



Sistema de Información Geográfica Avanzado

**Ingeniero Agrimensor
Tecnólogo en Cartografía**

Introducción (I)

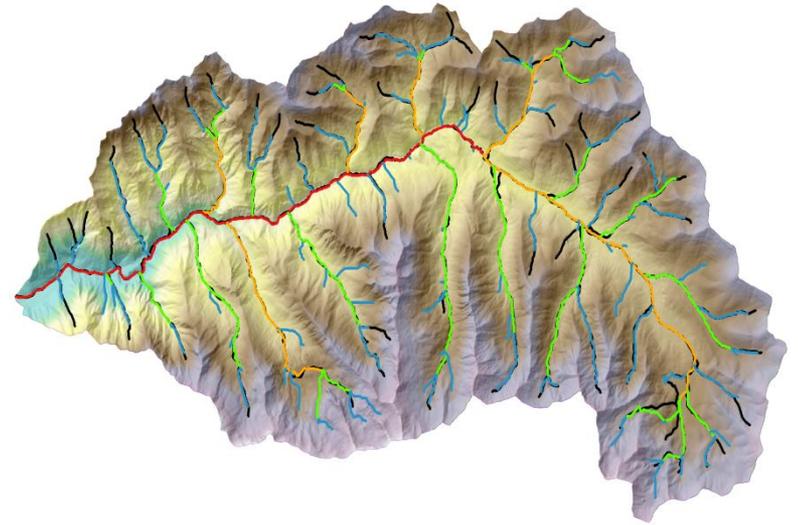
- Las **redes** ofrecen una manera de modelar redes e infraestructuras habituales del mundo real.
- A modo de ejemplo: **redes de distribución de agua, redes de líneas eléctricas, redes de conducción de gas, redes de telefonía, redes de flujo de agua de un río, redes de tránsito y circulación.**

¿Qué son las “redes”? (I)

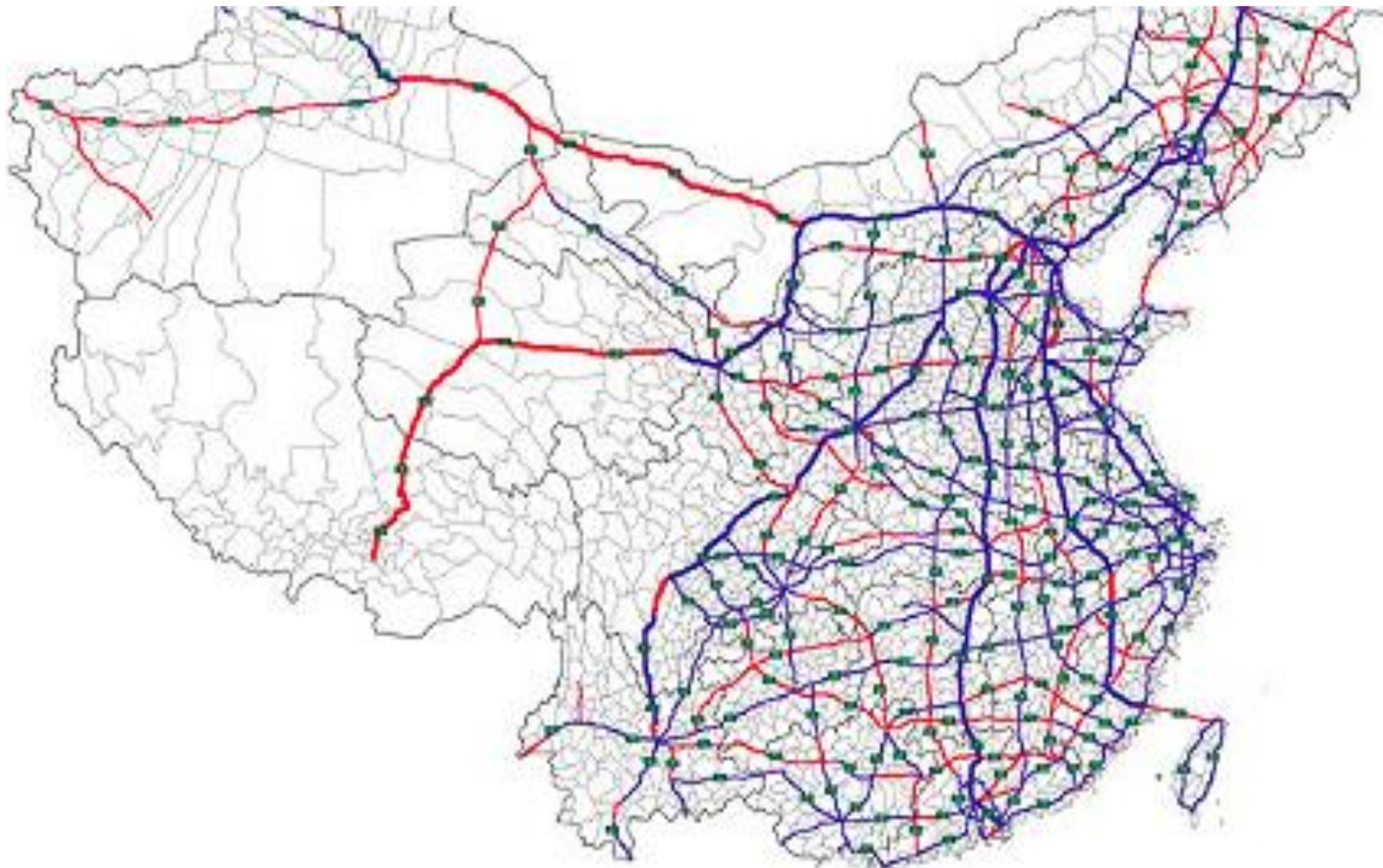
- Una **red** es un sistema de elementos interconectados, como **edges** (*bordes*: líneas) y **junctions** (*cruces de conexión*: puntos).
- Una red representa las posibles conexiones entre dos ubicaciones.



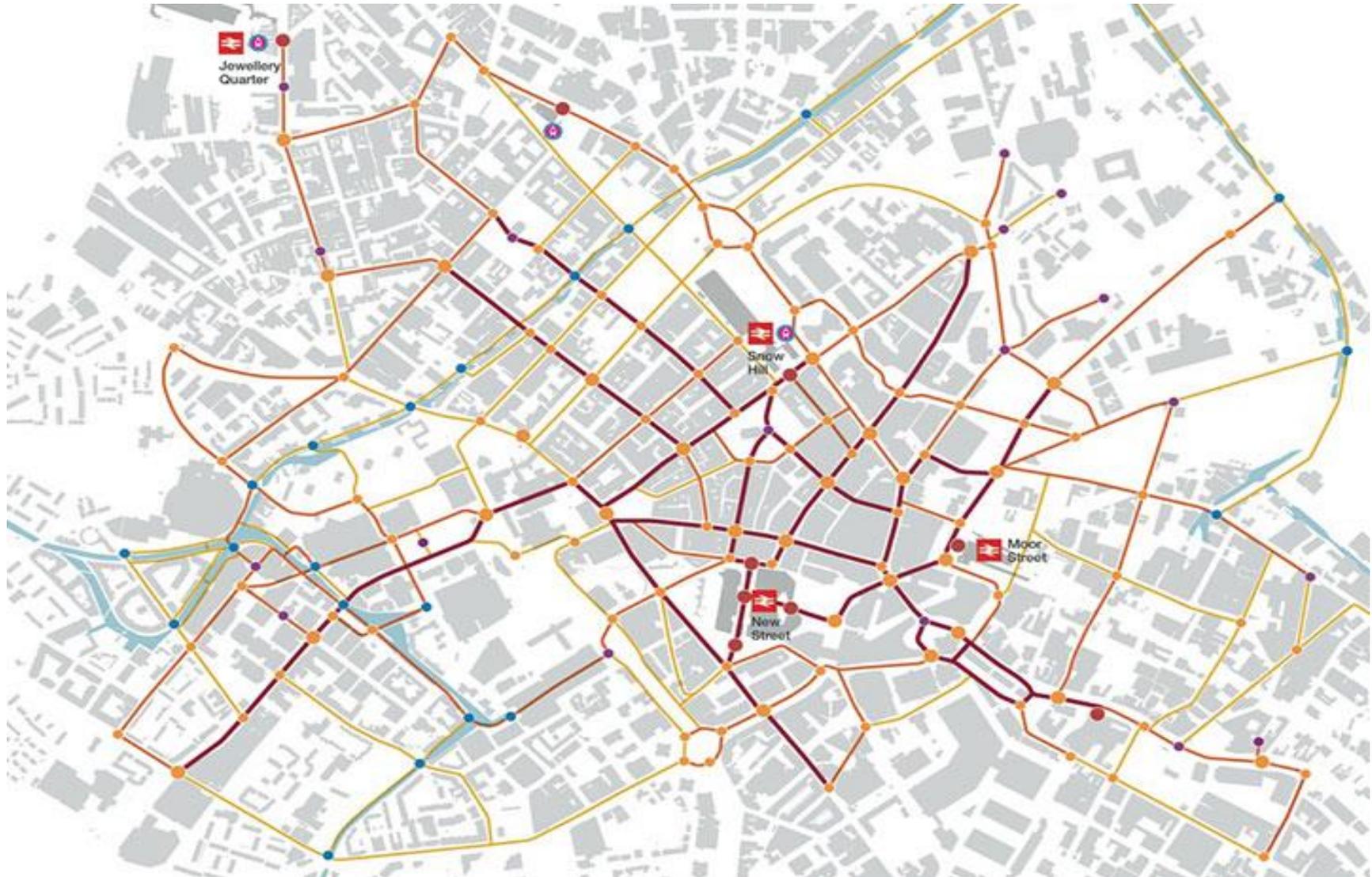
¿Qué son las “redes”? (II)



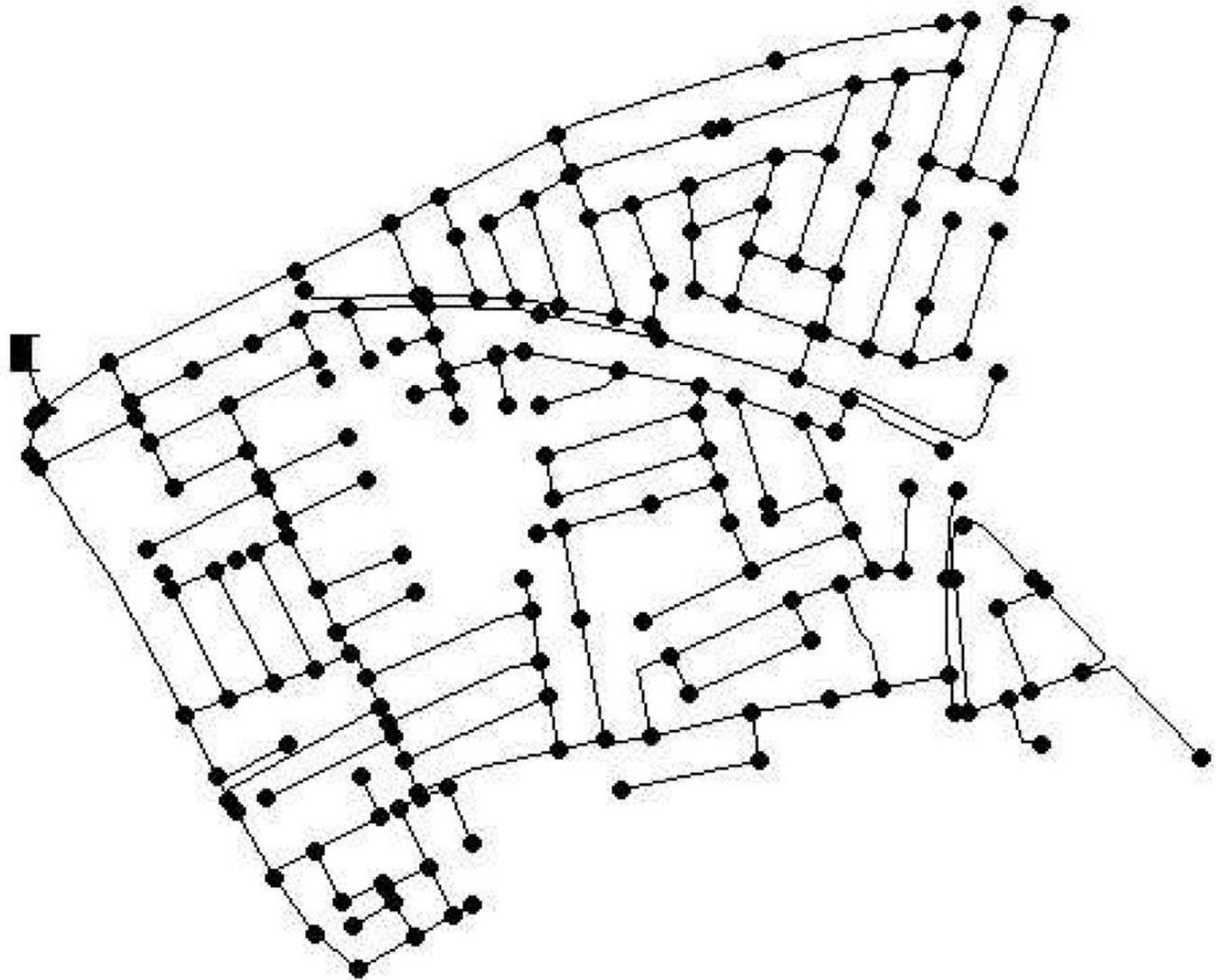
¿Qué son las “redes”? (III)



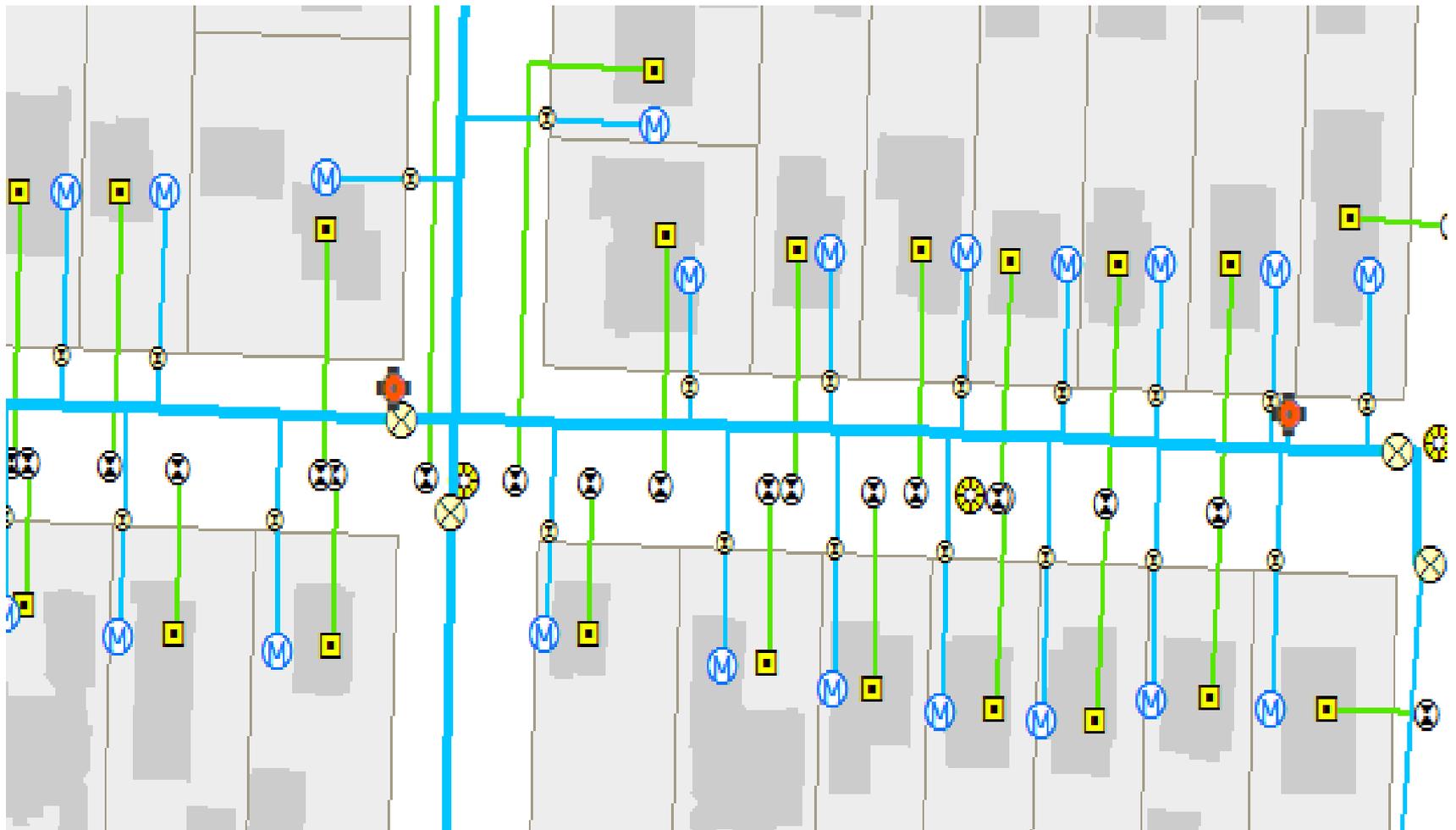
¿Qué son las “redes”? (IV)



¿Qué son las “redes”? (V)



¿Qué son las “redes”? (VI)



¿Qué son las “redes”? (VII)

- Las personas, recursos y **bienes tienden a circular a través de redes**: los autos y camiones circulan por carreteras, los aviones siguen rutas de vuelo predeterminadas, el petróleo circula por oleoductos.
- Si modelamos las potenciales rutas por una red, **podemos realizar análisis** relacionados con el movimiento del petróleo, camiones u otros agentes en la red.
- El análisis de red más común consiste en buscar la ruta más corta entre dos puntos.

Tipos de redes (I)

- Existen mayoritariamente dos tipos de redes: **redes de servicios** y **redes de transporte**.
- Las **redes de servicios o utilidades** modelan los ríos y servicios (electricidad, gas, desagües y conducciones de agua).
- Las **redes de transporte o tránsito** (redes de ferrocarril, peatones y calles, rutas aéreas) están orientadas a modelar y resolver los problemas de traslado de vehículos, personas y mercancías.

Principales funcionalidades (I)

- **Calcular la ruta más corta entre dos puntos:** las compañías de servicios públicos utilizan este método para inspeccionar la coherencia lógica de una red y comprobar la conexión entre dos puntos.
- **Buscar todos los elementos conectados o desconectados de la red:** las compañías eléctricas pueden ver qué parte de la red está desconectada y utilizar esa información para decidir cómo volverla a conectar.
- **Buscar bucles o circuitos en la red:** permite detectar un cortocircuito eléctrico.

Principales funcionalidades (II)

- **Determinar la dirección del flujo de bordes cuando se establecen fuentes o sumidero:** los ingenieros pueden ver el dirección del flujo a lo largo de los bordes.
- **Trazar elementos de red aguas arriba o aguas abajo de un punto:** las compañías de suministro de agua pueden determinar qué válvulas deben cerrar si estalla una tubería.
- **Calcular la ruta más corta de un punto a otro aguas arriba:** las estaciones de supervisión medioambiental pueden determinar el origen de la polución de los ríos.

Principales funcionalidades (III)

- **Buscar todos los elementos de red aguas arriba desde muchos puntos y determinar qué elementos son comunes a todos ellos:** las compañías eléctricas pueden utilizar las llamadas telefónicas de los clientes que sufren un corte de suministro para localizar transformadores o líneas con problemas.
- **Diseño de rutas:** permite encontrar la mejor manera de visitar varias ubicaciones.

Principales funcionalidades (IV)

- **Instalación más cercana:** Encontrar el hospital más cercano a un accidente, la patrulla de policía más próxima a la escena de un crimen o el almacén más cercano a la dirección de un cliente.
- **Áreas de servicio:** Encontrar áreas de servicio situadas en las proximidades de cualquier ubicación de una red. Un área de servicio es una región que abarca todas las calles accesibles, esto es, las calles que están dentro de un costo especificado.

Principales funcionalidades (V)

- **Matriz de costos:** Crear una matriz de coste origen-destino (OD) de múltiples orígenes a múltiples destinos. Un matriz de coste OD es una tabla que contiene los costos de red desde cada origen hasta cada destino.

“Edges” & “Junctions” (I)

Las redes se componen de dos elementos principales: **edges** y **junctions**.

- **Edges** (Bordes): es una entidad geográfica que tiene una longitud a través de la que fluye algo (producto, servicio, persona, vehículo). Los edges se crean a partir capas temáticas de tipo línea.
- Ejemplos: las cañerías del agua, las líneas de transmisión de la electricidad, las conducciones del gas, las líneas telefónicas, las vías de tren, las calles de una ciudad, las rutas de un país.

“Edges” & “Junctions” (II)

- **Junctions** (Cruces): un cruce es una entidad que permite que dos o más bordes se conecten y facilita la transferencia entre los bordes. Los junctions se crean a partir capas temáticas de tipo punto.
- Ejemplos: fusibles, conmutadores, grifos de servicio y válvulas, paradas de ómnibus, semáforos, cruces de calles.

“Edges” & “Junctions” (III)

- **Los edges y los junctions de una red están conectados topológicamente entre sí:** los edges deben conectarse a otros edges en junctions, y el flujo de los edges de la red se transfiere a otros edges a través de junctions.

Redes Lógicas (I)

- Un modelo de red tiene asociado una **red lógica**.
- Se usan para representar y modelar las relaciones de conectividad entre las distintas entidades.
- La red lógica es el gráfico de conectividad que se utiliza para las operaciones de trazado y de flujo. Toda la conectividad entre los edges y los junctions se mantienen en la red lógica.

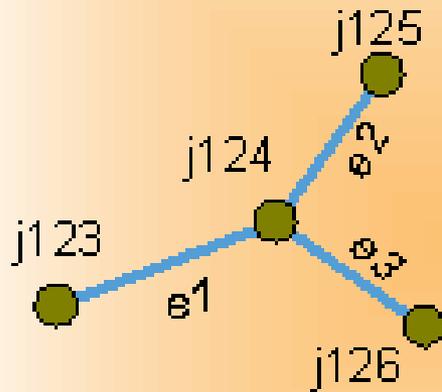
Redes Lógicas (II)

- Se administra como una colección de tablas alfanuméricas.
- Estas tablas registran cómo se conectan entre sí las entidades implicadas en una red.
- La red lógica permite a la red geométrica detectar y modelar rápidamente las relaciones de conectividad entre edges conectados y junctions en una red durante la edición y el análisis.

Redes Lógicas (III)

- Cuando se editan los edges y los junctions de una red, la red lógica correspondiente también se actualiza y se mantiene automáticamente.

Redes Lógicas (IV)



Junction feature table

id	geometry
j123	
j124	
j125	
j126	

Edge feature table

id	geometry
e1	
e2	
e3	

Connectivity table

Junction	Adjacent junction and edge		
j123	j124, e1		
j124	j123, e1	j125, e2	j126, e3
j125	j124, e2		
j126	j124, e3		

Costos de red (I)

- Una red puede tener un conjunto de **costos** (o pesos o impedancias) asociados.
- Se utilizan para representar el **costo de atravesar** un edge o junction en la red.
- **Por ejemplo:** en una red de abastecimiento de agua, cuando se atraviesa la longitud de una conducción de transmisión se pierde una cierta cantidad de presión debido a la fricción con la superficie interior de la tubería.

Costos de red (II)

- **Por ejemplo:** los costos de una red de transporte pueden ser los tiempos empleados en transitar los edges, los largos, los estados de conservación, etc. Incluso una combinación de todos.
- **Por ejemplo:** los costos de una red de ómnibus pueden ser los tiempos empleados en ir de un lugar a otro, el costo del boleto, la cantidad de paradas, etc..

Costos de red (III)

- **Los pesos de red se almacenan dentro de la red lógica.**
- Los valores de peso para cada elemento de red se **derivan de atributos** de la entidad correspondiente.
- Una red puede tener **cualquier número de costos** asociados.
- Cada clase de entidad de la red puede tener algunos, todos o ninguno de estos pesos asociados a sus atributos.

Costos de red (IV)

- El peso de cada entidad está determinado por un **atributo para esa entidad**.
- **Generalmente los pesos son positivos.**
- **Los valores negativos** se utilizan para indicar que los edges o junction no son transitables.

Costos de red (V)



Costos de red (VI)

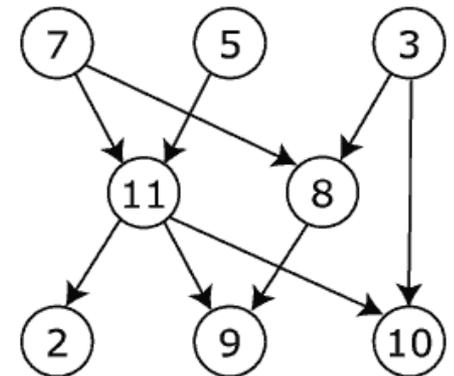


Teoría de Grafos (I)

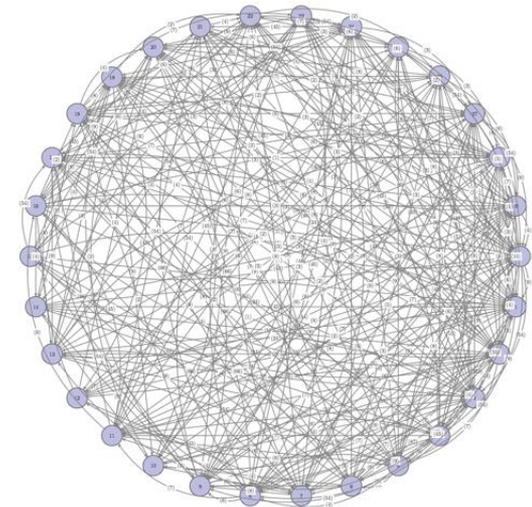
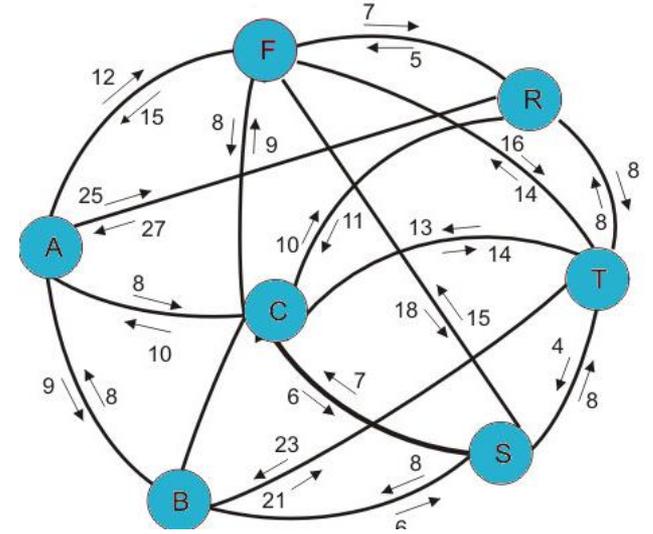
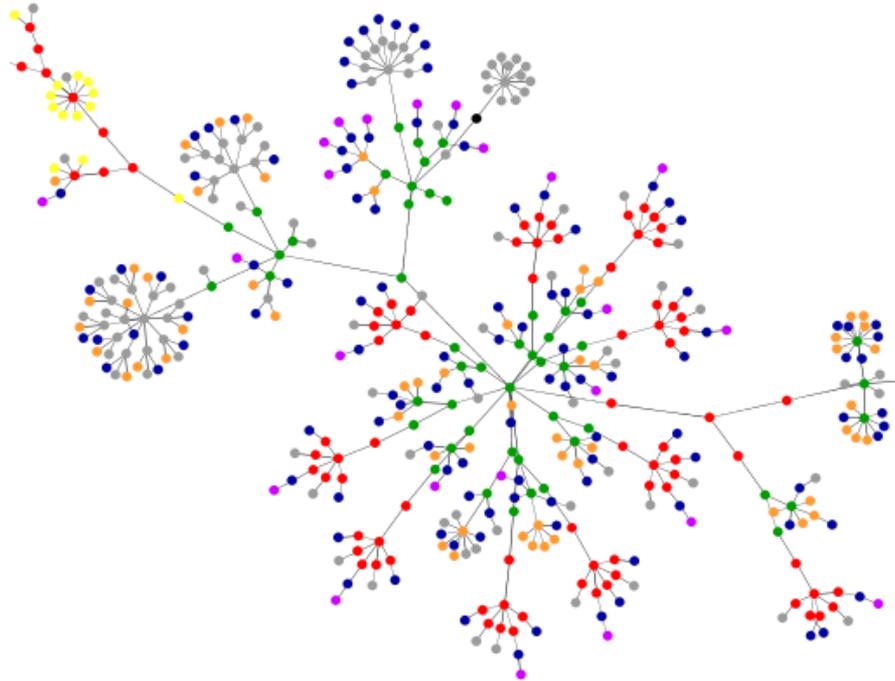
- Juegan un papel importante en la fundamentación matemática de las **Ciencias de la Computación**.
- Los grafos constituyen una herramienta básica para **modelar fenómenos discretos** y son fundamentales para la comprensión de las estructuras de datos y el análisis de algoritmos.

Teoría de Grafos (II)

- En matemáticas y ciencias de la computación, la teoría de grafos estudia las **propiedades de los grafos**, que son colecciones de objetos llamados **vértices** (o nodos) conectados por líneas llamadas **aristas** (o arcos) que pueden tener orientación (dirección asignada).



Teoría de Grafos (III)



Problema del camino más corto (I)

- Consiste en encontrar **un camino entre dos vértices (o nodos)** de tal manera que la **suma de los pesos de las aristas que lo constituyen es mínima**.
- Ejemplo: **encontrar el camino más rápido para ir de una ciudad**. En este caso, los vértices representarían las ciudades y las aristas las carreteras que las unen, cuya ponderación viene dada por el tiempo (o distancia) que se emplea en atravesarlas.

Problema del camino más corto (II)

- Este problema puede ser definido para grafos **no dirigidos**, **dirigidos** o **mixtos**.
- **Algoritmo de Dijkstra**: resuelve el problema de los caminos más cortos desde un único vértice origen hasta todos los otros vértices del grafo.
- **Algoritmo de Búsqueda A***: resuelve el problema de los caminos más cortos entre un par de vértices usando la heurística para intentar agilizar la búsqueda.

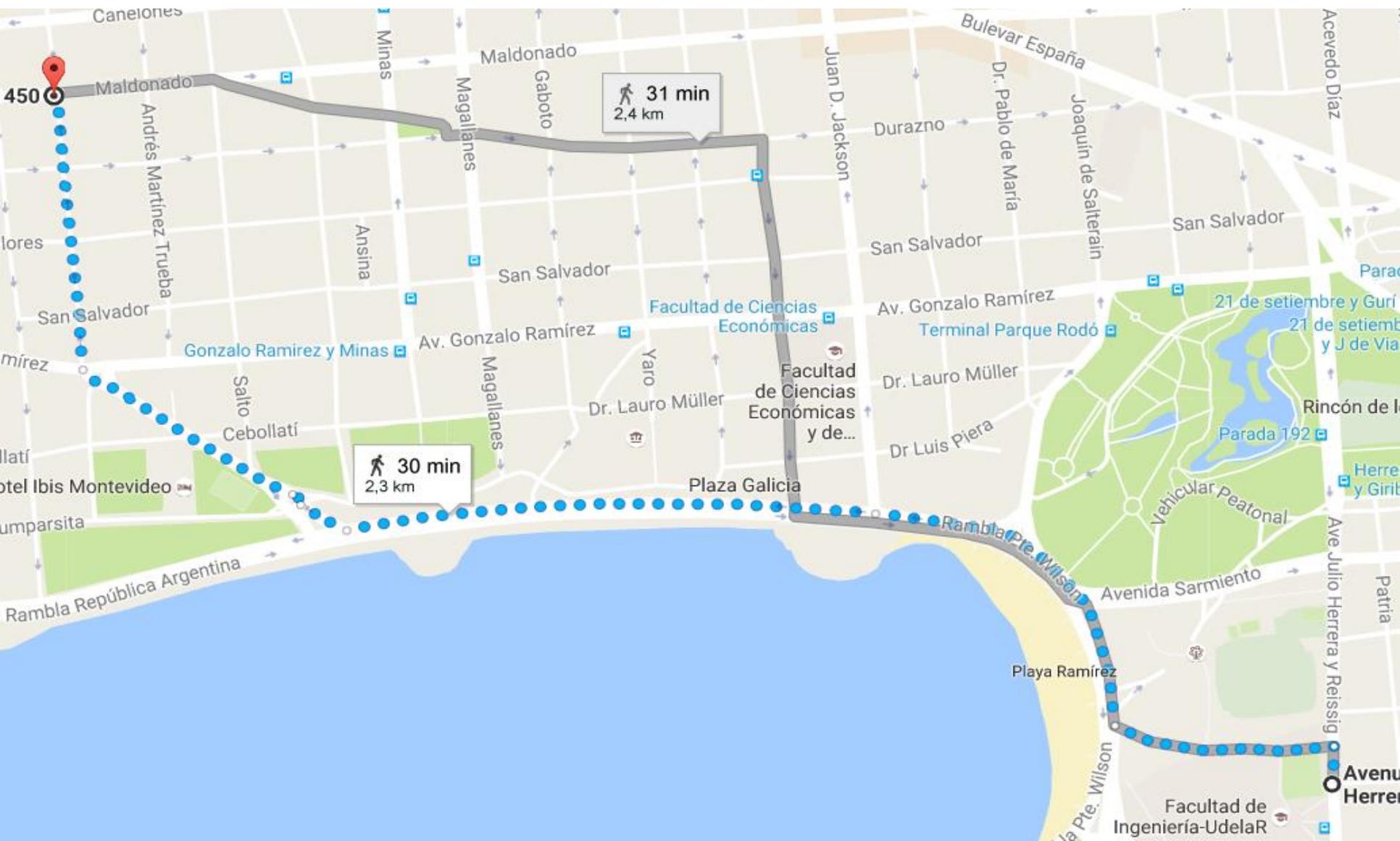
Problema del camino más corto (III)

- **Algoritmo de Floyd – Warshall:** resuelve el problema de los caminos más cortos entre todos los vértices.
- **Algoritmo de Viterbi:** resuelve el problema del camino estocástico más corto con un peso probabilístico adicional en cada vértice.

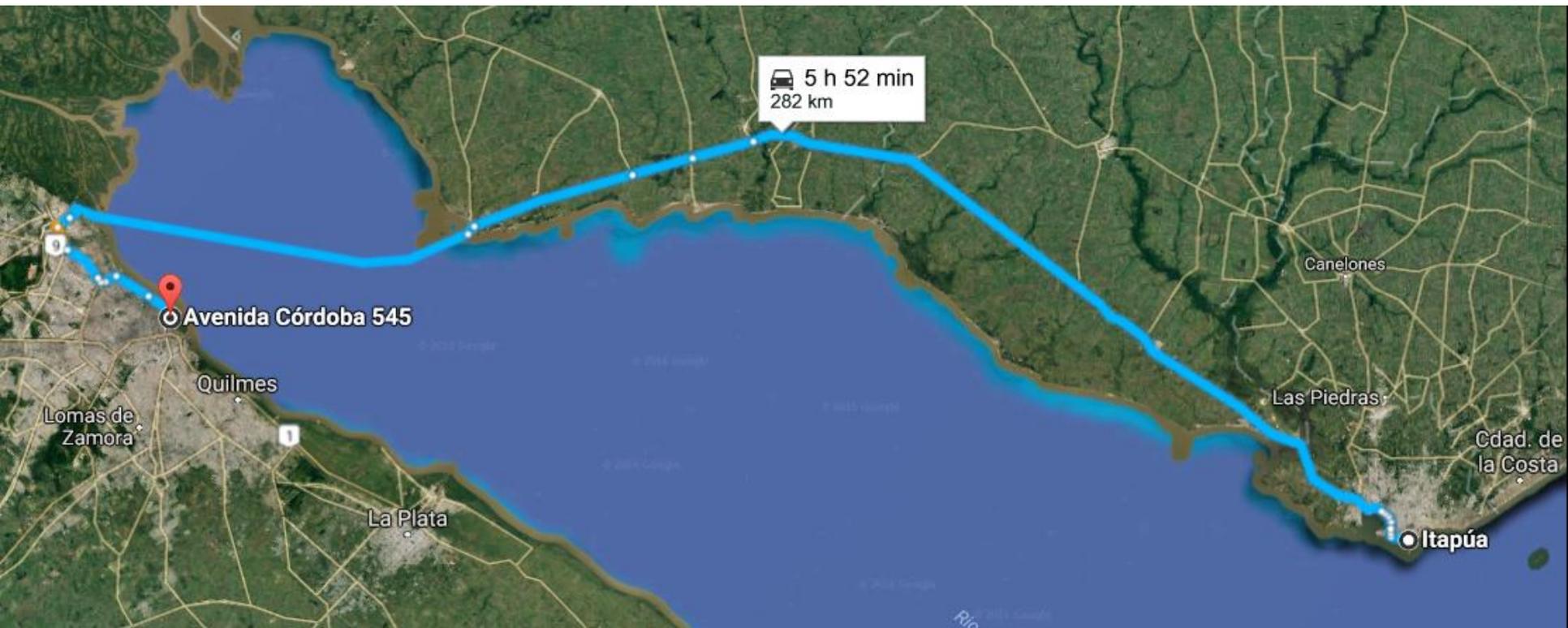
Problema del camino más corto (III)

- **Algoritmo:** Conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problemas.
- **Heurística:** dos objetivos fundamentales son encontrar algoritmos con buenos tiempos de ejecución y buenas soluciones, usualmente las óptimas. Una heurística es un algoritmo que abandona uno o ambos objetivos. Las heurísticas generalmente son usadas cuando no existe una solución óptima bajo las restricciones dadas (tiempo, espacio, etc.), o cuando no existe del todo o toman mucho tiempo.

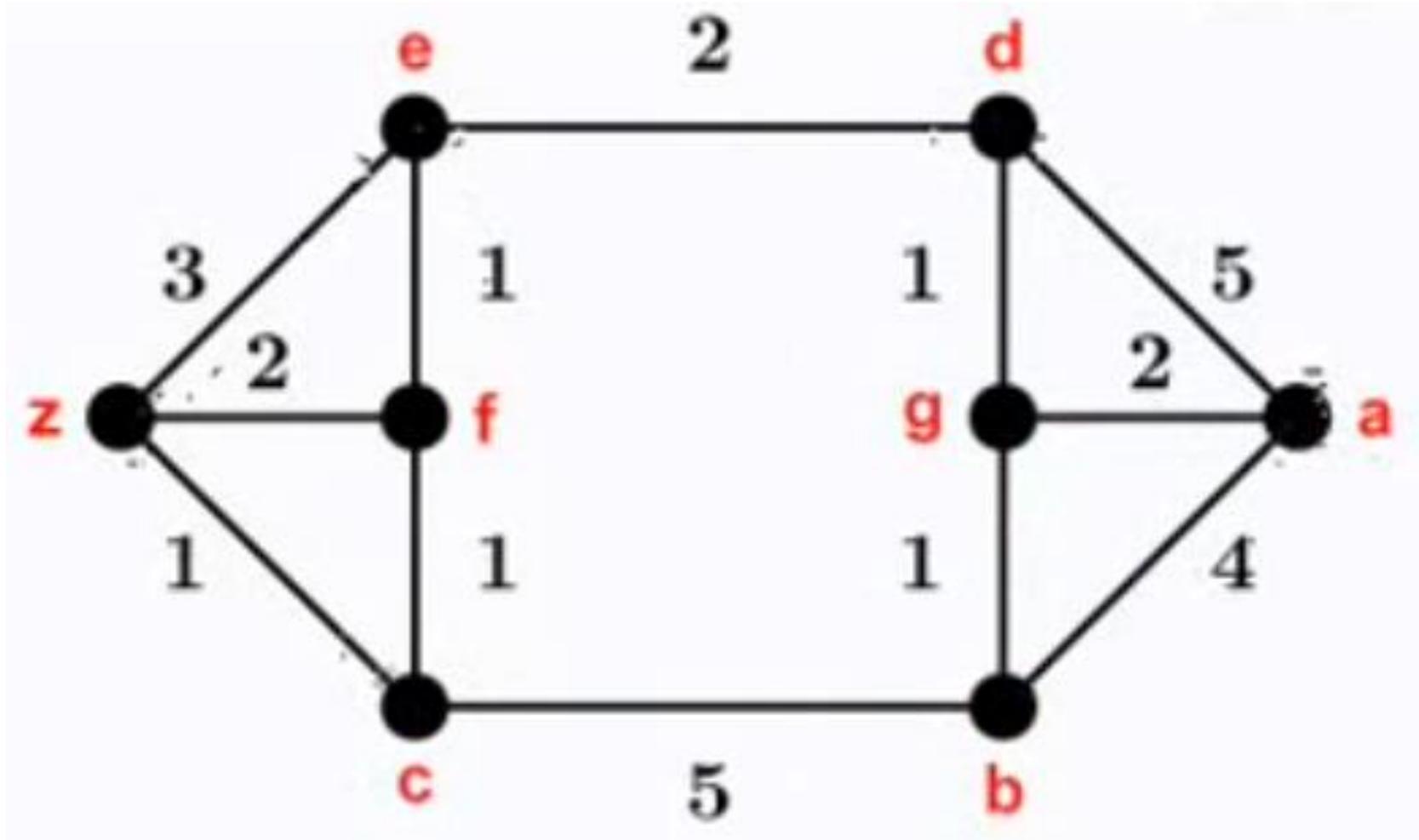
Problema del camino más corto (IV)



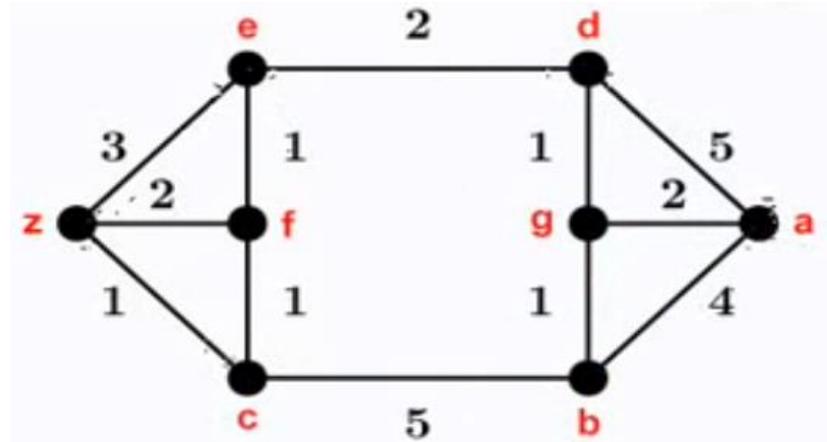
Problema del camino más corto (V)



Algoritmo de Dijkstra (I)



Algoritmo de Dijkstra (II)



Vértice	Paso 1	Paso 2	Paso 3	Paso 4	Paso 5	Paso 6	Paso 7	Paso 8
<i>a</i>	(0, <i>a</i>)	*	*	*	*	*	*	*
<i>b</i>	(4, <i>a</i>)	(3, <i>g</i>)	(3, <i>g</i>)	*	*	*	*	*
<i>c</i>	--	--	(8, <i>b</i>)	(8, <i>b</i>)	(8, <i>b</i>)	(7, <i>f</i>)	(7, <i>f</i>)	*
<i>d</i>	(5, <i>a</i>)	(3, <i>g</i>)	(3, <i>g</i>)	(3, <i>g</i>)	*	*	*	*
<i>e</i>	--	--	--	(5, <i>d</i>)	(5, <i>d</i>)	*	*	*
<i>f</i>	--	--	--	--	(6, <i>e</i>)	(6, <i>e</i>)	*	*
<i>g</i>	(2, <i>a</i>)	(2, <i>a</i>)	*	*	*	*	*	*
<i>z</i>	--	--	--	--	(8, <i>e</i>)	(8, <i>f</i>)	(8, <i>c</i>)	(8, <i>c</i>)

Algoritmo de Dijkstra (III)

Algoritmo (I):

- 1)** Inicializar todas las distancias en D con un valor infinito relativo ya que son desconocidas al principio, exceptuando la de x que se debe colocar en 0 debido a que la distancia de x a x sería 0.
- 2)** Sea $a = x$ (tomamos a como nodo actual).
- 3)** Recorremos todos los nodos adyacentes de a , excepto los nodos marcados, llamaremos a estos nodos no marcados v_i .

Algoritmo de Dijkstra (IV)

Algoritmo (parte II):

4) Para el nodo actual, calculamos la distancia tentativa desde dicho nodo a sus vecinos con la siguiente fórmula: $dt(v_i) = D_a + d(a, v_i)$. La distancia tentativa del nodo 'v_i' es la distancia que actualmente tiene el nodo en el vector D más la distancia desde dicho el nodo 'a' (el actual) al nodo v_i. Si la distancia tentativa es menor que la distancia almacenada en el vector, actualizamos el vector con esta distancia tentativa. Es decir: $Si dt(v_i) < Dv_i \rightarrow Dv_i = dt(v_i)$

Algoritmo de Dijkstra (III)

Algoritmo (parte III):

5) Marcamos como completo el nodo a .

6) Tomamos como próximo nodo actual el de menor valor en D y volvemos al paso 3 mientras existan nodos no marcados.

Algoritmo de Dijkstra (IV)

