CLASE GRAFOS

Este material es de uso exclusivo para clase de algoritmos y estructura de datos, la información de este documento fue tomada textualmente de varios libros por lo que está prohibida su impresión y distribución.

- Los grafos es una estructura de datos no lineal y que tiene gran número de aplicaciones.
- El estudio del análisis de grafos ha interesado a los matemáticos durante siglos y representa una parte importante de la teoría combinatoria en matemáticas.
- Para la teoría de grafos existen algoritmos que permiten su solución por computadora.

- Los árboles binarios representan estructuras jerárquicas con limitaciones de dos subárboles por cada nodo.
- Si se eliminan las restricciones de que cada nodo puede apuntar a otros nodos --como máximo- y que cada nodo puede estar apuntado por otro nodo --como máximo- nos encontramos con un grafo.

Definición

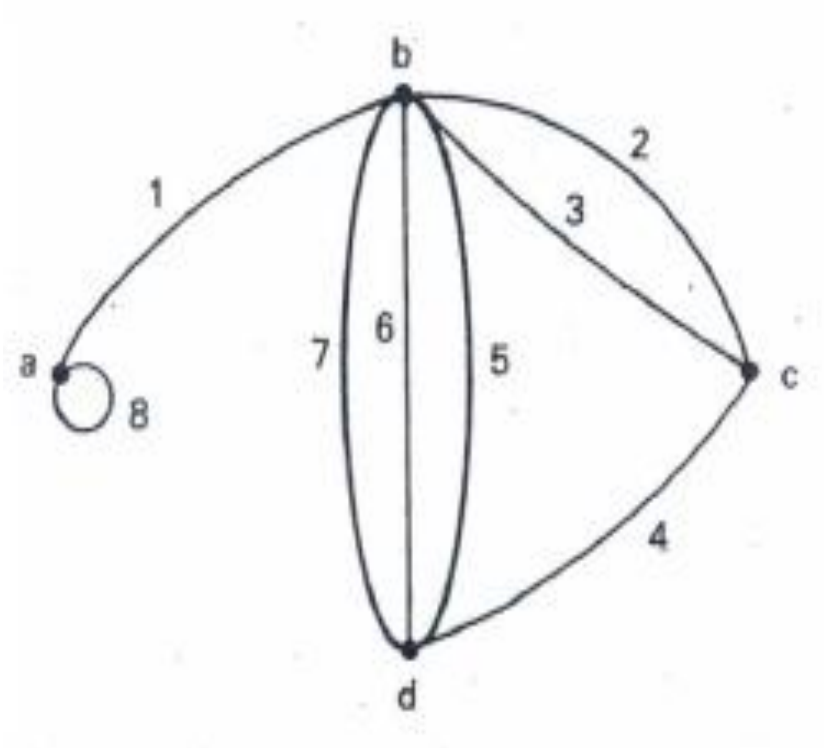
- un *grafo* es un conjunto de puntos(nodos) -una estructura de datos- y un conjunto de líneas, cada una de las cuales une un punto a otro. Los puntos se llaman *nodos* o *verticales* del grafo y las líneas se llaman *aristas* o *arcos* *(edges).*

Se representan el conjunto de vértices de un grafo dado G por V_G y el conjunto de arcos por A_G. Por ejemplo, en el grafo G.

V_G= {a,b,c,d}

y

A_G = {1,2,3,4,5,6,7,8}



Este material es de uso exclusivo para clase de algoritmos y estructura de datos, la información de este documento fue tomada textualmente de varios libros por lo que está prohibida su impresión y distribución.

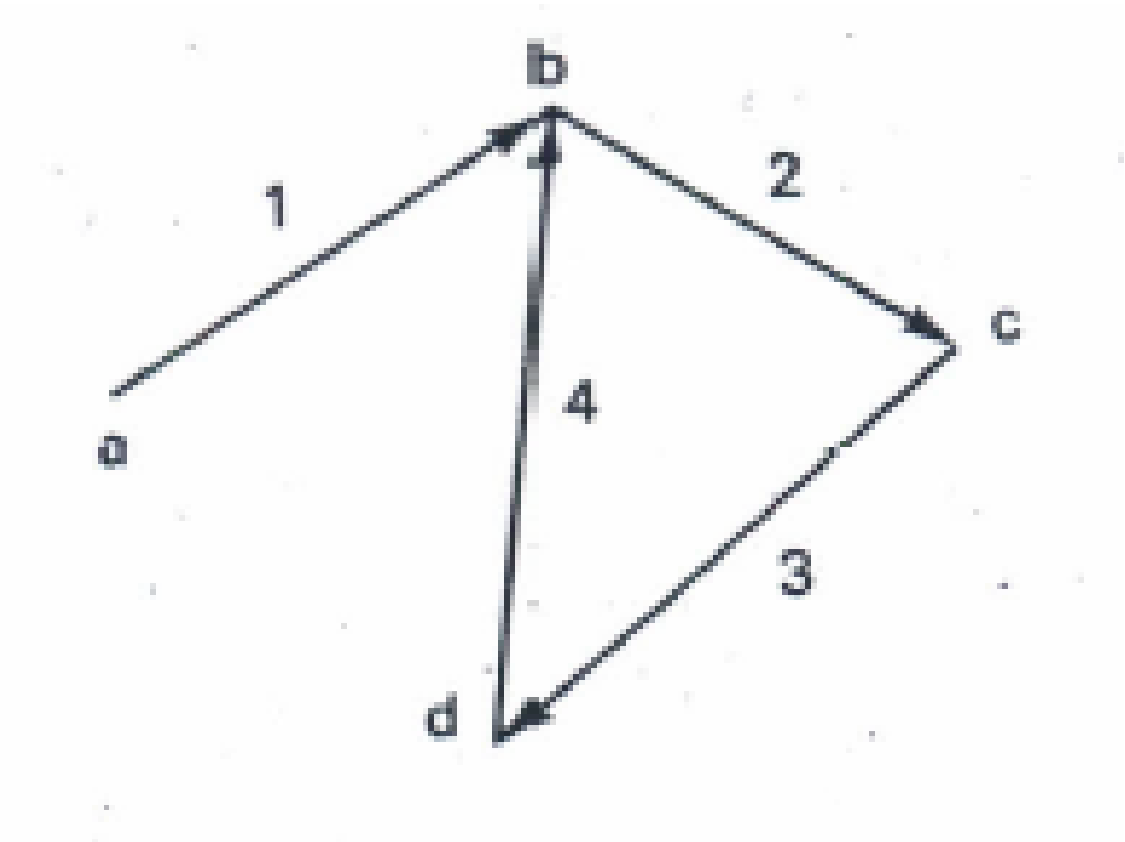
- El número de elementos de V_G se llama *orden* del grafo.
- Un *grafo nulo* es un grafo de orden cero.
- Una arista se representa por los vértices que conecta.

La arista 3 conecta los vértices b y c, y se representa por V(b,c). Algunos vértices pueden conectar un nodo consigo mismo; por ejemplo, el vértice 8 tiene el formato V(a,a). Estas aristas se denominan *bucles* o *lazos.*

Un grafo G se denomina *sencillo* si se cumplen las siguientes condiciones:

1. No tiene lazos, es decir, no existe un arco en A_G de la forma (V,V), donde está en V_G.
2. No existe más que un arco para unir dos nodos, es decir, no existe más qué un arco ($V_{1,}$ V_2 para cualquier par de vértices $V_{1,}$ $V_{2.}$

Grafo sencillo



Este material es de uso exclusivo para clase de algoritmos y estructura de datos, la información de este documento fue tomada textualmente de varios libros por lo que está prohibida su impresión y distribución.

Un grafo que no es sencillo se-denomina *grafo múltiple.*

Un *camino* es una secuencia de uno o más arcos que conectan dos nodos. Representaremos por C ($V_{i,}$ V_j) un camino que conecta los vértices V_1 y V_2.

La *longitud* de un camino es el número de arcos que lo comprenden. En el grafo anterior existen los siguientes caminos entre los nodos *b* y *d.*

C(b,d) = (b,c) (c,d) longitud = 2

C(b,d) = (b,c) (c,b) (b,c) (c,d) longitud = 4

C(b,d) = (b,d) longitud = 1

C(b,d) = (b,d) (c,b) (b,d) longitud = 3

- Dos vértices se dice que son *adyacentes* si hay un arco que los une.
- Sí, V_i y V_j son adyacentes si existe un camino que los une.
- Esta definición es muy general y normalmente se particulariza; si existe un camino desde A a B, decimos que A *es adyacente a* B y B *es adyacente desde* A

existen dos tipos de grafos.

- *dirigidos* los vértices apuntan unos a otros; los arcos están dirigidos o tienen dirección;
- *no-dirigidos* los vértices están relacionados, pero no se apuntan unos a otros; la dirección no es importante

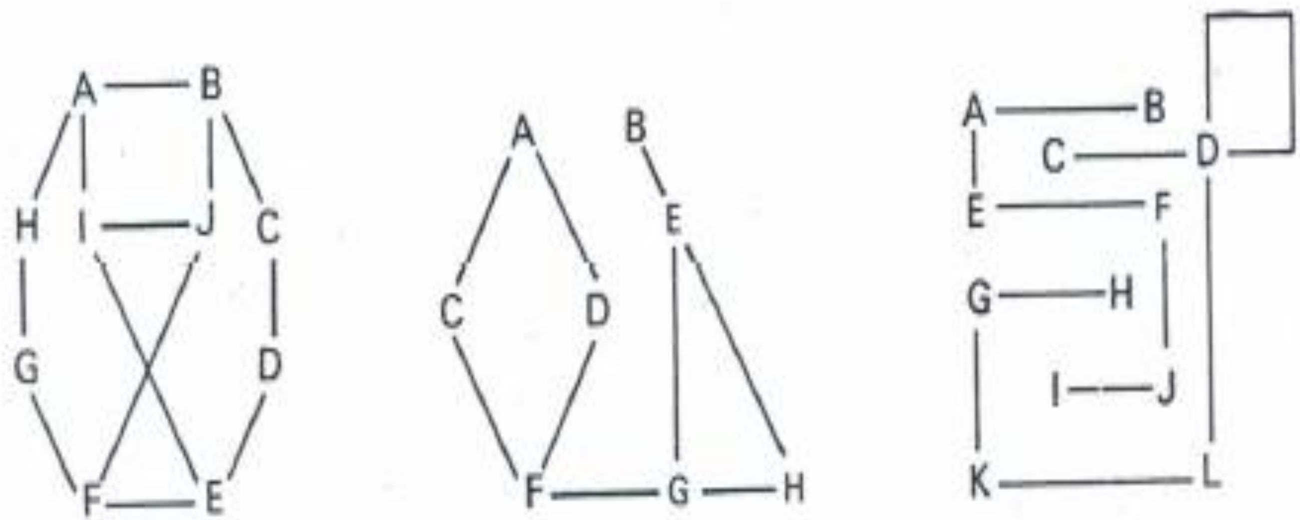
- Grafos Dirigidos y no dirigidos

(a), no dirigido

(b), dirigido

- *grafo conectado* existe siempre un camino que une dos vértices cualesquiera; .
- *grafo desconectado* existen vértices que no están unidos por un camino.

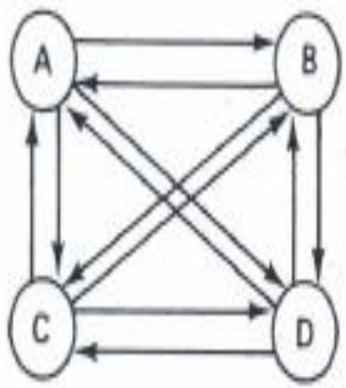
- Grafos conectados y no conectados



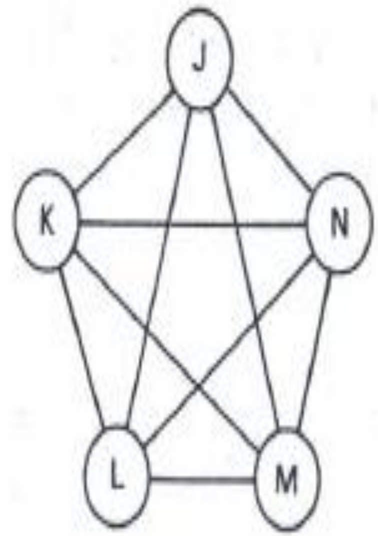
(a), conectado (b), conectado (c), no conectado

- Un *grafo completo* es aquel en que cada vértice está conectado con todos y cada uno de los restantes nodos. Si existen *n* vértices, habrá n(n -1) aristas en un grafo completo y dirigido, y n(n-1)/2 aristas en un grafo no dirigido completo

- Grafos completos



(a), grafo completo dirigido



(b), grafo completo no dirigido

- Un *grafo ponderado o con peso* es aquel en el que cada arista tiene un valor.
- Los grafos con peso suelen ser muy importantes, ya que pueden representar situaciones de gran interés; por ejemplo, los vértices pueden ser ciudades y las aristas distancias o precios del pasaje de ferrocarril o avión entre ambas ciudades.
- Eso nos puede permitir calcular cuál es el recorrido más económico entre dos ciudades, sumando los importes de los boletos de las ciudades existentes en el camino y así poder tomar una decisión acertada respecto al viaje e incluso estudiar el posible cambio de medio de transporte: avión o automóvil, si éstos resultan más económicos.

- La solución de encontrar el camino más corto, el de menor precio o más económico entre dos vértices de un grafo, es un algoritmo importante en la teoría de grafos. (El *algoritmo de Dijkstra* es un algoritmo tipo para la solución de dichos problemas.)

Representación de grafos

- Existen dos técnicas estándar para representar un grafo G: *la matriz de adyacencia* (mediante arreglos)
- y la *lista de adyacencia* (mediante apuntadores/listas enlazadas)

Matriz de adyacencia

- La matriz de adyacencia M es un arreglo de dos dimensiones que representa las conexiones entre pares de verticales.
- Sea un grafo G con un conjunto de nodos V_G y un conjunto de aristas A_G. Supongamos que el grafo es de orden N, donde N > = 1.
- La matriz de adyacencia M se representa por un arreglo n x n, donde:

$$M(i, j) = \begin{cases} 1 \text{ si existe una arista}(V_i, V_j) \text{ en } A_G, V_i \text{ es adyacente a } V_j \\ 0 \text{ en caso contrario} \end{cases}$$

- Las columnas y las filas de la matriz representan los vértices del grafo.
- Sí existe arista desde *i* a *j* (esto es, el vértice *i* es adyacente a *j*), el costo o peso de la arista de *i* a *j* se introduce; si no existe la arista, se introduce un 0; lógicamente, los elementos de la diagonal principal son todos ceros, ya que el costo de la arista *i* a *i* es 0.

Si G es un grafo no dirigido, la matriz es simétrica M (i, j) = M (j, i).

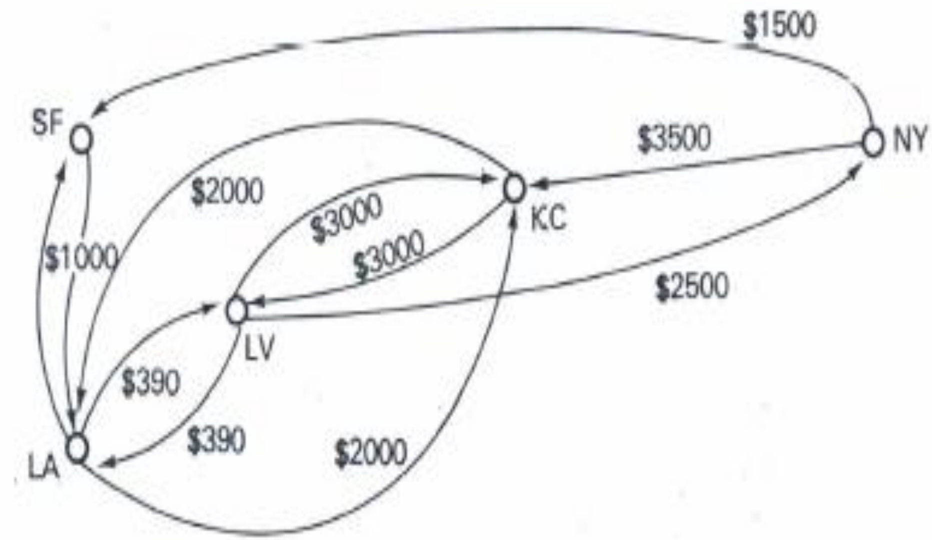
La matriz de adyacencia del grafo

i \ j	1	2	3	4	5	6
1	0	1	0	0	0	0
2	1	1	1	0	0	0
3	0	1	0	1	1	1
4	0	0	1	0	0	0
5	0	0	1	0	0	0
6	0	0	1	0	0	0

- Si el grafo fuese dirigido, su matriz resultante sería:

$i \backslash j$	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	1	1	0	0	0
3	0	0	0	0	1	1
4	0	0	1	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Si se tiene que deducir la matriz de adyacencia del grafo siguiente:



matriz de adyacencia resultante de este grafo, cuyos vértices representan ciudades y los pesos de las aristas, los precios de pasajes de avión en dólares es

	SF	LA	LV	KC	NY
SF		1000			
LA	1000		390	2000	
LV		390		3000	2500
KC		2200	3000		
NY	1500			3500	

Lista de adyacencia

El segundo método utilizado para representar grafos es útil cuando un grafo tiene muchos vértices y pocas aristas; es la *lista de adyacencia.*

Para representación se usa una lista enlazada por cada vértice *v* del grafo que tenga vértices adyacentes *desde* él.

El grafo completo incluye dos partes:

- un directorio
- un conjunto de listas enlazadas

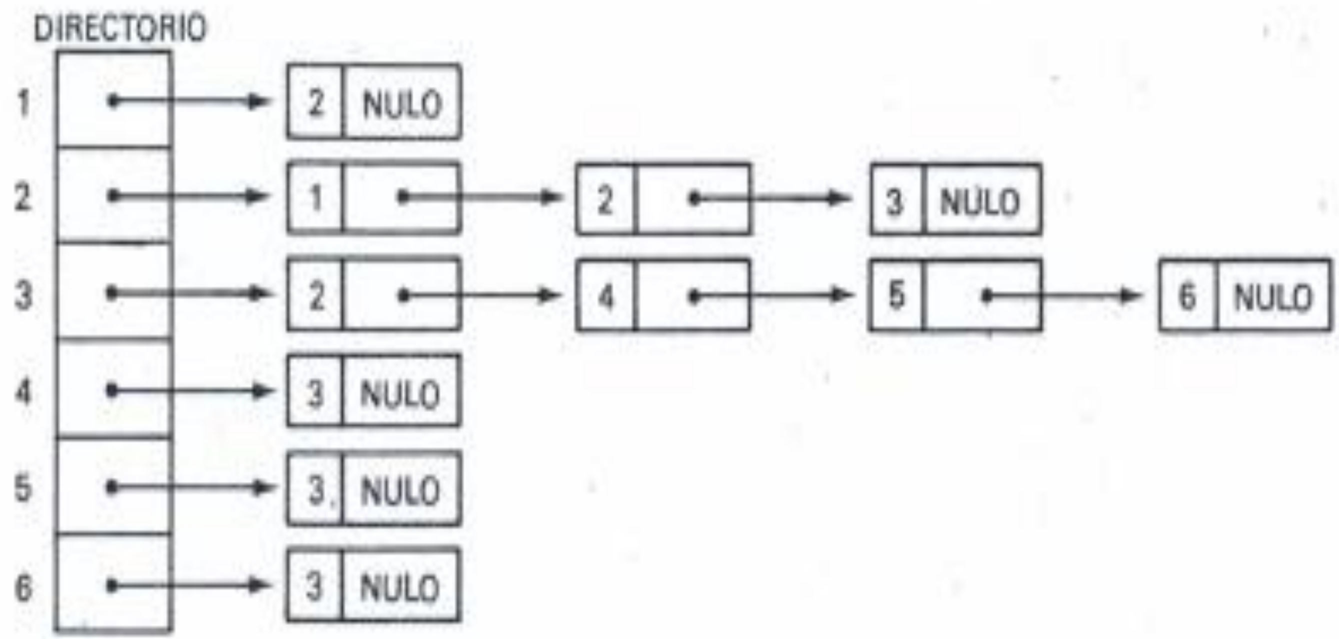
- Hay una entrada en el directorio por cada nodo del grafo.
- La entrada en el directorio del nodo *i* apunta a una lista enlazada que representa los nodos que son conectados al nodo *i.*

Cada registro de la lista enlazada tiene dos campos:

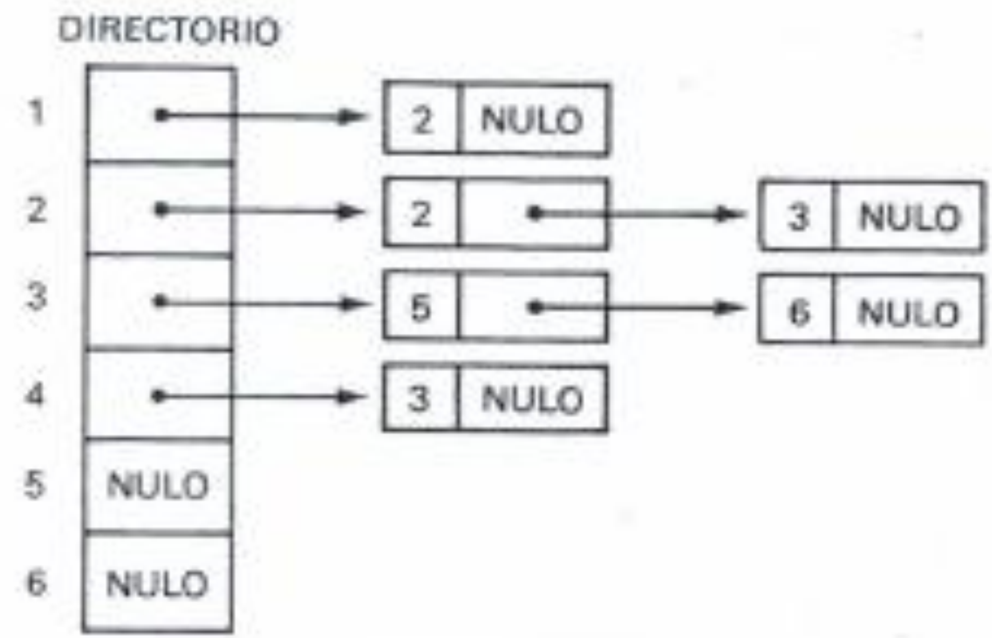
- ❖ un identificador de modo.
- ❖ un enlace al siguiente elemento de la lista

La lista enlazada representa arcos.

- la lista de adyacencia del grafo

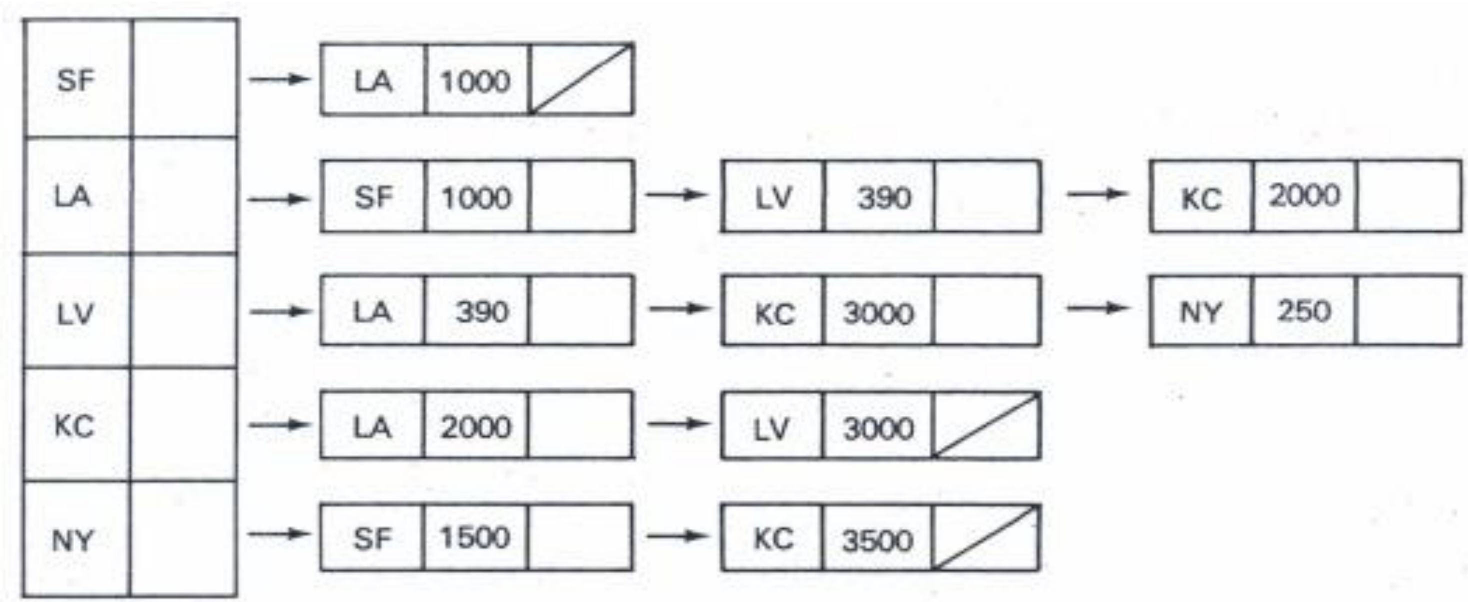


Un grafo no dirigido de orden N con A arcos requiere N entradas en el directorio y 2 * A entradas de listas enlazadas, excepto si existen ciclos que reducen el número de listas enlazadas en 1



Un grafo dirigido de orden N con A arcos requiere N entradas en el directorio y A entradas de listas enlazadas.

Por ejemplo en el ejercicio anterior la lista de adyacencia del grato seria:

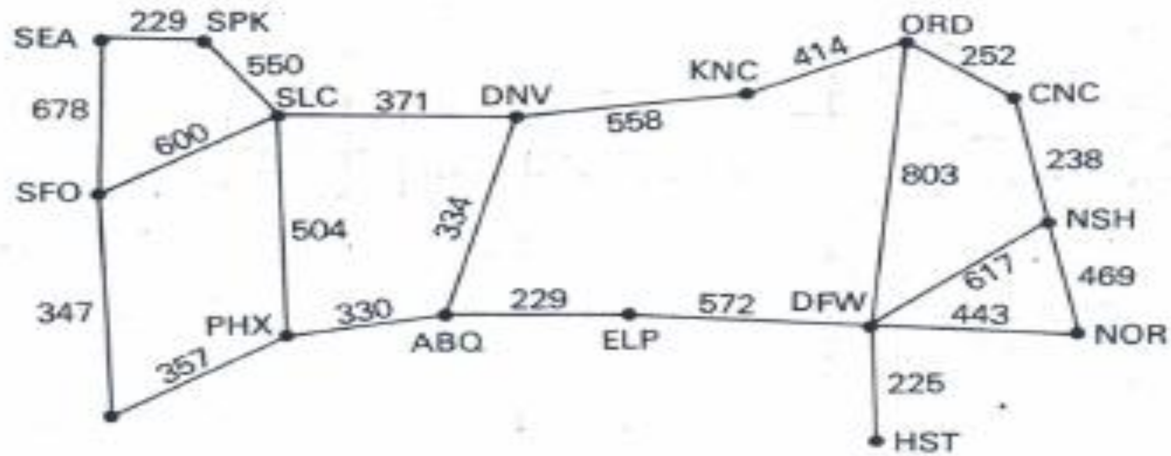


- La elección de la representación depende del algoritmo particular que se quiere implementar y si el grafo es "disperso" o "denso".
- Un grafo disperso es en el que el número de vértices N es mucho mayor que el número de arcos.
- En un grafo denso el número de arcos se acerca al máximo.

Ejercicio

Sea un grafo con aristas ponderadas. Los vértices representan ciudades y las aristas las rutas utilizadas por los camiones de una empresa de transporte de mercancías. Cada arista está rotulada con la distancia entre las parejas de ciudades enlazadas directamente

En este caso utilizaremos una matriz triangular



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1		0	0	334	229	0	0	0	0	0	0	330	0	0	0	0
2			0	0	0	0	0	0	238	0	252	0	0	0	0	0
3				0	572	225	0	0	617	443	803	0	0	0	0	0
4					0	0	558	0	0	0	0	0	0	0	371	0
5						0	0	0	0	0	0	0	0	0	0	0
6							0	0	0	0	0	0	0	0	0	0
7								0	0	0	414	0	0	0	0	0
8									0	0	0	357	0	347	0	0
9										469	0	0	0	0	0	0
10											0	0	0	0	0	0
11												0	0	0	0	0
12												0	0	0	0	0
13													0	0	504	0
14														678	0	229
15															600	0
16																550

1 = ABQ
2 = CNC
3 = DFW
4 = DNV
5 = ELP
6 = HST
7 = KNC
8 = LAX
9 = NSH
10 = NOR
11 = ORD
12 = PHX
13 = SEA
14 = SFO
15 = SLC
16 = SPK

NOTA: Obsérvese como consecuencia de este ejemplo que las aristas ponderadas tienen una gran aplicación.

En transporte comúnmente representan distancias, precios de boleto, tiempos.

En hidráulica, capacidades.

Por ejemplo, el caudal de un oleoducto entre diferentes ciudades litros/segundo.