UNIDAD II. ESTRUCTURA DE DATOS AVANZADAS (GRAFOS)

Francisco J. Hernández López

fcoj23@cimat.mx

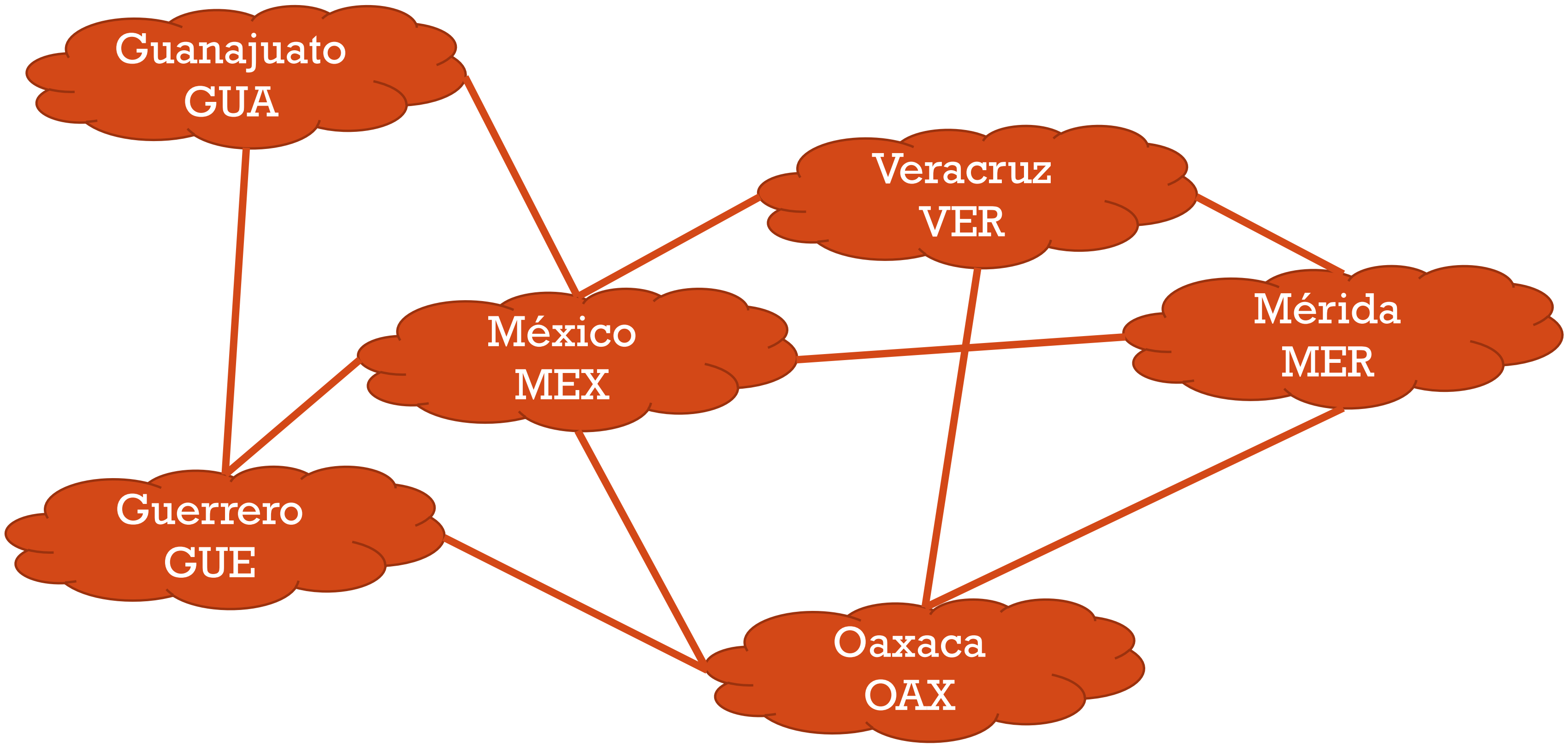
GRAFOS

- Estructura de datos no lineales donde cada componente o nodo puede tener uno o más predecesores (a diferencia de los árboles) y sucesores

- Un grafo esta formado por dos elementos:
  - Vértices (nodos, elementos)→Almacenan información
  - Aristas (bordes, arcos, enlaces)→Relaciones entre la información de los vértices

Estructura de datos, Cairó - Guardati, 3a. Edición, 2006.
Algorithms (4th Edition), Robert Sedgewick (Author), Kevin Wayne, Pearson Education Inc., 2011.

EJEMPLO DE UN GRAFO



DEFINICIONES

- Un grafo G tiene dos conjuntos:
  - $V(G)$ → Conjunto de Vértices
  - $A(G)$ → Conjunto de Aristas

- $G = (V, A)$ → Denota un grafo

- $a = (u, v)$→Arista que va del vértice u al v

- $grado(v)$ → Grado de un vértice: Número de aristas que contienen a v

- $a = (u, u)$→ Lazo o bucle: Una arista que conecta a un vértice consigo mismo

DEFINICIONES

- $P = (v_1, \dots, v_n)$ → Camino P de longitud n: Secuencia de n vértices que se debe seguir para llegar del vértice v_1 al vértice v_n
  - $v_1 = v_n$ → Camino cerrado
  - Si todos los vértices son distintos, con excepción del primero y el último (que pueden ser iguales) → Camino simple
  - Un camino simple cerrado de longitud $\mathrm{k} \geq 3$ → Ciclo o k-ciclo
- Grafo conexo: Si existe un camino simple entre cualesquiera dos de sus vértices
- Grafo árbol: Si G es un grafo conexo sin ciclos

DEFINICIONES

- Grafo completo: Si cada vértice v de G es adyacente a todos los demás vértices de G. Un grafo completo de n vértices tiene:

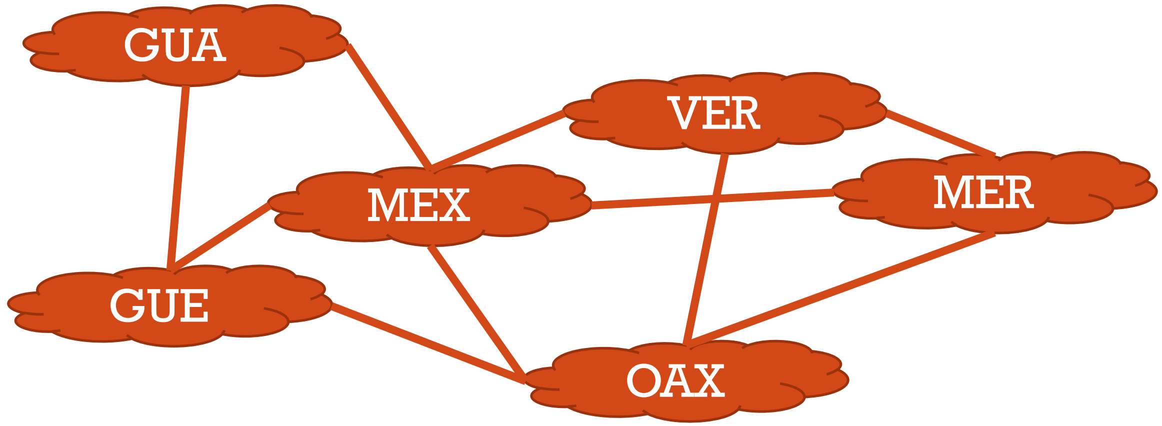
$$\frac{n(n-1)}{2} \text{ aristas}$$

- Grafo etiquetado: Si las aristas de G tienen asignado algún valor numérico no negativo $c(a)$, llamado costo, peso o longitud de a. Entonces cada camino P del grafo tendrá asociado un peso o longitud que será la suma de los pesos de las aristas que forman dicho camino

- Multígrafo: Si al menos dos de sus vértices están conectados entre sí por medio de dos aristas (aristas múltiples o paralelas)

- Subgrafo: Dado el grafo $G = (V, A)$, entonces $G' = (V', A')$ es un subgrafo de G si $V' \neq \emptyset$, $V' \subseteq V$ y $A' \subseteq A$, donde cada arista de A' es incidente (o conecta) con vértices de V'

EJEMPLO



$grado(MER) = 3$
$grado(MEX) = 5$

Un camino para llegar del vértice GUA al vértice MER puede ser:
$Q = (GUA, MEX, MER)$
$P = (GUA, MEX, OAX, MER)$

Camino simple: (GUA, MEX, VER)

Camino simple: $(GUA, MEX, OAX, GUE, GUA)$ → también es un camino cerrado

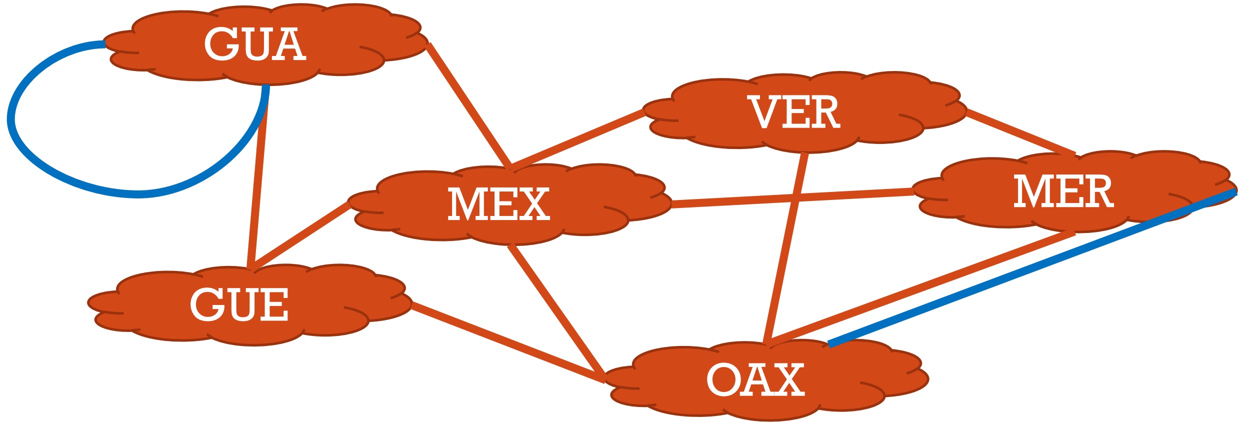
Camino cerrado: (GUA, MEX, GUE, GUA) → ciclo o 4-ciclo

Grafo conexo → Todos sus vértices tienen al menos un camino simple a otro vértice

Grafo no árbol → Es conexo, pero puede haber ciclos

Grafo no completo → numero de aristas $= 10 \neq n(n-1)/2$

EJEMPLO



Lazo o bucle: $a = (GUA, GUA)$

Multígrafo: ya que hay dos aristas que unen los vértices OAX y MER, las cuales se llaman aristas múltiples o aristas paralelas

TIPOS DE GRAFOS

- Grafo dirigido (también llamado dígrafo)
  - Cada arista está asociada a un par ordenado (u, v) de vértices
  - Una arista dirigida se le llama arco y se expresa como: $u \to v$

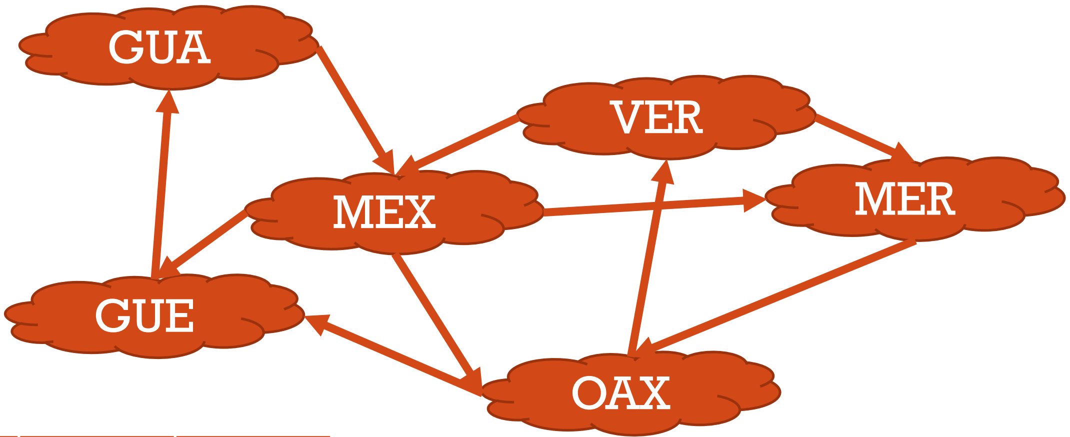
- Grafo no dirigido
  - Sus aristas son pares no ordenados de vértices, el camino del vértice u al vértice v es el mismo que de v a u
  - Se utilizan para modelar relaciones simétricas entre diferentes objetos, por ejemplo:

  El costo de un boleto para ir de MEX a MER será el mismo que de MER a MEX.

REPRESENTACIÓN DE UN GRAFO DIRIGIDO

- Matriz de adyacencia

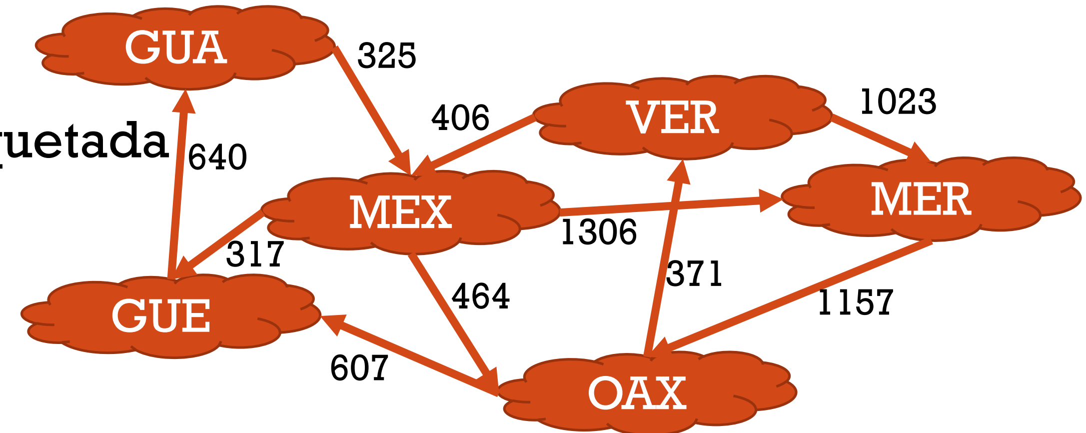
$$M(i,j) = \begin{cases} 1 & si\ \exists\ arco\ entre\ (i,j) \\ 0 & otro\ caso \end{cases}$$
$$con\ 1 \leq i \leq n\ y\ 1 \leq j \leq n$$



	GUA	GUE	MEX	OAX	VER	MER
GUA	0	0	1	0	0	0
GUE	1	0	0	0	0	0
MEX	0	1	0	1	0	1
OAX	0	1	0	0	1	0
VER	0	0	1	0	0	1
MER	0	0	0	1	0	0

REPRESENTACIÓN DE UN GRAFO DIRIGIDO

- Matriz de adyacencia etiquetada



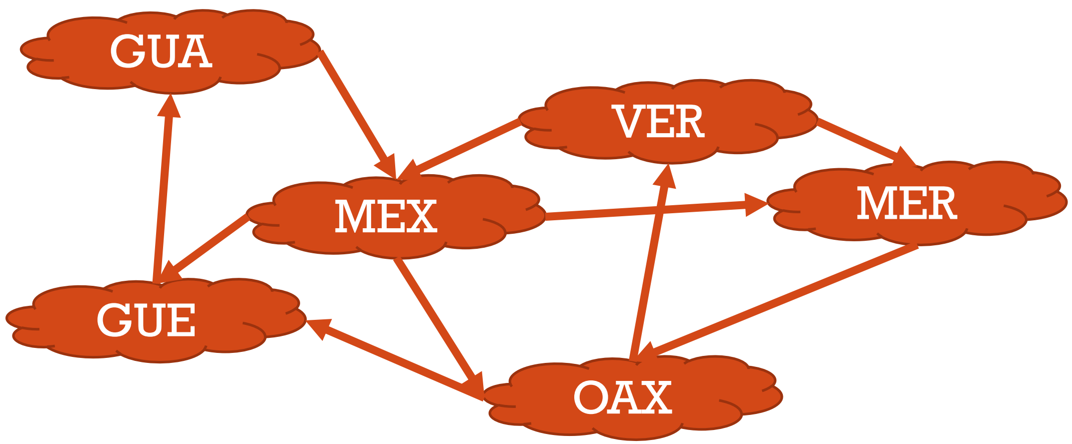
$$M(i,j) = \begin{cases} c_{ij} & si\ \exists\ arco\ entre\ (i,j) \\ 0 & otro\ caso \end{cases}$$
$$con\ 1 \leq i \leq n\ y\ 1 \leq j \leq n$$

	GUA	GUE	MEX	OAX	VER	MER
GUA	0	0	325	0	0	0
GUE	640	0	0	0	0	0
MEX	0	317	0	464	0	1306
OAX	0	607	0	0	371	0
VER	0	0	406	0	0	1023
MER	0	0	0	1157	0	0

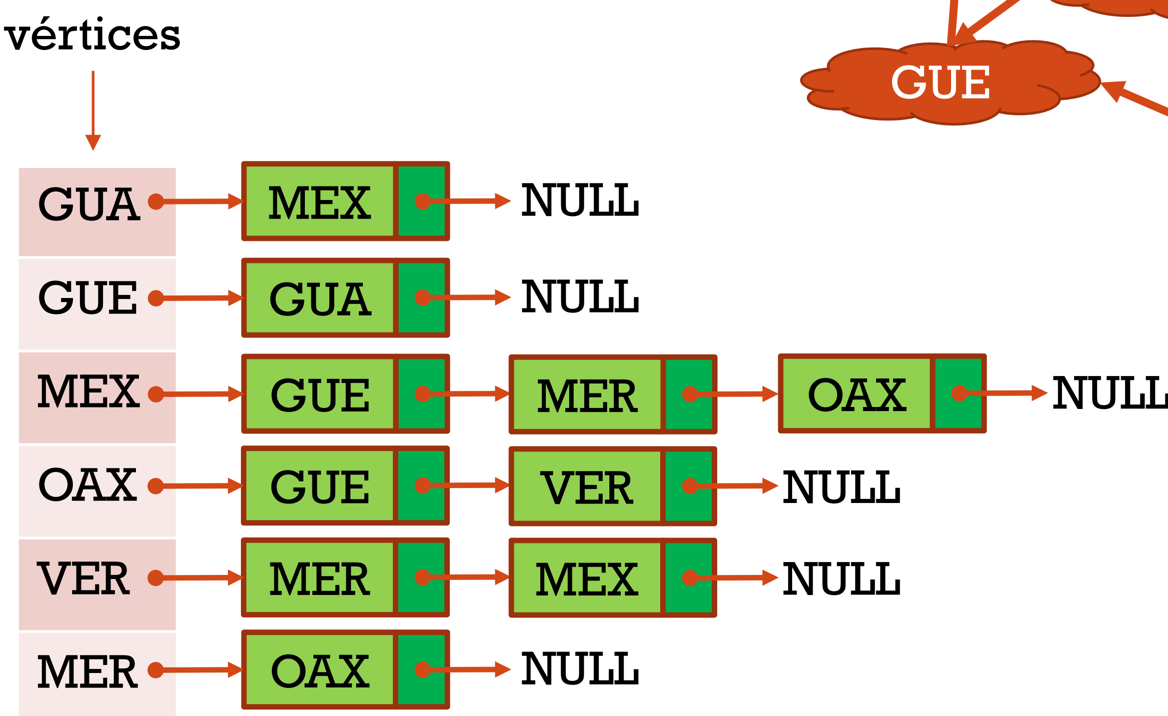
Ventaja:
Eficiente en acceder a cada elemento, no depende del tamaño de V y A.

Desventaja:
Espacio de almacenamiento de n^2, independientemente del número de aristas.

REPRESENTACIÓN DE UN GRAFO DIRIGIDO



- Lista de adyacencia



Ventaja:
Ahorra espacio de almacenamiento si el número de aristas es menor a n^2.

Desventaja:
Tiempo de búsqueda de una arista es de orden: $O(n)$.

ALGORITMOS PARA ENCONTRAR CAMINOS EN UN GRAFO DIRIGIDO

- Algoritmo de Dijkstra (1959): Encuentra el camino más corto de un vértice elegido a cualquier otro vértice del grafo.

- Algoritmo de Floyd (1962): Encuentra el camino más corto entre todos los vértices del grafo.

- Algoritmo de Warshall (1962): Encuentra si es posible, un camino entre cada uno de los vértices del grafo dirigido.

ALGORITMO DE DIJKSTRA

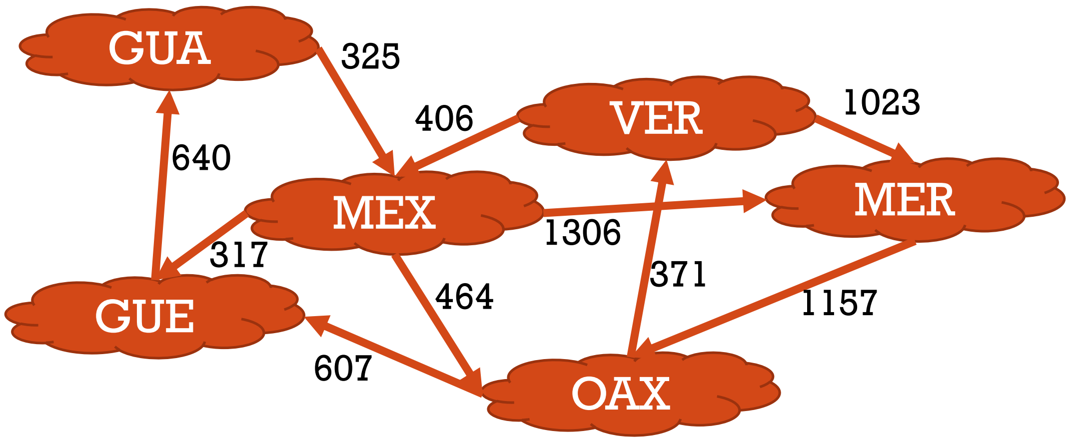
- Encuentra el camino más corto de un vértice a cualquier otro dentro de un grafo dirigido o no dirigido
- Consideraciones:
  - S es un arreglo formado por los vértices con distancia mínima entre ellos. Inicialmente solo tiene el vértice origen
  - D es un arreglo formado por la distancia (o costo) entre el vértice origen y los demás. $D[i]$→Almacena la menor distancia entre el origen y el vértice i
  - M es una matriz de distancias de $n \times n$ elementos, tal que $M[i,j]$ almacena la distancia entre los vértices i y j, si entre ambos existe una arista. En caso contrario $M[i,j]$ será un valor muy grande (∞)
  - Al terminar el algoritmo, D contendrá la distancia mínima entre el origen y cada uno de los otros vértices del grafo

ALGORITMO DE DIJKSTRA

1. Agregar el vértice inicial v_1 a S y calcular $D[k]\ \forall\ k = 2, .. N$
2. Repetir con i desde 2 hasta N
   1. Elegir un vértice $v \in (V - S)$ tal que $D[v]$ sea el mínimo valor
   2. Agregar v a S
   3. Repetir para cada vértice $w \in (V - S)$
      1. Hacer $D[w] \leftarrow minimo(D[w], D[v] + M[v, w])$
   4. Fin de ciclo
3. Fin del ciclo i

Si se utiliza:
Matriz de adyacencia → $O(N^2)$
Lista de adyacencia → $O(\log N)$

EJEMPLO



Encontrar el camino más corto para ir de MER a los otros estados:

Matriz de distancias (Km) M:

		1	2	3	4	5	6
		MER	**OAX**	**VER**	**MEX**	**GUE**	**GUA**
1	**MER**	0	1157	∞	∞	∞	∞
2	**OAX**	∞	0	371	∞	607	∞
3	**VER**	1023	∞	0	406	∞	∞
4	**MEX**	1306	464	∞	0	317	∞
5	**GUE**	∞	∞	∞	∞	0	640
6	**GUA**	∞	∞	∞	325	∞	0

ALGORITMO DE FLOYD

- Encuentra el camino más corto entre todos los vértices del grafo dirigido

- Consideraciones:
  - M → Matriz de distancias, si no existe un camino entre i y j entonces tendrá un valor muy grande (∞), y si $i = j$ entonces tendrá un valor cero (0)
  - En la k-ésima iteración $M[i,j]$ tendrá el camino de menor costo para llegar de i a j pasando por un número de vértices menor a k:

$$M_k[i,j] = \min\{M_{k-1}[i,j]\,, M_{k-1}[i,k] + M_{k-1}[k,j]\}$$

ALGORITMO DE FLOYD

1. Repetir con k desde 1 hasta N
   1. Repetir con i desde 1 hasta N
      1. Repetir con j desde 1 hasta N
         1. Si $\left(M_{ik} + M_{kj} < M_{ij}\right)$ entonces
            1. Hacer $M_{ij} \leftarrow M_{ik} + M_{kj}$
         2. Fin si
      2. Fin de ciclo j
   2. Fin de ciclo i
2. Fin de ciclo k

Orden: $O(N^3)$

Además podemos guardar la trayectoria:

$$T_{ij} \leftarrow k$$

ALGORITMO DE WARSHALL

- Encuentra si es posible, un camino entre cada uno de los vértices del grafo dirigido.
- La solución no presenta las distancias entre los vértices, solo muestra si hay o no camino entre ellos
- Consideraciones:
  - Sea M la matriz de adyacencia, entonces:
    - $M[i,j] = 1$ si hay un arco de i a j
    - $M[i,j] = 0$ si no hay arco de i a j
  - Sea C la matriz de cerradura transitiva de M tal que:
    - $C[i,j] = 1$ si hay un camino de longitud mayor o igual que 1 de i a j
    - $C[i,j] = 0$ en otro caso

ALGORITMO DE WARSHALL

1. Repetir con k desde 1 hasta N
    1. Repetir con i desde 1 hasta N
        1. Repetir con j desde 1 hasta N
            1. Si $(C[i,j] = 0)$ entonces
                1. Hacer $C[i,j] \leftarrow C[i,k]\ AND\ C[k,j]$
            2. Fin si
        2. Fin de ciclo j
    2. Fin de ciclo i
2. Fin de ciclo k

Orden: $O(N^3)$