

Tecnológico de Monterrey.

TC2038. Análisis y diseño de Algoritmos A

M.C. Ramona Fuentes Valdéz

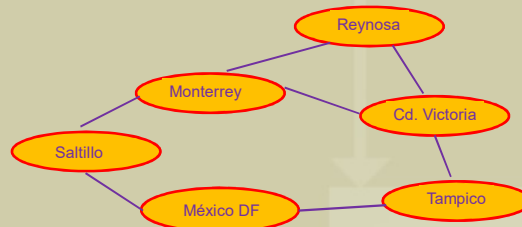
rfuentes@tec.mx

Grafos

Introducción

Es una estructura de tipo RED. En la jerarquía de las estructuras, es el caso más general que existe.

- Un grafo mantiene una relación de “muchos a muchos” (N:M) entre sus elementos.
- Una analogía fácil de entender...
 - La red de carreteras entre las diferentes ciudades de un país.

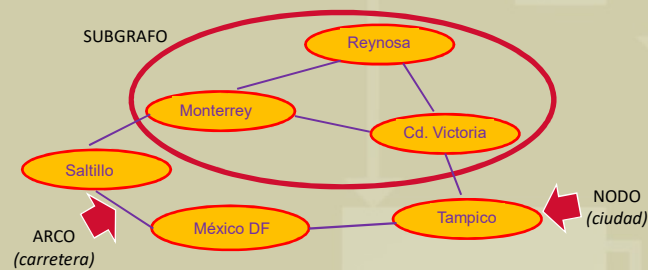


Presentación en canvas Modelo TEC21

Grafos

Terminología

- **Grafo:** Conjunto de Nodos y Arcos (también llamados Vértices y Aristas).
Un grafo es una tupla $G=\langle V, A \rangle$, donde V es un conjunto no vacío de **vértices** o **nodos** y A es un conjunto de **aristas** o **arcos**.
- **Nodo:** Elemento básico de información en un grafo.
- **Arco:** Liga que une dos nodos de un grafo (establece relación entre dos elementos). Cada arista es un par (v, w) donde $v, w \in V$.
- **Subgrafo:** Es un grafo que contiene a un subconjunto de Nodos y Arcos.
Un **subgrafo** $G=(V, A)$ es un grafo $G'=(V', A')$ tal que $V' \subseteq V$ y $A' \subseteq A$.



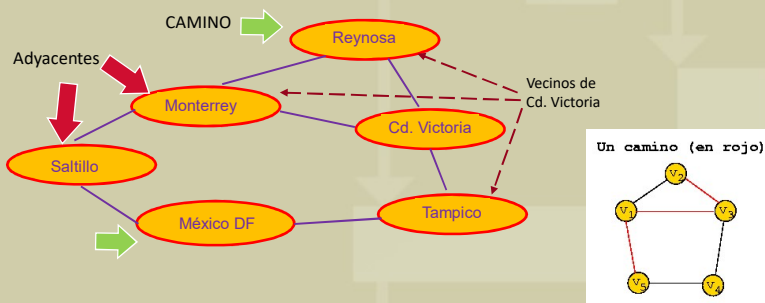
IMITA, Dr. Alberto González

Presentación en canvas Modelo TEC21

Grafos

Terminología

- **Nodos Adyacentes:** Nodos que tienen un arco que los conecta. $(v, w) \in A$
- **Vecinos de un Nodo:** Todos los nodos que son adyacentes al Nodo.
- **Camino o Trayectoria (PATH):** Secuencia de Nodos, de tal forma que cada par de nodos son adyacentes.
Camino de un vértice w_1 a w_q : es una secuencia $w_1, w_2, \dots, w_q \in V$, tal que todas las aristas $(w_1, w_2), (w_2, w_3), \dots, (w_{q-1}, w_q) \in A$.
- **Trayectoria Simple:** Camino donde todos los nodos contenidos son distintos.
Longitud de un camino: número de aristas del camino (*número de nodos-1*).



IMITA, Dr. Alberto González

Presentación en canvas Modelo TEC21

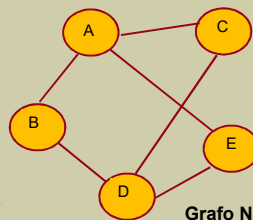
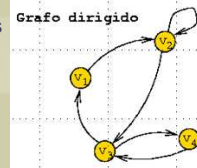
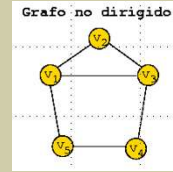
Grafos

Terminología

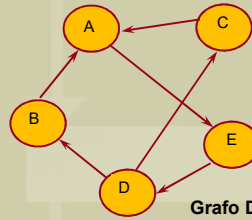
Tipos de grafos

- **Grafo No-Dirigido:**
 - Los arcos en el grafo no tienen ninguna dirección particular, es decir, se consideran bidireccionales.
 - Un arco de A a B es igual que uno de B a A. $(v,w) = (w,v)$
- **Grafo Dirigido:**
 - Los arcos tienen dirección. El primer elemento del arco es denominado el ORIGEN y el segundo el DESTINO.
 - Un arco de A a B es diferente de un arco de B a A.
$$\langle v,w \rangle \neq \langle w,v \rangle$$

$$\langle v,w \rangle \rightarrow v = \text{nodo origen}, w = \text{nodo destino}$$



Grafo No-Dirigido



Grafo Dirigido (Dígrafo)

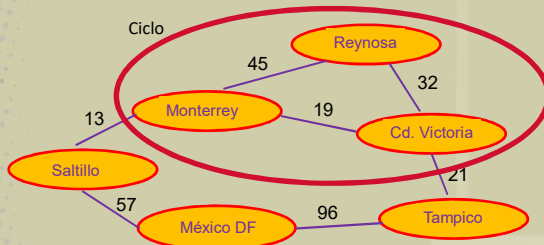
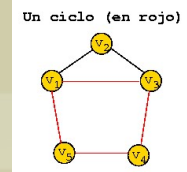
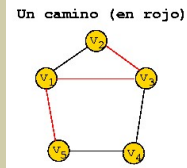
IMITA, Dr. Alberto González

Presentación en canvas Modelo TEC21

Grafos

Terminología

- **Grafo Ponderado:**
 - Los arcos en el grafo tienen asociado un valor (peso).
 - Típicamente relacionado con costo, distancia,...
- **Ciclo:**
 - Es una trayectoria en la que el nodo de inicio y el de terminación son iguales (*en grafos no dirigidos las aristas deben ser diferentes*).



IMITA, Dr. Alberto González

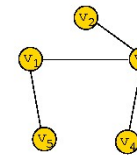
Presentación en canvas Modelo TEC21

Grafos

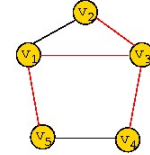
Terminología

- Dados dos vértices v, w se dice que están **conectados** si existe un camino de v a w .
- Un grafo es **conexo (o conectado)** si hay un camino entre cualquier par de vértices. Si es un grafo dirigido, se llama **fuertemente conectado**.
- Se dice que un grafo no dirigido es un **árbol** si es conexo y acíclico.
- Dado un grafo G no dirigido, conexo, se dice que un subgrafo T de G es un **árbol cobertor** si es un árbol y contiene el mismo conjunto de nodos que G .

Un árbol



Un árbol cobertor (en rojo)



Operaciones típicas

- Insertar un Nodo
- Insertar un nuevo Arco
- Borrar un Nodo
- Borrar un Arco
- Buscar un Nodo
- Recorrer el Grafo

Aplicaciones típicas

- ❖ Conectividad, Redes de Transporte
 - ¿Existe un camino entre dos Nodos?
 - ¿Cuál es el costo mínimo de conexión para todos los Nodos?
 - ¿Cuál es la ruta óptima para ir de un Nodo a otro?
- ❖ Autómatas o Diagramas de Estado

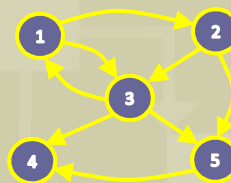
Grafos

Representación de grafos

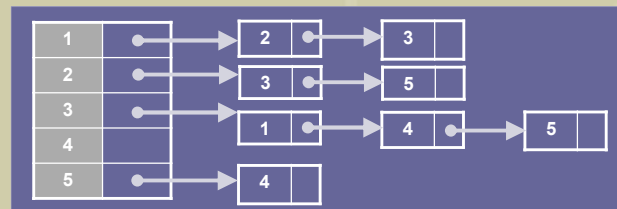
Existen muchas formas de representar a un grafo, las más comunes son:

Matriz de adyacencias

M	1	2	3	4	5
1	0	1	1	0	0
2	0	0	1	0	1
3	1	0	0	1	1
4	0	0	0	0	0
5	0	0	0	1	0



Listas de adyacencia



Listas de arcos

Grafos

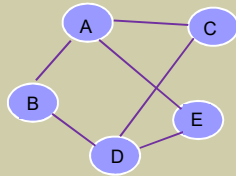
Representación de grafos

Matriz de adyacencias

Es una forma muy común de representación, aún y cuando puede resultar ineficiente.

- Utiliza una matriz estática de valores **booleanos**.
- Cada subíndice de la matriz está asociado a un nodo.
- La casilla $[i,j]$ de la matriz valdrá **true** si existe un arco que conecta al Nodo i con el Nodo j .

$$a_{ij} = \begin{cases} 1 & \text{if } (i, j) \in A \\ 0 & \text{otherwise} \end{cases}$$



M	A	B	C	D	E
A	F	T	T	F	T
B	T	F	F	T	F
C	T	F	F	T	F
D	F	T	T	F	T
E	T	F	F	T	F

- **VENTAJA:**
 - Las operaciones sobre grafos son muy sencillas de implementar.
- **DESVENTAJA:**
 - Se requiere conocer con anticipación la cantidad de elementos que conformarán al grafo.

Grafos

Representación de grafos

Matriz de adyacencias

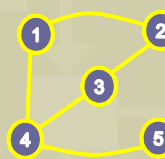
Grafos No Etiquetados

tipo GrafoNoEtiqu = array [1..n, 1..n] de 0..1

Sea M de tipo GrafoNoEtiqu, $G = (V, A)$.

$M[v, w] = \text{cierto} \Leftrightarrow (v, w) \in A$

- Grafo no dirigido \rightarrow matriz simétrica: $M[i, j] = M[j, i]$.
 - **Resultado:** se desperdicia mucho espacio (mitad de la memoria).
- Grafo dirigido \rightarrow se requiere toda la matriz.

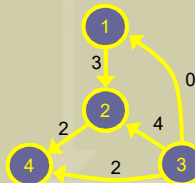


M	1	2	3	4	5
1	0	1	0	1	0
2	1	0	1	0	1
3	0	1	0	1	0
4	1	0	1	0	1
5	0	1	0	1	0

Grafos Etiquetados

tipo GrafoEtiqu[E] = array [1..n, 1..n] de E

- Se asigna el valor (peso) en lugar del valor booleano.
- El tipo E tiene un valor infinito, para el caso de no existir arista.



M	1	2	3	4
1	∞	3	∞	∞
2	∞	∞	∞	2
3	0	4	∞	2
4	∞	∞	∞	∞

Espacio

$A[N][N] - O(N^2)$

Grafos

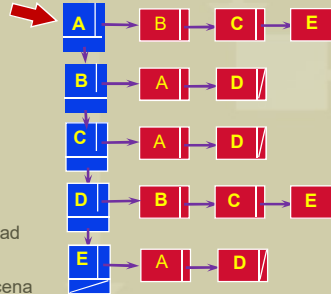
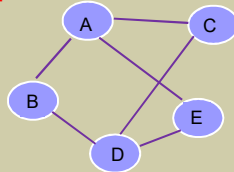
Representación de grafos

Lista de adyacencias

Esta representación dinámica trata de evitar el problema de restricción de espacio.

- Utiliza una lista encadenada de nodos vértice.
- Cada nodo vértice tiene asociada una lista encadenada donde almacena todos los nodos vértice que son adyacentes a él.

Ejemplo:



- Si el grafo es No-Dirigido, existe duplicidad de información.
- Si el grafo es Ponderado, además almacena el peso asociado a cada conexión.
- VENTAJA:
 - No necesita conocer con anterioridad la cantidad de nodos y de arcos que conforman al grafo.
- DESVENTAJA:
 - Duplica información.
 - Requiere más espacio de memoria debido a manejo de apuntadores.

IMITA, Dr. Alberto González

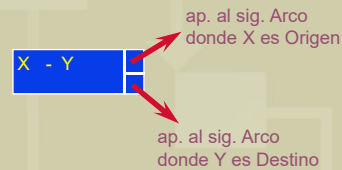
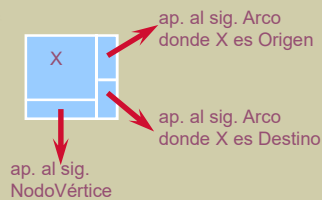
Presentación en canvas Modelo TEC21

Grafos

Representación de grafos

Lista de arcos

- Esta representación es más compleja de implementar.
- Gráficamente los Nodos Vértice y los elementos de la lista de Arcos se verían como:



IMITA, Dr. Alberto González

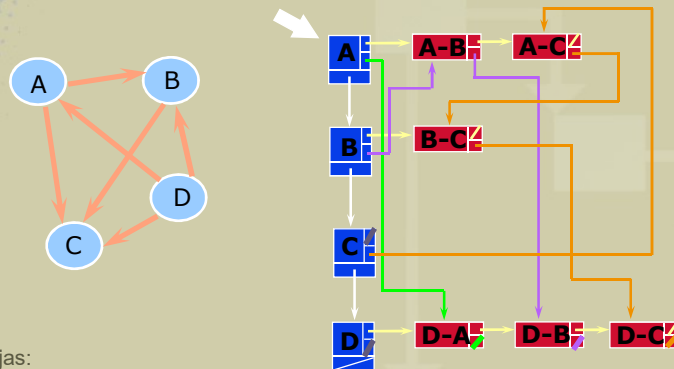
Presentación en canvas Modelo TEC21

Grafos

Representación de grafos

Lista de arcos

Ejemplo:



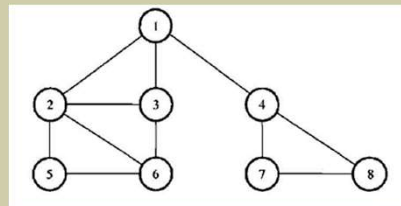
- Ventajas:
 - Es una representación muy eficiente.
- Desventajas:
 - Es bastante compleja de implementar.
 - Requiere más espacio de memoria debido al manejo de apuntadores.

Presentación en canvas Modelo TEC21

Grafos

Recorridos sobre un grafo

- Debido a la gran cantidad de arcos que pueden existir en un Grafo, es necesario plantear un modelo sistemático para poder recorrer "visitar" a todos los Nodos que lo conforman.
- Estos recorridos servirán posteriormente para auxiliar a aplicaciones tales como Análisis de Conectividad, Optimización, etc.
- Los recorridos sobre un grafo más utilizados son:
 - **BFS o Breadth First Search** (Primero en Amplitud)
 - **DFS o Depth First Search** (Primero en Profundidad)



IMTA, Dr. Alberto González

Presentación en canvas Modelo TEC21

Grafos

Recorrido en amplitud (BFS o *Breadth First Search*)

- A partir de un Nodo de Inicio, este algoritmo "visita" un Nodo y luego a todos sus Vecinos.
- El orden en el que visita a los Vecinos depende del "orden" de almacenamiento, según la representación que se esté utilizando.
- Este algoritmo utiliza una FILA auxiliar.
- Para evitar ciclos infinitos, todos los Nodos guardan un "status" (En Espera, Listo, Procesado).

Algoritmo

Inicializar el status de todos los Nodos a "En Espera".

Para cada Nodo del Grafo:

Si el status del Nodo es "En Espera", entonces:

- Insertar el Nodo en la FILA (con status "Listo")
- Mientras la FILA no esté vacía:
 - * Sacar Nodo de la FILA y procesarlo.
 - * Cambiar su status a "Procesado".
 - * Meter a la FILA todos los Vecinos del Nodo que tengan status "En Espera", cambiándolo a "Listo".

- En Espera
- Listo
- Procesado

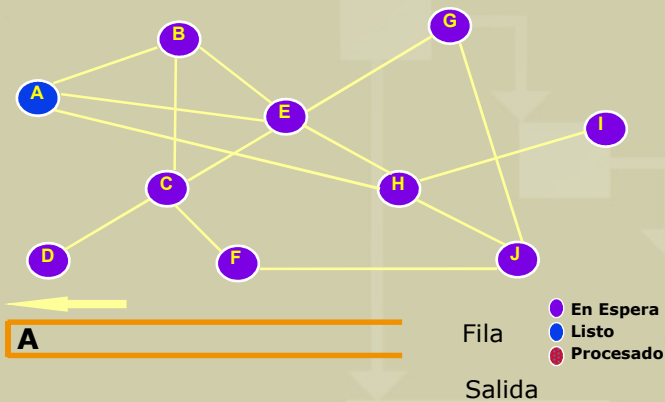
Presentación en canvas Modelo TEC21

Grafos

Recorridos sobre un grafo

Recorrido en amplitud (BFS o *Breadth First Search*)

Ejemplo:



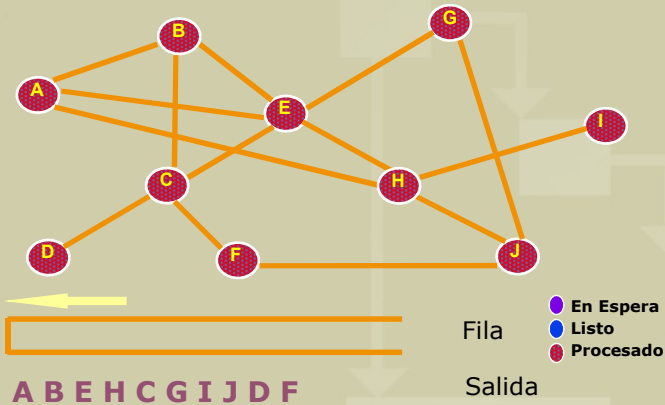
Presentación en canvas Modelo TEC21

Grafos

Recorridos sobre un grafo

Recorrido en amplitud (BFS o *Breadth First Search*)

Ejemplo:



Presentación en canvas Modelo TEC21

Grafos

Recorrido en amplitud (BFS o *Breadth First Search*)

Algoritmo

BFS se implementa con una EDA FIFO tipo Cola.

función BFS(Nodo v)

Q = Cola vacía // EDA tipo COLA

marca[v]=VISITADO

poner v en Q

mientras Q no esté vacía hacer

u = primero(Q)

quitar u de Q

para cada w adyacente a u hacer

si (marca[w]) != VISITADO entonces {

marca[w]=VISITADO

poner w en Q

}

Inicializar el status de todos los Nodos a "En Espera".

Para cada Nodo del Grafo:

Si el status del Nodo es "En Espera", entonces:

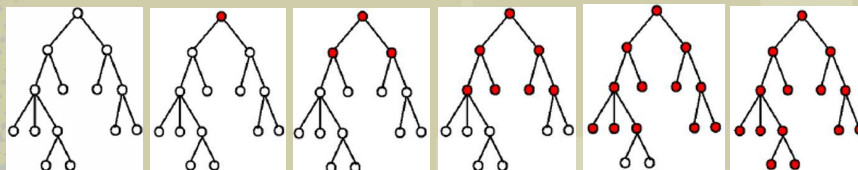
- Insertar el Nodo en la FILA (con status "Listo")

- Mientras la FILA no esté vacía:

* Sacar Nodo de la FILA y procesarlo.

* Cambiar su status a "Procesado".

* Meter a la FILA todos los Vecinos del Nodo que tengan status "En Espera", cambiándolo a "Listo"



IMITA, Dr. Alberto González

Grafos

Recorrido en profundidad (DFS o *Depth First Search*)

- A partir de un Nodo de Inicio, este algoritmo “visita” el Nodo y luego visita recursivamente a todos sus Vecinos NO procesados.
- El orden en el que visita a los Vecinos depende del “orden” de almacenamiento, según la representación que se esté utilizando.
- Este algoritmo utiliza una PILA auxiliar (**para simular la recursividad**).
- Para evitar ciclos infinitos, todos los Nodos guardan un “status” (En Espera, Procesado).

Algoritmo

Inicializar el status de todos los Nodos a “En Espera”.

Para cada Nodo del Grafo:

Si el status del Nodo es “En Espera”, entonces:

- Insertar el Nodo en la PILA.
- Mientras la PILA no esté vacía:
 - * Sacar Nodo de la PILA y procesarlo.
 - * Cambiar su status a “Procesado”.
 - * Meter a la PILA todos los Vecinos del Nodo que NO tengan status “Procesado”.
 - * Si un Nodo ya existía, se deja el último que llegó.

● En Espera
● Procesado

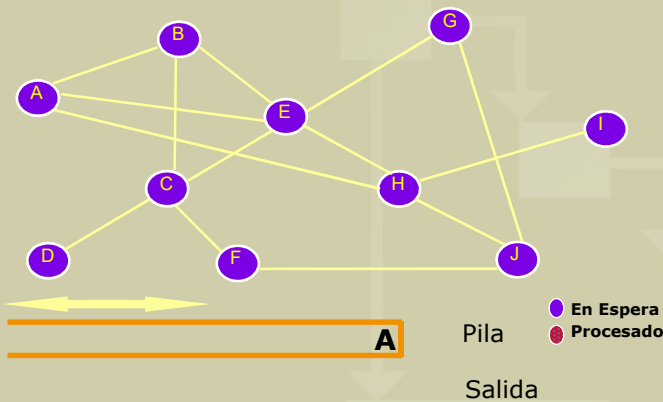
Presentación en canvas Modelo TEC21

Grafos

Recorridos sobre un grafo

Recorrido en profundidad (DFS o *Depth First Search*)

Ejemplo:



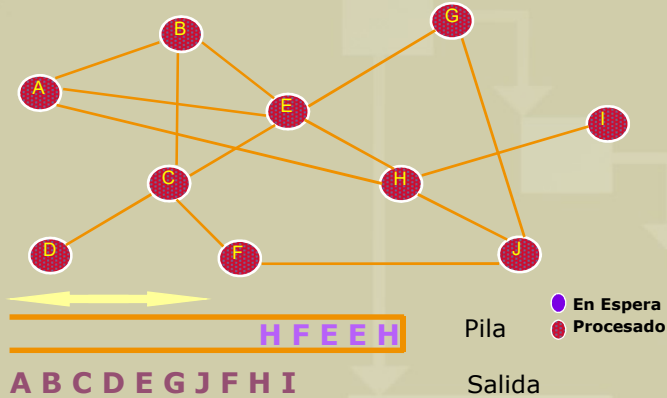
Presentación en canvas Modelo TEC21

Grafos

Recorridos sobre un grafo

Recorrido en profundidad (DFS o Depth First Search)

Ejemplo:



Presentación en canvas Modelo TEC21

Grafos

Recorrido en profundidad (DFS o Depth First Search)

Algoritmo

```
función MainDFS(Grafo G<V,A>)
  para cada v ∈ V hacer marca[v]=NO_VISITADO
  para cada v ∈ V hacer
    si (marca[v] != VISITADO ) entonces
      DFS(v)
```

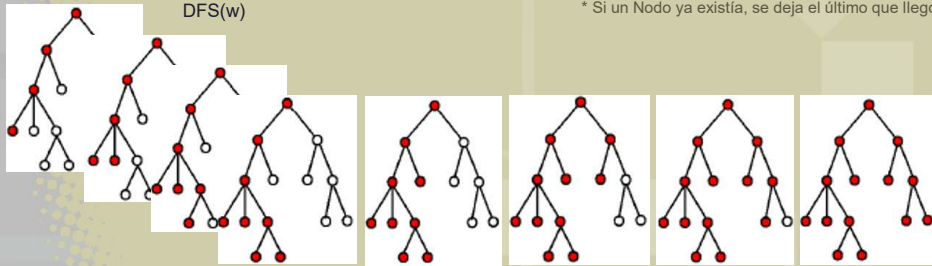
```
función DFS(Nodo v)
  // el nodo v no ha sido visitado
  // anteriormente
  marca[v]=VISITADO
  para cada nodo w adyacente a v hacer
    si (marca[w] != VISITADO ) entonces
      DFS(w)
```

Inicializar el status de todos los Nodos a "En Espera".

Para cada Nodo del Grafo:

Si el status del Nodo es "En Espera", entonces:

- Insertar el Nodo en la PILA.
- Mientras la PILA no esté vacía:
 - * Sacar Nodo de la PILA y procesarlo.
 - * Cambiar su status a "Procesado".
 - * Meter a la PILA todos los Vecinos del Nodo que NO tengan status "Procesado".
 - * Si un Nodo ya existía, se deja el último que llegó.



UMTA, Dr. Alberto González