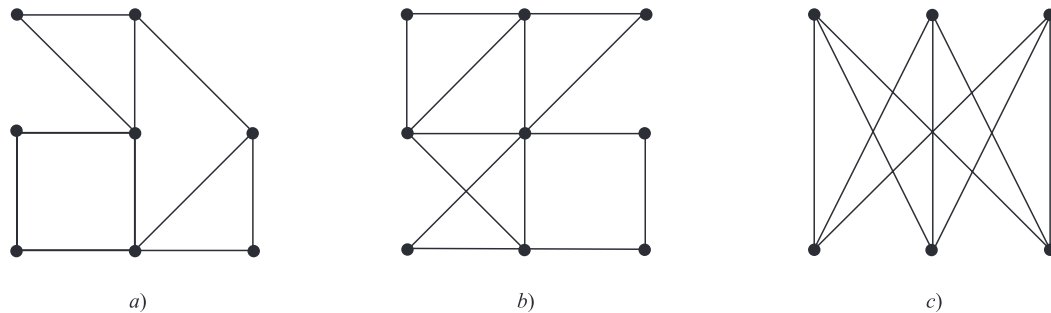


Figura 8-40

G es recorrible (tiene un camino euleriano) si sólo 0 o 2 vértices tienen grado impar, y G es euleriano (tiene un circuito euleriano) si todos los vértices son de grado par (teorema 8.3).

- a) Es recorrible, puesto que hay dos vértices impares. El camino recorrible debe empezar en uno de los vértices impares y terminar en el otro;
- b) Es recorrible, puesto que todos los vértices son pares. Por tanto, G tiene un circuito euleriano.
- c) Debido a que los seis vértices son de grado impar, G no es recorrible.

8.8 ¿Cuáles de los grafos en la figura 8-40 tienen algún circuito hamiltoniano? En caso de no tenerlo, ¿por qué?

Los grafos *a*) y *c*) tienen circuitos hamiltonianos. (El lector debe poder encontrarlos fácilmente.) Sin embargo, el grafo *b*) no tiene ningún circuito hamiltoniano. Si α es un circuito hamiltoniano, entonces α debe unir el vértice de enmedio con el vértice inferior derecho, luego proceder a lo largo del renglón inferior hacia el vértice inferior derecho, luego verticalmente hacia el vértice derecho de enmedio, donde es obligado a retroceder hacia el vértice central antes de visitar los vértices restantes.

8.9 Demostrar el teorema 8.3 (de Euler). Un grafo conexo finito G es euleriano si y solo si todo vértice tiene grado par.

El supuesto es que G es Euleriano y que T es un recorrido cerrado de Euler. Para cualquier vértice v de G , el recorrido T entra y sale de v el mismo número de veces sin repetir ninguna arista. Así, el grado de v es par.

A la inversa, cada vértice de G tiene grado par. Se construye un recorrido euleriano. Se empieza un recorrido T_1 en cualquier arista e . T_1 se extiende al agregar una arista después de otra. Si T_1 no está cerrado en ningún paso, por ejemplo, T_1 empieza en u pero termina en $v \neq u$, entonces sólo un número impar de las aristas que inciden sobre v aparecen en T_1 ; por tanto, T_1 puede extenderse por medio de otra arista que incida en v . Así es posible continuar extendiendo T_1 hasta que T_1 regresa a su vértice inicial u ; es decir, hasta que T_1 esté cerrado. Si T_1 incluye a todas las aristas de G , entonces T_1 es el recorrido euleriano buscado.

Ahora se considera que T_1 no incluye a todas las aristas de G ; es el caso del grafo H que se obtiene al eliminar en G todas las aristas de T_1 . Es posible que H no sea conexo, aunque cada vértice de H es de grado par, ya que T_1 contiene un número par de las aristas que inciden sobre cualquier vértice. Debido a que G es conexo, existe una arista e' de H que tiene un punto extremo u' en T_1 . Se construye un recorrido T_2 en H que empiece en u' y que use e' . Puesto que todos los vértices en H son de grado par, es posible continuar extendiendo a T_2 en H hasta que T_2 regresa a u' como se muestra en la figura 8-41. Resulta evidente que T_1 y T_2 se colocan juntos para formar un recorrido cerrado más largo en G . Este proceso continúa hasta que se usan todas las aristas de G . Finalmente se obtiene un recorrido euleriano, de modo que G es euleriano.

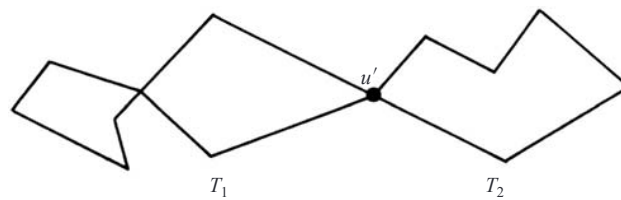


Figura 8-41

ÁRBOLES, ÁRBOLES DE EXPANSIÓN

8.10 Trazar todos los árboles que hay con exactamente seis vértices.

En la figura 8-42 hay seis árboles. El primero tiene un diámetro de 5, los dos siguientes un diámetro de 4, los dos siguientes de 3 y el último un diámetro de 2. Cualquier otro árbol con 6 nodos es isomorfo a alguno de estos árboles.

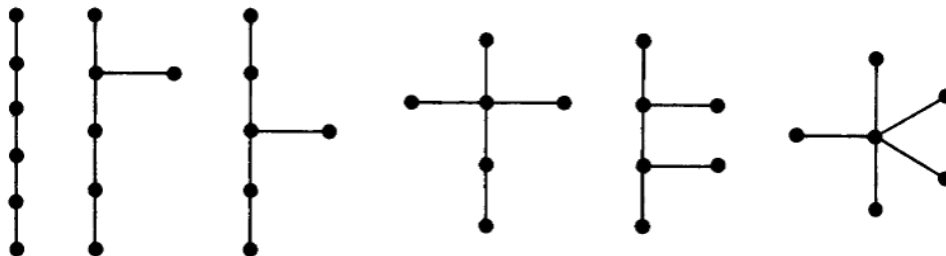


Figura 8-42

8.11 Encontrar todos los árboles de expansión del grafo G mostrada en la figura 8-43a).

Hay ocho árboles de expansión, como se muestra en la figura 8-43b). Cada árbol de expansión debe tener $4 - 1 = 3$ aristas, ya que G tiene cuatro vértices. Así, cada árbol puede obtenerse al eliminar dos de las cinco aristas de G . Esto puede hacerse

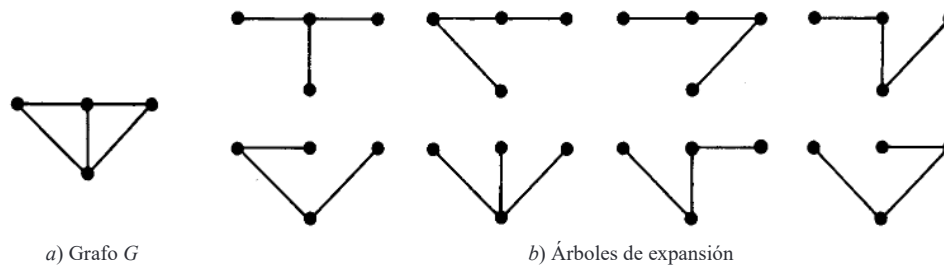


Figura 8-43

en 10 formas, excepto que dos de las formas producen grafos inconexos. Por tanto, los ocho árboles de expansión anteriores son todos los árboles de expansión de G .

8.12 Encontrar un árbol de expansión mínima T para el grafo ponderado G en la figura 8-44a).

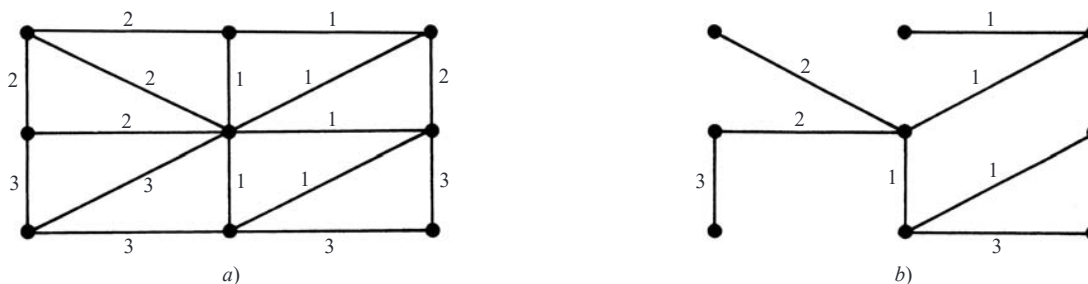


Figura 8-44

Puesto que G tiene $n = 9$ vértices, T debe tener $n - 1 = 8$ aristas. Se aplica el algoritmo 8.2; es decir, se eliminan las aristas con longitud máxima y sin desconectar el grafo hasta que sólo queden $n - 1 = 8$ aristas. En forma alterna, se aplica el algoritmo 8.3; es decir, empezando con los nueve vértices, se agregan aristas de longitud mínima y sin formar ningún ciclo hasta que se han agregado $n - 1 = 8$ aristas. Con ambos métodos se obtiene un árbol de expansión mínima como el que se muestra en la figura 8-44b).

8.13 Sea G un grafo con más de un vértice. Demostrar que las siguientes afirmaciones son equivalentes.

- i) G es un árbol.
 - ii) Cada par de vértices está unido por exactamente un camino simple.
 - iii) G es conexo; pero $G - e$ es inconexo para cualquier arista e de G .
 - iv) G es libre de ciclos, pero si a G se agrega cualquier arista, entonces el grafo resultante tiene exactamente un ciclo.
- i) *implica ii).* Sean u y v dos vértices en G . Puesto que G es un árbol, G es conexo, de modo que hay por lo menos un camino entre u y v . Por el problema 8.37, entre u y v sólo puede haber un camino simple; en caso contrario, G contiene un ciclo.
- ii) *implica iii).* Si se elimina una arista $e = \{u, v\}$ de G , e es un camino de u a v . Entonces, si el grafo resultante $G - e$ tiene un camino P de u a v . Por tanto, P y e son dos caminos distintos de u a v , lo que contradice la hipótesis. Por consiguiente, en $G - e$ no puede haber ningún camino entre u y v , de modo que $G - e$ es inconexo.
- iii) *implica iv).* Si en G hay un ciclo C que contiene una arista $e = \{u, v\}$; por hipótesis, G es conexo pero $G' = G - e$ es inconexo, donde u y v pertenecen a componentes distintos de G' (problema 8.41). Esto contradice que u y v son conexos por el camino $P = C - e$ que está en G' . Por tanto, G es libre de ciclos. Luego, sean x y y los vértices de G y sea H el grafo que se obtiene al adjuntar la arista $e = \{x, y\}$ a G . Debido a que G es conexo, en G hay una ruta P de x a y ; por tanto, $C = Pe$ forma un ciclo en H . Ahora, si H contiene otro ciclo C' , puesto que G es libre de ciclos, C' debe contener la arista e ; por ejemplo, $C' = P'e$. Entonces P y P' son dos caminos simples en G de x a y . (Vea la figura 8-45.) Por el problema 8.37, G contiene un ciclo, lo que contradice que G es libre de ciclos. Por consiguiente, H sólo contiene un ciclo.

iv) *implica i*). Puesto que al agregar cualquier arista $e = \{x, y\}$ a G se obtiene un ciclo, los vértices x y y ya deben estar unidos en G . Así, G es conexo y por hipótesis G es libre de ciclos; es decir, G es un árbol.

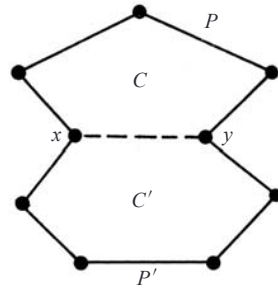


Figura 8-45

8.14 Demostrar el teorema 8.6: Sea G un grafo finito con $n \geq 1$ vértices. Entonces las afirmaciones siguientes son equivalentes. *i*) G es un árbol, *ii*) G es libre de ciclos y tiene $n - 1$ aristas, *iii*) G es conexo y tiene $n - 1$ aristas.

La demostración es por inducción sobre n . Ciertamente, el teorema es verdadero para el grafo con un solo vértice y que entonces no tiene aristas. Es decir, el teorema se cumple para $n = 1$. Ahora se supone que $n > 1$ y que el teorema es verdadero para grafos con menos de n vértices.

- i*) *implica ii*). Si G es un árbol, entonces G es libre de ciclos, de modo que sólo es necesario demostrar que G tiene $n - 1$ aristas. Por el problema 8.38, G tiene un vértice de grado 1. Al eliminar este vértice y su arista se obtiene un árbol T que tiene $n - 1$ vértices. El teorema se cumple para T , de modo que T tiene $n - 2$ aristas. Así, G tiene $n - 1$ aristas.
- ii*) *implica iii*). Si G es libre de ciclos y tiene $n - 1$ aristas. Sólo es necesario demostrar que G es conexo. Entonces, si G es inconexo y tiene k componentes, T_1, \dots, T_k que son árboles puesto que cada uno es conexo y libre de ciclos. Por ejemplo, T_i tiene n_i vértices donde $n_i < n$. Por tanto, el teorema se cumple para T_i , de modo que T_i tiene $n_i - 1$ aristas. Así,

$$n = n_1 + n_2 + \dots + n_k$$

y

$$n - 1 = (n_1 - 1) + (n_2 - 1) + \dots + (n_k - 1) = n_1 + n_2 + \dots + n_k - k = n - k$$

Por consiguiente, $k = 1$. Pero esto contradice la hipótesis que G es inconexo y que tiene $k > 1$ componentes. En consecuencia, G es conexo.

- iii*) *implica i*). Si G es conexo y tiene $n - 1$ aristas, sólo es necesario demostrar que G es libre de ciclos. Al suponer que en G hay un ciclo que contiene una arista e . Al eliminar e se obtiene el grafo $H = G - e$ que también es conexo. Sin embargo, H tiene n vértices y $n - 2$ aristas, lo cual contradice el problema 8.39. En consecuencia, G es libre de ciclos y, por tanto, es un árbol.

GRAFOS PLANOS

8.15 Dibujar una representación plana, en caso de ser posible, de los grafos *a*), *b*) y *c*) de la figura 8-46.

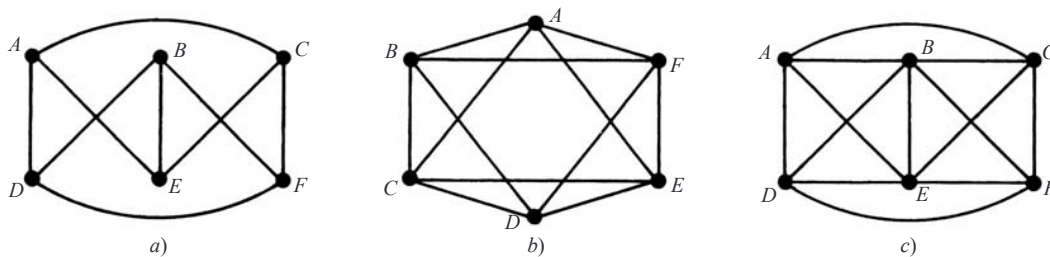


Figura 8-46

- a) Al volver a dibujar las posiciones de B y E se obtiene una representación plana del grafo como en la figura 8-47a).
- b) No es el grafo estrella $K_{3,3}$. Éste tiene una representación plana del grafo como en la figura 8-47b).
- c) Este grafo no es plano. El grafo de servicios $K_{3,3}$ es un subgrafo como se muestra en la figura 8-47c), donde se han vuelto a dibujar las posiciones de C y F .

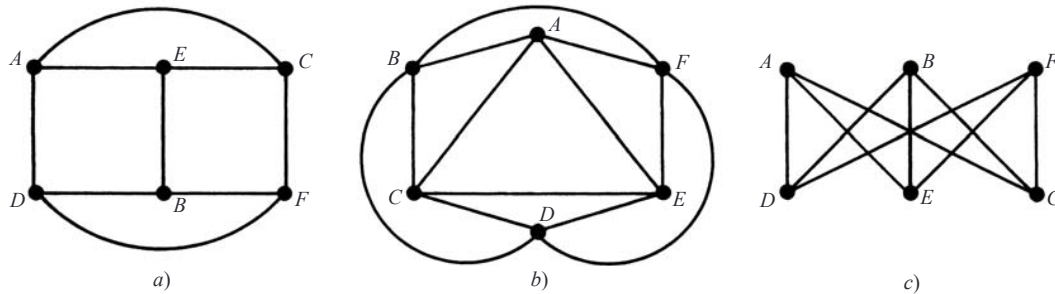


Figura 8-47

8.16 Contar el número V de vértices, el número E de aristas y el número R de regiones de cada mapa en la figura 8-48, y comprobar la fórmula de Euler. También, encontrar el grado d de la región exterior.

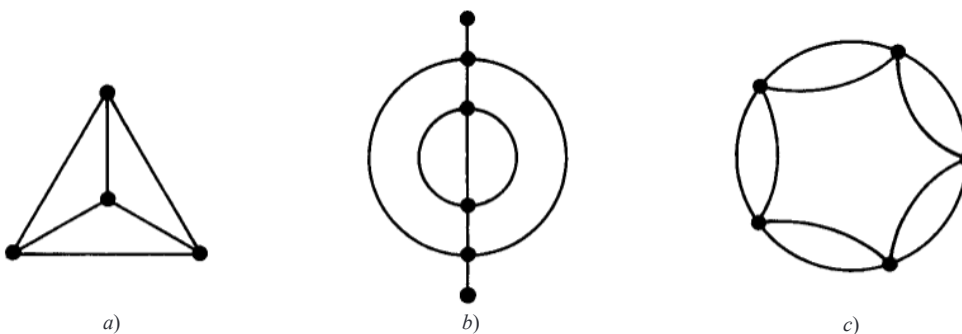


Figura 8-48

- a) $V = 4, E = 6, R = 4$. Por tanto $V - E + R = 4 - 6 + 4 = 2$. También $d = 3$.
- b) $V = 6, E = 9, R = 5$, de modo que $V - E + R = 6 - 9 + 5 = 2$. Aquí $d = 6$ puesto que dos aristas se contaron dos veces.
- c) $V = 4, E = 10, R = 7$. Por tanto $V - E + R = 4 - 10 + 7 = 2$. Aquí $d = 5$.

8.17 Encontrar el número mínimo n de colores necesarios para pintar cada mapa en la figura 8-48.

- a) $n = 4$; b) $n = 3$; c) $n = 2$.

8.18 Demostrar el teorema 8.8 (de Euler): $V - E + R = 2$.

Si el mapa conexo M consta de un solo vértice P como en la figura 8-49a), entonces $V = 1, E = 0$ y $R = 1$. Así, $V - E + R = 2$. En caso contrario, M puede establecerse a partir de un solo vértice por medio de las dos construcciones siguientes:

- 1) Se agrega un nuevo vértice Q_2 , que se une a un vértice existente Q_1 por medio de una arista que no cruce ninguna arista existente como en la figura 8-49b).
- 2) Unir dos vértices existentes Q_1 y Q_2 mediante un arista e que no cruce ninguna arista existente como en la figura 8-49c).

Ninguna de estas operaciones modifica el valor de $V - E + R$. Por tanto, M tiene el mismo valor de $V - E + R$ que el mapa que consta de un solo vértice; es decir, $V - E + R = 2$. Por consiguiente, ya se demostró el teorema.

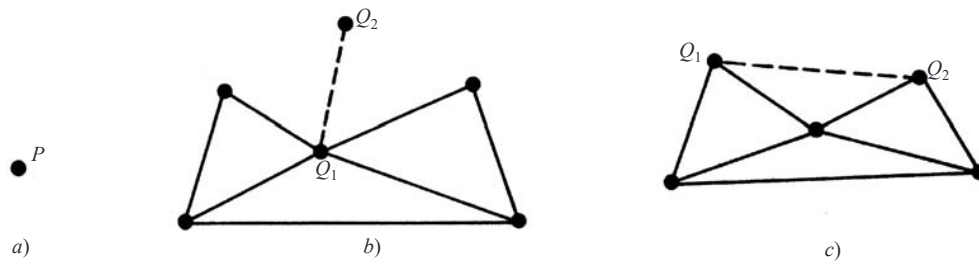


Figura 8-49

8.19 Demostrar el teorema 8.11: las afirmaciones siguientes son equivalentes para un grafo G : i) G es 2-coloreable. ii) G es bipartido. iii) Todo ciclo de G tiene longitud par.

- i) *implica ii*). Si G es 2-coloreable, sea M el conjunto de vértices pintados con el primer color y N el conjunto de vértices pintados con el segundo color. Entonces M y N forman una partición bipartida de los vértices de G puesto que ninguno de los vértices de M y ninguno de los vértices de N pueden ser adyacentes entre sí porque son del mismo color.
- ii) *implica iii*). Si G es bipartido y M y N forman una partición bipartida de los vértices de G , y si un ciclo empieza en un vértice u de, por ejemplo M , entonces irá a un vértice de N , y luego a un vértice de M , luego a uno de N y así continuará. En consecuencia, cuando el ciclo regresa a u debe ser de longitud par. Es decir, todo ciclo de G es de longitud par.
- iii) *implica i*). Por último, si cualquier ciclo de G es de longitud par, se escoge un vértice en cada componente conexo y se pinta con el primer color, por ejemplo rojo. Luego, se pintan todos los vértices como sigue: si un vértice está pintado de rojo, entonces cualquier vértice adyacente se pinta con el segundo color, por ejemplo azul. Si un vértice está pintado de azul, entonces cualquier vértice adyacente a él se pinta de rojo. Debido a que todo ciclo es de longitud par, ningún vértice adyacente se pinta del mismo color. Por consiguiente, G es 2-coloreable y se ha demostrado el teorema.

8.20 Demostrar el teorema 8.12: un grafo plano G es 5-coloreable.

La demostración es por inducción sobre el número p de vértices de G . Si $p \leq 5$, entonces resulta evidente que el teorema se cumple. Ahora bien, si $p > 5$ y el teorema es verdadero para grafos con menos de p vértices; por el problema precedente, G tiene un vértice v tal que $\text{grd}(v) \leq 5$. Por inducción, el subgrafo $G - v$ es 5-coloreable. Ahora hay que suponer uno de tales coloreados: si los vértices adyacentes a v usan menos de los cinco colores, entonces simplemente v se pinta con uno de los colores restantes y se obtiene un 5-coloreado de G . Queda aún pendiente el caso en que v es adyacente a cinco vértices que están pintados con diferentes colores. Por ejemplo, los vértices, en sentido contrario al movimiento de las manecillas del reloj, adyacentes a v son v_1, \dots, v_5 y están pintados con los colores c_1, \dots, c_5 . (Vea la figura 8-50a.)

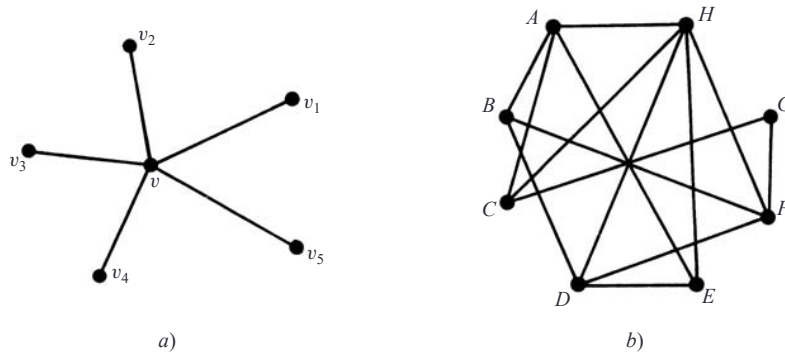


Figura 8-50

Luego se considera el subgrafo H de G generada por los vértices pintados con los colores c_1 y c_3 , donde H incluye a v_1 y v_3 . Si v_1 y v_3 pertenecen a componentes distintos de H , entonces es posible intercambiar los colores c_1 y c_3 en el componente que contiene a v_1 sin destruir el coloreado de $G - v$. Luego, v_1 y v_3 están pintados con c_3 y puede elegirse c_1 para pintar a v , y así se tiene un 5-coloreado de G . Por otra parte, si v_1 y v_3 pertenecen al mismo componente de H , entonces hay una ruta P de v_1 a v_3 cuyos vértices están pintados con c_1 o con c_3 . La ruta P junto con las aristas $\{v, v_1\}$ y $\{v, v_3\}$ forman

un ciclo C que abarca ya sea a v_2 o a v_4 . Luego se considera el subgrafo K generado por los vértices pintados con los colores c_3 o c_4 . Debido a que C incluye a v_2 o a v_4 , pero no a ambos, los vértices v_2 y v_4 pertenecen a componentes distintos de K . Por tanto, es posible intercambiar los colores c_2 y c_4 en el componente que contiene a v_2 sin destruir el coloreado de $G - v$. Entonces, v_2 y v_4 están pintados con el color c_4 y es posible escoger c_2 para pintar v y obtener un 5-coloreado de G . En consecuencia, G es 5-coloreable y se ha demostrado el teorema.

8.21 Aplicar el algoritmo 8.4, de Welch y Powell (figura 8-24), para pintar el grafo en la figura 8-50b).

Primero, los vértices se ordenan en orden decreciente de grado para obtener la secuencia

$$H, A, D, F, B, C, E, G$$

Al proceder en secuencia, el primer color se usa para pintar los vértices H, B y luego G . (No es posible pintar A, D o F con el primer color porque cada uno está unido a H , y no es posible pintar C o E con el primer color porque cada uno está unido a H o a B .) Al continuar en secuencia con los vértices sin pintar, el segundo color se usa para pintar los vértices A y D . Los vértices restantes F, C y E pueden pintarse con el tercer color. Así, el número cromático n no puede ser mayor que 3. Sin embargo, en cualquier coloreado, H, D y E deben pintarse con colores diferentes, ya que están unidos entre sí. Por tanto, $n = 3$.

8.22 Sea G un grafo plano conexo finito con por lo menos tres vértices. Demostrar que G tiene por lo menos un vértice de grado 5 o menos.

Sean p el número de vértices y q el número de aristas de G , y se supone que $\text{grd}(u) \geq 6$ para cada vértice u de G . Sin embargo, $2q$ es igual a la suma de los grados de los vértices de G (teorema 8.1); así, $2q \geq 6p$. En consecuencia,

$$q \geq 3p > 3p - 6$$

Esto contradice el teorema 8.9. Por consiguiente, algún vértice de G tiene grado 5 o menos.

REPRESENTACIÓN SECUENCIAL DE GRAFOS

8.23 Encontrar la matriz de adyacencia $A = [a_{ij}]$ de cada grafo G en la figura 8-51.

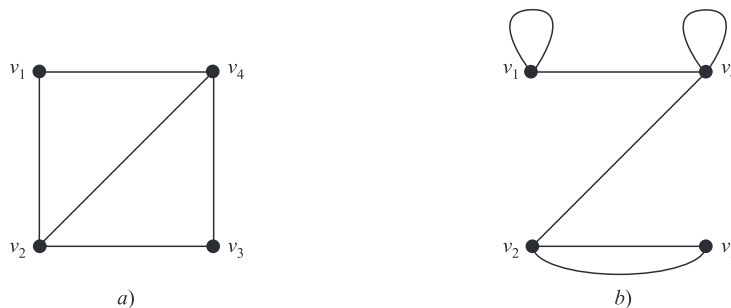


Figura 8-51

Sean $a_{ij} = n$ si hay n aristas $\{v_i, v_j\}$ y $a_{ij} = 0$ en caso contrario. Entonces:

$$a) A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}; \quad b) A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 2 & 1 \\ 0 & 2 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

(Puesto que a) no tiene arista múltiples ni lazos, las entradas de A son 0 o 1, y son 0 en la diagonal).

8.24 Dibujar el grafo G correspondiente a cada matriz de adyacencia:

$$a) A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}; \quad b) A = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 2 & 2 \\ 0 & 1 & 2 & 0 \end{bmatrix}$$

- a) Puesto que A es una matriz cuadrada de 5×5 , G tiene cinco vértices v_1, v_2, \dots, v_5 . Cuando $a_{ij} = 1$, se traza una arista de v_i a v_j . El grafo se muestra en la figura 8-52a).
- b) Puesto que A es una matriz cuadrada de 4×4 , G tiene cuatro vértices v_1, \dots, v_4 . Cuando $a_{ij} = n$, se trazan n aristas de v_i a v_j . También, cuando $a_i = n$ se trazan n lazos en v_i . El grafo se muestra en la figura 8-52b).

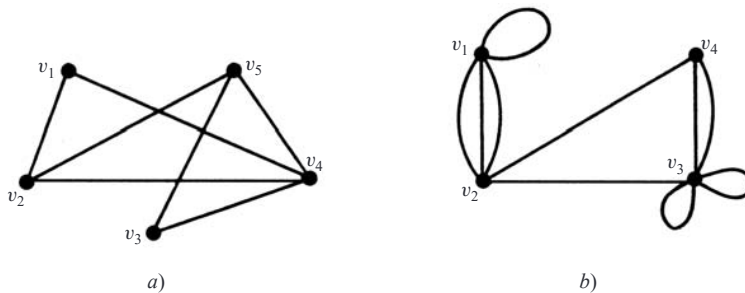


Figura 8-52

8.25 Encontrar la matriz de pesos $W = [w_{ij}]$ del grafo ponderado G en la figura 8-53a), donde los vértices están almacenados en el arreglo DATA como sigue: DATA: A, B, C, X, Y.

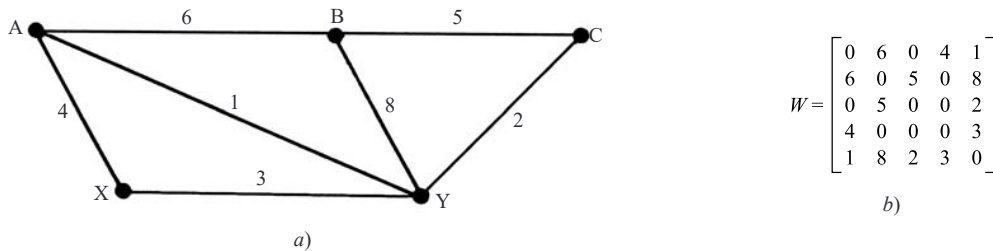


Figura 8-53

Los vértices se numeran según la forma en que se almacenan en el arreglo DATA; así, $v_1 = A, v_2 = B, \dots, v_5 = Y$. Luego se hace $W_{ij} = w$, donde w es el peso de la arista que va de v_i a v_j . Así se obtiene la matriz W en la figura 8-53b).

REPRESENTACIÓN ENLAZADA DE GRAFOS

8.26 Un grafo G con vértices A, B, \dots, F se almacena en la memoria mediante una representación enlazada con un archivo vértice y un archivo arista como en la figura 8-54.

- a) Enumerar los vértices en el orden en que aparecen en la memoria.
- b) Enumerar la lista de adyacencia $ady(v)$ de cada vértice v de G .
- a) Puesto que $START = 4$, la lista empieza con el vértice D . El NEXT-V indica dirigirse a $1(B)$, luego a $3(F)$, a $5(A)$, a $8(E)$ y a $7(C)$; es decir,

D, B, F, A, E, C

- b) Aquí $\text{ady}(D) = [5(A), 1(B), 8(E)]$. Específicamente, $\text{PTR}[4(D)] = 7$ y $\text{ADJ}[7] = 5(A)$ indican que $\text{ady}(D)$ empieza con A . Luego, $\text{NEXT}[7] = 3$ y $\text{ADJ}[3] = 1(B)$ indican que B es el siguiente vértice en $\text{ady}(D)$. Luego, $\text{NEXT}[3] = 10$ y $\text{ADJ}[10] = 8(E)$ indican que E es el siguiente vértice en $\text{ady}(D)$. Sin embargo, $\text{NEXT}[10] = 0$ indica que ya no hay más vecinos de D . En forma semejante,

$$\text{ady}(B) = [A, D], \quad \text{ady}(F) = [E], \quad \text{ady}(A) = [B, D], \quad \text{ady}(E) = [C, D, F], \quad \text{ady}(C) = [E]$$

En otras palabras, la estructura de adyacencia de G es la siguiente:

$$G = [A:B, D; \quad B:A, D; \quad C:E; \quad D:A, B, E; \quad E:C, D, F; \quad F:E]$$

		Archivo vértice								
		1	2	3	4	5	6	7	8	
START	4	VERTEX	B		F	D	A		C	E
		NEXT-V	3		5	1	8		0	7
		PTR	9		4	7	6		5	12

		Archivo arista													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
ADJ		4	4	1	8	8	1	5	3	5	8	4	7		
	NEXT	8	0	10	0	0	2	3	0	11	0	0	1		

Figura 8-54

- 8.27 Dibujar el diagrama del grafo G cuya representación enlazada se muestra en la figura 8-54.

Para dibujar el grafo G en la figura 8-55 se usan la lista de vértices obtenida en el problema 8.26a) y las listas de adyacencia obtenidas en el problema 8.26b).

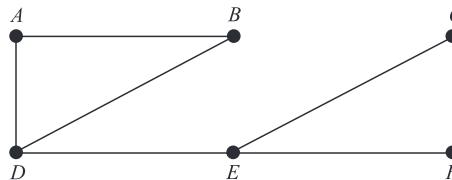


Figura 8-55

- 8.28 Mostrar la estructura de adyacencia (EA) del grafo G en la: a) figura 8-56a), b) figura 8-56b).

La estructura de adyacencia de un grafo G consta de las listas de adyacencia de los vértices, donde se usan dos puntos “:” para separar un vértice de su lista de adyacencia, y punto y coma “;” para separar las diversas listas. Así:

- a) $G = [A:B, C, D; \quad B:A, C, E; \quad C:A, B, D, E; \quad D:A, C; \quad E:B, C]$
 b) $G = [A:B, D; \quad B:A, C, E; \quad C:B, E, F; \quad D:A, E; \quad E:B, C, D, F; \quad F:C, E]$

ALGORITMOS DE GRAFOS

- 8.29 Considere el grafo G en la figura 8-56a) (donde los vértices están ordenados alfabéticamente).
- Encontrar la estructura de adyacencia de G .
 - Encontrar el orden en que se procesan los vértices de G mediante un algoritmo DFS (búsqueda en profundidad) empezando en el vértice A .
- a) Los vecinos de cada vértice se enumeran como sigue:

$$G = [A:B, C, D; \quad B:A, J; \quad C:A; \quad D:A, K; \quad J:B, K, M; \quad K:D, J, L; \quad L:K, M; \quad M:J, L]$$

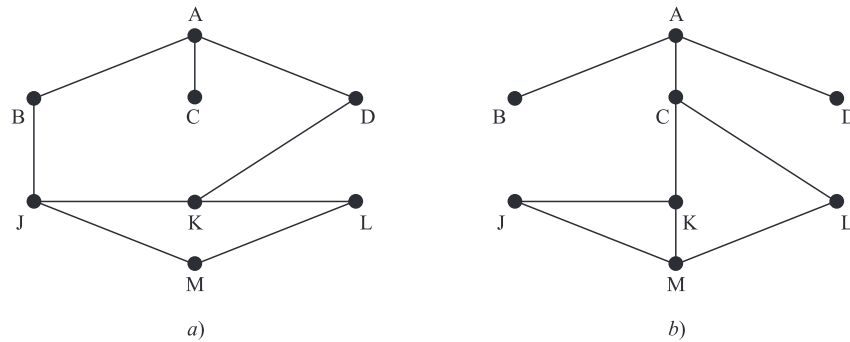


Figura 8-56

- b) Durante el algoritmo de búsqueda en profundidad, se procesa el primer vértice N en STACK y los vecinos de N (que no se han procesado antes) se colocan sobre STACK. Al principio, el vértice inicial A se coloca sobre STACK. A continuación se muestra la secuencia de las listas de espera en STACK y los vértices que están en proceso:

STACK	A	DCB	KCB	$LJCB$	$MJCB$	JCB	CB	B	\emptyset
Vértice	A	D	K	L	M	J	C	B	

En otras palabras, los vértices se procesan en el orden: A, D, K, L, M, J, C, B .

8.30 Repetir el problema 8.29 para el grafo G en la figura 8-56b).

- a) Los vecinos de cada vértice se enumeran como sigue:

$$G = [A:B, C, D; B:A; C:A, K, L; D:A; J:K, M; K:C, J, M; L:C, M; M:J, K, L]$$

- b) A continuación se muestra la secuencia de las listas de espera en STACK y los vértices que están en proceso:

STACK	A	DCB	CB	LKB	MKB	KJB	JB	B	\emptyset
Vértice	A	D	C	L	M	K	J	B	

En otras palabras, los vértices se procesan en el orden: A, D, C, L, M, K, J, B .

8.31 Si se empieza en el vértice A y se usa un algoritmo de búsqueda en anchura, encontrar el orden en que se procesan los vértices para el grafo G en la: a) figura 8-56a), b) figura 8-56b).

- a) La estructura de adyacencia de G se muestra en el problema 8-29. Durante la ejecución del algoritmo de búsqueda en profundidad, se procesa el primer vértice N en QUEUE y luego a QUEUE se agregan los vecinos de N (que no habían aparecido antes). Al principio, el vértice inicial A se asigna a QUEUE. A continuación se muestra la secuencia de las listas de espera en QUEUE y los vértices que están en proceso:

QUEUE	A	DCB	JDC	JD	KJ	MK	LM	L	\emptyset
Vértice	A	B	C	D	J	K	M	L	

En otras palabras, los vértices se procesan en el orden: A, B, C, D, J, K, M, L .

- b) La estructura de adyacencia de G aparece en el problema 8.30. A continuación se muestra la secuencia de las listas de espera en QUEUE y los vértices que están en proceso:

QUEUE	A	DCB	DC	LKD	LK	MJL	MJ	M	\emptyset
Vértice	A	B	C	D	K	L	J	M	

En otras palabras, los vértices se procesan en el orden: A, B, C, D, K, L, J, M .

EL PROBLEMA DEL AGENTE VIAJERO

- 8.32 Aplicar el algoritmo del vecino más próximo al grafo ponderado completo G en la figura 8-57, empezando en el: a) vértice A ; b) vértice B .

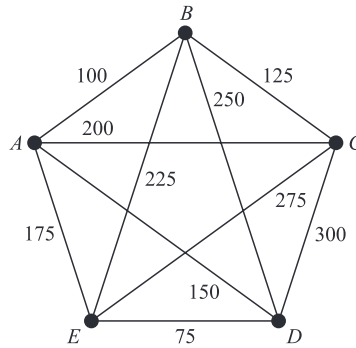


Figura 8-57

- a) Al empezar en A el vértice más próximo es B , con distancia 100; desde B el más próximo es C , con distancia 125; y desde C el más próximo es E , con distancia 275. Desde E es necesario ir a D con distancia 75 y, finalmente, desde D es necesario retroceder hacia A con distancia 150. En consecuencia, el algoritmo del vecino más próximo al empezar en A , produce el siguiente circuito hamiltoniano ponderado:

$$|ABCEDA| = 100 + 125 + 275 + 75 + 150 = 725$$

- b) Al empezar en D , es necesario ir hacia E , luego hacia A , de ahí hacia B , luego hacia C y finalmente de regreso a D . En consecuencia, el algoritmo del vecino más próximo al empezar en D , produce el siguiente circuito hamiltoniano ponderado:

$$|DEABCD| = 75 + 175 + 100 + 125 + 300 = 775$$

- 8.33 Demostrar el teorema 8.13. El grafo completo K_n con $n \geq 3$ vértices tiene $H = (n - 1)!/2$ circuitos hamiltonianos.

La convención para el conteo de circuitos hamiltonianos permite designar cualquier vértice en un circuito como el punto inicial. A partir del punto inicial es posible ir a cualquiera de los $n - 1$ vértices, y de ahí a cualquiera de los $n - 2$ vértices y así hasta que se llega al último vértice y luego se regresa al punto inicial. Por el principio de conteo básico, hay un total de $(n - 1)(n - 2) \cdots 2 \cdot 1 = (n - 1)!$ circuitos que pueden formarse a partir de un punto inicial. Para $n \geq 3$, cualquier circuito puede parearse con uno en la dirección opuesta que determine el mismo circuito hamiltoniano. En consecuencia, hay un total de $H = (n - 1)!/2$ circuitos hamiltonianos.

PROBLEMAS SUPLEMENTARIOS

TERMINOLOGÍA DE GRAFOS

- 8.34 Considere el grafo G en la figura 8-58. Encuentre:
- a) el grado de cada vértice (y compruebe el teorema 8.1);
 - b) todos los caminos simples de A a L ;
 - c) todos los recorridos (aristas distintas) de B a C ;
 - d) $d(A, C)$, la distancia de A a C ;
 - e) $\text{diám}(G)$, el diámetro de G .

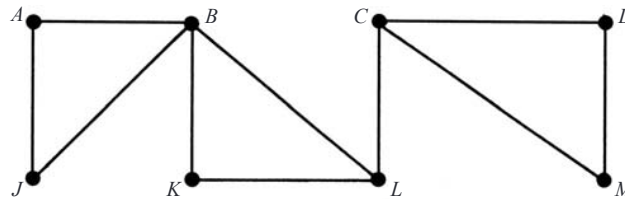


Figura 8-58

- 8.35 Considere el grafo en la figura 8-58. Encuentre (en caso de haberlos): a) todos los ciclos; b) todos los puntos de corte; c) todos los puentes.
- 8.36 Considere el grafo en la figura 8-58. Encuentre el subgrafo $H = H(V', E')$ de G , donde V' es igual a:
- a) $\{B, C, D, J, K\}$ b) $\{A, C, J, L, M\}$ c) $\{B, D, J, M\}$ d) $\{C, K, L, M\}$
- ¿Cuáles son isomorfos y cuáles son homeomorfos?
- 8.37 Suponga que un grafo G contiene dos caminos distintos de un vértice u a un vértice v . Demuestre que G tiene un ciclo.
- 8.38 Suponga que G es un grafo finito libre de ciclos con por lo menos una arista. Demuestre que G tiene por lo menos dos vértices de grado 1.
- 8.39 Demuestre que un grafo conexo G con n vértices debe tener por lo menos $n - 1$ aristas.
- 8.40 Encuentre el número de grafos conexos que hay con cuatro vértices. (Dibújelos.)
- 8.41 Sea G un grafo conexo. Demuestre:
- a) Si en G hay un ciclo C que contiene una arista e , entonces $G - e$ sigue siendo conexo.
 - b) Si $e = \{u, v\}$ es una arista tal que $G - e$ es inconexo, entonces u y v pertenecen a componentes distintos de $G - e$.
- 8.42 Suponga que G tiene V vértices y E aristas. Sean M y m que denotan, respectivamente, el máximo y el mínimo de los grados de los vértices en G . Demuestre que $m \leq 2E/V \leq M$.
- 8.43 Considere los dos pasos siguientes en un grafo G : 1) Eliminar una arista. 2) Eliminar un vértice y todas las aristas que contienen a ese vértice. Demuestre que todo subgrafo H de un grafo finito G puede obtenerse mediante una secuencia que consta de estos dos pasos.

GRAFOS RECORRIBLES, CIRCUITOS EULERIANOS Y HAMILTONIANOS

- 8.44 Considere los grafos K_5 , $K_{3,3}$ y $K_{2,3}$ en la figura 8-59. Encuentre un camino euleriano (recorrible) o un circuito euleriano de cada grafo, si existe. En caso de no existir, explique por qué.
- 8.45 Considere cada grafo en la figura 8-59. Encuentre un camino o un circuito hamiltoniano, si existe. En caso de no existir, explique por qué.
- 8.46 Demuestre que K_n tiene $H = (n - 1)!/2$ circuitos hamiltonianos. En particular, encuentre el número de circuitos hamiltonianos para el grafo K_5 en la figura 8-59a).
- 8.47 Suponga que G y G^* son grafos homeomorfos. Demuestre que G es recorrible (euleriano) si y sólo si G^* es recorrible (euleriano).

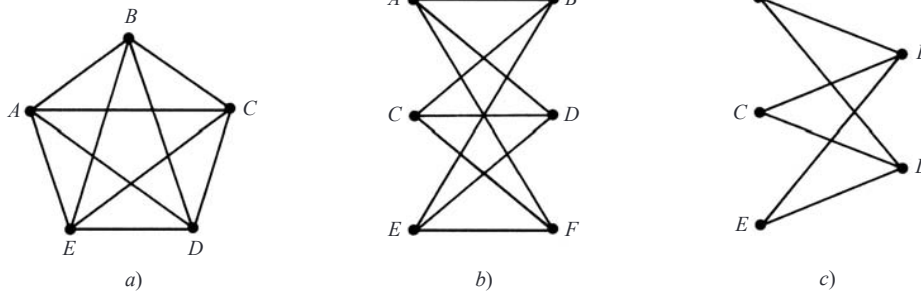


Figura 8-59

GRAFOS ESPECIALES

8.48 Dibuje dos grafos 3-regular con a) ocho vértices; b) nueve vértices.

8.49 Considere el grafo completo K_n .

- a) Encuentre el diámetro de K_n .
- b) Encuentre el número m de aristas en K_n .
- c) Encuentre el grado de cada vértice en K_n .
- d) Encuentre los valores de n para los que K_n es: i) recorrible; ii) regular.

8.50 Considere el grafo completo $K_{m,n}$.

- a) Encuentre el diámetro de $K_{m,n}$.
- b) Encuentre el número E de aristas en $K_{m,n}$.
- c) Encuentre los $K_{m,n}$ que son recorribles.
- d) ¿Cuáles de los grafos $K_{m,n}$ son isomorfos y cuáles son homeomorfos?

8.51 El n -cubo, denotado por Q_n , es el grafo cuyos vértices son las 2^n cadenas de bits de longitud n , y donde dos vértices son adyacentes si sólo difieren por una posición. En las figuras 8-60a) y b) se muestran los n -cubos Q_2 y Q_3 .

- a) Encuentre el diámetro de Q_n .
- b) Encuentre el número m de aristas en Q_n .
- c) Encuentre el grado de cada vértice en Q_n .
- d) Encuentre los valores de n para los que Q_n es recorrible.
- e) Encuentre un circuito hamiltoniano (denominado *código Gray*) para i) Q_3 ; ii) Q_4 .

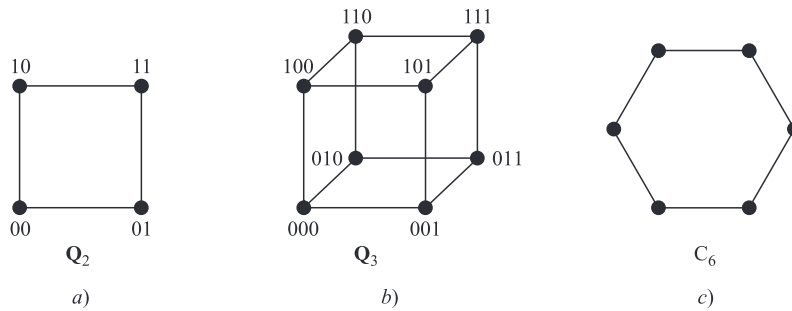


Figura 8-60

- 8.52 El n -ciclo, denotado por C_n , es el grafo que consta de un solo ciclo de longitud n . En la figura 8-60c) se muestra el 6-ciclo C_6 . a) Encuentre el número de vértices y aristas en C_n . b) Encuentre el diámetro de C_n .
- 8.53 Describa los grafos conexos que son bipartidos y regulares.

ÁRBOLES

- 8.54 Dibuje todos los árboles con cinco vértices o menos.
- 8.55 Encuentre el número de árboles con siete vértices.
- 8.56 Encuentre el número de árboles de expansión en la figura 8-61a).
- 8.57 Encuentre el peso de un árbol de expansión mínima en la figura 8-61b).

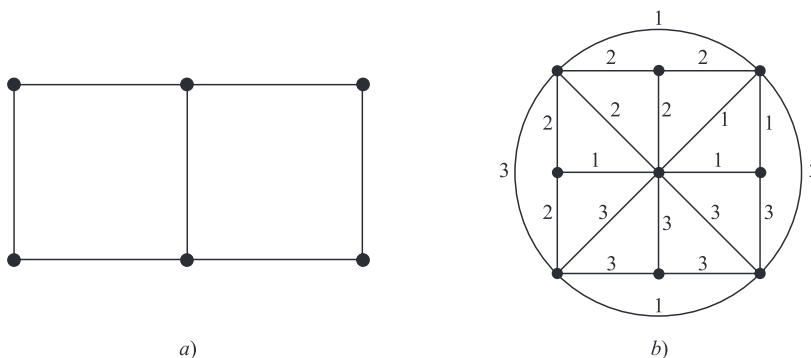


Figura 8-61

- 8.58 Demuestre que cualquier árbol es un grafo bipartido.
- 8.59 ¿Cuáles grafos bipartidos completos $K_{m,n}$ son árboles?

GRAFOS PLANAS, MAPAS, COLOREADOS

- 8.60 De ser posible dibuje una representación plana de cada grafo G en la figura 8-62; en caso contrario, demuestre que tiene un subgrafo homeomorfo a K_5 o a $K_{3,3}$.

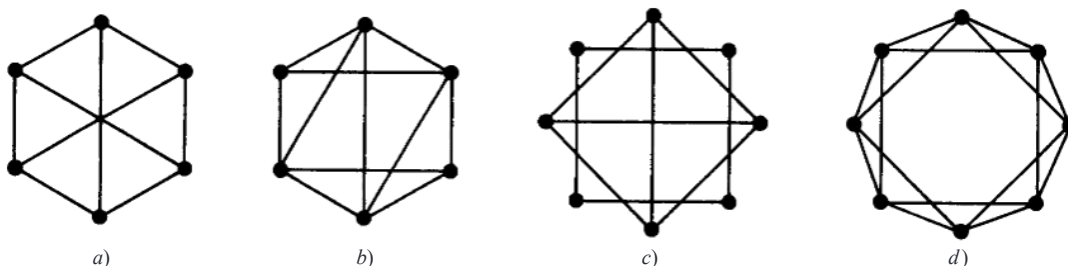


Figura 8-62

- 8.61 Demuestre que el 3-cubo Q_3 [figura 8-60b)] es plano.
- 8.62 Para el mapa en la figura 8-63, encuentre el grado de cada región y compruebe que la suma de los grados de las regiones es igual al doble del número de aristas.
- 8.63 Cuente el número V de vértices, el número E de aristas y el número R de regiones de cada uno de los mapas en la figura 8-64, y compruebe la fórmula de Euler.

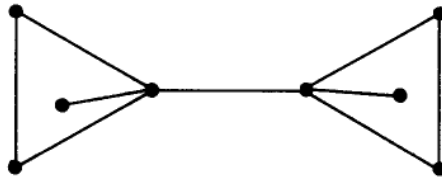


Figura 8-63

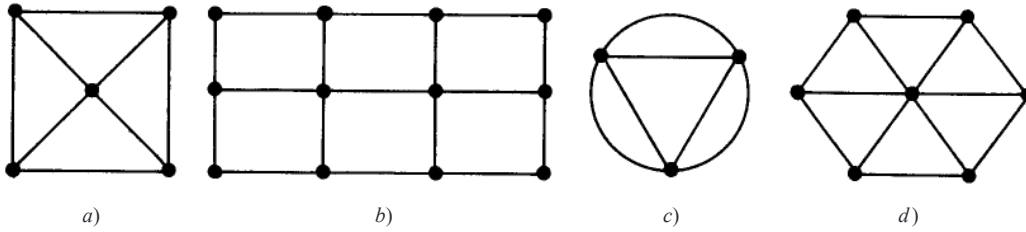


Figura 8-64

- 8.64 Encuentre el número mínimo de colores necesarios para pintar las regiones de cada mapa en la figura 8-64.
- 8.65 Dibuje el mapa dual a cada mapa en la figura 8-64.
- 8.66 Aplique el algoritmo de Welch y Powell para pintar cada grafo en la figura 8-65. Encuentre el número cromático n del grafo.

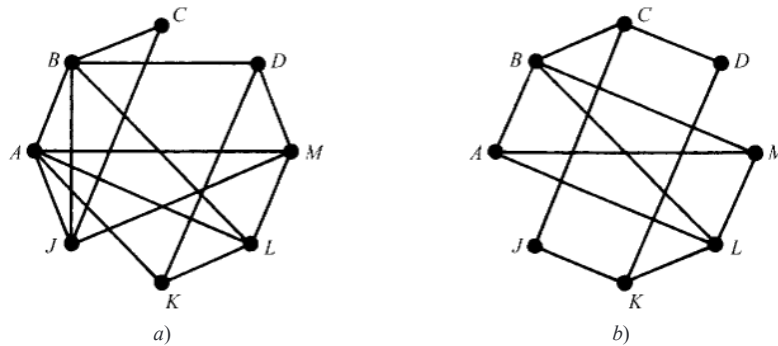


Figura 8-65

REPRESENTACIÓN SECUENCIAL DE GRAFOS

- 8.67 Encuentre la matriz de adyacencia A de cada multigrafo en la figura 8-66.

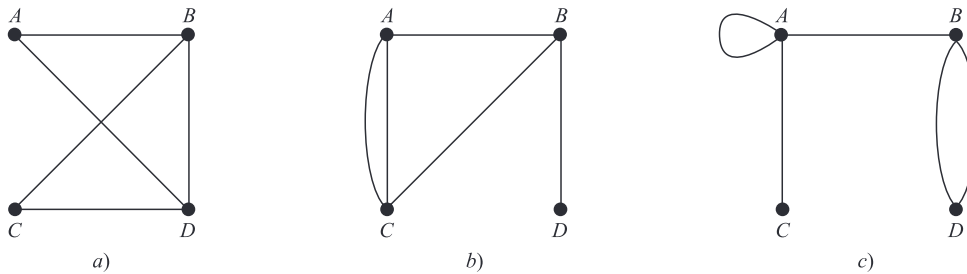


Figura 8-66

8.68 Dibuje el multigrafo G correspondiente a cada una de las siguientes matrices de adyacencia:

$$a) A = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}; \quad b) A = \begin{bmatrix} 1 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 \\ 2 & 0 & 2 & 2 \end{bmatrix}$$

8.69 Suponga que un grafo G es bipartido. Demuestre que es posible ordenar los vértices de G de modo que su matriz de adyacencia A tenga la forma: $A = \begin{bmatrix} 0 & B \\ C & 0 \end{bmatrix}$

REPRESENTACIÓN ENLAZADA DE GRAFOS

8.70 Suponga que un grafo G se almacena en la memoria como en la figura 8-67.

		Archivo vértice								
		1	2	3	4	5	6	7	8	
START	7	VERTEX	C		F	E	A		B	D
		NEXT-V	0		5	1	8		3	4
		PTR	2		11	6	12		4	1

		Archivo arista											
		1	2	3	4	5	6	7	8	9	10	11	12
ADJ	7	7	7	4	5		7	1		8	3	1	7
	0	10	0	7		0	9		3	0	0	0	0

Figura 8-67

- a) Enumere los vértices en el orden en que aparecen en la memoria.
- b) Encuentre la estructura de adyacencia de G ; es decir, encuentre la lista de adyacencia $ady(v)$ de cada vértice v de G .

8.71 Muestre la estructura de adyacencia (EA) para cada grafo G en la figura 8-59.

8.72 En la figura 8.68a) se muestra un grafo G que representa seis ciudades A, B, \dots, F unidas por siete autopistas numeradas 22, 33, ..., 88. Muestre la forma en que G puede mantenerse en la memoria mediante una presentación ligada con arreglos ordenados para las ciudades y las autopistas numeradas. (Observe que VERTEX es un arreglo ordenado, de modo que el campo NEXT-V no es necesario).

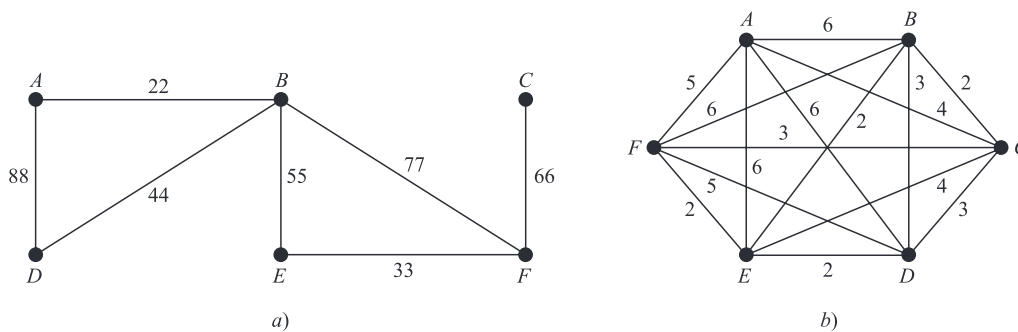


Figura 8-68

PROBLEMA DEL AGENTE VIAJERO

- 8.73 Aplique el algoritmo del vecino más próximo al grafo ponderado completo G en la figura 8-68b) empiece en: a) el vértice A ; b) el vértice B .
- 8.74 Considere el grafo ponderado completo G en la figura 8-57 con 5 vértices.
 - a) Empiece en el vértice A y enumere los $H = (n - 1)!/2 = 12$ circuitos hamiltonianos de G , y encuentre el peso de cada uno.
 - b) Encuentre un circuito hamiltoniano de peso mínimo.

ALGORITMOS DE GRAFS

- 8.75 Considere el grafo G en la figura 8-57 (donde los vértices están ordenados alfabéticamente).
 - a) Encuentre la estructura de adyacencia (EA) de G .
 - b) Use el algoritmo de búsqueda en profundidad 8.5 sobre G , empiece en el vértice C , encuentre la secuencia STACK y el orden en que se procesan los vértices.
 - c) Repita el inciso b); ahora empiece en el vértice K .
- 8.76 Use el algoritmo de búsqueda en anchura 8.6 sobre el grafo G en la figura 8-57 para encontrar la secuencia QUEUE y el orden en que se procesan los vértices, empiece en: a) vértice C ; b) vértice K .
- 8.77 Repita el problema 8.75 para el grafo G en la figura 8-65a).
- 8.78 Repita el problema 8.76 para el grafo G en la figura 8-65a).
- 8.79 Repita el problema 8.75 para el grafo G en la figura 8-65b).
- 8.80 Repita el problema 8.76 para el grafo G en la figura 8-65b).

Respuestas a los problemas suplementarios

- 8.34 a) 2, 4, 3, 2, 2, 3, 2; b) $ABL, ABKL, AJBL, AJBKL$; c) $BLC, BKLC, BAJBLC, BAJBKLC$; d) 3; e) 4.
- 8.35 a) $AJBA, BKLB, CDMC$; b) B, C, L ; c) sólo $\{C, L\}$.
- 8.36 a) $E' = \{BJ, BK, CD\}$; b) $E' = \{AJ, CM, LC\}$; c) $E' = \{BJ, DM\}$; d) $E' = \{KL, LC, CM\}$. También, a) y b) son isomorfos, y a), b) y c) son homeomorfos.
- 8.38 *Sugerencia:* Considere un camino simple máximo α , y demuestre que sus puntos extremos tienen grado 1.
- 8.40 Hay cinco de ellos, como se muestra en la figura 8-69.

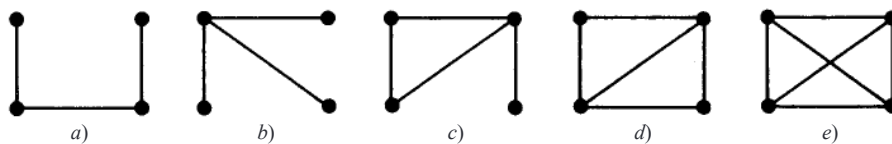


Figura 8-69

- 8.42 *Sugerencia:* Use el teorema 8.1.
- 8.43 Primero elimine todas las aristas en G que no están en H ; luego, elimine todos los vértices en G que no están en H .
- 8.44 a) euleriano, puesto que todos los vértices son pares: $ABCDEACEBDA$. b) Ninguno, puesto que cuatro vértices son impares. c) Camino euleriano que empieza en B y termina en D (o viceversa): $BADCBED$.

- 8.45 a) $ABCDEA$; b) $ABCDEF A$; c) ninguno, puesto que B o D deben visitarse dos veces en cualquier camino cerrado que incluya todos los vértices.
- 8.46 $(5 - 1)!/2 = 12$.
- 8.47 *Sugerencia:* Agregar un vértice al dividir una arista no modifica el grado de los vértices originales y simplemente agrega un vértice de grado par.
- 8.48 a) Los dos grafos 3-regular en la figura 8-70 no son isomorfos; b) tiene un 5-ciclo, pero a) no. b) No hay ninguno. La suma de los grados de un grafo r -regular con s vértices es igual a rs , y rs debe ser par.

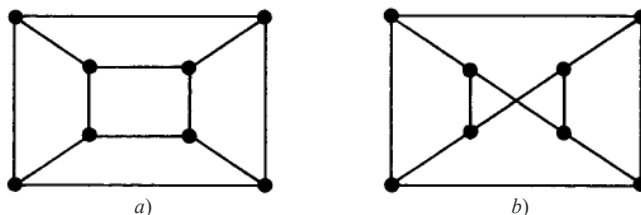


Figura 8-70

- 8.49 a) $\text{diám}(K_1) = 0$; todas las demás tienen diámetro 1; b) $m = C(n, 2) = n(n - 1)/2$; c) $n - 1$; d) i) $n = 2$ y n impar; ii) toda n .
- 8.50 a) $\text{diám}(K_{1,1}) = 1$; todas las demás tienen diámetro 2; b) $E = mn$; c) $K_{1,1}$ y $K_{1,2}$, y todo $K_{m,n}$ donde m y n sean pares; d) ninguno es isomorfo; sólo $K_{1,1}$ y $K_{1,2}$ son homeomorfos.
- 8.51 a) n ; b) $n2^{n-1}$; c) n ; d) $n = 1$, par, e) considere la matriz de 4×16 :

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

que muestra la forma en que Q_4 (las columnas de M) se obtiene a partir de Q_3 . Es decir, la submatriz superior izquierda de 3×8 de M es Q_3 , la submatriz superior derecha de 3×8 de M es Q_3 escrita al revés, y el último renglón consta de ocho ceros seguidos de ocho unos.

- 8.52 a) n y n ; b) $n/2$ cuando n es par, $(n + 1)/2$ cuando n es impar.
- 8.53 $K_{m,m}$ es bipartido y m -regular. También, al empezar con $K_{m,m}$, se eliminan m aristas ajenas para obtener un grafo bipartido que es $(m - 1)$ -regular, se eliminan otros m aristas ajenas para obtener un grafo bipartido que es $(m - 2)$ -regular y así sucesivamente. Estos grafos pueden ser inconexos, pero sus componentes conexos poseen las propiedades deseadas.
- 8.54 Hay ocho árboles así, como se muestra en la figura 8-71. El grafo con un vértice y ninguna arista se denomina *árbol trivial*.

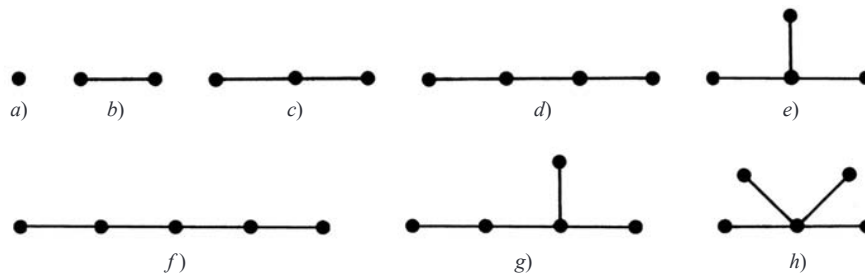


Figura 8-71

- 8.55 10
 8.56 15
 8.57 $1 + 1 + 1 + 1 + 1 + 2 + 2 + 3 = 12$.
 8.59 $m = 1$.
 8.60 Sólo *a*) no es plano, y $K_{3,3}$ es un subgrafo.
 8.61 La figura 8-70*a*) es una representación plana de Q_3 .
 8.62 La región exterior tiene grado 8, y las otras dos regiones tienen grado 5.
 8.63 *a*) 5, 8, 5; *b*) 12, 17, 7; *c*) 3, 6, 5; *d*) 7, 12, 7.
 8.64 *a*) 3; *b*) 3; *c*) 2; *d*) 3.
 8.65 Vea la figura 8-72.

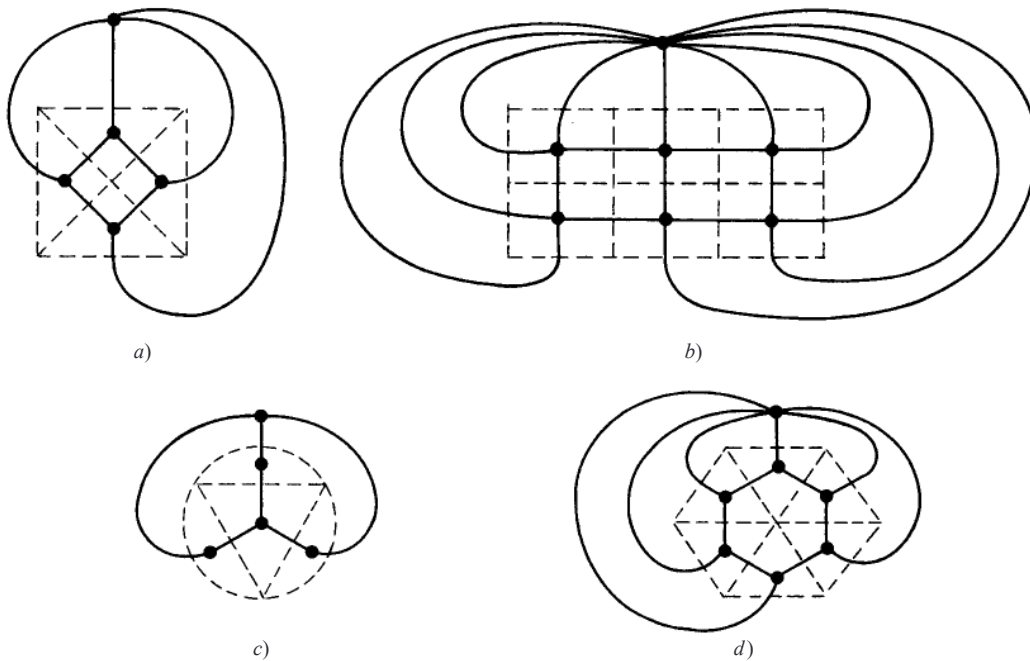


Figura 8-72

- 8.66 *a*) $n = 3$; *a*) $n = 4$.

8.67 *a*) $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$; *b*) $\begin{bmatrix} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$; *c*) $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$

- 8.68 Vea la figura 8-73.
 8.69 Sean M y N los dos conjuntos ajenos de vértices que determinan el grafo bipartido G . Primero se ordenan los vértices en M y luego se ordenan los vértices en N .
 8.70 *a*) B, F, A, D, E, C .
b) $G = [A:B; B:A, C, D, E; C:F; D:B; E:B; F:C]$.

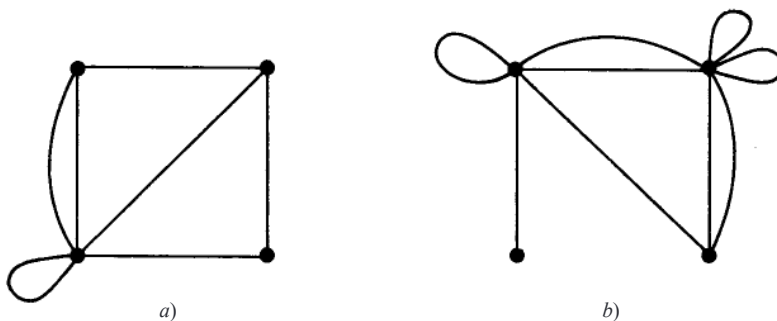


Figura 8-73

- 8.71 a) Cada vértice es adyacente a los otros cuatro vértices.
 b) $G = [A:B, D, F; B:A, C, E; C:B, D, F; D:A, C, E; E:B, D, F; F:A, C, E]$.
 c) $G = [A:B, D; B:A, C, E; C:B, D; D:A, C, E; E:B, D]$.

8.72. Vea la figura 8-74.

		Archivo vértice							
		1	2	3	4	5	6	7	8
VERTEX		A	B	C	D	E	F		
PTR		1	2	9	14	8	12		

		Archivo arista														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NUMBER		22	22	33	33	44	44	55	55	66	66	77	77	88	88	
ADJ		2	1	6	5	4	2	5	2	6	3	6	2	4	1	
NEXT		13	5	0	0	7	0	11	3	0	4	0	10	0	6	

Figura 8-74

- 8.73 a) $|ACBEDFA| = 20$ o $|A CBEFDA| = 21$; B) $|BCFEDAB| = 21$ o $|BCDEFAB| = 20$
- 8.74 a) $|ABCDEA| = 775$, $|ABCEDA| = 725$, $|ABDCEA| = 1100$, $|ABDECA| = 900$, $|ABECDA| = 1050$,
 $|ABEDCA| = 900$, $|ACBDEA| = 825$, $|ACBEDA| = 775$, $|ACDBEA| = 1150$, $|ACEBDA| = 1100$,
 $|ADBCEA| = 975$;
 b) $|ABCEDA| = 725$
- 8.75 a) $G = [A:BJ; B:AJKL; C:DLM; D:CM; J:AB; K:BL; L:BCK; M:CD]$
 b) [STACK : C, MLD, DL, L, KB, B, J, A], CMDLKBJA
 c) [STACK : K, LB, CB, MDB, DB, B, JA, A], KLCMDBJA
- 8.76 a) [QUEUE : C, MLD, ML, L, KB, JAK, JA, J], CDMLBKAJ
 b) [QUEUE : K, LB, JAL, CJA, CJ, C, MD, M], KBLAJCDM
- 8.77 a) $G = [A:BMJKL; B:ACDJL; C:BJ; D:BKM; J:ABCM; K:ADL; L:ABKM; M:ADJL]$
 b) [STACK : C, JB, MBA, LDAB, KBAD, DAB, AB, B], CJMLKDAB
 c) [STACK : K, LDA, MBAD, JDAB, CBAD, BAD, AD, D], KLMJCBAD
- 8.78 a) [QUEUE : C, JB, LDAJ, MLDA, KMLD, KML, KM, K], CBJADLMK
 b) [QUEUE : K, LDA, JMBLD, JMBL, CJMB, CJM, CJ, C], KADLBMJC

- 8.79 a) $G = [A:BLM; B:ACLM; C:BDJ; D:CK; J:CK; K:DJL; L:ABKM; M:ABL]$
b) [STACK : C, JDB, KDB, LDB, MBAD, BAD, AD, D], CJKLMBAD
c) [STACK : K, LJD, MBAJD, BAJD, CAJD, JDA, DA, A], KLMBCJDA
- 8.80 a) [QUEUE : C, JDB, MLAJD, KMLAJ, KMLA, KML, KM, K], CBDJALMK
b) [QUEUE : K, LJD, CLJ, CL, MBAC, MBA, MB, M], KDJLCABM

9

Grafos dirigidos

CAPÍTULO

9.1 INTRODUCCIÓN

Los *grafos dirigidos* son grafos con aristas orientadas en una dirección. Dichos grafos son útiles en sistemas dinámicos como computadoras digitales o sistemas de flujo y es esta característica agregada lo que hace más difícil la determinación de ciertas propiedades de los grafos. Es decir, el procesamiento de estos grafos puede ser semejante a recorrer una ciudad con muchas calles de un solo sentido.

En este capítulo se proporcionan las definiciones y propiedades básicas de los grafos dirigidos. Muchas de las definiciones son semejantes a las del capítulo precedente sobre grafos (no dirigidos). Sin embargo, por razones pedagógicas, este capítulo es esencialmente independiente del capítulo precedente.

9.2 GRAFOS DIRIGIDOS

Un *grafo dirigido* G , o *digrafo* (o simplemente *grafo*), consta de dos partes:

- i) Un conjunto V cuyos elementos ordenados se denominan *vértices*, *nodos* o *puntos*.
- ii) Un conjunto E de pares *ordenados* (u, v) de vértices que se denominan *arcos*, *aristas dirigidas*, o simplemente *aristas*.

Cuando se desea recalcar las dos partes de G , se escribe $G(V, E)$. También se escribe $V(G)$ y $E(G)$ para denotar, respectivamente, el conjunto de vértices y el conjunto de aristas de un grafo G . (En caso de que no se plantee explícitamente, el contexto suele determinar si un grafo es o no un grafo dirigido.)

Suponga que $e = (u, v)$ es una arista en un grafo dirigido G . Entonces se usa la siguiente terminología:

- a) e empieza en u y termina en v .
- b) u es el *origen* o *punto inicial* de e , y v es el *destino* o *punto terminal* de e .
- c) v es un *sucesor* de u .
- d) u es *adyacente a* v , y v es *adyacente a* u .

Si $u = v$, entonces e se denomina *lazo*.

El conjunto de todos los sucesores de un vértice u es importante; se denota y define formalmente por

$$\text{suc}(u) = \{v \in V \mid \text{existe una arista } (u, v) \in E\}$$

Se denomina *lista de sucesores* o *lista de adyacencia* de u .

Una *ilustración* de un grafo dirigido G es una representación de G en el plano. Es decir, cada vértice u de G se representa por un punto (o un círculo pequeño) y cada arista (dirigida) $e = (u, v)$ se representa por una flecha o una curva dirigida desde el punto inicial u de e hasta el punto terminal v . Un grafo dirigido G suele representarse por su ilustración, más que mediante la enumeración explícita de sus vértices y aristas.

Si las aristas y/o los vértices de un grafo dirigido G se etiquetan con algún tipo de datos, entonces G se denomina *grafo dirigido etiquetado*.

Se dice que un grafo dirigido $\{V, E\}$ es *finito* si su conjunto V de vértices y su conjunto E de aristas son finitos.

EJEMPLO 9.1

- a) Considere el grafo dirigido G que se muestra en la figura 9-1a). Consta de cuatro vértices A, B, C, D , es decir, $V(G) = \{A, B, C, D\}$ y las siete aristas siguientes:

$$E(G) = \{e_1, e_2, \dots, e_7\} = \{(A, D), (B, A), (B, A), (D, B), (B, C), (D, C), (B, B)\}$$

Se dice que las aristas e_2 y e_3 son *paralelas*, puesto que ambas empiezan en B y terminan en A . La arista e_7 es un *lazo*, ya que empieza y termina en B .

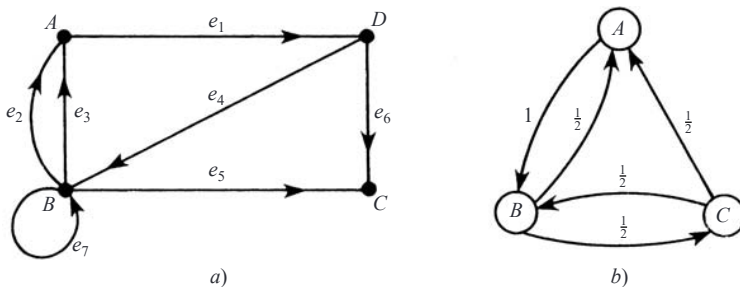


Figura 9-1

- b) Suponga que tres muchachos A, B, C , se lanzan una pelota entre sí de modo que A siempre la lanza a B , pero B y C tienen la misma probabilidad de lanzar la pelota a A , así como uno al otro. Este sistema dinámico se representa en la figura 9-1b), donde las aristas están etiquetadas con las probabilidades respectivas; es decir, A lanza la pelota a B con probabilidad 1, B la lanza a A y a C con probabilidad igual a $1/2$ para cada uno, y C la lanza a A y a B con probabilidad igual a $1/2$ para cada uno.

Subgrafos

Sea $G = G(V, E)$ un grafo dirigido, y sea V' un subconjunto del conjunto V de vértices de G . Suponga que E' es un subconjunto de E tal que los puntos terminales de las aristas de E' pertenecen a V' . Entonces $H(V', E')$ es un grafo dirigido, y se denomina un *subgrafo* de G . En particular, si E' contiene todas las aristas en E cuyos puntos terminales pertenecen a V' , entonces $H(V', E')$ se denomina subgrafo de G *generado* o *determinado* por V' . Por ejemplo, para el grafo $G = G(V, E)$ en la figura 9-1a), $H(V', E')$ es el subgrafo de G determinado por el conjunto de vértices V' , donde

$$V' = \{B, C, D\} \quad \text{y} \quad E' = \{e_4, e_5, e_6, e_7\} = \{(D, B), (B, C), (D, C), (B, B)\}$$

9.3 DEFINICIONES BÁSICAS

En esta sección se analizan las cuestiones de grados de los vértices, caminos y conectividad en un grafo dirigido.

Grados

Suponga que G es un grafo dirigido. El *grado de salida* (*outdegree*) de un vértice v de G , que se escribe $\text{outdeg}(v)$, es el número de aristas que empiezan en v , y el *grado de entrada* (*indegree*) de v , que se escribe $\text{indeg}(v)$, es el número de aristas que terminan en v . Puesto que cada arista empieza y termina en un vértice, de inmediato se obtiene el siguiente teorema.

Teorema 9.1: La suma de los grados de salida de los vértices de un grafo dirigido G es igual a la suma de los grados de entrada de los vértices, que es igual al número de aristas en G .

Un vértice v con grado de entrada cero se denomina *fuentes*, y un vértice con grado de salida cero se denomina *sumidero*.

EJEMPLO 9.2 Considere el grafo G en la figura 9-1a). Se tiene

$$\begin{aligned} \text{outdeg}(A) = 1, \quad \text{outdeg}(B) = 4, \quad \text{outdeg}(C) = 0, \quad \text{outdeg}(D) = 2, \\ \text{indeg}(A) = 2, \quad \text{indeg}(B) = 2, \quad \text{indeg}(C) = 2, \quad \text{indeg}(D) = 1. \end{aligned}$$

Como era de esperar, la suma de los grados de salida es igual a la suma de los grados de entrada, que es igual al número 7 de aristas. El vértice C es un sumidero, puesto que ninguna arista empieza en C . El grafo no tiene fuentes.

Caminos

Sea G un grafo dirigido. Los conceptos de camino, camino simple, recorridos y ciclo válidos para grafos no dirigidos se aplican a los grafos dirigidos G , excepto que las direcciones de las aristas deben coincidir con la dirección del camino. Con más precisión:

i) Un *camino (dirigido)* P en G es una secuencia alterna de vértices y aristas dirigidas; Por ejemplo,

$$P = (v_0, e_1, v_1, e_2, v_2, \dots, e_n, v_n)$$

tal que la arista e_i empieza en v_{i-1} y termina en v_i . Si no hay ambigüedad, P se denota por su secuencia de vértices o su secuencia de aristas.

ii) La *longitud* del camino P es n , su número de aristas.

iii) Un *camino simple* es un camino con vértices distintos. Un *recorrido* es un camino con aristas distintas.

iv) Un *camino cerrado* tiene los mismos vértices inicial y final.

v) Un *camino de expansión* contiene todos los vértices de G .

vi) Un *ciclo* (o *circuito*) es un camino cerrado con vértices distintos (excepto el primero y el último).

vii) Un *semicamino* es lo mismo que un camino, excepto que la arista e_i puede empezar en v_{i-1} o en v_i y terminar en el otro vértice. Los *semirecorridos* y los *caminos semisimples* se definen en forma análoga.

Un vértice v es *alcanzable* desde un vértice u si hay una camino de u a v . Si v es alcanzable desde u , entonces (al eliminar las aristas redundantes) debe haber un camino simple de u a v .

EJEMPLO 9.3 Considere el grafo G en la figura 9-1a).

a) La secuencia $P_1 = (D, C, B, A)$ es un *semicamino*, no un camino, puesto que (C, B) no es una arista; es decir, la dirección de $e_5 = (C, B)$ no coincide con la dirección de P_1 .

b) La secuencia $P_2 = (D, B, A)$ es un camino de D a A puesto que (D, B) y (B, A) son aristas. Por tanto, A es alcanzable desde D .

Conectividad

En un grafo dirigido G hay tres tipos de conectividad:

- i) G es *fuertemente conexo* o *fuerte* si para cualquier par de vértices u y v en G , hay un camino de u a v y un camino de v a u ; es decir, cada uno es alcanzable desde el otro.
- ii) G es *unilateralmente conexo* o *unilateral* si para cualquier par de vértices u y v en G , hay un camino de u a v o un camino de v a u ; es decir, uno de ellos es alcanzable desde el otro.
- iii) G es *débilmente conexo* o *débil* si entre cualquier par de vértices u y v en G hay un semicamino.

Sea G' el grafo (no dirigido) que se obtiene a partir de un grafo dirigido G al dejar que todas las aristas de G sean no dirigidas. Resulta evidente que G es débilmente conexo si y sólo si el grafo G' es conexo.

Observe que fuertemente conexo implica unilateralmente conexo, lo cual implica débilmente conexo. Se dice que G es *estrictamente unilateral* si es unilateral pero no fuerte, y se dice que G es *estrictamente débil* si es débil pero no unilateral.

La conectividad puede caracterizarse en términos de los caminos de expansión como sigue:

Teorema 9.2: Sea G un grafo dirigido finito. Entonces:

- i) G es fuerte si y sólo si G tiene un camino de expansión cerrado.
- ii) G es unilateral si y sólo si G tiene un camino de expansión.
- iii) G es débil si y sólo si G tiene un semicamino de expansión.

EJEMPLO 9.4 Considere el grafo G en la figura 9-1a). Es débilmente conexo puesto que el grafo no dirigido subyacente es conexo. No hay ningún camino desde C hasta cualquier otro vértice; es decir, C es un sumidero, de modo que G no es fuertemente conexo. Sin embargo, $P = (B, A, D, C)$ es un camino de expansión, de modo que G es unilateralmente conexo.

Los grafos con fuentes y sumideros aparecen en muchas aplicaciones (como diagramas de flujo y redes). Una condición suficiente para que tales vértices existan es la siguiente.

Teorema 9.3: Suponga que un grafo dirigido finito G es libre de ciclos; es decir, que no contiene ciclos (dirigidos). Entonces G contiene una fuente y un sumidero.

Demostración: Sea $P = (v_0, v_1, \dots, v_n)$ un camino simple de longitud máxima, que existe porque G es finito. Entonces el último vértice v_n es un sumidero; en caso contrario, una arista (v_n, u) extiende a P o forma un ciclo si $u = v_i$ para alguna i . En forma semejante, el primer vértice v_0 es una fuente.

9.4 ÁRBOLES CON RAÍZ

Recuerde que un grafo de un árbol es un grafo conexo libre de ciclos; es decir, un grafo conexo sin ningún ciclo. Un *árbol T con raíz* es un grafo de un árbol con un vértice designado r al que se le denomina *raíz* del árbol. Puesto que existe un camino simple único de la raíz r a cualquier otro vértice v en T , esto determina una dirección hacia las aristas de T . Por tanto, T puede considerarse como un grafo dirigido. Observe que cualquier árbol puede hacerse un árbol con raíz con la simple elección de uno de los vértices como la raíz.

Considere un árbol T con raíz cuya raíz es r . La longitud del camino de la raíz r a cualquier vértice v se denomina *nivel* (o *profundidad*) de v , y el máximo nivel de vértice se denomina *profundidad* del árbol. Los vértices con grado 1, que no sean la raíz r , se denominan *hojas* de T , y un camino dirigido de un vértice a una hoja se denomina *rama*.

Por lo general, un árbol T con raíz se ilustra con la raíz en la parte superior del árbol. En la figura 9-2a) se muestra un árbol T con raíz r y 10 vértices más. El árbol tiene cinco hojas d, f, h, i y j . Observe que $nivel(a) = 1$, $nivel(f) = 2$, $nivel(j) = 3$. Además, la profundidad del árbol es 3.

El hecho de que un árbol T con raíz proporcione una dirección a las aristas significa que es posible asignar una relación de precedencia entre los vértices. De manera más precisa, se dice que un vértice u *precede* a un vértice v o que v *sigue a* u si hay un camino (dirigido) de v a u . En particular, se dice que v *sigue inmediatamente a* u si (u, v) es

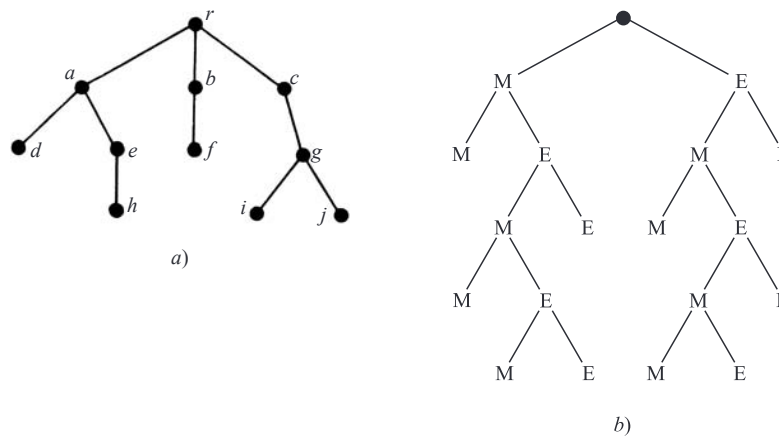


Figura 9-2

una arista; es decir, si v sigue a u y v es adyacente a u . Observe que cualquier vértice v , que no sea la raíz, sigue inmediatamente a un vértice único, aunque v puede ser seguido inmediatamente por más de un vértice. Por ejemplo, en la figura 9-2a), el vértice j sigue a c aunque inmediatamente sigue a g . Ambos vértices i y j siguen inmediatamente a g .

Un árbol T con raíz también es un dispositivo útil para enumerar todas las posibilidades lógicas de una secuencia de eventos donde cada evento puede ocurrir en una forma finita de formas. Este hecho se ilustra con el siguiente ejemplo.

EJEMPLO 9.5 Suponga que Marcos y Eric juegan en un torneo de tenis de modo que la primera persona en ganar dos juegos seguidos o quien gane un total de tres juegos gana el torneo. Encuentre el número de formas en que puede desarrollarse el torneo.

El árbol con raíz en la figura 9-2b) muestra las diferentes formas en que puede desarrollarse el torneo. Hay 10 hojas que corresponden a las 10 formas en que puede ocurrir el torneo:

MM, MEMM, MEMEM, MEMEE, MEE, EMM, EMEMM, EMEME, EMEE, EE

Específicamente, el camino de la raíz a la hoja describe quién ganó cuáles juegos en el torneo.

Árboles con raíz ordenados

Considere un árbol T con raíz en el que las aristas que salen de cada vértice están ordenadas. Entonces se tiene el concepto de *árbol con raíz ordenado*. Los vértices de un árbol así pueden etiquetarse (o *direccionarse*) en forma sistemática como: primero se asigna 0 a la raíz r . Luego se asigna 1, 2, 3, ..., a los vértices que siguen de inmediato a r según la forma en que se ordenaron las aristas. Enseguida se etiquetan los vértices restantes: si a es la etiqueta de un vértice v , entonces $a.1, a.2, \dots$ se asignan a los vértices que siguen de inmediato a v según la forma en que se ordenaron las aristas. Este sistema de direcciones se ilustra en la figura 9-3a), donde las aristas se representan de izquierda a derecha según su orden. Observe que el número de puntos decimales en cualquier etiqueta es uno menos que el nivel del vértice. Este sistema de identificación se denomina *sistema universal de direcciones* para un árbol con raíz ordenado.

El sistema universal de direcciones constituye una forma importante para describir (o almacenar) linealmente un árbol con raíz ordenado. De manera más concisa, dadas las direcciones a y b , se hace $a < b$ si $b = a.c$ (es decir, a es un segmento inicial de b), o si hay enteros positivos m y n con $m < n$ tales que

$$a = r.m.s \quad y \quad b = r.n.t$$

Este orden se denomina *orden lexicográfico* puesto que es semejante a la forma en que las palabras están dispuestas en un diccionario. Por ejemplo, las direcciones en la figura 9-3a) están ordenadas linealmente según se representa en la figura 9-3b). Ese orden lexicográfico es idéntico al orden que se obtiene al moverse hacia abajo a partir de la rama

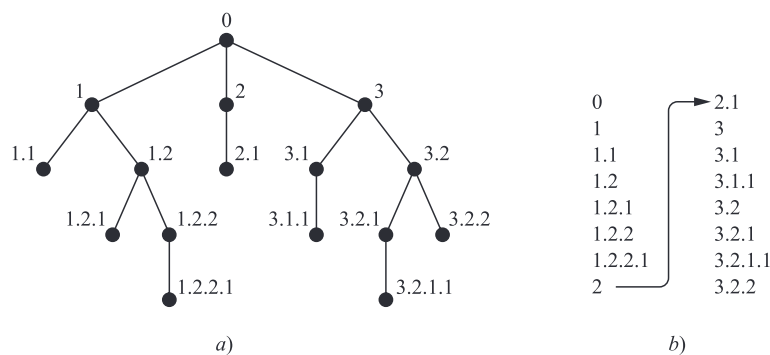


Figura 9-3

más a la izquierda del árbol, en seguida hacia la siguiente rama a la derecha, luego la segunda rama a la derecha y así sucesivamente.

9.5 REPRESENTACIÓN SECUENCIAL DE GRAFOS DIRIGIDOS

Hay dos formas fundamentales para mantener un grafo dirigido G en la memoria de una computadora. Una forma, denominada *representación secuencial* de G , es por medio de su matriz de adyacencia A . La otra forma, denominada *representación enlazada* de G , es por medio de listas ligadas de vecinos. En esta sección se estudia la primera representación. La representación enlazada se analizará en la sección 9.7.

Suponga que un grafo G tiene m vértices (nodos) y n aristas. Se dice que G es *denso* si $m = O(n^2)$ y *disperso* si $m = O(n)$ o inclusive si $m = O(n \log n)$. Cuando G es denso suele usarse la representación matricial de G , y cuando G es disperso suelen usarse las listas ligadas. Sin importar la forma en que un grafo G se mantiene en la memoria, el grafo G suele introducirse en la computadora por medio de su definición formal; es decir, como una colección de vértices y una colección de aristas (pares de vértices).

Observación: Para evitar casos especiales de los resultados se supone, a menos que se especifique otra cosa, que $m > 1$, donde m es el número de vértices en el grafo G . En consecuencia, G no es conexo si G no tiene aristas.

Diagramas y relaciones, matriz de adyacencia

Sea $G(V, E)$ un grafo dirigido *simple*; es decir, un grafo sin aristas paralelas. Entonces E es simplemente un subconjunto de $V \times V$, y entonces E es una relación sobre V . A la inversa, si R es una relación sobre un conjunto V , entonces $G(V, R)$ es un grafo dirigido simple. Por tanto, los conceptos de relaciones sobre un conjunto y de grafos dirigidos simples son uno y el mismo. De hecho, en el capítulo 2, ya se presentó el grafo dirigido correspondiente a una relación sobre un conjunto.

Suponga que G es un grafo dirigido simple con m vértices y que los vértices de G se han ordenado y que se denominan v_1, v_2, \dots, v_m . Entonces la *matriz de adyacencia* $A = [a_{ij}]$ de G es la matriz de $m \times m$ definida como sigue:

$$a_{ij} = \begin{cases} 1 & \text{si existe una arista } (v_i, v_j) \\ 0 & \text{en otro caso} \end{cases}$$

Esta matriz A , que sólo contienen entradas 0 o 1, se denomina *matriz de bits* o *matriz booleana*. (Aunque la matriz de adyacencia de un grafo no dirigido es simétrica, esto no es cierto aquí para un grafo dirigido.)

La matriz de adyacencia A del grafo G depende del ordenamiento de los vértices de G . Sin embargo, las matrices resultantes de dos ordenamientos distintos están estrechamente relacionadas en el sentido de que una puede obtenerse a partir de la otra al intercambiar simplemente renglones y columnas. A menos que se establezca otra cosa, se supone que los vértices de la matriz tienen un ordenamiento fijo.

Observación: La matriz de adyacencia $A = [a_{ij}]$ puede extenderse a grafos dirigidos con aristas paralelas al hacer:

$$a_{ij} = \text{número de aristas que empiezan en } v_i \text{ y terminan en } v_j$$

Así, las entradas de A son enteros no negativos. A la inversa, toda matriz A $m \times m$ con entradas enteras no negativas define de manera única un grafo dirigido con m vértices.

EJEMPLO 9.6 Sea G el grafo dirigido en la figura 9-4a) con vértices v_1, v_2, v_3, v_4 . Entonces la matriz de adyacencia A de G se muestra en la figura 9-4b). Observe que el número de unos en A es igual al número (ocho) de aristas.

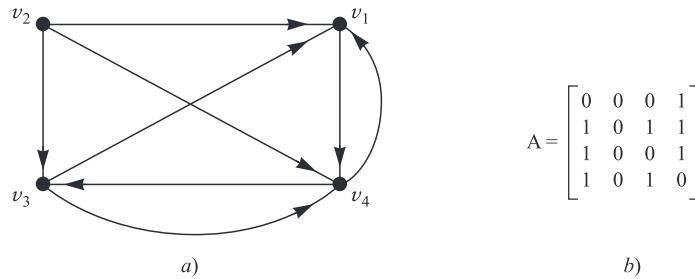


Figura 9-4

Considere las potencias A, A^2, A^3, \dots , de la matriz de adyacencia $A = [a_{ij}]$ de un grafo G . Sea

$$a_K(i, j) = \text{la entrada } ij \text{ en la matriz } A^K$$

Observe que $a_1(i, j) = a_{ij}$ proporciona el número de caminos de longitud 1 del vértice v_i al vértice v_j . Puede demostrarse que $a_2(i, j)$ proporciona el número de caminos de longitud 2 de v_i a v_j . De hecho, en el problema 9.17 se demuestra el siguiente resultado general.

Proposición 9.4: Sea A la matriz de adyacencia de un grafo G . Entonces $a_K(i, j)$, la entrada ij en la matriz A^K , proporciona el número de caminos de longitud K de v_i a v_j .

EJEMPLO 9.7 Considere de nuevo el grafo G y su matriz de adyacencia A que se muestran en la figura 9-4. A continuación se proporcionan las potencias A^2, A^3 y A^4 :

$$A^2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 2 & 0 & 1 & 2 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 2 \end{bmatrix}, \quad A^3 = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 3 & 0 & 2 & 3 \\ 2 & 0 & 1 & 2 \\ 2 & 0 & 2 & 1 \end{bmatrix}, \quad A^4 = \begin{bmatrix} 2 & 0 & 2 & 1 \\ 5 & 0 & 3 & 5 \\ 3 & 0 & 2 & 3 \\ 3 & 0 & 1 & 4 \end{bmatrix}$$

Observe que $a_2(4, 1) = 1$, de modo que hay un camino de longitud 2 de v_4 a v_1 . También, $a_3(2, 3) = 2$, de modo que hay dos caminos de longitud 3 de v_2 a v_3 ; y $a_4(2, 4) = 5$, de modo que hay cinco caminos de longitud 4 de v_2 a v_4 .

Observación: Sea A la matriz de adyacencia de un grafo G , y sea B_r la matriz definida por:

$$B_r = A + A^2 + A^3 + \dots + A^r$$

Entonces la entrada ij de la matriz B_r proporciona el número de caminos de longitud r o menos del vértice v_i al vértice v_j .

Matriz de caminos

Sea $G = G(V, E)$ un grafo dirigido simple con m vértices v_1, v_2, \dots, v_m . La *matriz de caminos* o *matriz de alcanzabilidad* de G es la matriz cuadrada $P = [p_{ij}]$ definida como:

$$p_{ij} = \begin{cases} 1 & \text{si hay un camino de } v_i \text{ a } v_j \\ 0 & \text{en otro caso} \end{cases}$$

(La matriz de caminos puede considerarse como la cerradura transitiva de la relación E sobre V .)

Ahora suponga que en un grafo G con m vértices hay un camino del vértice v_i al vértice v_j . Entonces debe haber un camino simple de v_i a v_j cuando $v_i \neq v_j$, o debe haber un ciclo de v_i a v_j cuando $v_i = v_j$. Puesto que G tiene m vértices, este camino simple debe tener longitud $m - 1$ o menor, o tal ciclo debe tener longitud m o menor. Esto significa que en la matriz B_m (definida antes) hay una entrada ij distinta de cero, donde A es la matriz de adyacencia de G . En consecuencia, la matriz de caminos P y B_m tienen las mismas entradas diferentes de cero. Este resultado se plantea formalmente como sigue.

Proposición 9.5: Sea A la matriz de adyacencia de un grafo G con m vértices. Entonces la matriz de caminos P y B_m tienen las mismas entradas diferentes de cero, donde

$$B_m = A + A^2 + A^3 + \dots + A^m$$

Recuerde que un grafo dirigido G es *fuertemente conexo* si, para cualquier par de vértices u y v en G , hay un camino de u a v y de v a u . Por consiguiente, G es fuertemente conexo si y sólo si la matriz de caminos P de G no tiene entradas cero. Este hecho junto con la proposición 9.5 proporciona el siguiente resultado.

Proposición 9.6: Sea A la matriz de adyacencia de un grafo G con m vértices. Entonces G es fuertemente conexo si y sólo si B_m no tiene entradas cero, donde

$$B_m = A + A^2 + A^3 + \dots + A^m$$

EJEMPLO 9.8 Considere el grafo G y su matriz de adyacencia A , que se muestran en la figura 9-4. Aquí G tiene $m = 4$ vértices. Al sumar la matriz A y las matrices A^2, A^3, A^4 en el ejemplo 9.7 se obtiene la siguiente matriz B_4 y también una matriz de caminos (de alcanzabilidad) P al sustituir las entradas diferentes de cero en B_4 por 1:

$$B_4 = \begin{bmatrix} 4 & 0 & 3 & 4 \\ 11 & 0 & 7 & 11 \\ 7 & 0 & 4 & 7 \\ 7 & 0 & 4 & 7 \end{bmatrix} \quad \text{y} \quad P = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

Al analizar la matriz B_4 o P , se observan entradas iguales a cero; por tanto, G no es fuertemente conexo. En este caso se observa que el vértice v_2 no es alcanzable desde ninguno de los otros vértices.

Observación: La matriz de adyacencia A y la matriz de camino P de un grafo G se consideran matrices lógicas (booleanas) cuando 0 representa “falso” y 1 representa “verdadero”. Así, las operaciones lógicas de \wedge (AND) y \vee (OR) pueden aplicarse a las entradas de A y P donde estas operaciones, usadas en la siguiente sección, se definen en la figura 9-5.

\wedge	0	1
0	0	0
1	0	1

a) AND.

\vee	0	1
0	0	1
1	1	1

b) OR.

Figura 9-5

Cerradura transitiva y la matriz de caminos

Sea R una relación sobre un conjunto finito V con m elementos. Como ya se observó, la relación R puede identificarse con el grafo dirigido simple $G = G(V, R)$. Se observa que la relación composición $R^2 = R \times R$ consta de todos los pares (u, v) tales que hay un camino de longitud 2 de u a v . En forma semejante:

$$R^K = \{(u, v) \mid \text{hay un camino de longitud } K \text{ de } u \text{ a } v\}.$$

La cerradura transitiva R^* de la relación R sobre V ahora puede considerarse como un conjunto de pares ordenados (u, v) tales que hay en el grafo G un camino de u a v . Además, por el análisis anterior, sólo es necesario buscar caminos simples de longitud $m - 1$ o menor y ciclos de longitud m o menor. En consecuencia, se tiene el siguiente resultado, que caracteriza la cerradura transitiva R^* de R .

Teorema 9.7: Sea R una relación sobre un conjunto V con m elementos. Entonces:

- i) $R^* = R \cup R^2 \cup \dots \cup R^m$ es la cerradura transitiva de R .
- ii) La matriz de caminos P de $G(V, R)$ es la matriz de adyacencia de $G'(V, R^*)$.

9.6 ALGORITMO DE WARSHALL, CAMINOS MÁS CORTOS

Sea G un grafo dirigido con m vértices v_1, v_2, \dots, v_m . Suponga que desea encontrar la matriz de caminos P del grafo G . Warshall proporcionó un algoritmo mucho más eficiente que calcular las potencias de la matriz de adyacencia A . Tal algoritmo se define en esta sección, y un algoritmo semejante se utiliza para encontrar los caminos más cortos en G cuando G es ponderado.

Algoritmo de Warshall

Primero se definen las matrices booleanas cuadradas $m \times m$ P_0, P_1, \dots, P_m , donde $P_k[i, j]$ denota la entrada ij de la matriz P_k :

$$P_k[i, j] = \begin{cases} 1 & \text{si hay un camino simple de } v_i \text{ a } v_j \text{ que no use ningún otro vértice,} \\ & \text{excepto quizá } v_1, v_2, \dots, v_k. \\ 0 & \text{en otro caso.} \end{cases}$$

Por ejemplo,

$$P_k[i, j] = 1 \quad \text{si hay un camino simple de } v_i \text{ a } v_j \text{ que no use ningún otro vértice,} \\ \text{excepto quizá } v_1, v_2, v_3.$$

Observe que la primera matriz $P_0 = A$ es la matriz de adyacencia de G . Además, puesto que G sólo tiene m vértices, la última matriz $P_m = P$ es la matriz de caminos de G .

Warshall observó que $P_k[i, j] = 1$ puede pasar sólo si ocurre uno de los dos casos siguientes:

- 1) Hay un camino simple de v_i a v_j que no usa ningún otro vértice, excepto quizá v_1, v_2, \dots, v_{k-1} ; por tanto,

$$P_{k-1}[i, j] = 1$$

- 2) Hay un camino simple de v_i a v_k y un camino simple de v_k a v_j donde cada camino simple no usa ningún otro vértice, excepto quizá v_1, v_2, \dots, v_{k-1} ; por tanto,

$$P_{k-1}[i, k] = 1 \quad \text{y} \quad P_{k-1}[k, j] = 1$$

Estos dos casos se representan como sigue:

$$1) v_i \rightarrow \dots \rightarrow v_j; \quad 2) v_i \rightarrow \dots \rightarrow v_k \rightarrow \dots \rightarrow v_j$$

donde $\rightarrow \dots \rightarrow$ denota parte de un camino simple que no usa ningún otro vértice, excepto quizá v_1, v_2, \dots, v_{k-1} . En consecuencia, los elementos de P_k pueden obtenerse como sigue:

$$P_k[i, j] = P_{k-1}[i, j] \vee (P_{k-1}[i, k] \wedge P_{k-1}[k, j])$$

donde se usan las operaciones lógicas de \wedge (AND) y \vee (OR). En otras palabras, cada entrada en la matriz P_k puede obtenerse buscando sólo tres entradas en la matriz P_{k-1} . El algoritmo de Warshall se muestra en la figura 9-6.

Algoritmo 9.1 (de Warshall): Un grafo dirigido G con M vértices se mantiene en la memoria por medio de su matriz de adyacencia A . Este algoritmo encuentra la matriz de caminos (booleana) P del grafo de G .

Paso 1. Repetir para $I, J = 1, 2, \dots, M$; [Inicializa P].
 Si $A[I, J] = 0$, entonces: Establecer $P[I, J] := 0$;
 O bien: Establecer $P[I, J] := 1$.
 [Fin del ciclo].

Paso 2. Repetir los pasos 3 y 4 para $K = 1, 2, \dots, M$: [Actualiza P].

Paso 3. Repetir el paso 4 para $I = 1, 2, \dots, M$:

Paso 4. Repetir para $J = 1, 2, \dots, M$:
 Establecer $P[I, J] := P[I, J] \vee P[I, K] \wedge P[K, J]$.
 [Fin del ciclo].
 [Fin del ciclo del paso 3].
 [Fin del ciclo del paso 2].

Paso 5. Salir.

Figura 9-6

Algoritmo del camino más corto

Sea G un grafo dirigido simple con m vértices, v_1, v_2, \dots, v_m . Suponga que G es ponderado; es decir, suponga que a cada arista e de G se asigna un número no negativo $w(e)$ denominado *peso* o *longitud* de e . Entonces G puede mantenerse en la memoria por medio de su matriz de pesos $W = [w_{ij}]$ definida como sigue:

$$w_{ij} = \begin{cases} w(e) & \text{si hay una arista } e \text{ de } v_i \text{ a } v_j \\ 0 & \text{si no hay una arista de } v_i \text{ a } v_j \end{cases}$$

La matriz de caminos P indica si entre los vértices hay o no caminos. Ahora se desea encontrar una matriz Q que indique las longitudes de los caminos más cortos entre los vértices o, más exactamente, una matriz $Q = [q_{ij}]$ donde

$$[q_{ij}] = \text{longitud del camino más corto de } v_i \text{ a } v_j$$

A continuación se describe una modificación del algoritmo de Warshall que encuentra de manera eficiente la matriz Q .

Aquí se define una secuencia de matrices Q_0, Q_1, \dots, Q_m (semejante a las matrices anteriores P_0, P_1, \dots, P_m) donde $Q_k[i, j]$, la entrada ij de Q_k , se define como sigue:

$Q_k[i, j] =$ la menor longitud del camino precedente de v_i a v_j o la suma de las longitudes de los caminos precedentes de v_i a v_k y de v_k a v_j .

Más exactamente,

$$Q_k[i, j] = \text{MIN}(Q_{k-1}[i, j], Q_{k-1}[i, k] + Q_{k-1}[k, j])$$

La matriz inicial Q_0 es la misma que la matriz de pesos W , excepto que cada 0 en w se sustituye por ∞ (o por un número muy, muy grande). La matriz final Q_m es la matriz buscada Q .

EJEMPLO 9.9 En la figura 9-7 se muestran un grafo ponderado G y su matriz de pesos W , donde se supone que $v_1 = R, v_2 = S, v_3 = T, v_4 = U$.

Suponga que el modelo modificado del algoritmo de Warshall se aplica al grafo ponderado G en la figura 9-7. Se obtienen las matrices Q_0, Q_1, Q_3 y Q_4 en la figura 9-8. (A la derecha de cada matriz Q_k en la figura 9-8 se muestra la matriz de caminos que

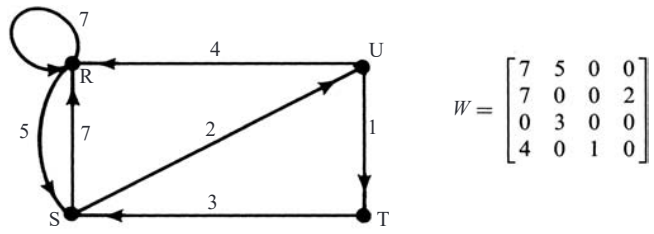


Figura 9-7

corresponde a las longitudes en la matriz Q_k .) Las entradas en la matriz Q_0 son las mismas que en la matriz de pesos W , excepto que cada 0 en W se sustituye por ∞ (un número muy, muy grande). A continuación se indica cómo se obtuvieron las entradas encerradas en un círculo:

$$\begin{aligned}
 Q_1[4, 2] &= \text{MÍN} (Q_0[4, 2], \quad Q_0[4, 1] + Q_0[1, 2]) = \text{MÍN}(\infty, 4 + 5) = 9 \\
 Q_2[1, 3] &= \text{MÍN} (Q_1[1, 3], \quad Q_1[1, 2] + Q_1[2, 3]) = \text{MÍN}(\infty, 5 + \infty) = \infty \\
 Q_3[4, 2] &= \text{MÍN} (Q_2[4, 2], \quad Q_2[4, 3] + Q_2[3, 2]) = \text{MÍN}(9, 3 + 1) = 4 \\
 Q_4[3, 1] &= \text{MÍN} (Q_3[3, 1], \quad Q_3[3, 4] + Q_3[4, 1]) = \text{MÍN}(10, 5 + 4) = 9
 \end{aligned}$$

La última matriz $Q_4 = Q$, la matriz buscada con el camino más corto.

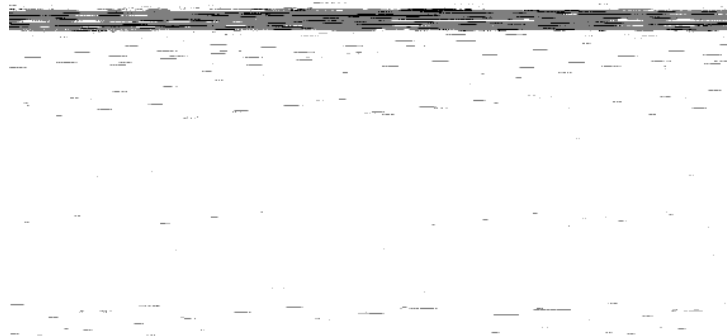


Figura 9-8

9.7 REPRESENTACIÓN LIGADA DE GRAFOS DIRIGIDOS

Sea G un grafo dirigido con m vértices. Suponga que el número de aristas de G es $O(m)$, o incluso $O(m \log m)$; es decir, suponga que G es disperso. Entonces la matriz de adyacencia A de G contiene muchos ceros; por tanto, se desperdicia

bastante espacio de memoria. En consecuencia, cuando G es disperso, G suele representarse en la memoria por medio de algún tipo de *representación enlazada*, también denominada *estructura de adyacencia*, que se describe a continuación con un ejemplo.

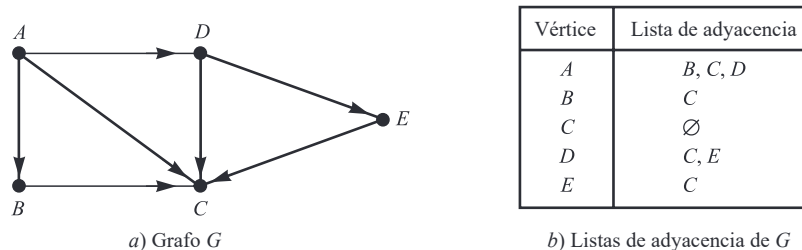


Figura 9-9

Considere el grafo dirigido G en la figura 9-9a). Observe que G puede definirse en forma equivalente por la tabla en la figura 9-9b), donde se muestra cada vértice en G seguido por su *lista de adyacencia*, también denominada de sus *sucesores* o *vecinos*. Aquí el símbolo \emptyset denota una lista vacía. Observe que cada arista de G corresponde a un vértice único en una lista de adyacencia y viceversa. Aquí, G tiene siete aristas y en las listas de adyacencia hay siete vértices. Esta tabla también puede presentarse en la siguiente forma abreviada, donde dos puntos “:” separan un vértice de su lista de vecinos y un punto y coma “;” separa las diversas listas:

$$G = [A : B, C, D; B : C; C : \emptyset; D : C, E; E : C]$$

La *representación ligada* de un grafo dirigido G mantiene a G en la memoria mediante el uso de listas ligadas para sus listas de adyacencia. Con más brevedad, la representación enlazada normalmente contiene dos archivos (conjuntos de registros), uno denominado Vertex File y el otro Edge File, como sigue.

- a) **Vertex File:** Este archivo contiene la lista de vértices del grafo G usualmente mantenida por medio de un arreglo o por una lista ligada. Cada registro del archivo tiene la forma

VERTEX	NEXT-V	PTR	
--------	--------	-----	--

Aquí VERTEX es el nombre del vértice, NEXT-V apunta al siguiente vértice en la lista de vértices en el Vertex File, y PTR apunta al primer elemento en la lista de adyacencia del vértice que aparece en el Edge File. El área sombreada indica que en el registro correspondiente al vértice puede haber otra información.

- b) **Edge File:** Este archivo contiene las aristas de G y todas las listas de adyacencia de G , donde cada lista se mantiene en la memoria por medio de una lista ligada. Cada registro del Edge File representa una arista única en G y, por tanto, corresponde a un vértice único en una lista de adyacencia. Normalmente, el registro tiene la forma

EDGE	BEG-V	END-V	NEXT-E	
------	-------	-------	--------	--

Aquí:

- 1) EDGE es el nombre de la arista (en caso de tener una).
- 2) BEG-V apunta a la ubicación del Vertex File del vértice inicial de la arista.
- 3) END-V apunta a la ubicación del Vertex File del vértice terminal de la arista. Las listas de adyacencia aparecen en este campo.
- 4) NEXT-E apunta a la ubicación en el Edge File del siguiente vértice en la lista de adyacencia.

Recuerde que las listas de adyacencia constan de vértices terminales, por lo que se mantienen mediante el campo END-V. El área sombreada indica que en el registro correspondiente a la arista puede haber otra información. Se observa que el orden de los vértices en una lista de adyacencia no depende del orden en que las aristas (pares de vértices) aparecen en los datos de entrada.

En la figura 9-10 se muestra la forma en que el grafo G en la figura 9-9a) puede aparecer en la memoria. Aquí los vértices de G se mantienen en la memoria por medio de una lista ligada usando la variable START para apuntar al primer vértice. (En forma alterna, podría usarse un arreglo lineal para la lista de vértices, y así no sería necesario NEXT-V.) La elección de ocho ubicaciones para el Vertex File y 10 localizaciones para el Edge File es arbitraria. El espacio adicional en los archivos se usa en caso de que en el grafo se inserten vértices o aristas adicionales. En la figura 9-10 también se muestra, con flechas, la lista de adyacencia $[B, C, D]$ del vértice A .

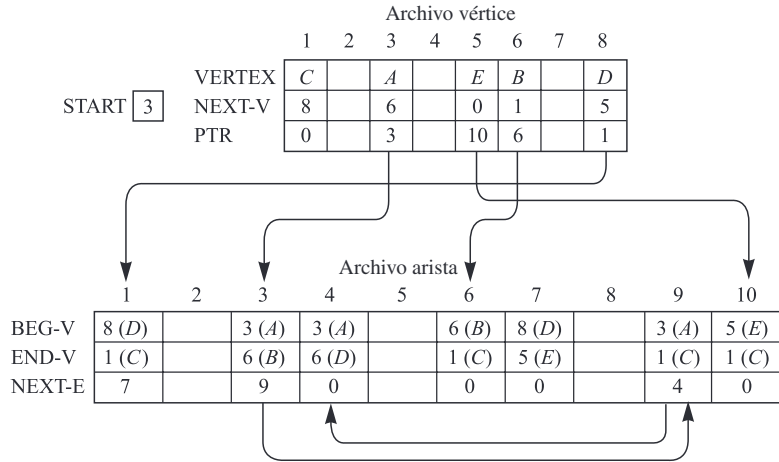


Figura 9-10

9.8 ALGORITMOS DE GRAFOS: BÚSQUEDAS EN PROFUNDIDAD Y EN ANCHURA

En esta sección se analizan dos importantes algoritmos de grafos para un grafo dado G . Cualquier algoritmo de grafos particular puede depender de la forma en que G se mantiene en la memoria. Aquí se supone que G se mantiene en la memoria por medio de su estructura de adyacencia. El grafo de prueba G con su estructura de adyacencia se muestran en la figura 9-11.

Muchas aplicaciones de grafos requieren el examen sistemático de los vértices y las aristas de un grafo G . Hay dos formas normales para hacer lo anterior. Una forma se denomina *búsqueda en profundidad* (DFS: depth-first search) y la otra, *búsqueda en anchura* (BFS: breadth-first search). (Estos algoritmos son esencialmente idénticos a los correspondientes para grafos no dirigidos del capítulo 8.)

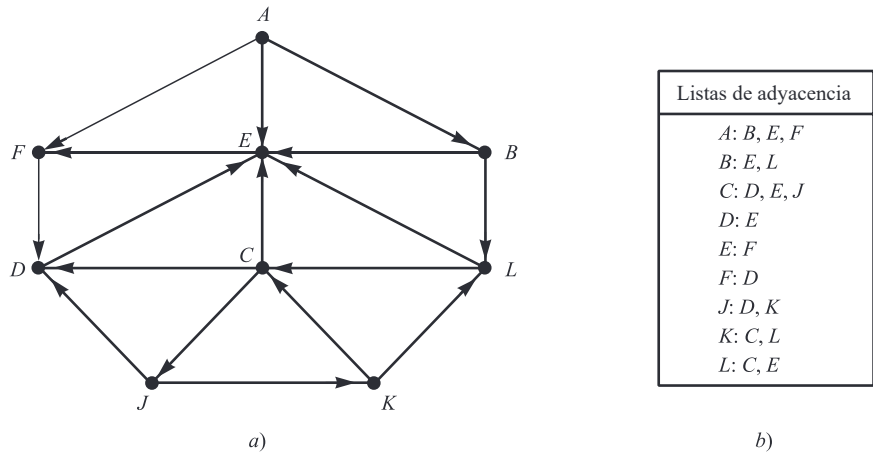


Figura 9-11

Durante la ejecución de los algoritmos, cada vértice (nodo) N de G se encuentra en uno de tres estados, denominados *status* de N , como sigue:

STATUS = 1: (Estado Ready) El estado inicial del vértice N .

STATUS = 2: (Estado Waiting) El vértice N está en una lista (de espera), en espera de ser procesado.

STATUS = 3: (Estado Processed) El vértice N ha sido procesado.

La lista de espera para la búsqueda en profundidad es una STACK —modificada— (que se escribe horizontalmente con la parte superior de STACK a la izquierda), mientras la lista de espera para la búsqueda en anchura es una QUEUE.

- a) **Búsqueda en profundidad:** La idea general detrás de una búsqueda en profundidad que empieza en un vértice inicial A es: primero se procesa el vértice inicial A . Luego se procesa cada vértice N a lo largo de un camino P que empiece en A ; es decir, se procesa un vecino de A , luego un vecino de un vecino de A y así en lo sucesivo. Después de llegar a un “punto muerto”; es decir, a un vértice sin vecino no procesado, se retrocede sobre el camino P hasta que es posible continuar a lo largo de otro camino P' . Y se continúa del mismo modo. El retroceso se logra usando una STACK para mantener los vértices iniciales de futuros caminos posibles. También se requiere un campo STATUS que indica el estado actual de cualquier vértice, de modo que ningún vértice sea procesado más de una vez. El algoritmo se muestra en la figura 9-12.

Algoritmo 9.2 (De búsqueda en profundidad): Este algoritmo ejecuta una búsqueda en profundidad sobre un grafo dirigido G , empezando en un vértice inicial A .

Paso 1. Todos los vértices se inicializan en el estado ready (STATUS = 1).

Paso 2. El vértice inicial A se introduce en STACK y el status de A cambia al estado waiting (STATUS = 2).

Paso 3. Repetir los pasos 4 y 5 hasta que STACK esté vacía.

Paso 4. El vértice superior N se saca de STACK. Se procesa N , y se establece STATUS (N) = 3, el estado processed.

Paso 5. Examinar cada vecino J de N .

a) Si STATUS (J) = 1 (estado ready), J se coloca sobre STACK y se restablece STATUS (J) = 2 (estado waiting).

b) Si STATUS (J) = 2 (estado waiting), el J previo se elimina de STACK y el J actual se coloca sobre STACK.

c) Si STATUS (J) = 3 (estado processed), se ignora el vértice J .

[Fin del ciclo del paso 3.]

Paso 6. Salir.

Figura 9-12

El algoritmo 9.2 procesa sólo aquellos vértices que son alcanzables desde un vértice inicial A . Suponga que se desea procesar todos los vértices en el grafo G . Así, el algoritmo debe modificarse de modo que vuelva a empezar con otro vértice que aún se encuentre en el estado ready (STATE = 1). Este nuevo vértice, por ejemplo B , puede obtenerse al recorrer la lista de vértices.

Observación: Técnicamente, la estructura STACK en el algoritmo 9.2 no es una pila ya que, en el paso 5b), se permite la eliminación de un vértice J y luego su inserción en el frente de la pila. (Aunque se trata del mismo vértice J , representa una arista distinta.) Si el J previo no se elimina en el paso 5b), entonces se obtiene un algoritmo de recorrido alterno.

EJEMPLO 9.10 Considere el grafo de prueba G en la figura 9-11. Suponga que desea encontrar e imprimir todos los vértices alcanzables desde el vértice J (incluso a J). Una forma de hacerlo es aplicar un algoritmo en profundidad de G empezando en el vértice J .

Al aplicar el algoritmo 9.2, los vértices se procesan e imprimen en el orden siguiente:

$$J, K, L, E, F, D, C$$

En la figura 9-13a) se muestra la secuencia de las listas de espera en STACK y los vértices que están en proceso. (La línea diagonal / indica que un vértice se elimina de la lista de espera.) Recuerde que cada vértice, excepto J , proviene de una lista de adyacencia, y por tanto, es el vértice terminal de una arista única del grafo. La arista se ha indicado al etiquetar el vértice terminal con el vértice inicial de la arista como un subíndice. Por ejemplo,

$$D_j$$

significa que D está en la lista de adyacencia de J , y entonces que D es el vértice terminal de una arista que empieza en J . Estas aristas constituyen un árbol T con raíz cuya raíz es J , lo cual se muestra en la figura 9-13b). (Los números indican el orden en que las aristas se agregan al árbol.) Este árbol T genera el subgrafo G' de G que consta de los vértices alcanzables desde J .

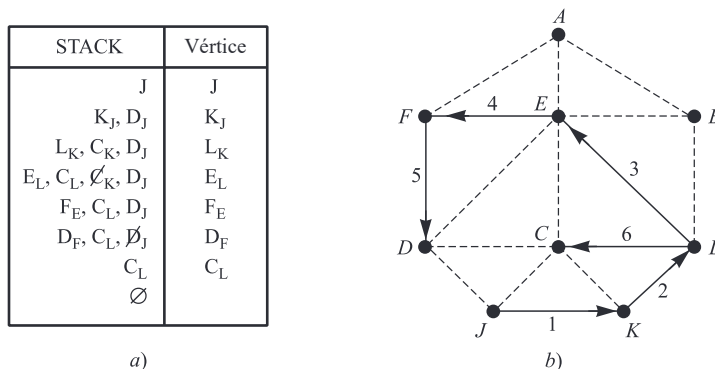


Figura 9-13

b) **Búsqueda en anchura:** La idea general detrás de una búsqueda en anchura que empieza en un vértice inicial A es: primero se procesa el vértice inicial A . Luego se procesan todos los vecinos de A y enseguida se procesan todos los vecinos de los vecinos de A . Y así se continúa. Resulta evidente que es necesario mantener la pista de los vecinos de un vértice, así como garantizar que ningún vértice sea procesado dos veces. Esto se logra usando una QUEUE para mantener los vértices que están en espera de ser procesados; y mediante un campo STATUS que indica el estado actual de un vértice. El algoritmo se muestra en la figura 9-14.

El algoritmo 9.3 sólo procesa aquellos vértices que son alcanzables desde un vértice inicial A . Suponga que se desea procesar todos los vértices en un grafo G . Entonces es necesario modificar el algoritmo de modo que nuevamente empiece con otro vértice que aún se encuentre en el estado ready (STATE = 1). Este nuevo vértice B , por ejemplo, puede obtenerse al recorrer la lista de vértices.

EJEMPLO 9.11 Considere el grafo de prueba G en la figura 9-11. Suponga que G representa los vuelos diarios entre ciudades y que desea volar de la ciudad A a la ciudad J con el menor número de escalas. Es decir, se quiere encontrar un camino más corto P de A a J (donde cada arista tiene un peso de 1). Una forma de hacer lo anterior es usar una búsqueda en anchura de G empezando en el vértice A , y detenerse tan pronto como se encuentre J .

En la figura 9-15a) se muestra la secuencia de las listas de espera en QUEUE y los vértices que se están procesando hasta el momento en que se encuentra el vértice J . Luego se trabaja hacia atrás a partir de J para obtener el siguiente camino deseado que se muestra en la figura 9-15b):

$$J_C \leftarrow C_L \leftarrow L_B \leftarrow B_A \leftarrow A \quad \text{o} \quad A \rightarrow B \rightarrow L \rightarrow C \rightarrow J$$

Así, un vuelo de la ciudad A a la ciudad J hará tres escalas intermedias en B, L y C . Observe que el camino no incluye todos los vértices procesados por el algoritmo.

Algoritmo 9.3 (de búsqueda en anchura): Este algoritmo ejecuta una búsqueda en anchura sobre un grafo dirigido G , empezando en un vértice inicial A .

Paso 1. Todos los vértices se inicializan en el estado ready (STATUS = 1).

Paso 2. El vértice inicial A se introduce en QUEUE y el status de A se cambia al estado waiting (STATUS = 2).

Paso 3. Repetir los pasos 4 y 5 hasta que QUEUE esté vacía.

Paso 4. Sacar el primer vértice N de QUEUE. Se procesa N , y se establece STATUS (N) = 3, el estado processed.

Paso 5. Se examina cada vecino J de N .

a) Si STATUS (J) = 1 (estado ready), J se coloca en la parte trasera de QUEUE y se restablece STATUS (J) = 2 (estado waiting).

b) Si STATUS (J) = 2 (estado waiting) o STATUS(J) = 3 (estado processed), se ignora el vértice J .

[Fin del ciclo del paso 3.]

Paso 6. Salir.

Figura 9-14

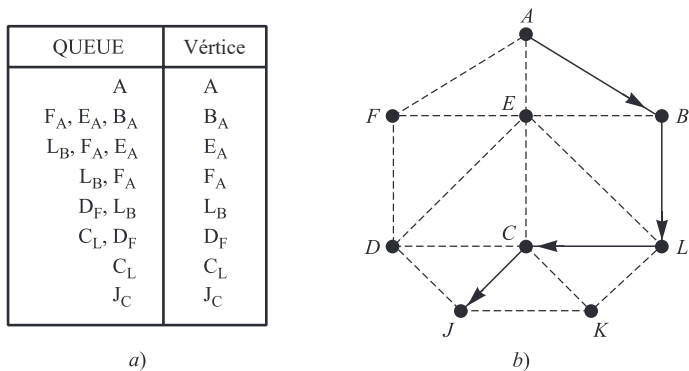


Figura 9-15

9.9 GRAFOS DIRIGIDOS LIBRES DE CICLOS, ORDENACIÓN TOPOLOGICA

Sea S un grafo dirigido con las dos propiedades siguientes:

- 1) Cada vértice v_i de S representa una tarea.
- 2) Cada arista (dirigida) (u, v) de S significa que la tarea u debe completarse antes de empezar la tarea v .

Se observa que un grafo S así no puede contener ningún ciclo, como $P = (u, v, w, u)$, puesto que, en caso contrario, sería necesario completar u antes de empezar v , completar v antes de empezar w y completar w antes de empezar u . Es decir, no es posible comenzar ninguna de las tres tareas del ciclo.

Se dice que un grafo S así, que representa tareas y una relación prerrequisito y que no puede tener ningún ciclo, es *libre de ciclos* o *acíclico*. La forma abreviada de denominar a un grafo acíclico dirigido (libre de ciclos) es *gad*. En la figura 9-16 se muestra un ejemplo de un grafo así.

Una operación fundamental sobre un *gad* S consiste en procesar los vértices uno después de otro de modo que el vértice u siempre sea procesado antes que el vértice v siempre que (u, v) sea una arista. Un ordenamiento lineal T así de los vértices de S , que puede no ser único, se denomina *ordenamiento topológico*.

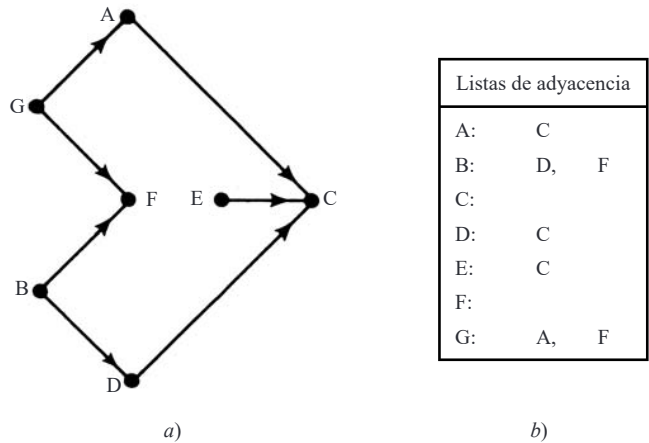


Figura 9-16

En la figura 9-17 se muestran dos ordenamientos topológicos del grafo S en la figura 9-16. En la figura 9-17 se han incluido las aristas de S para mostrar que coinciden con la dirección del ordenamiento lineal.

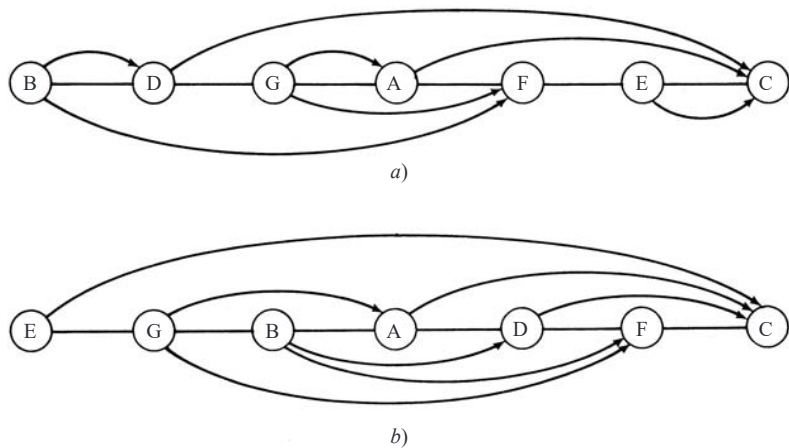


Figura 9-17 Dos ordenamientos topológicos

A continuación se presenta el resultado teórico más importante de esta sección.

Teorema 9.8: Sea S un grafo libre de ciclos dirigido finito. Entonces existe un ordenamiento topológico T del grafo S .

Observe que el teorema sólo establece que existe un ordenamiento topológico. A continuación se proporciona un algoritmo que encuentra un ordenamiento topológico. La idea más importante del algoritmo es que cualquier vértice (nodo) N con grado de entrada cero puede escogerse como el primer elemento en el ordenamiento T . En esencia el algoritmo repite los dos pasos siguientes hasta que S está vacía:

- 1) Encontrar un vértice N con grado de entrada cero.
- 2) Eliminar N y sus aristas del grafo S .

Se usa una QUEUE auxiliar para mantener temporalmente todos los vértices con grado cero. El algoritmo se muestra en la figura 9-18.

Algoritmo 9.4: El algoritmo encuentra un ordenamiento topológico T de un grafo libre de ciclos dirigido S .

Paso 1. Encontrar el grado de entrada $\text{INDEG}(N)$ de cada vértice N de S .

Paso 2. Insertar en QUEUE todos los vértices con grado cero.

Paso 3. Repetir los pasos 4 y 5 hasta que QUEUE esté vacía.

Paso 4. Eliminar y procesar el vértice frontal N de QUEUE.

Paso 5. Repetir para cada vecino M del vértice N .

a) Establecer $\text{INDEG}(M) := \text{INDEG}(M) - 1$.
[Así se elimina la arista de N a M .]

b) Si $\text{INDEG}(M) = 0$, agregar M a QUEUE.
[Fin del ciclo.]

[Fin del ciclo del paso 3.]

Paso 6. Salir.

Figura 9-18

EJEMPLO 9.12 Suponga que el algoritmo 9.4 se aplica al grafo S en la figura 9-16. Se obtiene la siguiente secuencia de los elementos de QUEUE y la secuencia de los vértices que están en proceso:

QUEUE	GEB	DGE	DG	FAD	FA	CF	C	\emptyset
Vértice	B	E	G	D	A	F	C	

Así, los vértices se procesan en el orden: B, E, G, D, A, F .

9.10 ALGORITMO DE PODA PARA EL CAMINO MÁS CORTO

Si G es un grafo libre de ciclos dirigido ponderado, se busca el camino más corto entre dos vértices, por ejemplo, u y w . Se supone que G es finito, de modo que en cada paso hay un número finito de movimientos. Puesto que G es libre de ciclos, todos los caminos entre u y w se proporcionan mediante un árbol con raíz cuya raíz sea u . En la figura 9-19b) se enumeran todos los caminos entre u y w en el grafo en la figura 9-19a).

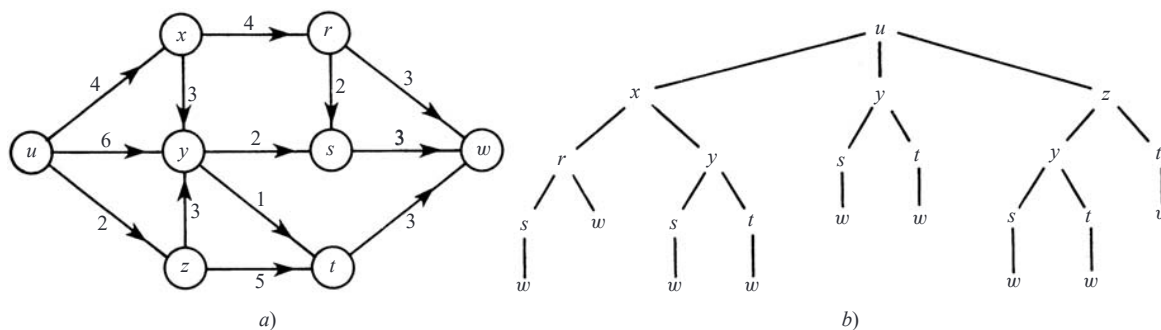


Figura 9-19

Una forma de encontrar el camino más corto entre u y w es calcular las longitudes de todos los caminos que corresponden a las raíces del árbol. Por otra parte, suponga que dos caminos parciales conducen a un vértice intermedio v . A partir de entonces, sólo es necesario considerar el camino parcial más corto; es decir, el árbol se poda en el vértice correspondiente al camino parcial más largo. El algoritmo de poda se describe a continuación.

Algoritmo de poda

Este algoritmo encuentra el camino más corto entre un vértice u y un vértice w en un grafo G dirigido libre de ciclos. El algoritmo posee las siguientes propiedades:

- a) Durante el algoritmo a cada vértice v' de G se le asignan:
 - 1) Un número $\ell(v')$ que denota la longitud mínima actual de un camino de u a v' .
 - 2) Un camino $p(v')$ de u a v' de longitud $\ell(v')$.
- b) Al inicio se hace $\ell(u) = 0$ y $p(u) = u$. A cualquier otro vértice v al inicio se le asigna $\ell(v) = \infty$ y $p(v) = \emptyset$.
- c) En cada paso el algoritmo examina una arista $e = (v', v)$ de v' a v con, por ejemplo, longitud k . Se calcula $\ell(v') + k$.
 - 1) Suponga que $\ell(v') + k < \ell(v)$. Entonces se ha encontrado un camino más corto de u a v . Así, se actualiza:

$$\ell(v) = \ell(v') + k \quad \text{y} \quad p(v) = p(v')v$$
 (Esto siempre es cierto cuando $\ell(v) = \infty$; es decir, cuando el vértice v se introduce por primera vez.)
 - 2) En caso contrario, no se modifican $\ell(v)$ ni $p(v)$.
Si ninguna otra arista no examinada entra en v , se dice que se ha determinado $p(v)$.
- d) El algoritmo termina cuando se ha determinado $p(w)$.

Observación: La arista $e = (v', v)$ en el inciso c) sólo puede escogerse si v' ha sido visitado previamente; es decir, si $p(v')$ no está vacío. Además, suele ser mejor examinar una arista que empieza en un vértice v' cuyo camino $p(v')$ ha sido determinado.

EJEMPLO 9.13 El algoritmo de poda se aplica al grafo G en la figura 9-19a).

Desde u : los vértices sucesivos son x, y y z , que se introducen por primera vez. Así:

- 1) se hace $\ell(x) = 4$, $p(x) = ux$.
- 2) se hace $\ell(y) = 6$, $p(y) = uy$.
- 3) se hace $\ell(z) = 2$, $p(z) = uz$.

Observe que se han determinado $p(x)$ y $p(z)$.

Desde x : los vértices sucesivos son r , introducido por primera vez, y y . Así:

- 1) Se hace $\ell(r) = 4 + 4 = 8$ y $p(r) = p(x)r = uxr$.
- 2) Se calcula:

$$\ell(x) + k = 4 + 3 = 7 \quad \text{que no es menor que} \quad \ell(y) = 6.$$

Por tanto, $\ell(y)$ y $p(y)$ se dejan solos.

Observe que se ha determinado $p(r)$.

Desde z : los vértices sucesivos son t , introducido por primera vez, y y . Así:

- 1) Se hace $\ell(t) = \ell(z) + k = 2 + 5 = 7$ y $p(t) = p(z)t = urt$.
- 2) Se calcula:

$$\ell(z) + k = 2 + 3 = 5 \quad \text{que es menor que} \quad \ell(y) = 6.$$

Se ha encontrado un camino más corto hacia y , de modo que se actualizan $\ell(y)$ y $p(y)$; se hace

$$\ell(y) = \ell(z) + k = 5 \quad \text{y} \quad p(y) = p(z)y = uzy$$

Ahora se ha determinado $p(y)$.

Desde y: los vértices sucesivos son s , introducido por primera vez, y t . Así:

- 1) Se hace $\ell(s) = \ell(y) + k = 5 + 2 = 7$ y $p(s) = p(y)s = uzys$.
- 2) Se calcula:

$$\ell(y) + k = 5 + 1 = 6 \quad \text{que es menor que} \quad \ell(t) = 7.$$

Por tanto, se cambian $\ell(t)$ y $p(t)$ para leer:

$$\ell(t) = \ell(y) + 1 = 6 \quad \text{y} \quad p(t) = p(y)t = uzyt.$$

Ahora se ha determinado $p(t)$.

Desde r: los vértices sucesivos son w , introducido por primera vez, y s . Así:

- 1) Sea $\ell(w) = \ell(r) + 3 = 11$ y $p(w) = p(r)w = uxrw$.
- 2) Se calcula:

$$\ell(r) + k = 8 + 2 = 10 \quad \text{que es menor que} \quad \ell(s) = 7.$$

Por tanto, $\ell(s)$ y $p(s)$ se dejan solos.

Observe que se ha determinado $p(s)$.

Desde s: el vértice sucesivo es w . Se calcula:

$$\ell(s) + k = 7 + 3 = 10 \quad \text{que es menor que} \quad \ell(w) = 11.$$

Por tanto, se cambian $\ell(w)$ y $p(w)$ para leer:

$$\ell(w) = \ell(s) + 3 = 10 \quad \text{y} \quad p(w) = p(s)w = uzysw.$$

Desde t: el vértice sucesivo es w . Se calcula:

$$\ell(t) + k = 6 + 3 = 9 \quad \text{que es menor que} \quad \ell(w) = 10.$$

Por tanto, se actualizan $\ell(w)$ y $p(w)$ como sigue:

$$\ell(w) = \ell(t) + 3 = 9 \quad \text{y} \quad p(w) = p(t)w = uzytw$$

Ahora se ha determinado $p(w)$.

El algoritmo ha terminado puesto que se ha determinado $p(w)$. Por tanto, $p(w) = uzytw$ es el camino más corto de u a w y $\ell(w) = 9$.

Las aristas que se analizaron en el ejemplo precedente constituyen el árbol con raíz en la figura 9-20. Es el árbol de la figura 9-19b) que ha sido podado en los vértices que pertenecen a caminos parciales más largos. Observe que sólo fue necesario examinar 13 de las 23 aristas originales del árbol.

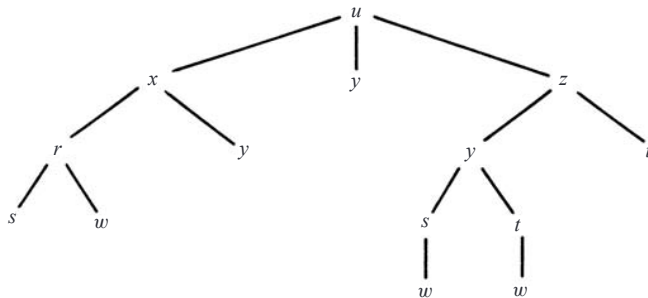


Figura 9-20

PROBLEMAS RESUELTOS

TERMINOLOGÍA DE GRAFOS

9.1 Sea G el grafo dirigido en la figura 9-21a).

- a) Describa formalmente a G .
 - b) Encuentre todos los caminos simples de X a Z .
 - c) Encuentre todos los caminos simples de Y a Z .
 - d) Encuentre todos los ciclos en G .
 - e) ¿ G es unilateralmente conexo?
 - f) ¿ G es fuertemente conexo?
- a) El conjunto de vértices V tiene cuatro vértices y el conjunto de aristas E tiene siete aristas (dirigidas) como sigue:

$$V = \{X, Y, Z, W\} \text{ y } E = \{(X, Y), (X, Z), (X, W), (Y, W), (Z, Y), (Z, W), (W, Z)\}$$

- b) Hay tres caminos simples de X a Z , que son (X, Z) , (X, W, Z) y (X, Y, W, Z) .
- c) De Y a Z sólo hay un camino simple, que es (Y, W, Z) .
- d) En G sólo hay un ciclo, que es (Y, W, Z, Y) .
- e) G es unilateralmente conexo, ya que (X, Y, W, Z) es un camino de expansión.
- f) G no es fuertemente conexo porque no hay ningún camino de expansión.

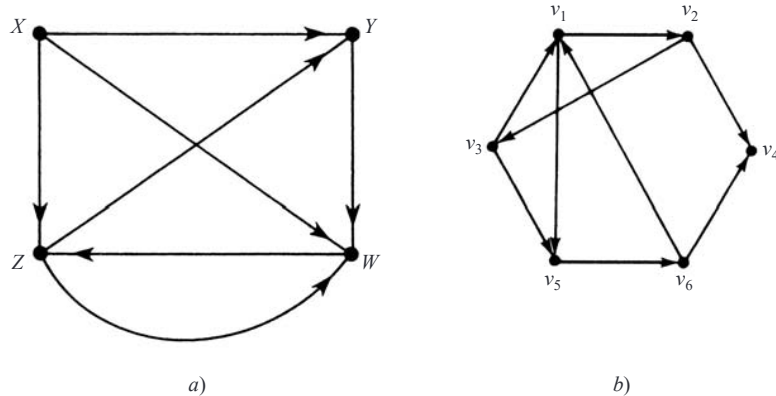


Figura 9-21

9.2 Sea G el grafo dirigido en la figura 9-21a).

- a) Encuentre el grado de entrada y el grado de salida de cada vértice de G .
 - b) Encuentre la lista de sucesores de cada vértice de G .
 - c) ¿Hay alguna fuente o algún sumidero?
 - d) Encuentre el subgrafo H de G determinado por el conjunto de vértices $V' = X, Y, Z$.
- a) Se cuenta el número de aristas que empiezan y terminan en un vértice v para obtener, respectivamente, $\text{indeg}(v)$ y $\text{outdeg}(v)$. Lo anterior produce los datos siguientes:

$$\begin{aligned} \text{indeg}(X) = 0, & \quad \text{indeg}(Y) = 2, & \quad \text{indeg}(Z) = 2, & \quad \text{indeg}(W) = 3, \\ \text{outdeg}(X) = 3, & \quad \text{outdeg}(Y) = 1, & \quad \text{outdeg}(Z) = 2, & \quad \text{outdeg}(W) = 1, \end{aligned}$$

(Como era de esperar, la suma de los grados de entrada y la suma de los grados de salida es igual —cada una— a 7, el número de aristas).

- b) El vértice v se agrega a la lista de sucesores (u) para cada arista (u, v) en G . Así se obtiene:

$$\text{suc}(X) = [Y, Z, W], \quad \text{suc}(Y) = [W], \quad \text{suc}(Z) = [Y, W], \quad \text{suc}(W) = [Z]$$

- c) X es una fuente a la que no entra ninguna arista; es decir, $\text{indeg}(X) = 0$. No hay sumideros, ya que cada vértice es el punto inicial de una arista; es decir, tiene grado de salida distinto de cero.
- d) Sea E' que consta de todas las aristas de G cuyos puntos terminales están en V' . Así se obtiene $E' = \{(X, Y), (X, Z), (Z, Y)\}$. Entonces $H = H(V', E')$.

9.3 Sea G el grafo dirigido en la figura 9-21b).

- a) Encuentre dos caminos simples de v_1 a v_6 . ¿Es $\alpha = (v_1, v_2, v_4, v_6)$ un camino simple?
- b) Encuentre todos los ciclos en G que incluyen a v_3 .
- c) ¿ G es unilateralmente conexo? ¿Fuertemente conexo?
- d) Encuentre la lista de sucesores de cada vértice de G .
- e) ¿Hay alguna fuente en G ? ¿Algún sumidero?
- a) En un camino simple todos los vértices son distintos. Así, (v_1, v_5, v_6) y $(v_1, v_2, v_3, v_5, v_6)$ son dos caminos simples de v_1 a v_6 . La secuencia ni siquiera es un camino puesto que la arista que une v_4 a v_6 no empieza en v_4 .
- b) Hay dos ciclos así: (v_3, v_1, v_2, v_3) y $(v_3, v_5, v_6, v_1, v_2, v_3)$.
- c) G es unilateralmente conexo puesto que $(v_1, v_2, v_3, v_5, v_6, v_4)$ es un camino de expansión. G no es fuertemente conexo, porque no hay ningún camino de expansión cerrado.
- d) El vértice v se agrega a la lista de sucesores $\text{suc}(u)$ para cada arista (u, v) en G . Así se obtiene:

$$\begin{aligned} \text{suc}(v_1) &= [v_2, v_5], & \text{suc}(v_2) &= [v_3, v_4], & \text{suc}(v_3) &= [v_1, v_5] \\ \text{suc}(v_4) &= \emptyset, & \text{suc}(v_5) &= [v_6], & \text{suc}(v_6) &= [v_1, v_4] \end{aligned}$$

(Como era de esperar, el número de sucesores es igual a 9, que es el número de aristas).

- e) No hay fuentes, ya que todo vértice es el punto terminal de alguna arista. Sólo v_4 es un sumidero puesto que ninguna arista empieza en v_4 ; es decir, $\text{suc}(v_4) = \emptyset$, el conjunto vacío.

9.4 Sea G el grafo dirigido con conjunto de vértices $V(G) = (a, b, c, d, e, f, g)$ y conjunto de aristas:

$$E(G) = \{(a, a), (b, e), (a, e), (e, b), (g, c), (a, e), (d, f), (d, b), (g, g)\}$$

- a) Identifique cualquier lazo o aristas paralelas.
- b) ¿Hay alguna fuente en G ?
- c) ¿Hay algún sumidero en G ?
- d) Encuentre el subgrafo H de G determinado por el conjunto de vértices $V' = \{a, b, c, d\}$.
- a) Un lazo es una arista cuyos puntos inicial y terminal son los mismos; por tanto, (a, a) y (g, g) son lazos. Dos aristas son paralelas si sus puntos inicial y terminal son los mismos. Así, (a, e) y (a, e) son aristas paralelas.
- b) El vértice d es una fuente, ya que ninguna arista termina en d ; es decir, d no aparece como el segundo elemento en ninguna arista. No hay otras fuentes.
- c) Tanto c como f son sumideros, ya que ninguna arista empieza en c o en f ; es decir, ni c ni f aparecen como el primer elemento en ninguna arista. No hay otros sumideros.
- d) Sea E' que consta de todas las aristas de G cuyos puntos terminales están en $V' = \{a, b, c, d\}$. Así se obtiene $E' = \{(a, a), (d, b)\}$. Entonces, $H = H(V', E')$.

ÁRBOLES CON RAÍZ, ÁRBOLES CON RAÍZ ORDENADOS

9.5 Sea T el árbol con raíz en la figura 9-22.

- a) Identifique el camino α de la raíz R a cada uno de los vértices siguientes, y encuentre el número de nivel n del vértice: i) H ; ii) F ; iii) M .
- b) Encuentre los hermanos de E .
- c) Encuentre las hojas de T .

a) Los vértices del árbol se enumeran procediendo a partir de R hacia el vértice. El número de vértices, que no sean R , es el número de nivel:

$$i) \alpha = (R, A, C, H), n = 3; \quad ii) \alpha = (R, B, F), n = 2; \quad iii) \alpha = (R, B, G, L, M), n = 4.$$

b) Los hermanos de E son F y G , puesto que tienen el mismo padre.

c) Las hojas son los vértices sin hijos; es decir, H, D, I, J, K, M, N

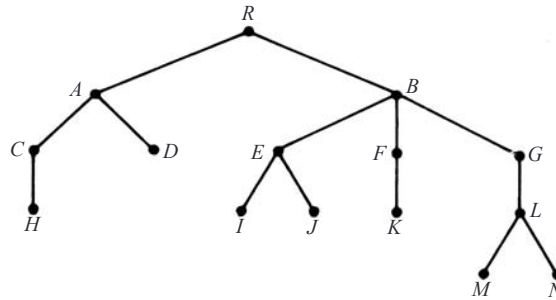


Figura 9-22

9.6 Sea T el árbol con raíz ordenado en la figura 9-23 cuyos vértices se han etiquetado mediante el sistema universal de direcciones. Encuentre el orden lexicográfico de las direcciones del árbol T .

Un árbol T con raíz ordenado suele trazarse de modo que las aristas estén ordenadas de izquierda a derecha como en la figura 9-23. El orden lexicográfico se obtiene al leer la rama que está más a la izquierda, luego la segunda rama a la izquierda y así en lo sucesivo.

Al leer la rama que está más a la izquierda de T se obtiene:

$$0, 1, 1.1, 1.1.1$$

La rama siguiente es 1.2, 1.2.1, 1.2.1.1, de modo que esto se agrega a la lista para obtener

$$0, 1, 1.1, 1.1.1, 1.2, 1.2.1, 1.2.1.1$$

Al proceder se esta manera, finalmente se obtiene

$$0, 1, 1.1, 1.1.1, 1.2, 1.2.1, 1.2.1.1, 1.2.2, 1.3, 2, 2.1, 2.2.1$$

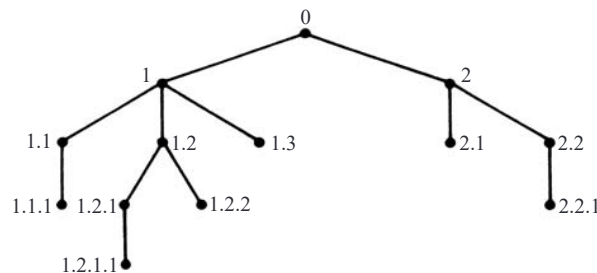


Figura 9-23

REPRESENTACIÓN SECUENCIAL DE GRAFOS

9.7 Considere el grafo G en la figura 9-21a), y suponga que los vértices están almacenados en la memoria en el arreglo:

$$\text{DATA: } X, Y, Z, W$$

- a) Encuentre la matriz de adyacencia A del grafo G y las potencias A^2, A^3, A^4 .
 b) Encuentre la matriz de caminos P de G con las potencias de A . ¿ G es fuertemente conexo?
 a) Los vértices suelen ordenarse según la forma en que aparecen en la memoria; es decir, se supone $v_1 = X, v_2 = Y, v_3 = Z, v_4 = W$. La matriz de adyacencia $A = [a_{ij}]$ se obtiene al hacer $a_{ij} = 1$ si hay una arista de v_i a v_j ; y 0 en caso contrario. A continuación se muestran la matriz A y sus potencias:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, A^2 = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}, A^3 = \begin{bmatrix} 0 & 1 & 2 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, A^4 = \begin{bmatrix} 0 & 2 & 2 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

- b) Puesto que G tiene 4 vértices, sólo es necesario encontrar la matriz $B_4 = A + A^2 + A^3 + A^4$ y luego la matriz de caminos $P = [p_{ij}]$ se obtiene al hacer $p_{ij} = 1$ siempre que en la matriz B_4 haya una entrada diferente de cero, y 0 en caso contrario. A continuación se muestran las matrices B_4 y P :

$$B_4 = \begin{bmatrix} 0 & 5 & 6 & 8 \\ 0 & 1 & 2 & 3 \\ 0 & 3 & 3 & 5 \\ 0 & 2 & 3 & 5 \end{bmatrix} \quad \text{y} \quad P = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

La matriz de caminos P indica que no hay caminos desde ningún nodo hacia v_1 . Por tanto, G no es fuertemente conexo.

9.8 Considere la matriz de adyacencia A del grafo G en la figura 9-19a) obtenida en el problema 9.7. Encuentre la matriz de caminos P de G con el algoritmo de Warshall en lugar de las potencias de A .

Al inicio se hace $P_0 = A$. Luego, P_1, P_2, P_3, P_4 se obtienen recursivamente al hacer

$$P_k[i, j] = P_{k-1}[i, j] \vee (P_{k-1}[i, k] \wedge P_{k-1}[k, j])$$

donde $P_k[i, j]$ denota el ij -ésimo elemento de la matriz P_k . Es decir, al hacer

$$P_k[i, j] = 1 \quad \text{si} \quad P_{k-1}[i, j] = 1 \quad \text{o si ambas} \quad P_{k-1}[i, k] = 1 \quad \text{y} \quad P_{k-1}[k, j] = 1$$

Las matrices P_1, P_2, P_3, P_4 son las siguientes:

$$P_1 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P_2 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P_3 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}, P_4 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Observe que $P_1 = P_2 = A$. Los cambios en P_3 ocurren por las razones siguientes:

$$P_3[4, 2] = 1 \quad \text{porque} \quad P_2[4, 3] = 1 \quad \text{y} \quad P_2[3, 2] = 1$$

$$P_3[4, 4] = 1 \quad \text{porque} \quad P_2[4, 3] = 1 \quad \text{y} \quad P_2[3, 4] = 1$$

9.9 Dibuje una representación del grafo ponderado G que se mantiene en la memoria mediante el siguiente arreglo de vértices DATA y la matriz de pesos W :

$$\text{DATA: } X, Y, S, T; \quad W = \begin{bmatrix} 0 & 0 & 3 & 0 \\ 5 & 0 & 1 & 7 \\ 2 & 0 & 0 & 4 \\ 0 & 6 & 8 & 0 \end{bmatrix}$$

La representación se muestra en la figura 9-24a). Los vértices se etiquetaron con las entradas en DATA.

Si se supone que $v_1 = X, v_2 = Y, v_3 = S, v_4 = T$, el orden de los vértices aparece en el arreglo DATA, se traza una arista de v_i a v_j con peso w_{ij} cuando $w_{ij} \neq 0$.

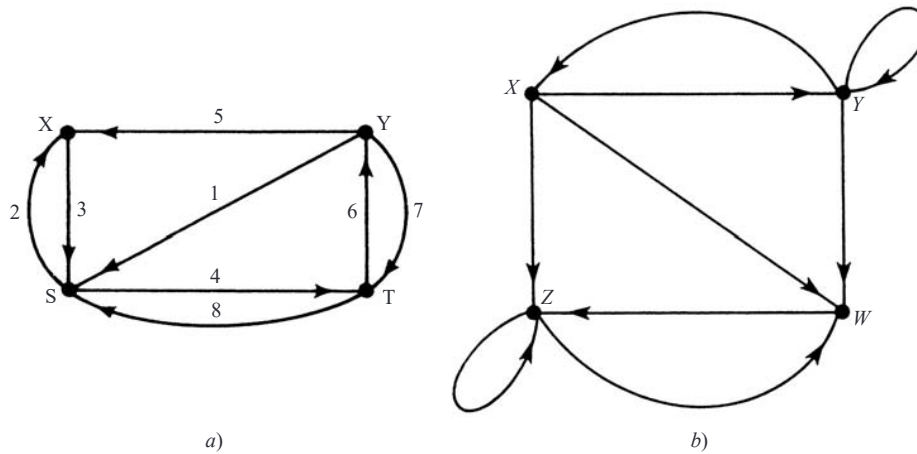


Figura 9-24

REPRESENTACIÓN LIGADA DE GRAFOS

9.10 Sea G el grafo presentado por medio de la tabla siguiente:

$$G = [X : Y, Z, W; Y : X, Y, W; Z : Z, W; W : Z]$$

- a) Encuentre el número de vértices y aristas en G .
 - b) Trace el grafo de G .
 - c) ¿Hay alguna fuente o algún sumidero?
- a) La tabla indica que hay cuatro vértices, X, Y, Z, W . Los grados de salida de los vértices son 3, 3, 2, 1, respectivamente. Por tanto, hay $3 + 3 + 2 + 1 = 9$ aristas.
- b) Con las listas de adyacencia en la figura 9-24b) se traza el grafo.
- c) Ningún vértice tiene grado de salida cero, por lo que no hay sumideros. Asimismo, ningún vértice tiene grado de entrada cero; es decir, cada vértice es un sucesor y no hay fuentes.

9.11 Un grafo ponderado G con seis vértices, A, B, \dots, F , se almacena en la memoria mediante una representación ligada con un archivo de vértices y un archivo de aristas como en la figura 9-25a).

Archivo vértice

	1	2	3	4	5	6	7	8
VERTEX	D		B	F	A		C	E
NEXT-V	7		1	5	0		8	4
PTR	9		3	0	6		10	1

START 3

Archivo arista

	1	2	3	4	5	6	7	8	9	10
BEG-V	8	5	3		5	5	3		1	7
END-V	1	4	7		3	1	1		8	8
NEXT-E	0	5	7		0	2	0		0	0
WEIGHT	3	4	2		6	3	1		2	5

Figura 9-25

- a) Enumere los vértices en el orden en que aparecen en la memoria.
- b) Encuentre la lista de sucesores $\text{suc}(v)$ de cada vértice v .
- c) Trace el grafo de G .

- a) Puesto que $START = 3$, la lista empieza con el vértice B . Luego, $NEXT-V$ indica ir hacia $1(D)$, enseguida a $7(C)$, a $8(E)$, a $4(F)$ y luego a $5(A)$; es decir,

$$B, D, C, E, F, A$$

- b) Aquí $suc(A) = [1(D), 4(F), 3(B)] = [D, F, B]$. Específicamente, $PTR[5(A)] = 6$ y $END-V[6] = 1(D)$ indican que $suc(A)$ empieza con D . Luego, $NEXT-E[6] = 2$ y $END-V[2] = 4(F)$ indican que F es el siguiente vértice en $suc(A)$. Luego, $NEXT-E[2] = 5$ y $END-V[5] = 3(B)$ indican que B es el siguiente vértice en $suc(A)$. Sin embargo, $NEXT-E[5] = 0$ indica que ya no hay más sucesores de A . En forma semejante,

$$suc(B) = [C, D], \quad suc(C) = [E], \quad suc(D) = [E], \quad suc(E) = [D]$$

Además, $suc(F) = \emptyset$, puesto que $PTR[4(F)] = 0$. En otras palabras,

$$G = [A:D, F, B; B:C, D; C:E; D:E; E:D; F:\emptyset]$$

- c) Utilice la lista de sucesores obtenida en el inciso b) y los pesos de las aristas en el archivo de aristas en la figura 9-25a) para trazar el grafo en la figura 9-25b).

9.12 Suponga que una aerolínea tiene nueve vuelos diarios como sigue:

103	Atlanta a Houston	203	Boston a Denver	305	Chicago a Miami
106	Houston a Atlanta	204	Denver a Boston	308	Miami a Boston
201	Boston a Chicago	301	Denver a Reno	401	Reno a Chicago

Describa los datos por medio de un grafo dirigido etiquetado G .

Los datos se describen mediante el grafo en la figura 9-26a) (donde los números de vuelo se han omitido por conveniencia en la notación).

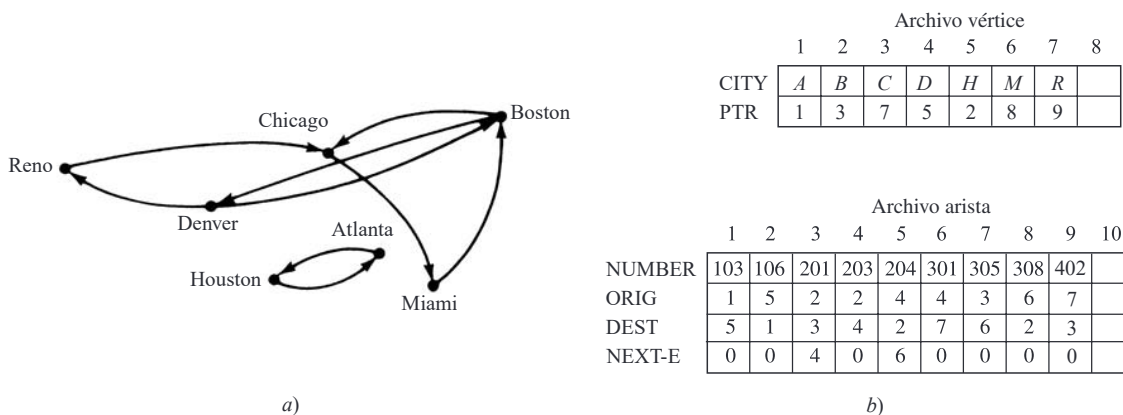


Figura 9-26

9.13 Describa cómo el grafo en el problema 9.12 puede aparecer en la memoria mediante una representación ligada donde las ciudades y los vuelos aparezcan en arreglos lineales ordenados.

Vea la figura 9-26b) (donde A, B, \dots , denotan, respectivamente, Atlanta, Boston, ...). No hay necesidad de una variable $START$, puesto que las ciudades constituyen un arreglo, no una lista ligada. También se usa $ORIG$ (origen) y $DEST$ (destino) en lugar de $BEG-V$ y $END-V$.

9.14 Resulta evidente que los datos del problema 9.12 pueden almacenarse de manera eficiente en un archivo en el que cada registro contiene sólo tres campos:

Número de vuelo, Ciudad de origen, Ciudad de destino

Sin embargo, cuando hay demasiados vuelos, esta representación no contesta fácilmente las siguientes preguntas naturales:

- i) ¿Hay un vuelo directo de la ciudad X a la ciudad Y ?
- ii) ¿Es posible volar de la ciudad X a la ciudad Y ?
- iii) ¿Cuál es el camino más directo (número mínimo de escalas) de la ciudad X a la ciudad Y ?

Muestre cómo la respuesta, por ejemplo en el inciso ii), puede obtenerse más fácilmente si los datos se almacenan en la memoria con la representación ligada en el grafo de la figura 9-26b).

Una forma de contestar al inciso ii) es usar un algoritmo de búsqueda en anchura o en profundidad para decidir si la ciudad Y es alcanzable desde la ciudad X . Estos algoritmos requieren las listas de adyacencia, que pueden obtenerse fácilmente a partir de la representación ligada de un grafo, pero no a partir de la representación anterior, que sólo usa tres campos.

PROBLEMAS DIVERSOS

9.15 Sea $A = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ la matriz de adyacencia de un multigrafo G . Dibuje una representación de G .

Puesto que A es una matriz de 4×4 , G tiene cuatro vértices v_1, v_2, v_3, v_4 . Para cada entrada a_{ij} en A , se trazan a_{ij} arcos (aristas dirigidas) del vértice v_i al vértice v_j para obtener el grafo de la figura 9-27a).

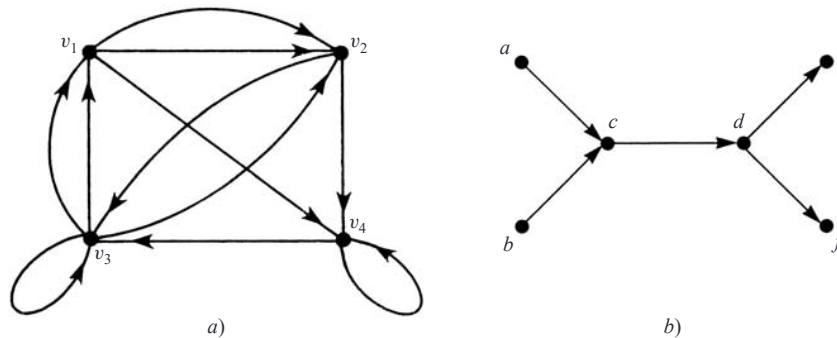


Figura 9-27

9.16 Sea S el grafo libre de ciclos en la figura 9-27b). Encuentre todos los ordenamientos topológicos posibles de S .

Hay cuatro ordenamientos topológicos posibles de S : cada ordenamiento T debe empezar con a o con b , debe terminar con e o con f y c y d deben ser los elementos tercero y cuarto, respectivamente. Los cuatro ordenamientos son los siguientes:

$$T_1 = [a, b, c, d, e, f], \quad T_2 = [b, a, c, d, e, f]$$

$$T_3 = [a, b, c, d, f, e], \quad T_4 = [b, a, c, d, f, e]$$

9.17 Demuestre la proposición 9.4: sea A la matriz de adyacencia de un grafo G . Entonces $a_K[i, j]$, la ij -ésima entrada en la matriz A^K , proporciona el número de caminos de longitud K de v_i a v_j .

La demostración es por inducción sobre K . Un camino de longitud 1 de v_i a v_j es precisamente una arista (v_i, v_j) . Por definición de la matriz de adyacencia A , $a_1[i, j] = a_{ij}$ proporciona el número de aristas de v_i a v_j . Así, la proposición es verdadera para $K = 1$.

Se supone $K > 1$. (Es decir, que G tiene m nodos). Puesto que $A^K = A^{K-1}A$,

$$a_K[i, j] = \sum_{s=1}^m a_{K-1}[i, s] a_1[s, j]$$

Por inducción, $a_{K-1}[i, s]$ proporciona el número de caminos de longitud $K - 1$ de v_i a v_s y $a_1[s, j]$ proporciona el número de caminos de longitud 1 de v_s a v_j . Por tanto, $a_{K-1}[i, s]a_1[s, j]$ proporciona el número de caminos de longitud K de v_i a v_j donde v_s es el penúltimo vértice. Por lo que, todos los caminos de longitud K de v_i a v_j pueden obtenerse al sumar el producto $a_{K-1}[i, s]a_1[s, j]$ para toda s . En consecuencia, $a_K[i, j]$ es el número de caminos de longitud K de v_i a v_j . Así, se ha demostrado la proposición.

PROBLEMAS SUPLEMENTARIOS

TERMINOLOGÍA DE GRAFOS

9.18 Considere el grafo G en la figura 9-28a).

- a) Encuentre el grado de entrada y el grado de salida de cada vértice.
- b) ¿Hay alguna fuente o algún sumidero?
- c) Encuentre todos los caminos simples de v_1 a v_4 .
- d) Encuentre todos los ciclos en G .
- e) Encuentre todos los caminos de longitud 3 o menores de v_1 a v_3 .
- f) ¿ G es unilateral o fuertemente conexo?

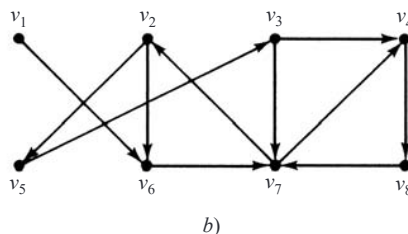
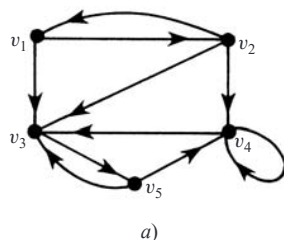


Figura 9-28

9.19 Considere el grafo G en la figura 9-28b).

- a) ¿Hay alguna fuente o algún sumidero?
- b) Encuentre todos los caminos simples de v_1 a v_4 .
- c) Encuentre un camino no simple de v_1 a v_4 .
- d) Encuentre todos los ciclos en G que incluyen a v_4 .

9.20 Considere el grafo G en la figura 9-28b).

- a) Encuentre: $\text{suc}(v_1)$, $\text{suc}(v_3)$, $\text{suc}(v_5)$, $\text{suc}(v_7)$.
- b) Encuentre el subgrafo H de G generado por i) $\{v_1, v_3, v_5, v_6\}$; ii) $\{v_2, v_3, v_6, v_7\}$.

9.21 Sea G el grafo con conjunto de vértices $V(G) = \{A, B, C, D, E\}$ y conjunto de aristas

$$E(G) = \{(A, D), (B, C), (C, E), (D, B), (D, D), (D, E), (E, A)\}$$

- a) Exprese G mediante su tabla de adyacencia.
- b) ¿ G tiene lazos o aristas paralelas?
- c) Encuentre todos los caminos simples de D a E .
- d) Encuentre todos los ciclos en G .
- e) ¿ G es unilateral o fuertemente conexo?
- f) Encuentre el número de subgrafos de G con vértices C, D, E .
- g) Encuentre el subgrafo H de G generado por C, D, E .

9.22 Sea G el grafo con conjunto de vértices $V(G) = \{a, b, c, d, e\}$ y las siguientes listas de sucesores:

$$\text{suc}(a) = [b, c] \quad \text{suc}(b) = \emptyset \quad \text{suc}(c) = [d, e] \quad \text{suc}(d) = [a, b, e] \quad \text{suc}(e) = \emptyset$$

- a) Enumere las aristas de G . b) ¿ G es débil, unilateral o fuertemente conexo?

9.23 Sea G el grafo en la figura 9-29a).

- a) Exprese G mediante su tabla de adyacencia.
- b) ¿ G tiene fuentes o sumideros?
- c) Encuentre todos los caminos simples de A a E .
- d) Encuentre todos los ciclos en G .
- e) Encuentre un camino de expansión en G .
- f) ¿ G es fuertemente conexo?

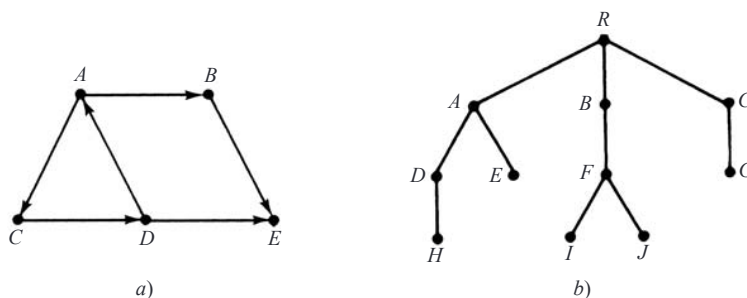


Figura 9-29

ÁRBOLES CON RAÍZ, ÁRBOLES CON RAÍZ ORDENADOS

- 9.24 Sea T el árbol con raíz en la figura 9-29b).
- Identifique el camino α de la raíz R a cada uno de los siguientes vértices, y encuentre el número de nivel del vértice: i) D ; ii) J ; iii) G .
 - Encuentre las hojas de T .
 - Suponga que T es un árbol con raíz ordenado y encuentre la dirección universal de cada hoja de T .
- 9.25 Las siguientes direcciones están en orden aleatorio:
 2.1.1, 3.1, 2.1. 1, 2.2.1.2, 0, 3.2, 2.2, 1.1, 2, 3.1.1, 2.2.1, 3, 2.2.1.1
- Escriba las direcciones en orden lexicográfico.
 - Dibuje el árbol con raíz correspondiente.

REPRESENTACIÓN SECUENCIAL DE GRAFOS

- 9.26 Sea G el grafo en la figura 9-30a).
- Encuentre la matriz de adyacencia A y la matriz de caminos P para G .
 - Para toda $k > 0$, encuentre n_k , donde n_k denota el número de caminos de longitud k de v_1 a v_4 .
 - ¿ G es débil, unilateral o fuertemente conexo?

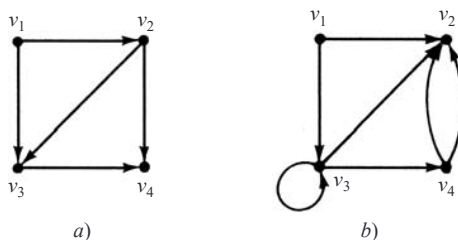


Figura 9-30

- 9.27 Repita el problema 9.26 para el grafo G en la figura 9-30b).
- 9.28 Sea P la matriz de caminos de un grafo G . Describa P cuando G es: a) fuertemente conexo; b) unilateralmente conexo.
- 9.29 Sea G el grafo en la figura 9-31a), donde los vértices se mantienen en la memoria mediante el arreglo DATA: X, Y, Z, S, T .
- Encuentre la matriz de adyacencia A y la matriz de caminos P de G .
 - Encuentre todos los ciclos en G .
 - ¿ G es unilateralmente conexo? ¿Fuertemente conexo?
- 9.30 Sea G el grafo ponderado en la figura 9-31b), donde los vértices se mantienen en la memoria mediante el arreglo DATA: X, Y, S, T .
- Encuentre la matriz ponderada W de G .
 - Use el algoritmo de Warshall para encontrar la matriz Q de los caminos más cortos.

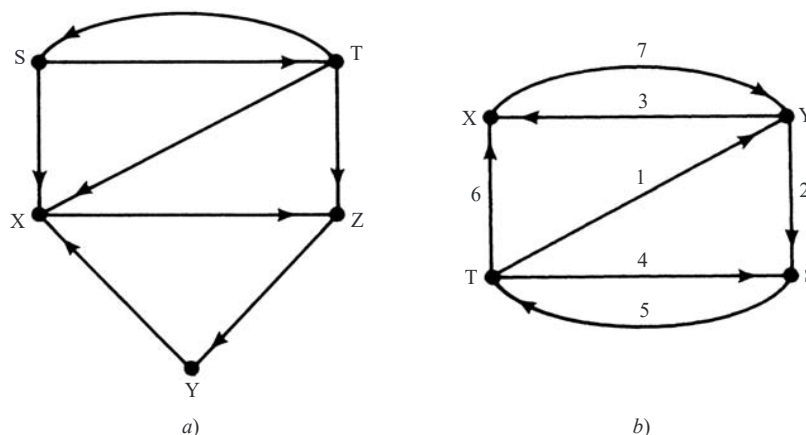


Figura 9-31

REPRESENTACIÓN LIGADA DE GRAFOS

9.31 El grafo ponderado G con seis vértices A, B, \dots, F se almacena en la memoria mediante una representación ligada con un archivo de vértices y un archivo de aristas como en la figura 9-32.

- a) Enumere los vértices en el orden en que aparecen en la memoria.
- b) Encuentre la lista de sucesores $\text{suc}(v)$ de cada vértice v en G .
- c) Dibuje una representación de G .

		Archivo vértice							
		1	2	3	4	5	6	7	8
START	7								
VERTEX		D		B	F	A		C	E
NEXT-V		3		8	1	0		4	5
PTR		7		5	9	2		3	0

		Archivo arista											
		1	2	3	4	5	6	7	8	9	10	11	12
BEG-V		5	5	7		3	7	1		4	1	4	7
END-V		8	7	5		1	1	5		8	4	3	8
NEXT-E		0	1	12		0	0	10		11	0	0	6
WEIGHT		5	2	1		3	2	4		1	3	4	1

Figura 9-32

- 9.32 Sea G el grafo presentado por la tabla: $G = [A : B, C; B : C, D; C : C; D : B; E : \emptyset]$.
- a) Encuentre el número de vértices y aristas en G .
 - b) Dibuje una representación de G .
 - c) ¿Hay alguna fuente o algún sumidero?
 - d) ¿ G es débil, unilateral o fuertemente conexo?
- 9.33 Repita el problema 9.32 para la tabla: $G = [A : D; B : C; C : E; D : B, D, E; E : A]$.
- 9.34 Repita el problema 9.32 para la tabla: $G = [A : B, C, D, K; B : J; C : \emptyset; D : \emptyset; J : B, D, L; K : D, L; L : D]$.
- 9.35 Suponga que una aerolínea tiene ocho vuelos diarios que sirven a las ciudades Atlanta, Boston, Chicago, Denver, Houston, Filadelfia y Washington. Suponga que los datos sobre los vuelos se almacenan en la memoria como en la figura 9-33; es decir, que se usa una representación ligada donde las ciudades y los vuelos aparecen en arreglos ordenados linealmente. Dibuje un grafo dirigido etiquetado G que describa los datos.

Archivo vértice								
	1	2	3	4	5	6	7	8
CITY	A	B	C	D	H	P	W	
PTR	1	2	3	8	9	5	7	

Archivo arista										
	1	2	3	4	5	6	7	8	9	10
NUMBER	101	102	201	202	203	301	302	401	402	
ORIG	1	2	3	1	6	6	7	4	5	
DEST	2	3	6	7	3	1	6	5	4	
NEXT-E	4	0	0	0	6	0	0	0	0	

Figura 9-33

- 9.36 Use los datos en la figura 9-33 para escribir un procedimiento con entrada CITY X y CITY Y que encuentre el número de un vuelo directo de la ciudad X a la ciudad Y , en caso de existir. Use lo siguiente para probar el procedimiento:
- a) $X = \text{Atlanta}, Y = \text{Filadelfia};$ c) $X = \text{Houston}, Y = \text{Chicago};$
 b) $X = \text{Filadelfia}, Y = \text{Atlanta};$ d) $X = \text{Washington}, Y = \text{Chicago}.$
- 9.37 Use los datos en la figura 9-33 para escribir un procedimiento con entrada CITY X y CITY Y que encuentre el camino más directo (número mínimo de escalas) de la ciudad X a la ciudad Y , en caso de existir. Pruebe el procedimiento con los datos de entrada del problema 9.36.

PROBLEMAS DIVERSOS

- 9.38 Use el algoritmo de poda para encontrar el camino más corto de s a t en la figura 9-34.

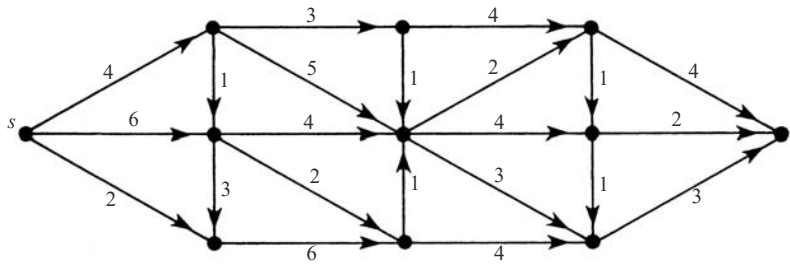


Figura 9-34

- 9.39 Encuentre un ordenamiento topológico T de cada uno de los siguientes grafos:
- a) $G = [A:Z; B:T; C:B; D:\emptyset; X:D; Y:X; Z:B, X; S:C, Z; T:\emptyset]$
 b) $G = [A:X, Z; B:A; C:S, T; D:Y; X:S, T; Y:B; Z:\emptyset; S:Y; T:\emptyset]$
 c) $G = [A:C, S; B:T, Z; C:\emptyset; D:Z; X:A; Y:A; Z:X, Y; S:\emptyset; T:Y]$
- 9.40 Dibuje un grafo etiquetado G que represente la siguiente situación. Tres hermanas, Bárbara, Rosa y Susana, llaman por teléfono, cada una, en forma regular a su madre, Gertrudis, aunque Gertrudis sólo llama a Rosa. Susana no llama a Rosa, aunque Rosa continúa llamando a Susana. Bárbara y Susana se llaman mutuamente, y Bárbara y Rosa se llaman entre sí.
- 9.41 Sea R la relación (grafo dirigido) sobre $V = \{2, 3, 4, 9, 15\}$ definido por “ x es menor que y primo relativo de y ”. a) Dibuje el diagrama del grafo R . b) ¿ R es débilmente conexo? ¿Unilateralmente conexo? ¿Fuertemente conexo?
- 9.42 Un grafo dirigido G es *completo* si, para cada par de vértices distintos u y v , se cumple que (u, v) es un arco o que (v, u) es un arco. Demuestre que un grafo dirigido completo finito G tiene un camino que incluye todos los vértices. (Resulta evidente que esto se cumple para grafos completos no dirigidos.) Por tanto, G es unilateralmente conexo.

- 9.43 Suponga que un grafo G se introduce por medio de un entero M , que representa los vértices $1, 2, \dots, M$ y una lista de N pares ordenados de enteros que representan las aristas de G . Escriba un procedimiento que efectúe lo siguiente:
- Encuentre la matriz de adyacencia A $M \times M$ del grafo G .
 - Utilice A y el algoritmo de Warshall para encontrar la matriz de caminos P de G .
- Use los datos siguientes para probar el procedimiento:
- $M = 5; N = 8; (3, 4), (5, 3), (2, 4), (1, 5), (3, 2), (4, 2), (3, 1), (5, 1)$
 - $M = 6; N = 10; (1, 6), (2, 1), (2, 3), (3, 5), (4, 5), (4, 2), (2, 6), (5, 3), (4, 3), (6, 4)$
- 9.44 Suponga que un grafo G se introduce por medio de un entero M , que representa los vértices $1, 2, \dots, M$ y una lista de N tripletas ordenadas (a_i, b_i, w_i) de enteros tales que (a_i, b_i) es una arista de G y w_i es su peso. Escriba un procedimiento que efectúe lo siguiente:
- Encuentre la matriz de pesos W de $M \times M$ del grafo G .
 - Utilice W y el algoritmo de Warshall para encontrar la matriz Q de caminos más cortos entre los vértices de G .
- Use los datos siguientes para probar el procedimiento:
- $M = 4; N = 7; (1, 2, 5), (2, 4, 2), (3, 2, 3), (1, 1, 7), (4, 1, 4), (4, 3, 1)$
 - $M = 5; N = 8; (3, 5, 3), (4, 1, 2), (5, 2, 2), (1, 5, 5), (1, 3, 1), (2, 4, 1), (3, 4, 4), (5, 4, 4)$
- 9.45 Considere el grafo G en la figura 9-11. Muestre la secuencia de listas de espera en STACK y la secuencia de vértices procesados mientras se lleva a cabo una búsqueda en profundidad (DFS) de G que empiece en el vértice: a) B ; b) E ; c) K .
- 9.46 Considere el grafo G en la figura 9-11. Como se hizo en el ejemplo 9.11, use una búsqueda en anchura de G para encontrar el camino más corto de K a F . En particular, muestre la secuencia de listas de espera en QUEUE durante la búsqueda.

Respuestas a los problemas suplementarios

Notación: $M = [R_1; R_2; \dots; R_n]$ denota una matriz con renglones R_1, R_2, \dots, R_n .

- 9.18 a) Grados de entrada: 1, 1, 4, 3, 1; grados de salida: 2, 3, 1, 2, 2.
 b) Ninguno.
 c) $(v_1, v_2, v_4), (v_1, v_3, v_5, v_4), (v_1, v_2, v_3, v_5, v_4)$
 d) (v_3, v_5, v_4, v_3)
 e) $(v_1, v_3), (v_1, v_2, v_3), (v_1, v_2, v_4, v_3), (v_1, v_2, v_1, v_3), (v_1, v_3, v_5, v_7)$
 f) unilateralmente conexo, pero no fuertemente conexo.
- 9.19 a) Fuentes: v_1
 b) $(v_1, v_6, v_7, v_4), (v_1, v_6, v_7, v_2, v_5, v_3, v_4)$
 c) $(v_1, v_6, v_7, v_2, v_6, v_7, v_4)$
 d) $(v_4, v_8, v_7, v_4), (v_4, v_8, v_7, v_2, v_5, v_3, v_4)$
- 9.20 a) $(\text{suc}1) = [6], (\text{suc}3) = [4, 7], (\text{suc}5) = [3], (\text{suc}7) = [2, 4]$.
 b) i) $(1, 6), (5, 3)$; ii) $(2, 6), (6, 7), (7, 2), (3, 7)$.
- 9.21 a) $G = [A : D; B : C; C : E; D : B, D, E; E : A]$
 b) Lazo: D, D
 c) $(D, E), (D, B, C, E)$
 d) $(A, D, E, A), (A, D, B, C, E, A)$
 e) Unilateral y fuertemente conexo.
 f) y g) H tiene tres aristas: $(C, E), (D, E), (D, D)$. Hay $8 = 2^3$ formas de escoger alguno de las tres aristas; y con cada elección se obtiene un subgrafo.
- 9.22 a) $(a, b), (a, c), (c, d), (c, e), (d, a), (d, b), (d, e)$
 b) Puesto que b y e son sumideros, de b a e o de e a b no hay ningún camino, de modo que G no es unilateral ni

fuertemente conexo. G es débilmente conexo, ya que cc, a, b, d, e es un semicamino de expansión.

- 9.23 a) $G = [A : B, C : B : E; C : D; E : \emptyset]$; b) Sumidero: E ; c) $(A, B, E), (A, C, D, E)$; d) (A, C, D, A) ; e) (C, D, A, B, E) ; f) No.
- 9.24 a) i) $(R, A, D), 2$; ii) $(R, B, F, J), 3$; iii) $R, C, G), 2$.
 b) H, E, I, J, G
 c) $H : 1.1.1, E : 1.2, I : 2.1.1, J : 2.1.2, G : 3.1$
- 9.25 a) $0, 1, 1.1, 2, 2.1, 2.1.1, 2.2, 2.2.1, 2.2.1.1, 2.2.1.2, 3, 3.1, 3.1.1, 3.2$. b) Fig. 9-35a).
- 9.26 a) $A = [0, 1, 1, 0; 0, 0, 1, 1; 0, 0, 0, 1; 0, 0, 0, 0]$;
 $P = [0, 1, 1, 1; 0, 0, 1, 1; 0, 0, 0, 1; 0, 0, 0, 0]$;
 b) $0, 2, 1, 0, 0, \dots$; c) Débil y unilateralmente conexo.
- 9.27 a) $A = [0, 1, 1, 0; 0, 0, 0, 0; 0, 1, 1, 1; 0, 2, 0, 0]$;
 $P = [0, 1, 1, 1; 0, 0, 0, 0; 0, 1, 1, 1; 0, 1, 0, 0]$;
 b) $0, 1, 1, 1, \dots$; c) Débil y unilateralmente conexo.
- 9.28 Sea $P = [p_{ij}]$. Para $i \neq j$: a) $p_{ij} \neq 0$; b) cualquiera $p_{ij} \neq 0$ o $p_{ji} \neq 0$.
- 9.29 a) $A = [0, 0, 1, 0, 0; 1, 0, 0, 0, 0; 0, 1, 0, 0, 0; 1, 0, 0, 0, 1; 1, 0, 1, 1, 0]$;
 $P = [1, 1, 1, 0, 0; 1, 1, 1, 0, 0; 1, 1, 1, 0, 0; 1, 1, 1, 1, 1; 1, 1, 1, 1, 1]$
 b) $(X, Z, Y, X); (S, T, S)$ c) Unilateralmente.
- 9.30 a) $A = [0, 7, 0, 0; 3, 0, 2, 0; 0, 0, 0, 5; 6, 1, 4, 0]$
 b) $Q = [XYX, XY, XYS, XYST; YX, YSTY, YS, YST; STYX, STY, STYS, ST; TX, TY, TYS, TYST]$
- 9.31 a) C, F, D, B, E, A ; b) $[A : C, E; B : D; C : D, E, A; D : A, F; E : \emptyset; F : B, E]$; c) Vea la figura 9-35b).
- 9.32 a) $5, 6$; b) fuente: A ; c) Vea la figura 9-36a); ninguno.

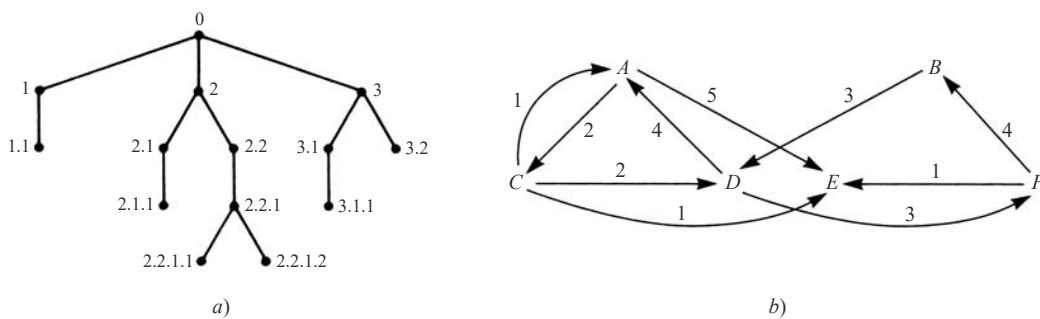


Figura 9-35

9.33 a) 5, 1; b) ninguno: A; c) Vea la figura 9-36b); d) los tres.

9.34 a) 7, 11; b) fuente: A; sumideros: C, D; c) Vea la figura 9-36c); d) sólo débilmente.

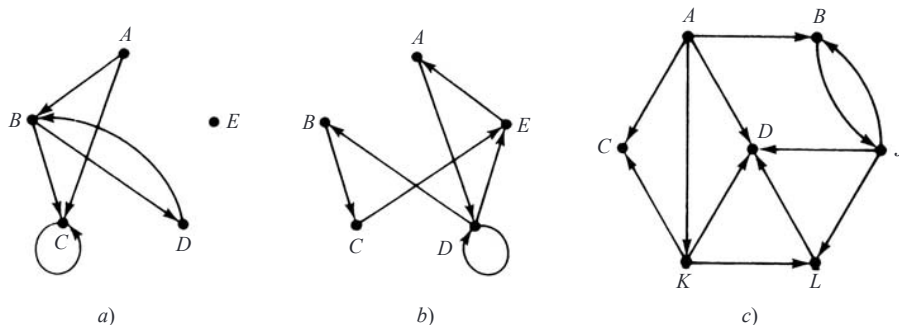


Figura 9-36

9.35 Vea la figura 9-37.

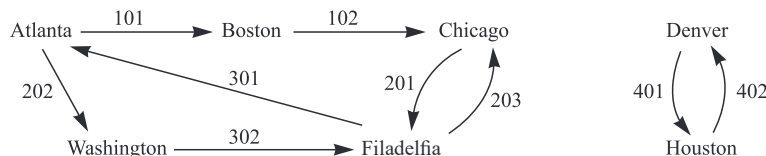


Figura 9-37

9.36 a) No; b) sí; c) no; d) no.

9.37 a) AWP; b) PA; c) ninguno; d) WPC.

9.38 (s, 4, 1, 2, 1, 2, 1, 2, t)

9.39 **Sugerencia:** Primero trace el grafo. a) ASYCZBXTD; b) ninguno, el grafo no es libre de ciclos; por ejemplo, YBAXSY es un ciclo; c) BTYXACSDZ.

9.40 Vea la figura 9-38a).

9.41 a) Vea la figura 9-38b). b) Sólo débilmente conexo.

9.42 **Sugerencia:** Suponga que $(\alpha = v_1, \dots, v_m)$ es un camino más largo en G y que no incluye al vértice u . Si (u_1, v_1) es un arco, entonces $\beta = (u, \alpha)$ extiende a α . Por tanto, (v_1, u) es un arco. Si (u, v_2) también es un arco, entonces $\beta = (v_1, u, v_2, \dots, v_m)$ extiende a α ; por tanto, (v_2, u) es un arco. En forma semejante, $(v_3, u), \dots, (v_m, u)$ son arcos. Por tanto, $\beta = (\alpha, u)$ extiende a α . Esto contradice la maximalidad de α .



Figura 9-38

- 9.43 i) $A = [0, 0, 0, 0, 0; 0, 0, 0, 1, 0; 1, 1, 0, 1, 0; 0, 1, 0, 0, 0; 1, 0, 1, 0, 0]$
 $P = [1, 1, 1, 1, 1; 0, 1, 0, 1, 0; 1, 1, 1, 1, 1; 0, 1, 0, 1, 0; 1, 1, 1, 1, 1]$
 ii) $A = [0, 0, 0, 0, 0, 1; 1, 0, 1, 0, 0, 1; 0, 0, 0, 0, 1, 0; 0, 1, 1, 0, 1, 0; 0, 0, 1, 0, 0, 0; 0, 0, 0, 1, 0, 0]$
 $P = [1, 1, 1, 1, 1, 1; 1, 1, 1, 1, 1, 1; 0, 0, 1, 0, 1, 0; 1, 1, 1, 1, 1, 1; 0, 0, 1, 0, 1, 0; 1, 1, 1, 1, 1, 1]$
- 9.44 i) $W = [7, 5, 0, 0; 0, 0, 0, 2; 0, 3, 0, 0; 4, 0, 1, 0];$
 $Q = [AA, AB, ABCD, ABD; BDA, BDCB, BDC, BD; CBDA, CB, CBDC, CBD; DA, DCB, DC, DCBD],$
 donde A, B, C, D son los vértices.
 ii) $W = [0, 0, 1, 0, 5; 0, 0, 0, 1, 0; 0, 0, 0, 4, 3; 2, 0, 0, 0, 0; 0, 2, 0, 4, 0];$

- $Q = [ACDA, ACEB, AC, ACD, ACE; BDA, BDACEB, BDAC, BD, BDACE; CDA, CEB, CDAC, CD, CE; DA, DACEB, DAC, DACD, DACEB; EDA, EB, EDAC, ED, EDACE]$ donde A, B, C, D, E son los vértices.
- 9.45 a) STACK: $B, L_B E_B, E_L C_L E_B, F_E C_L, D_F C_L, C_L, J_C, K_J, \emptyset$; Vértice: $B, L_B, E_L, F_E, D_F, C_L, J_C, K_J$
 b) STACK: E, F_E, D_F, \emptyset ; Vértice: E, F_E, D_F
 c) STACK: $K, L_K C_K, E_L C_L, C_K, C_L, D_F C_L, C_L J_C, \emptyset$; Vértice: $K, L_K, E_L, F_E, D_F, C_L, J_C$
- 9.46 QUEUE: $K, L_K C_K, J_C E_C D_C L_K, J_C E_C D_C, J_C E_C, F_E$; Vértice: $K, C_K, L_K, D_C, E_C, J_C, F_E$; Camino mínimo: $F_E \leftarrow E_C \leftarrow C_K \leftarrow K \rightarrow C_K \rightarrow E_C \rightarrow F_E$.