

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación



IIC2115 - Programación como Herramienta para la Ingeniería

Manejo de redes

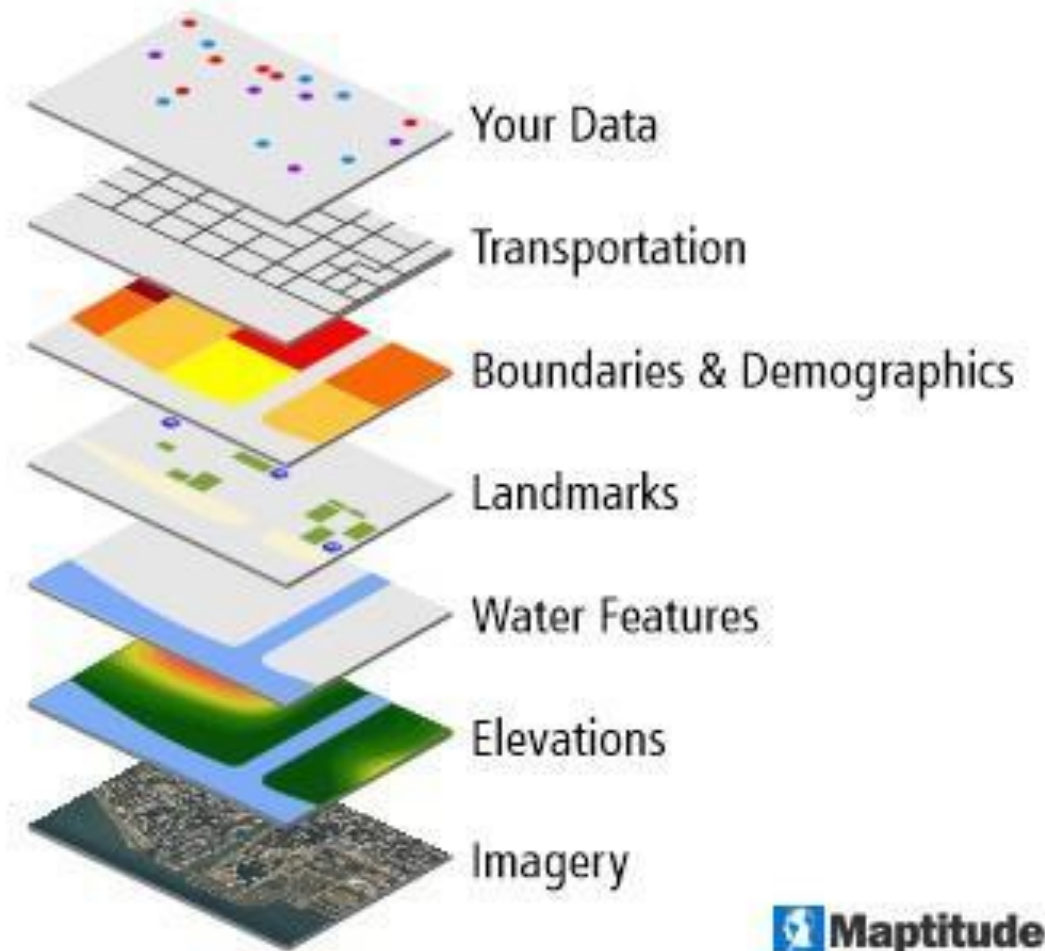
Profesor: Felipe Gutiérrez
Prof. Coordinador: Hans Löbel

¿Cómo podríamos encontrar la ruta de evacuación **más corta** en caso de inundación?
(manteniendo la idea del análisis exploratorio de datos)



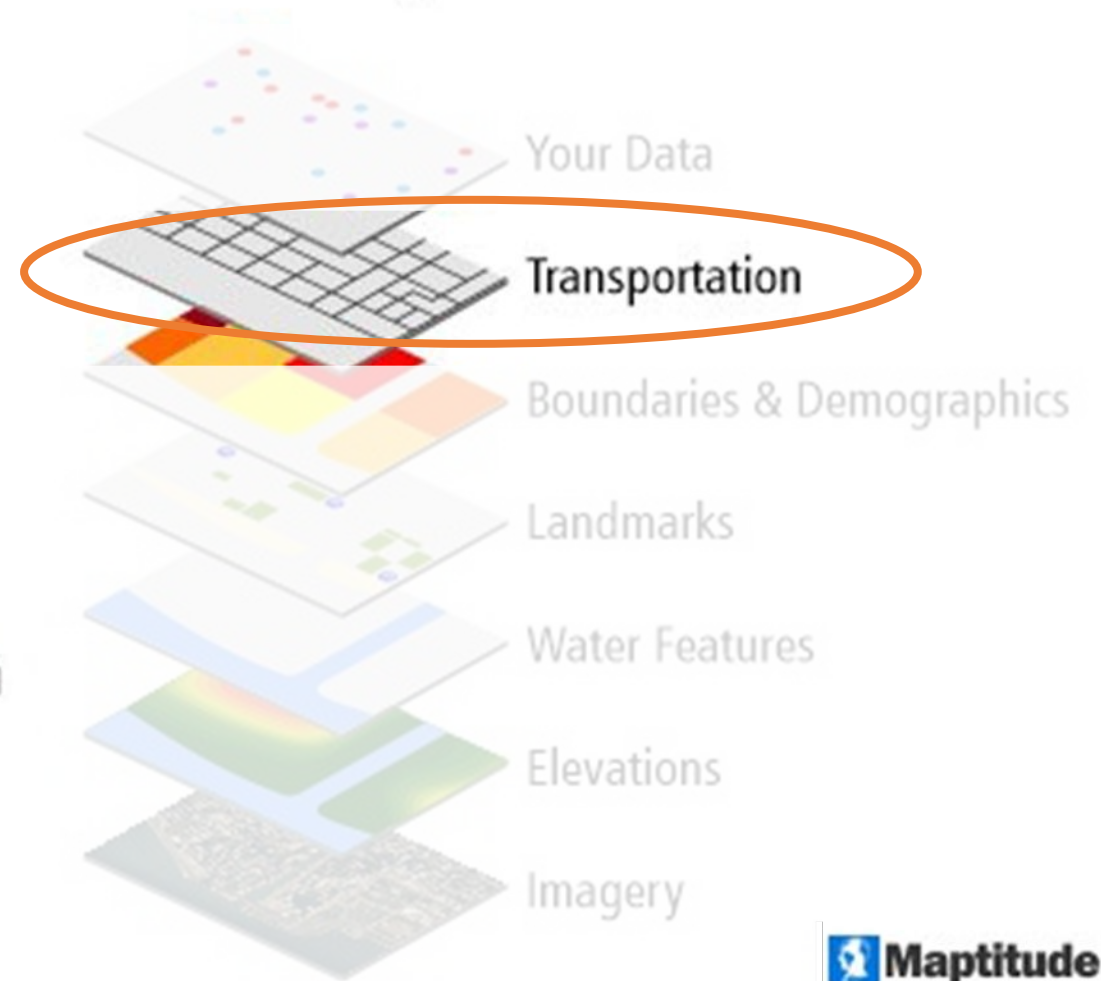
¿Cómo podríamos encontrar la ruta de evacuación **más corta** en caso de inundación?
(manteniendo la idea del análisis exploratorio de datos)

Real World → **GIS Data Layers**



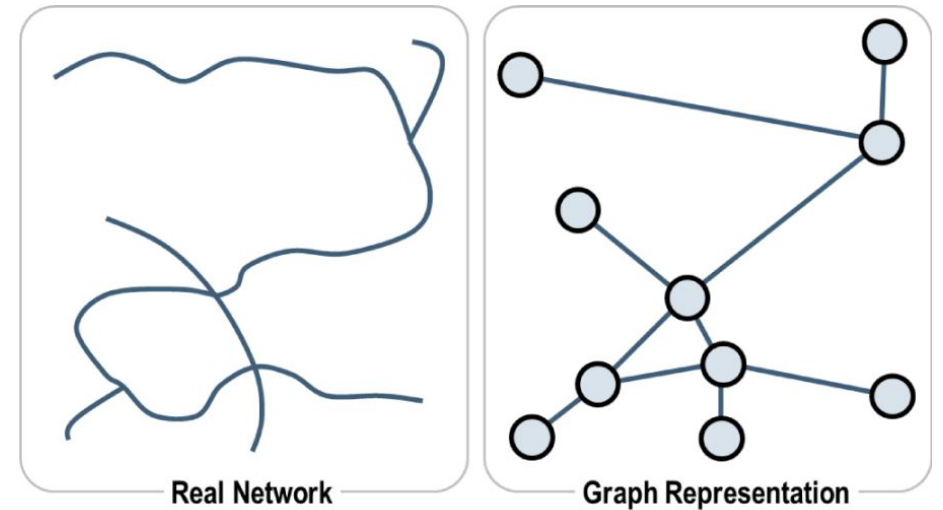
¿Cómo podríamos encontrar la ruta de evacuación **más corta** en caso de inundación?
(manteniendo la idea del análisis exploratorio de datos)

Real World → **GIS Data Layers**



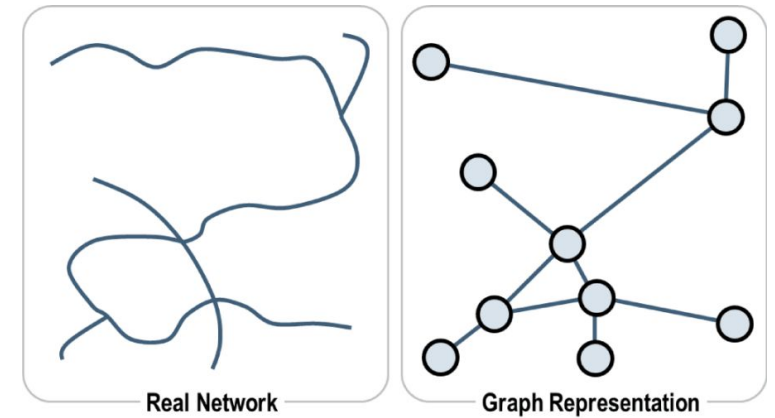
No nos basta información geométrica de la red de transporte/vial

- Necesitamos una abstracción más fácil de operar que los GeoDataframes
- **Grafos** son una solución eficiente
- Existen múltiples algoritmos para solucionar problemas de ruta mínima, flujo, ruteo, entre otros.
- Para hacer esto, nos basaremos en 2 librerías: NetworkX y OpenStreetMap



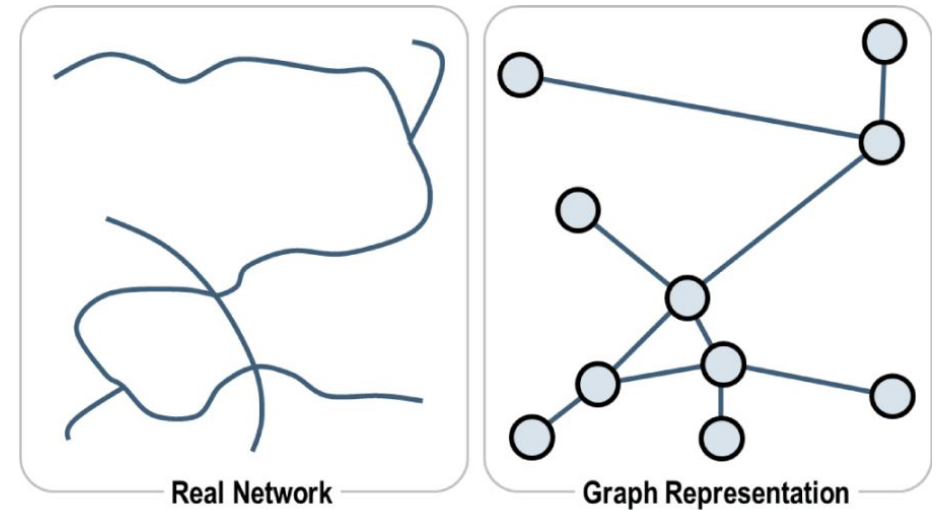
Outline: ¿qué veremos hoy?

- ¿Qué es un grafo?
- Tipos de grafos
- Problemas clásicos de grafos
- Ejemplo práctico



¿Qué es un grafo?

- **Abstracción** matemática para **representar** redes.
- Consta de **nodos** y **aristas**, que representan entidades y las relaciones entre ellas, respectivamente.
- Aplicable a **múltiples dominios**: redes de transporte, redes sociales, redes de comunicación, moléculas, etc.
- Para redes de transporte, podemos utilizar distintos grafos dependiendo del problema.

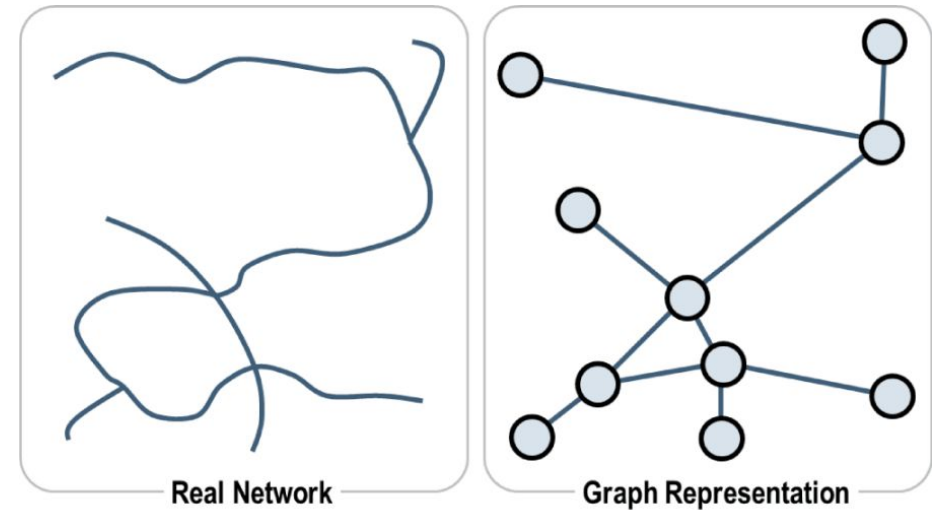


¿Qué es un grafo?

Recordemos la definición matemática:

$$\mathbf{G} = (\mathbf{V}, \mathbf{E})$$

- \mathbf{G} representa al **grafo**, es una **tupla** de dos conjuntos:
 - \mathbf{V} es el conjunto de **vértices o nodos**
 - \mathbf{E} es el conjunto de **aristas o arcos**
- Las arista pueden **tener o no una dirección**
- Tanto artistas como nodos pueden **guardar información**
 - Ejemplo: distancia, tiempo de transporte, ubicación, etc...



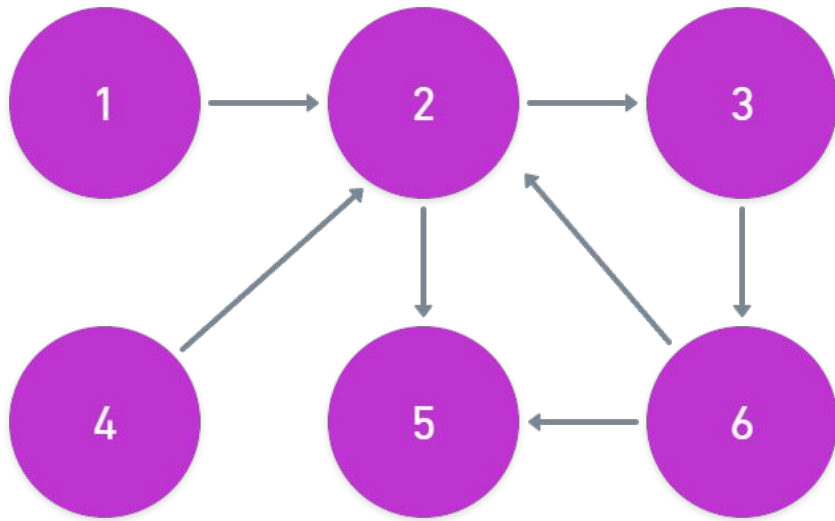
¿Qué es un grafo?

Conjunto de Vértices

$$\mathbf{V} = \{v_0, \dots, v_n\}$$

Conjunto de Aristas

$$\mathbf{E} = \{(v_h, v_i), \dots, (v_j, v_k)\}$$



Propiedades

- **Incidencia:** nodo destino de una arista
 - ¿Qué pasa si el grafo no tiene dirección?
- **Grado de un nodo:** cantidad de aristas que inciden en un nodo
- **Grado del grafo:** el grado de nodo máximo

Pensemos... ¿Cuál es el grado del grafo mostrado?

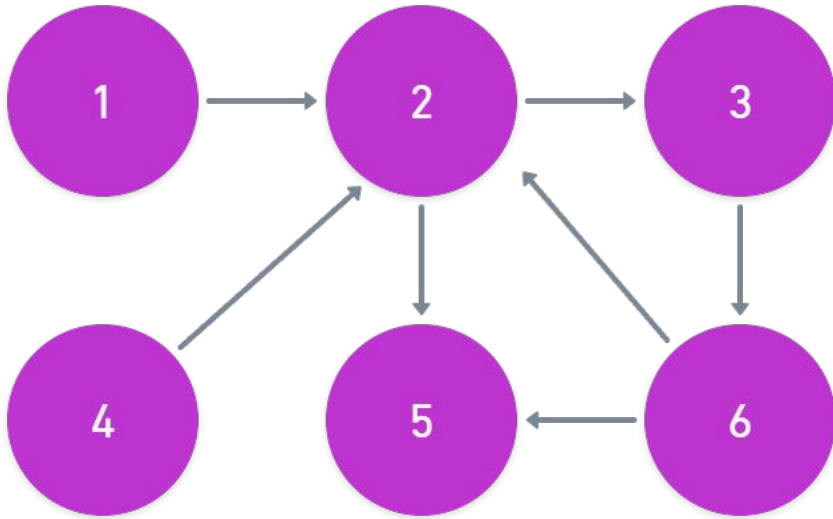
¿Qué es un grafo?

Conjunto de Vértices

$$\mathbf{V} = \{v_0, \dots, v_n\}$$

Conjunto de Aristas

$$\mathbf{E} = \{(v_h, v_i), \dots, (v_j, v_k)\}$$



En computación, los grafos se representan con:

- **Matrices de Adyacencia:** lista 2D en la que

$A[i][j] = 1$ indica si el nodo v_i incide en v_j

$A[i][j] = 0$ indica que v_i no incide en v_j

- El valor puede ser distinto de 1 para indicar pesos

- **Listas de Adyacencia:** lista de tuplas en la que si

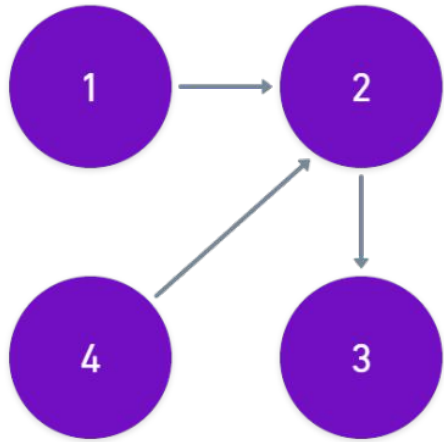
la tupla (v_j, v_k) está presente, v_i incide en v_j

- Se puede agregar un 3er valor para indicar peso de la arista

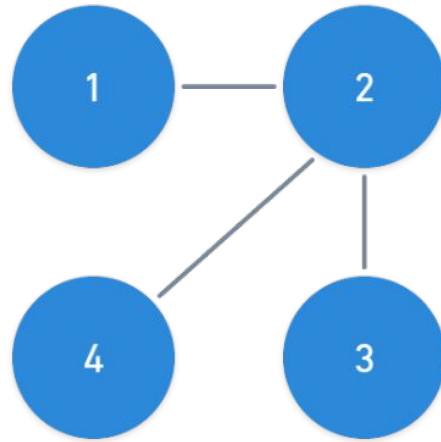
Tipos de grafos

Algunos ejemplos...

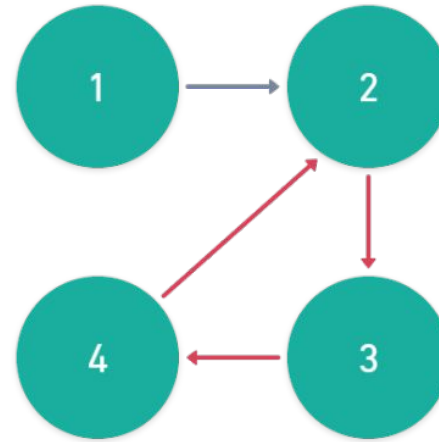
Dirigido



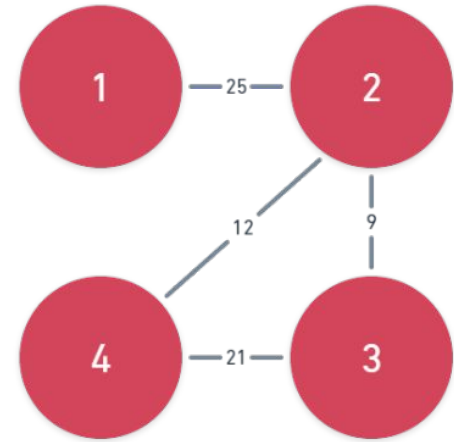
No Dirigido



Cíclico



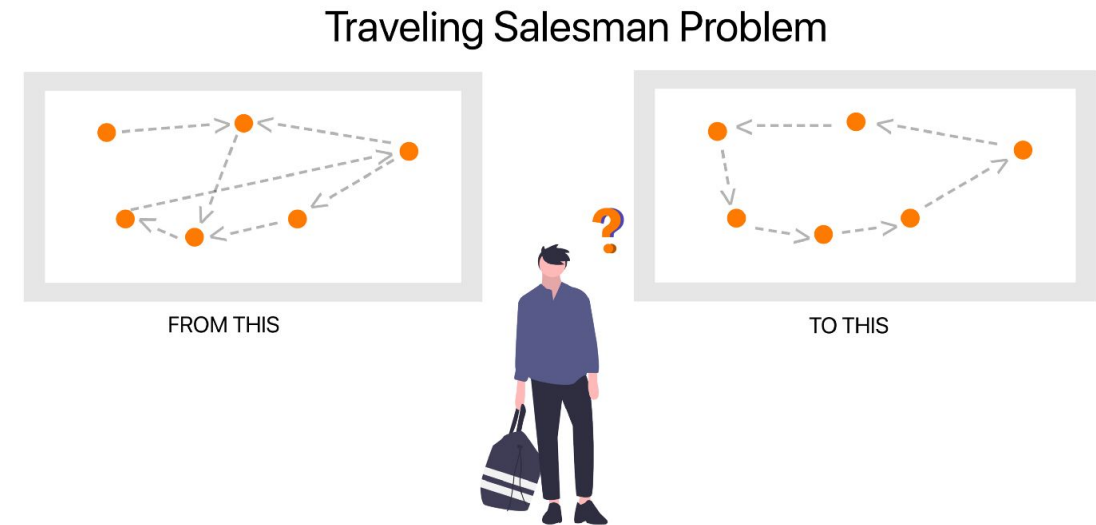
Con Pesos



¿Sabías que los árboles son un tipo de grafo? ¿De qué tipo pueden ser?

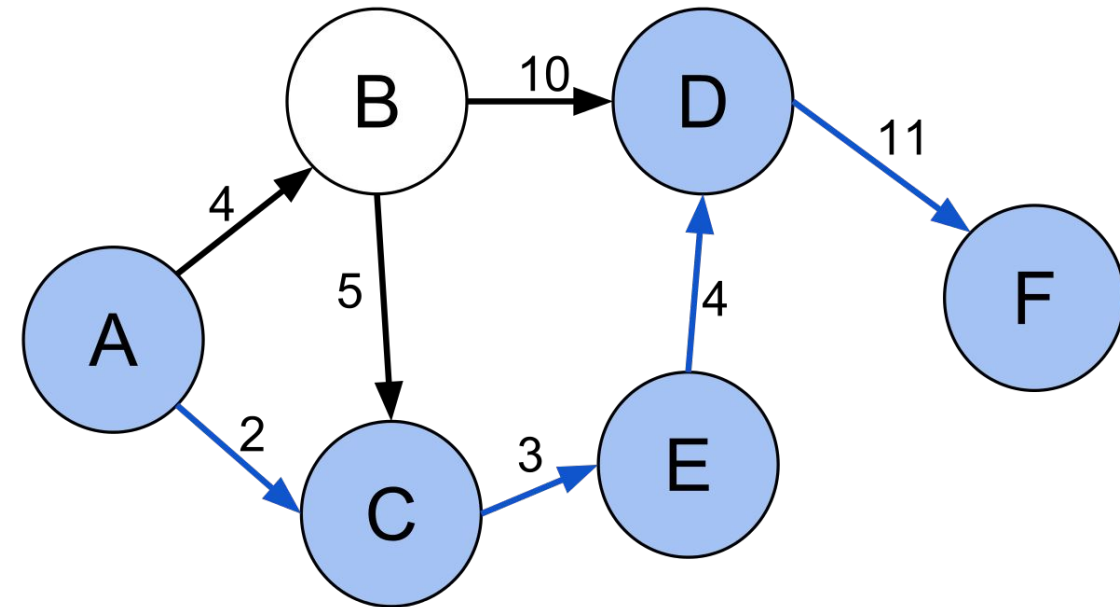
Problemas clásicos de grafos

- **Vendedor Viajero (Traveling Salesman Problem):** encontrar la ruta más corta/eficiente para visitar todos los nodos de una lista
- **Rutas más cortas:** encontrar la ruta más corta entre dos nodos
- **Flujo máximo:** dada una capacidad máxima para cada arista, cuál es la cantidad de unidades máximas que pueden transitar
- Buscar ciclos, determinar conectividad, isomorfismo, etc..
 - Ver más algoritmos en la librería de [NetworkX](#)



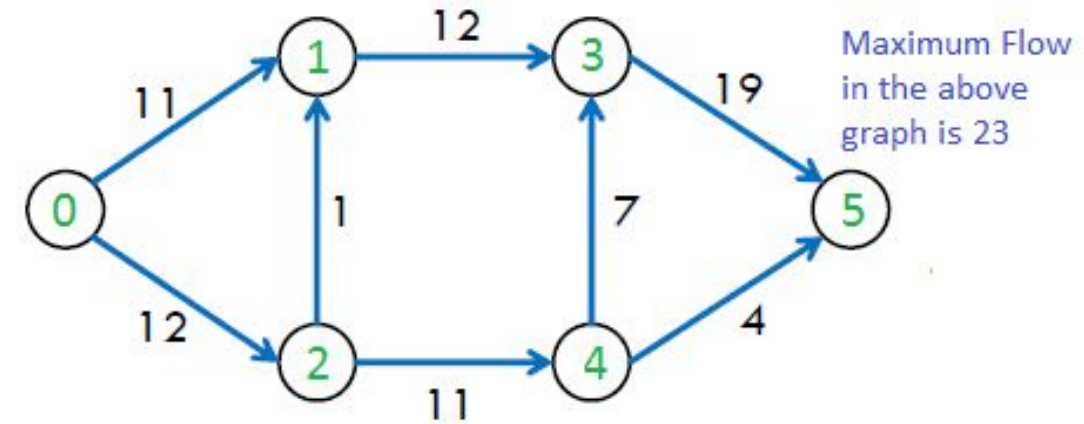
Problemas clásicos de grafos

- **Vendedor Viajero (Traveling Salesman Problem):** encontrar la ruta más corta/eficiente para visitar todos los nodos de una lista
- **Rutas más cortas:** encontrar la ruta más corta entre dos nodos
- **Flujo máximo:** dada una capacidad máxima para cada arista, cuál es la cantidad de unidades máximas que pueden transitar
- Buscar ciclos, determinar conectividad, isomorfismo, etc..
 - Ver más algoritmos en la librería de [NetworkX](#)



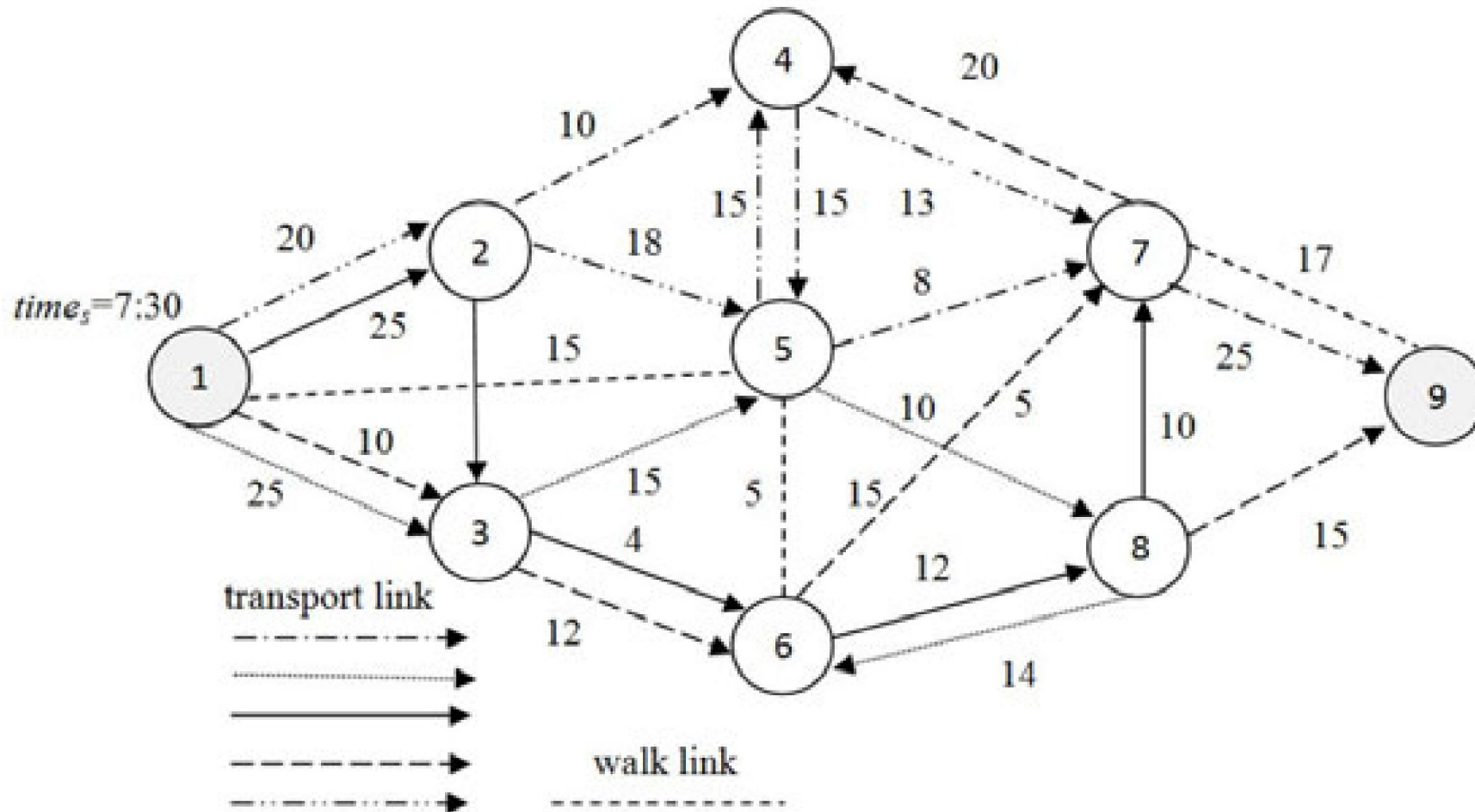
Problemas clásicos de grafos

- **Vendedor Viajero (Traveling Salesman Problem):** encontrar la ruta más corta/eficiente para visitar todos los nodos de una lista
- **Rutas más cortas:** encontrar la ruta más corta entre dos nodos
- **Flujo máximo:** dada una capacidad máxima para cada arista, cuál es la cantidad de unidades máximas que pueden transitar
- Buscar ciclos, determinar conectividad, isomorfismo, etc..
 - Ver más algoritmos en la librería de [NetworkX](#)



Problemas clásicos de grafos

Principalmente, nos interesará obtener/generar el grafo adecuado al problema y luego movernos eficientemente por él



Vamos a Colab...



NetworkX



OpenStreetMap

OSMnx

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación



IIC2115 - Programación como Herramienta para la Ingeniería

Manejo de redes

Profesora: Felipe Gutiérrez
Prof. Coordinador: Hans Löbel