



ESTRUCTURA DE DATOS

Contenido

TEMA 4. – Grafos

- 4.1. Grafos
 - 4.1.1. Definición
 - 4.1.2. Conceptos
- 4.2. Modelado de problemas típicos
- 4.3. Representación de un grafo a través de su matriz de:
 - 4.3.1. Adyacencia
 - 4.3.2. Incidencia
- 4.4. Representación a través de una lista de sucesores
- 4.5. Recorridos de grafos
 - 4.5.1. Anchura
 - 4.5.2. Profundidad

4,1,1 Definición

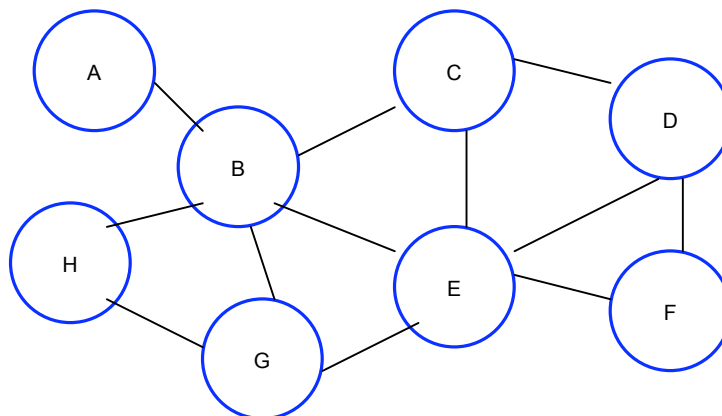
Un grafo, G , es un par, compuesto por dos conjuntos V y A . Al conjunto V se le llama conjunto de vértices o nodos del grafo. A es un conjunto de pares de vértices, estos pares se conocen habitualmente con el nombre de arcos o ejes del grafo. Se suele utilizar la notación $G = (V, A)$ para identificar un grafo.

Ejemplo:

$G = (V, A)$

$V = \{A, B, C, D, E, F, G, H\}$

$A = \{ \{A, B\}, \{B, C\}, \{B, E\}, \{B, G\}, \{B, H\}, \{C, D\}, \{C, E\}, \{D, F\}, \{D, E\}, \{E, G\}, \{E, F\}, \{G, H\} \}$



Los grafos permiten representar conjuntos de objetos arbitrariamente relacionados. Se puede asociar el conjunto de vértices con el conjunto de objetos y el conjunto de arcos con las relaciones que se establecen entre ellos.

Los grafos son modelos matemáticos de numerosas situaciones reales: un mapa de

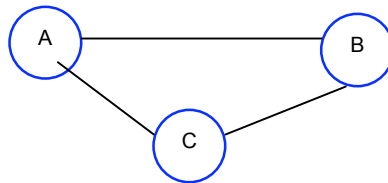
carreteras, una figura geométrica, la red de ferrocarriles, el plano de un circuito eléctrico, el esquema de la red telefónica de una compañía, etc.

El número de distintos pares de vértices $(v(i), v(j))$, con $v(i) \neq v(j)$, en un grafo con n vértices es $n(n-1)/2$. Este es el número máximo de arcos en un grafo no dirigido de n vértices. Un grafo no dirigido que tenga exactamente $n(n-1)/2$ arcos se dice que es un **grafo completo**. En el caso de un grafo dirigido de n vértices el número máximo de arcos es $n(n-1)$.

Tipos de grafos

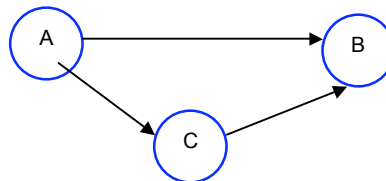
Grafo no dirigido (grafo)

En un grafo no dirigido el par de vértices que representa un arco no está ordenado. Por lo tanto, los pares $(v1, v2)$ y $(v2, v1)$ representan el mismo arco.



Grafo dirigido (digrafo)

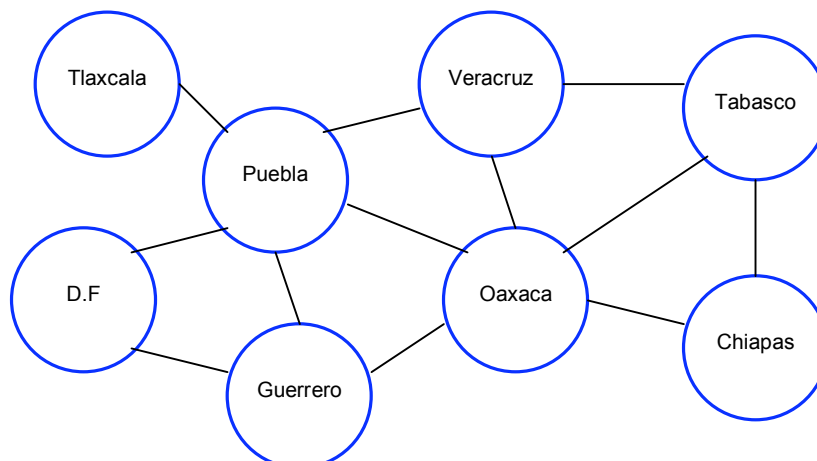
En un grafo dirigido cada arco está representado por un par ordenado de vértices, de forma que $(v1, v2)$ y $(v2, v1)$ representan dos arcos diferentes.



Grafo etiquetado

Un grafo cuyos nodos tienen asociada información. Un ejemplo común es que los nodos contengan los nombres de ciudades en un mapa aéreo.

Ejemplo: Conexiones entre algunos estados de la república mexicana.

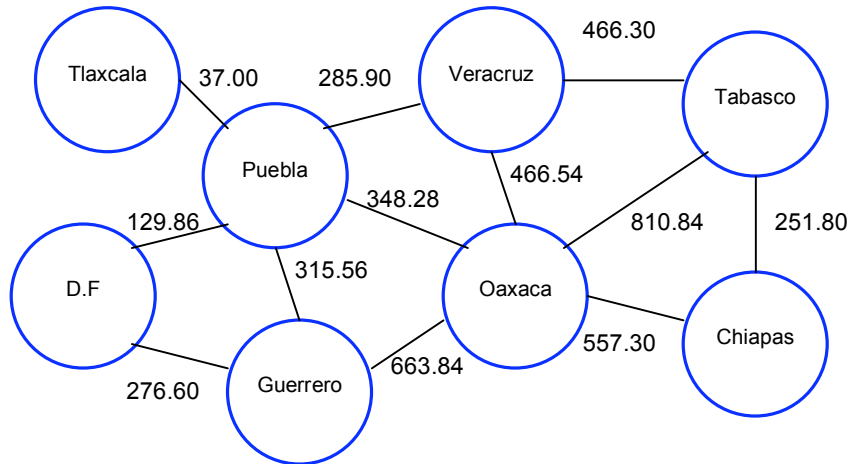


Grafo ponderado

Un grafo donde cada arco tiene asociado un valor numérico o peso. Un grafo donde se indica las distancias de diferentes rutas en un mapa aéreo, es un ejemplo típico.

Ejemplo:

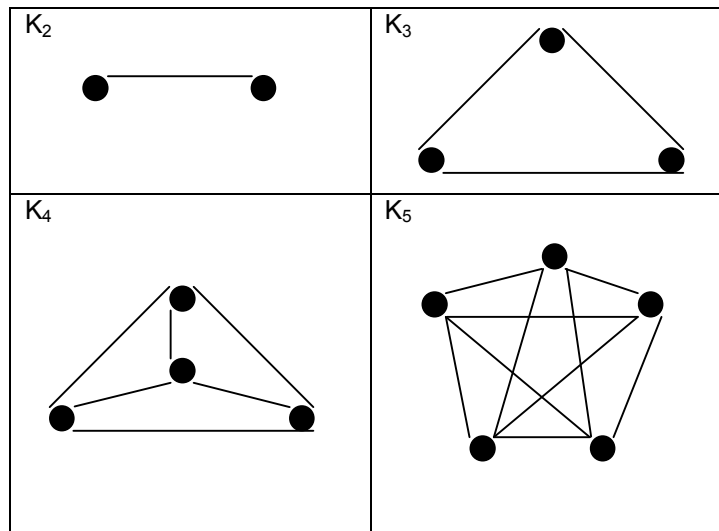
Conexiones entre algunas ciudades de la república mexicana, donde el peso de cada arista significa la distancia en kilómetros desde la ciudad capital de un estado al otro.



Conceptos-Terminología

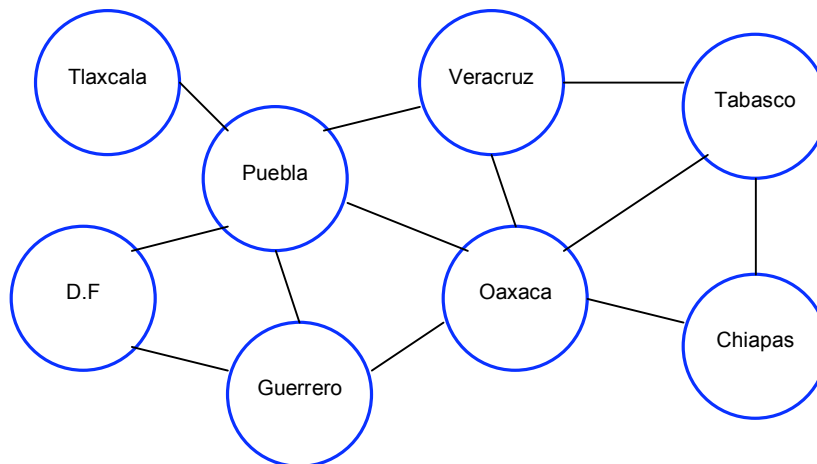
- **Orden de un grafo:** es el número de nodos (vértices) del grafo.
- **Grado de un nodo:** llamado también valencia de un nodo, es el número de ejes (arcos) que inciden sobre el nodo. Se denota como $\delta(v)$. Una nota importante es que la suma de los grados de los vértices es igual al doble del número de aristas.
- **Grafo simétrico:** es un grafo dirigido tal que si existe la relación entonces existe , con u, v pertenecientes a V .
- **Grafo no simétrico:** es un grafo que no cumple la propiedad anterior.
- **Grafo reflexivo:** es el grafo que cumple que para todo nodo u de V existe la relación (u, u) de A .
- **Grafo transitivo:** es aquél que cumple que si existen las relaciones (u, v) y (v, z) de A entonces existe (u, z) de A .
- **Grafo completo:** es el grafo que contiene todos los posibles pares de relaciones, es decir, para cualquier par de nodos u, v de V , $(u \neq v)$, existe (u, v) de A .
- **Camino:** un camino en el grafo G es una sucesión de vértices y arcos: $v(0), a(1), v(1), a(2), v(2), \dots, a(k), v(k)$; tal que los extremos del arco $a(i)$ son los vértices $v(i-1)$ y $v(i)$.
- **Longitud de un camino:** es el número de arcos que componen el camino. Para un grafo ponderado, la suma de sus pesos es la longitud del camino.
- **Camino simple:** camino donde sus vértices son distintos dos a dos, salvo a lo sumo los extremos.
- **Ciclo o camino cerrado** (llamado también circuito): camino en el que coinciden los vértices extremos ($v(0) = v(k)$). Y en el que no se repiten aristas
- **Grafo conexo:** es un grafo no dirigido tal que para cualquier par de nodos existe al menos un camino que los une.
- **Grafo fuertemente conexo:** es un grafo dirigido tal que para cualquier par de nodos existe un camino que los une.

Ejemplos de grafos completos, si $k = n-1$:



- Punto de articulación: es un nodo que si desaparece provoca que se cree un grafo no conexo.
- Componente conexas: subgrafo conexo maximal de un grafo no dirigido (parte más grande de un grafo que sea conexas).

Ejemplo 1:



Orden del grafo: Cantidad de nodos:

8

Grado del nodo Oaxaca:

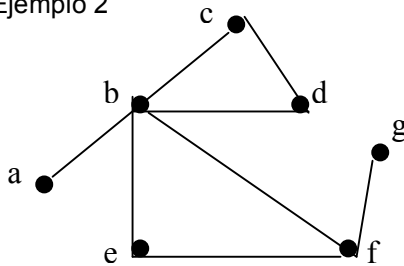
5

Caminos simples del D.F. a Chiapas

D.F-Puebla-Oaxaca-Chiapas
 D.F-Puebla-Oaxaca-Tabasco-Chiapas
 D.F-Puebla-Veracruz-Tabasco-Chiapas
 D.F-Guerrero-Oaxaca-Veracruz-Tabasco-Chiapas
 D..F-Guerrero-Puebla-Oaxaca-Veracruz-Tabasco-Chiapas
 D.F-Guerrero-Oaxaca-Chiapas
 D.F-Guerrero-Puebla-Oaxaca-Chiapas
 D.F-Guerrero-Puebla-Oaxaca-Tabasco-Chiapas
 D.F-Guerrero-Puebla-Veracruz-Tabasco-Chiapas

Longitud del camino D.F-Guerrero-Puebla-Oaxaca-Chiapas: 4

Ejemplo 2



Sucesión de lados	Camino	Camino Simple	Ciclo	Ciclo Simple
(a, b, d, f, g)	No	No	No	No
(f, e, b, d, c, b, a)	Si	No	No	No
(f, e, b, d)	Si	Si	No	No
(b, f, e, b, d, c, b)	Si	No	Si	No
(e, f, b, e)	Si	No	Si	Si

- Recordar que: Un camino puede repetir vértices NO lados, mientras que un camino simple NO puede repetir vértices.

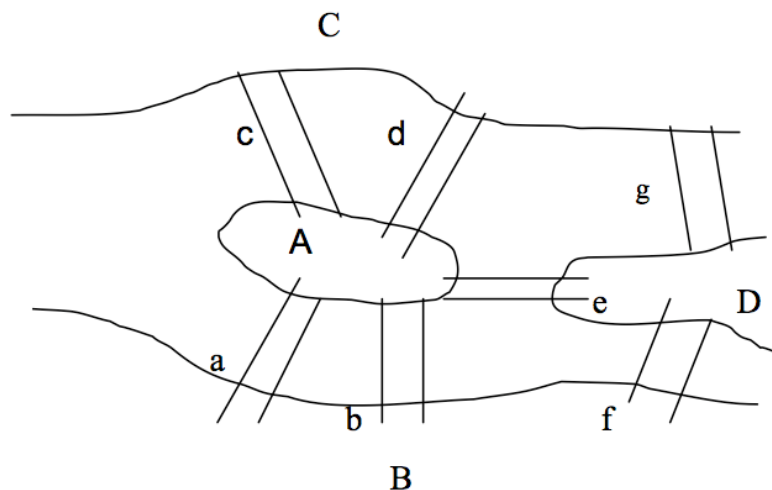
Modelado de problemas típicos

Los grafos pueden representar, como se vio anteriormente, un mapa de carreteras, una figura geométrica, la red de ferrocarriles, el plano de un circuito eléctrico, el esquema de la red telefónica de una compañía, etc.

Ejemplo:

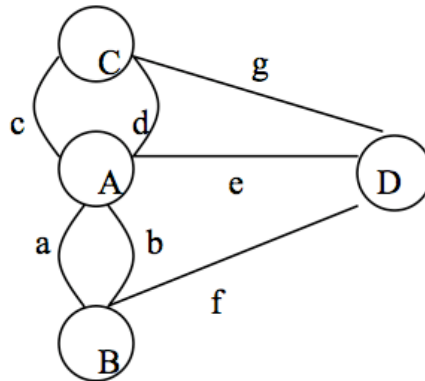
El problema de los puentes de Königsberg (Leonhard Euler)

En la ciudad de Königsberg (antes en Prusia Oriental; llamada ahora Kaliningrado), el río Pregel fluye alrededor de la isla de Kneiphof y después se divide en dos partes dejando 4 zonas de tierra etiquetadas de A a D. Se trata de determinar si comenzando en una de las zonas de tierra y pasando por cada uno de los puentes una sola vez, es posible llegar al punto de partida.



Euler resolvió el problema presentando las zonas de tierra como vértices y los puentes como aristas de un grafo. La solución fue negativa:

Definiendo el grado de un vértice como el número de aristas que inciden en él, Euler demostró que hay un camino que comienza en un vértice, pasa por todas las aristas una sola vez y vuelve al punto de partida si y sólo si el grado de cada vértice es par (circuito Euleriano).



Aplicaciones:

En México, el sitio web de la Secretaría de comunicaciones y trasportes

Sección: ruta punto a punto

http://aplicaciones4.sct.gob.mx/sibuac_internet/ControllerUI?action=cmdEscogeRuta

Representación de los grafos

Existen 4 formas:

1. Matriz de adyacencia.
2. Listas de adyacencia.
3. Multilistas de adyacencia(no se verán en este curso).
4. Matriz de incidencia.

4.3.1 Matriz de adyacencia

Es una matriz booleana que representa las conexiones entre pares de vértices. La matriz de adyacencia de un grafo es simétrica. Si un vértice es aislado entonces la correspondiente fila (columna) está compuesta sólo por ceros. Si el grafo es simple entonces la matriz de adyacencia contiene solo ceros y unos (matriz binaria) y la diagonal está compuesta sólo por ceros.

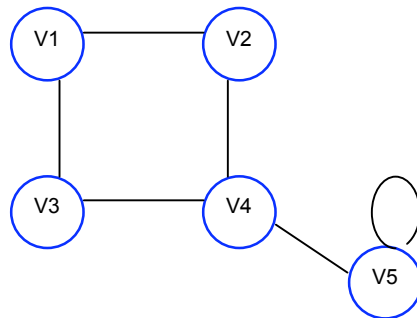
La matriz de adyacencia de un dígrafo no es simétrica. Es una matriz binaria. El número de unos que aparecen en una fila es igual al grado de salida del correspondiente vértice y el número de unos que aparecen en una determinada columna es igual al grado de entrada del correspondiente vértice.

Algoritmo para construir la matriz de adyacencia de un grafo.

- Paso 1: Se le asignan un orden arbitrario a los vértices.
- Paso 2: Se construye una matriz de dimensión $n \times n$, cardinalidad (# vértices) por cardinalidad (# vértices).
- Paso 3: En la posición i -ésima se coloca 1 si el vértice i es adyacente al vértice j ; 0 en el otro caso.

Ejemplo:

Dado el siguiente grafo, construir la matriz de adyacencia (MA)



MA =

	V ₁	V ₂	V ₃	V ₄	V ₅
v ₁	0	1	1	0	0
v ₂	1	0	0	1	0
v ₃	1	0	0	1	0
v ₄	0	1	1	0	1
v ₅	0	0	0	1	1

Se obtiene la valencia de cada uno de los vértices si se suma la fila o columna correspondiente a dicho vértice.

MA =

	V ₁	V ₂	V ₃	V ₄	V ₅	δ
v ₁	0	1	1	0	0	2
v ₂	1	0	0	1	0	2
v ₃	1	0	0	1	0	2
v ₄	0	1	1	0	1	3
v ₅	0	0	0	1	1	2
δ	2	2	2	3	2	

Notas: Con la matriz de adyacencia **NO** se pueden representar los lados paralelos.

Para un grafo no dirigido en la matriz de adyacencia la diagonal principal contiene ceros. Esto puede llegar a aprovecharse para ahorrar tiempo en algunos algoritmos. La representación por medio de matriz se prefiere para algoritmos donde el número de arcos es grande en proporción al número de vértices. Si sucediera lo contrario se prefiere la representación por medio de listas de adyacencia.

$$MA^2 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 2 & 0 \\ 0 & 2 & 2 & 0 & 1 \\ 0 & 2 & 2 & 0 & 1 \\ 2 & 0 & 0 & 3 & 1 \\ 0 & 1 & 1 & 1 & 2 \end{bmatrix}$$

En la diagonal de la matriz cuadrada se obtiene el grado (o la valencia) de los vértices, y los demás elementos de esta matriz cuadrada indican el número de lados de longitud 2 de v_i a v_j , por ejemplo:

$$MA^2[1,4] = 2 \quad (v_1, v_2) (v_2, v_4) \text{ y } (v_1, v_3) (v_3, v_4)$$

$$MA^2[2,3] = 2 \quad (v_2, v_1) (v_1, v_3) \text{ y } (v_2, v_4) (v_4, v_3)$$

$$MA^2[2,5] = 1 \quad (v_2, v_4) (v_4, v_5)$$

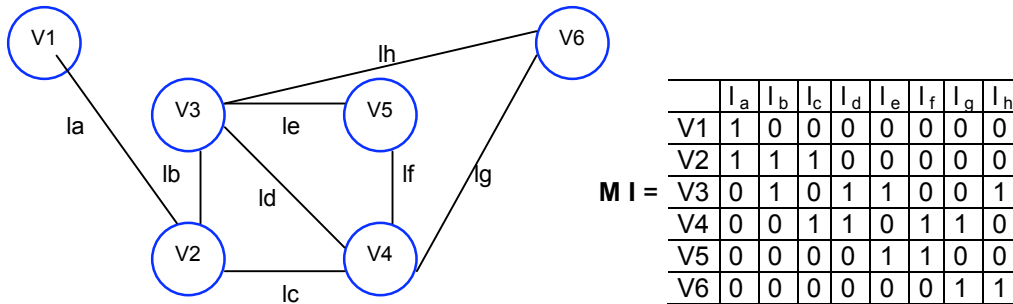
Desventajas

- Poco eficiente si el número de vértices varía a lo largo del tiempo de vida del grafo.
- Puede darse el caso que el número de vértices sea mayor del número inicial.
- Poco eficiente cuando el grafo tiene pocos arcos (la matriz es dispersa).

4.3.2 Matriz de incidencia

La matriz de incidencia sólo contiene ceros y unos (matriz binaria). Como cada arista incide exactamente en dos vértices, cada columna tiene exactamente dos unos. El número de unos que aparece en cada fila es igual al grado del vértice correspondiente.

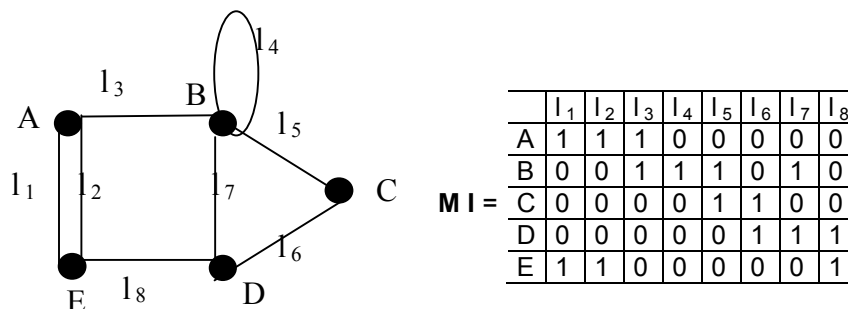
Una fila compuesta sólo por ceros corresponde a un vértice aislado.



La matriz de adyacencia de un dígrafo no es simétrica. Es una matriz binaria. El número de unos que aparecen en una fila es igual al grado de salida del correspondiente vértice y el número de unos que aparecen en una determinada columna es igual al grado de entrada del correspondiente vértice.

Algoritmo para construir la matriz de incidencia de un grafo

- **Paso1:** Se le asigna un orden arbitrario a cada uno de los vértices y lados del grafo.
- **Paso2:** Se construye la matriz de incidencia de dimensión $n*m$, cardinalidad (No. de vértices) por cardinalidad (número de lados o aristas).
- **Paso3:** Se asigna un 1 al elemento i, j si el lado j es incidente al vértice i , 0 en otro caso.



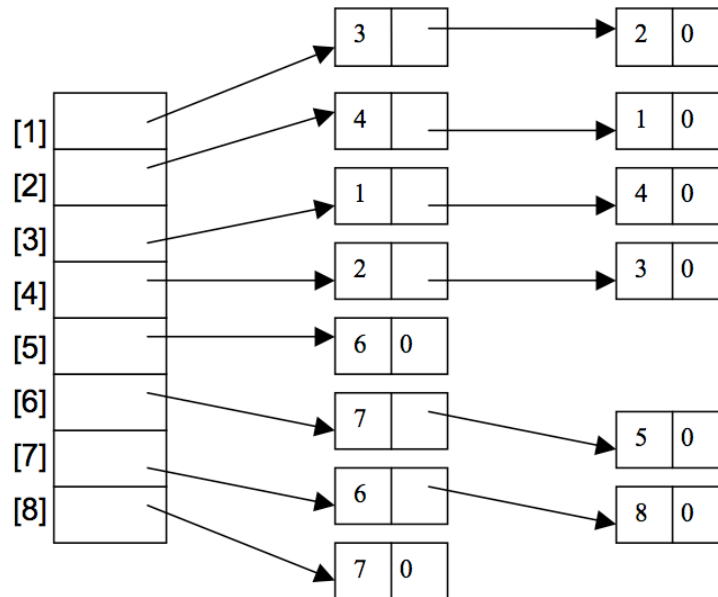
Notas:

- La suma de la fila en una matriz de incidencia representa el número de lados incidentes en ese vértice.
- Si la suma de la columna es uno (1) en una matriz de incidencia entonces ese lado es un lazo.
- La matriz de incidencia nos permite representar grafos con lados paralelos y lazos simultáneamente.

4.4 Lista de adyacencia

Representa una lista de todos los vértices

Cada objeto vértice guarda una lista de adyacencia con un objeto arista para cada vértice alcanzable desde él. Es una lista de listas.



Las **operaciones básicas** para manipulación de los grafos son:

- `arista(u,v)` Añade el arco o arista(u,v) al grafo.
- `aristaPeso(u,v,w)` Para un grafo con pesos, añade el arco(u,v) al grafo y el peso w.
- `borrarArco(u,v)` Elimina del gafo el arco (u,v).
- `adyacente(u,v)` Operación que devuelve 1 si los vértices u,v forman un arco.
- `nuevoVertice(u)` Añade el vértice u al grafo G.
- `borrarVertice(u)` Borra el vértice u al grafo G.
- `numeroNodos()` Devuelve el número de nodos en el grafo G.
- `gradoNodo(u)` Devuelve el número de arcos incidentes en el vértice u.
- `camino(v1,v2)` Para un origen v1 a un destino v2, se listan los vértices que forman esa ruta.

Recorridos en grafos

Los recorridos se aplican de manera similar a los recorridos en árboles, la idea es visitar todos los nodos que lo forman, moviéndose por las aristas, para lo cuál se debe llevar la cuenta de los nodos visitados y no visitados.

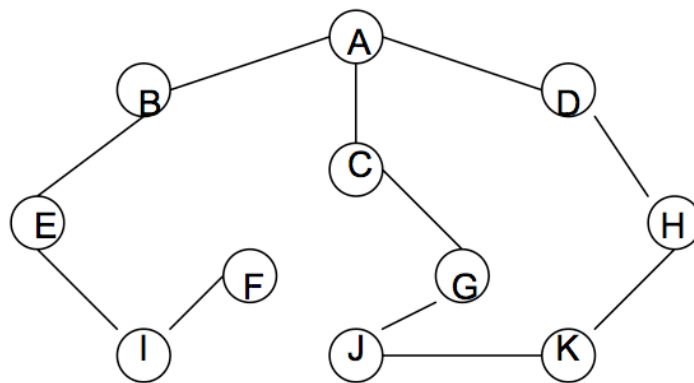
1 DFS (depth first search, búsqueda del primero en altura)

A partir de un nodo dado se localizan sus vértices adyacentes y para cada uno de ellos se inicia nuevamente el recorrido dfs.

Características

- Equivale a un recorrido en preorden de un árbol.
- Se sigue la técnica de la búsqueda en un laberinto, se avanza de sala en sala, marcando cada sala visitada. La búsqueda siempre avanza hacia una sala no marcada internándose “profundamente” en el laberinto sin repetir una sala.
- El conjunto de nodos marcados se tratan con una pila.

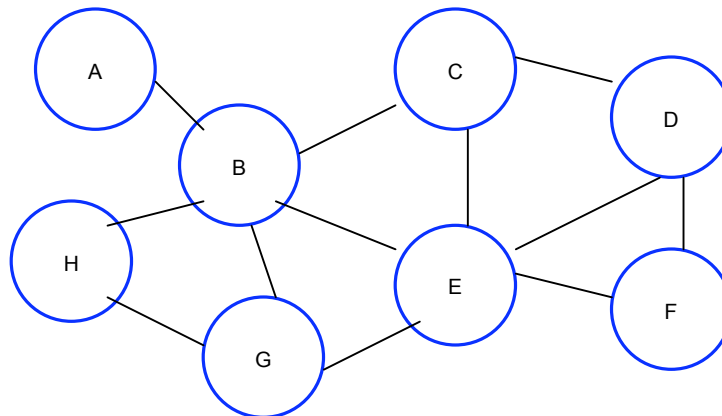
Ejemplo 1:



Recorrido en profundidad: A-B-E-I-F-C-G-J-K-H-D

Recorrido en amplitud: A-B-C-D-E-G-H-I-J-K-F

Ejemplo 2:



Recorrido en profundidad: A-B-C-D-E-F-G-H

Recorrido en amplitud: A-B-C-H-E-D-G-F

Nota. El recorrido no es el único ya que depende del nodo inicial. Y del orden que se elija para procesar los nodos.

2 BFS (breadth first search, búsqueda del primero en anchura)

A diferencia del dfs a partir de nodo dado visita todos sus adyacentes inmediatamente se usa también el vector visitados con el mismo significado que el dfs. Búsqueda en amplitud también es otro nombre para este recorrido.

Características

- Equivale a recorrer un árbol por niveles.
- Es adecuado para solucionar problemas de optimización.
- El conjunto de nodos marcados se trata como una cola.

Algoritmo de recorrido en profundidad

RecProf(Nodo v)

Variables Pila[Nodo] P, Nodo v,u

BorraMarcas

Para v=1,..., n hacer

si marca[y] == noVisitado entonces
 rp(v)

fin si

fin para

rp(Nodo v)

 marca[v] =visitado

 insertarPila(,v,P)

para cada nodo u adyacente a v hacer

si marca[u] == noVisitado entonces
 rp(v)

fin si

fin para

Algoritmo de recorrido en amplitud

Recorrido en Anchura

 BorrarMarcas // marcas del vector o lista , que indican si se paso o no por un vértice.

 Para v = 1,... n hacer

 Si (marca[v]== noVisitado) entonces RecAn(v) // noVisitado es una constante con valor 0

RecAn(Nodo v)

Variables Cola[Nodo] C, Nodo: x,y

 Marca[v]= visitado // se marca el vértice de inicio

 insertaCola(v,C) //Mete en cola el vértice de inicio

mientras No VaciaCola(C) hacer

 x =FrenteCola(C)

```
    suprimirCola( C )
  para cada nodo y adyacente a x hacer
    si marca[y] == noVisitado entonces
      marca[y] = visitado
      InsertarCola( C )
    fin si
  fin para
fin mientras
```

Circuito de Euler y de Hamilton

- Un **circuito euleriano** es un ciclo(camino cerrado) que contiene todas las aristas un sola vez y sus vértices extremos coinciden.
- Un grafo euleriano es aquel que admite un circuito euleriano .
- En un grafo euleriano todos los vértices tienen grado par.
- Un grafo conexo es un grafo euleriano sí y sólo si cada vértice tiene grado par.
- Un grafo conexo admite un camino euleriano no cerrado si y sólo si tiene exactamente 2 vértices de grado impar.
- Un camino hamiltoniano es un camino simple que contiene todos los vértices del grafo (sin repetir ninguno).
- Un **ciclo hamiltoniano** es una camino hamiltoniano cerrado
- Un grafo hamiltoniano es un grafo que contiene un ciclo hamiltoniano, y por lo tanto todo grafo completo contiene un ciclo hamiltoniano.

Algoritmo para construir el ciclo de Euler

Paso 1: Verifique que sea conexo y que todos sus vértices tengan valencia par, si es verdadero continúe si no ese grafo no tiene un ciclo de Euler y termina.

Paso 2: Seleccionar un vértice arbitrariamente.

Paso 3: Encontrar un ciclo que nos lleve a ese mismo vértice.

Paso 4: Eliminar este ciclo.

Paso 5: Si existen vértices ir al paso 1, si no continúe

Paso 6: Encadene todos los ciclos para obtener el ciclo de Euler.

Paso 7: Fin.