Large-Scale and Multi-Structured Databases

Graph Databases

Recap of Graph Theory

Prof. Pietro Ducange

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

IN SUPREMÆ DIGNITATIS 1343

UNIVERSITÀ DI PISA

CROSSLAB Innovation for industry 4.0

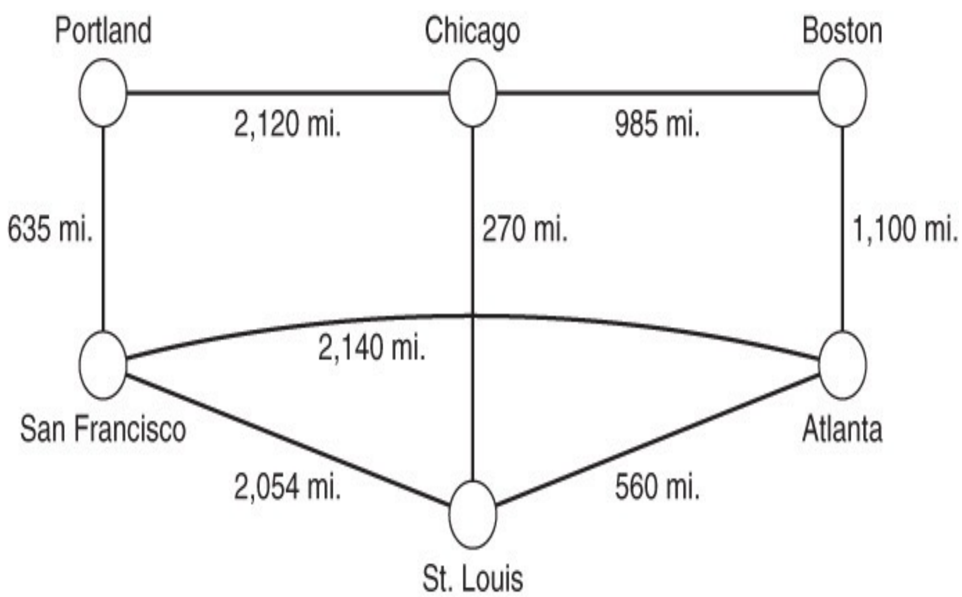
Vertex

In the figure, we show ***instances*** of an ***entity (city)*** represented by a set of vertices. We are in the context of ***highways networks***.

The ***properties*** that the vertices, namely the cities, may have are:

- Population
- Coordinates
- Name
- Geographic Region

Edges: Relationships between Vertices



The ***properties*** that the edge in the highways network, may have are:

- Distance
- Speed limit
- Number of lanes

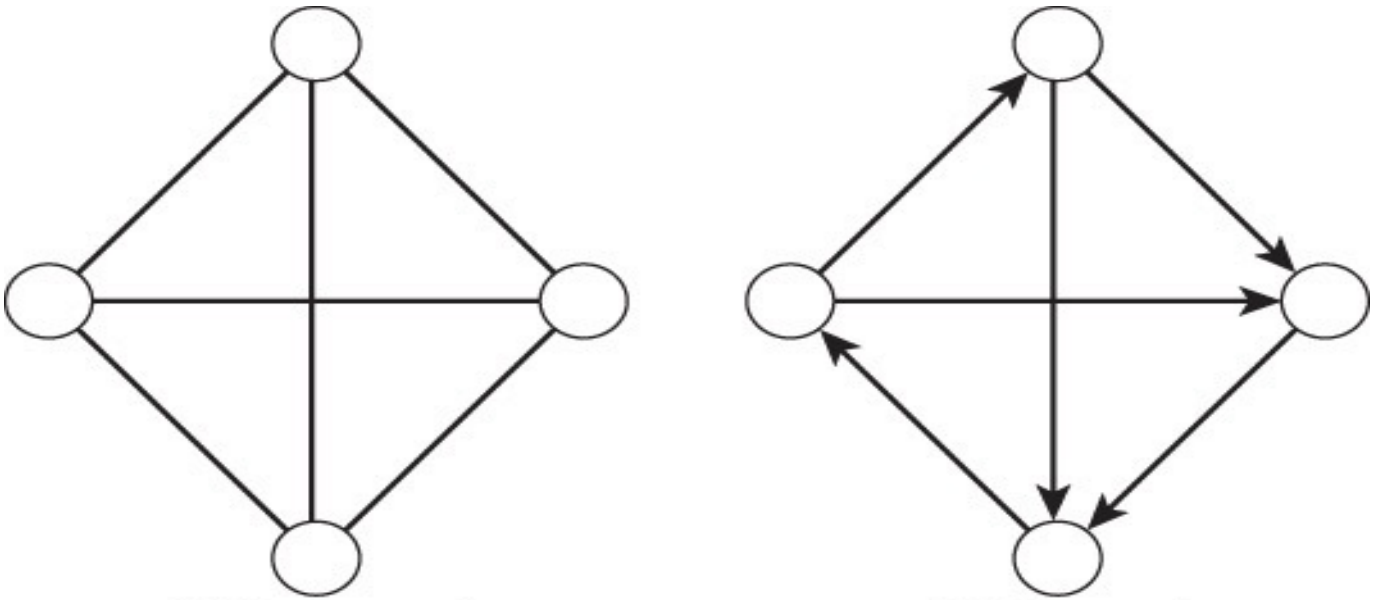
A Weight is a commonly used property of edge.

It represents some value about the ***strength*** of the relationship (cost, distance, etc.)

Undirected and Directed Graphs

Directed edges have a ***direction***, whereas undirected edge does not have a direction.

The ***presence*** or not of a ***direction*** in edges depends on ***specific applications***.



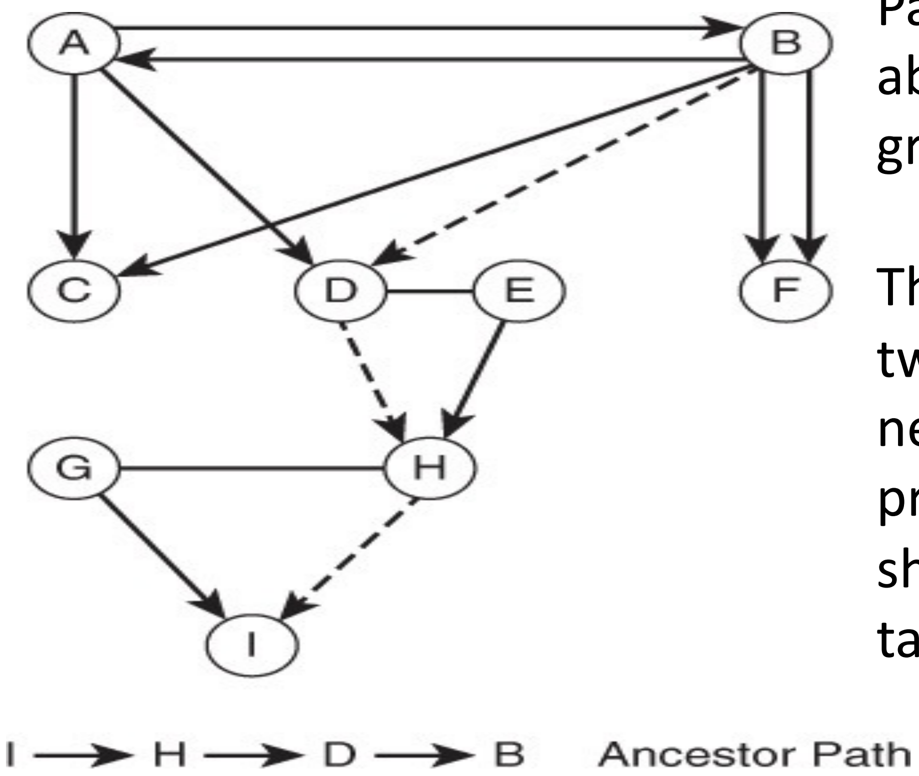
(a) Undirected (b) Directed

Example: Highways Network Graphs

Example: Family Relation Graphs

Path in a Graph

A ***path*** through a graph is a ***set of vertices*** along with the ***edges*** between those vertices.

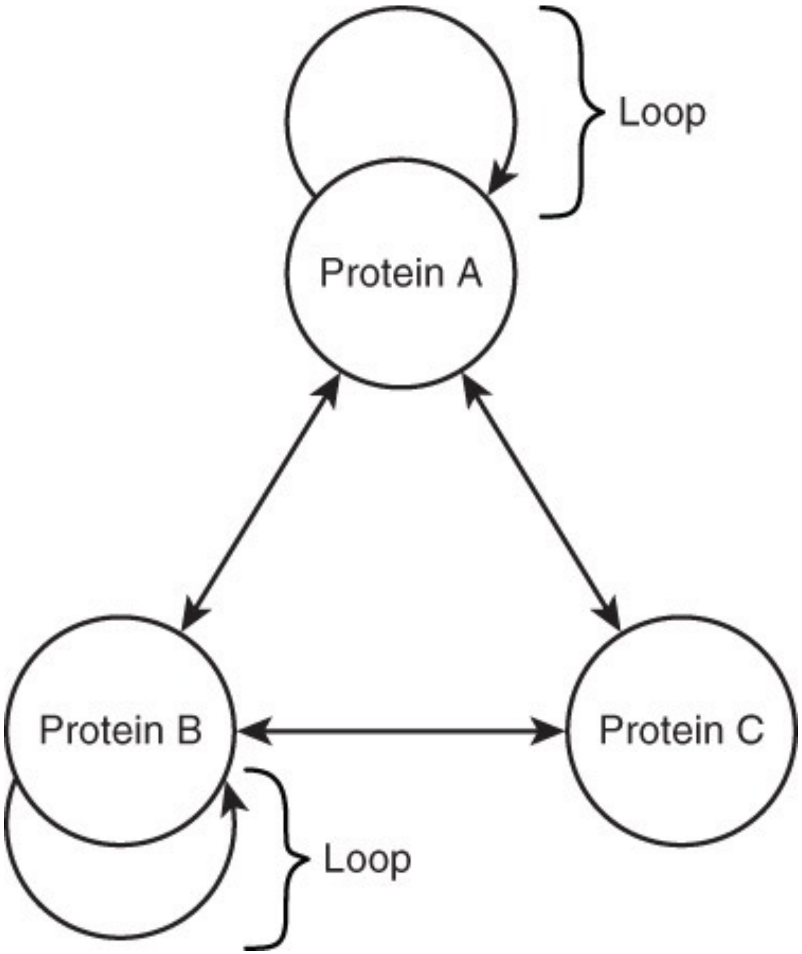


Paths allow us to ***capture*** information about the ***relationships*** of vertices in a graph.

There may be ***multiple paths*** between two vertices. In this case (highways networks, for instance), often the problem ***of finding the best*** (the shortest, the cheapest, etc.) path is taken into consideration.

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
UNIVERSITÀ DI PISA
CROSSLAB Innovation for industry 4.0

Loops in Graphs

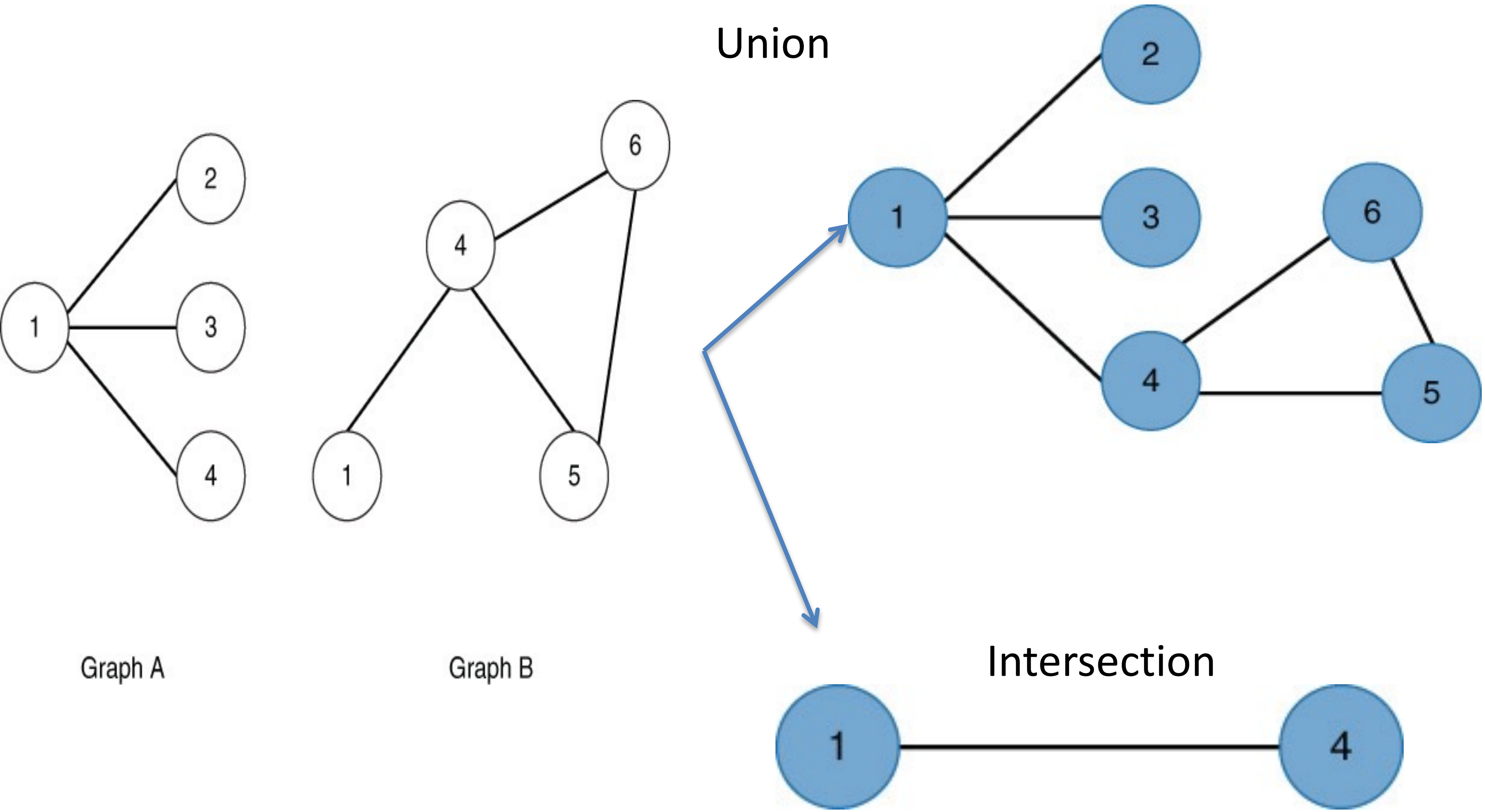


A loops represents a ***relation*** that an entities has with ***itself***.

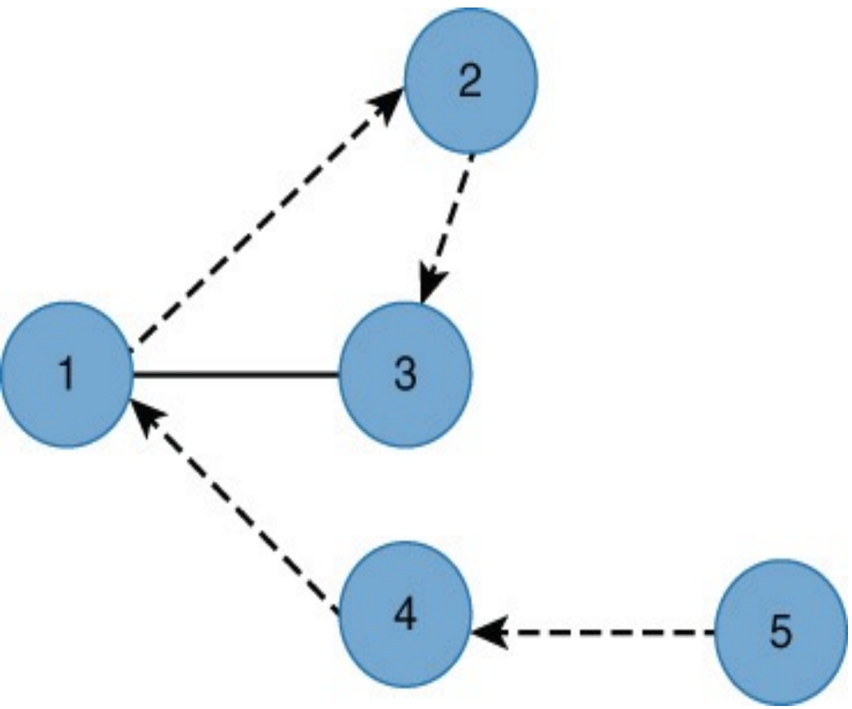
Proteins, for example, may interacts with proteins of the same type.

The presence of loop may ***not have sense*** in some specific domains, such as family tree graphs.

Union and Intersection of Graphs



Graph Traversal

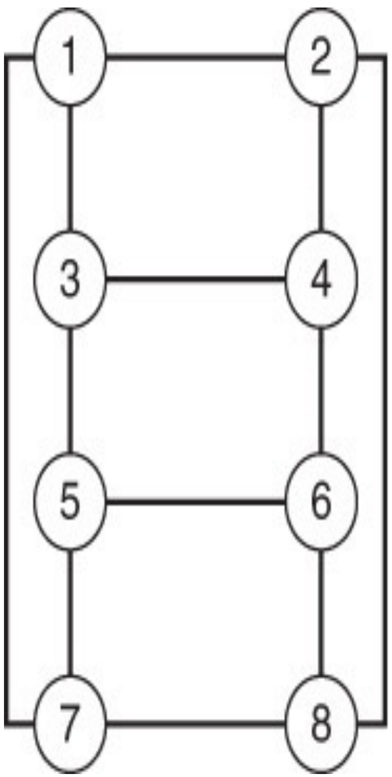


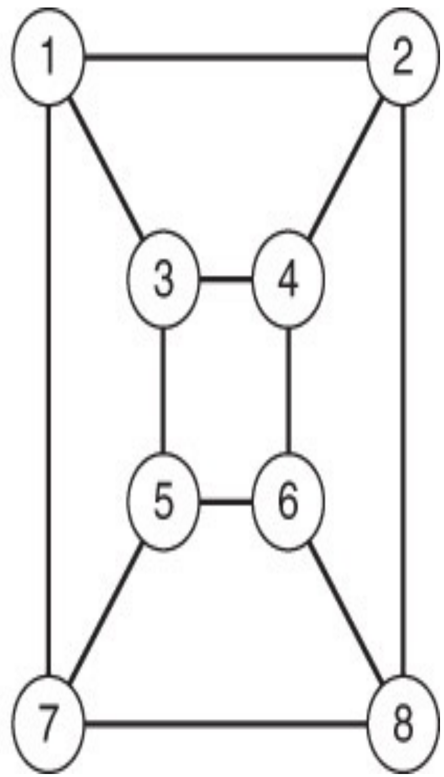
Graph traversal is the process of ***visiting all vertices*** in a graph.

The purpose of this is usually to either ***set*** or ***read*** some property value in a graph.

The traversal of the graph is often carried out in a particular way, such us ***minimizing the cost*** of the path.

Isomorphism





Two graphs are ***isomorphic*** if:

- ***for each vertex*** in the first graph, there is a ***corresponding vertex*** in the other graph

- ***for each edge*** between a pair of vertices in the first graph, there is a ***corresponding edge*** between the corresponding vertices of the other graph.

Detecting isomorphism in graphs or sub graphs allows us to identify ***useful patterns***.

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
IN SUPREMÆ DIGNITATIS 1343
UNIVERSITÀ DI PISA
CROSSLAB Innovation for industry 4.0

Order and Size of Graphs

Order: the number of vertices

Size: the number of edges

These two measures are very important to evaluate time and memory occupancy required to handle specific operations.

Example: to find the largest subset of people in a social network that know each other.

The higher the order and the size of the graph, the harder will be to solve the query!

Degree and Closeness of a Vertex

Degree: is the number of edges linked to a vertex. It measures the ***importance*** of any given vertex in a graph.

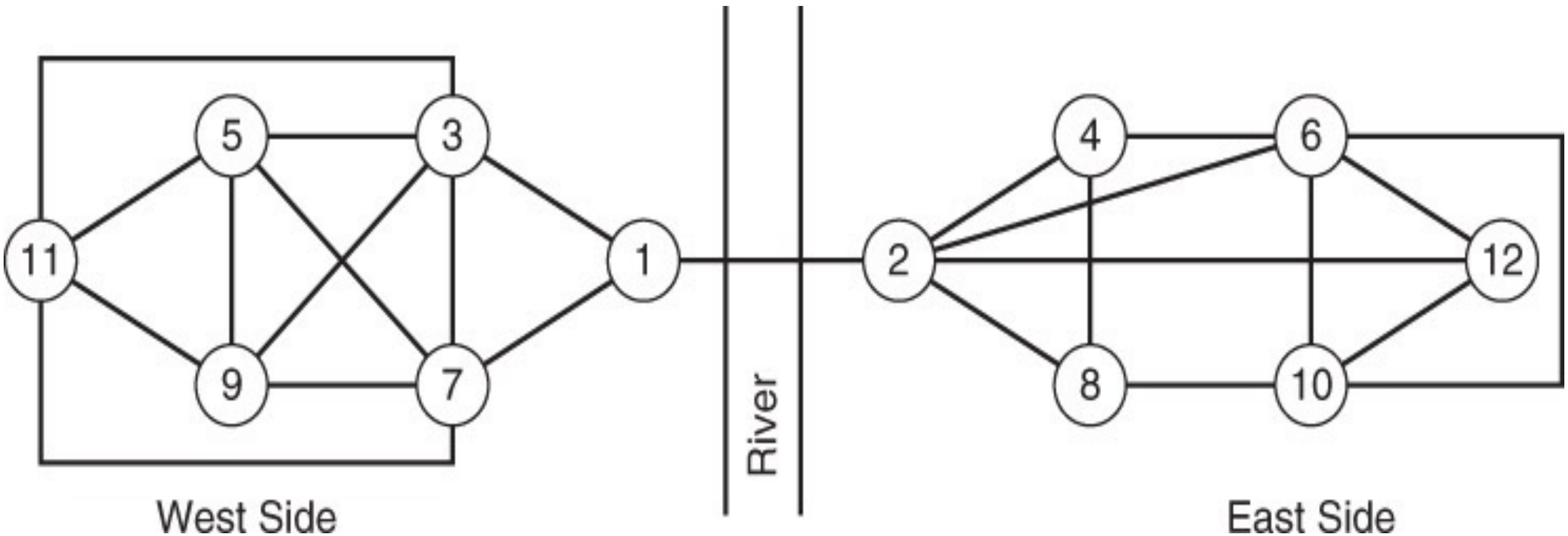
Example: Consider a person with many family members and friends that he/she sees regularly (high degree vertex). What if that person contracts a contagious disease?

Closeness: is the measure of how far the vertex is from all others in the graph**.** It can be calculated, in a fully connected graph, as the ***reciprocal of the sum*** of the length of the ***shortest paths*** between the node and all other nodes in the graph.

It is useful to evaluate ***the speed of information spreading in a network,*** starting from a specific node**.**

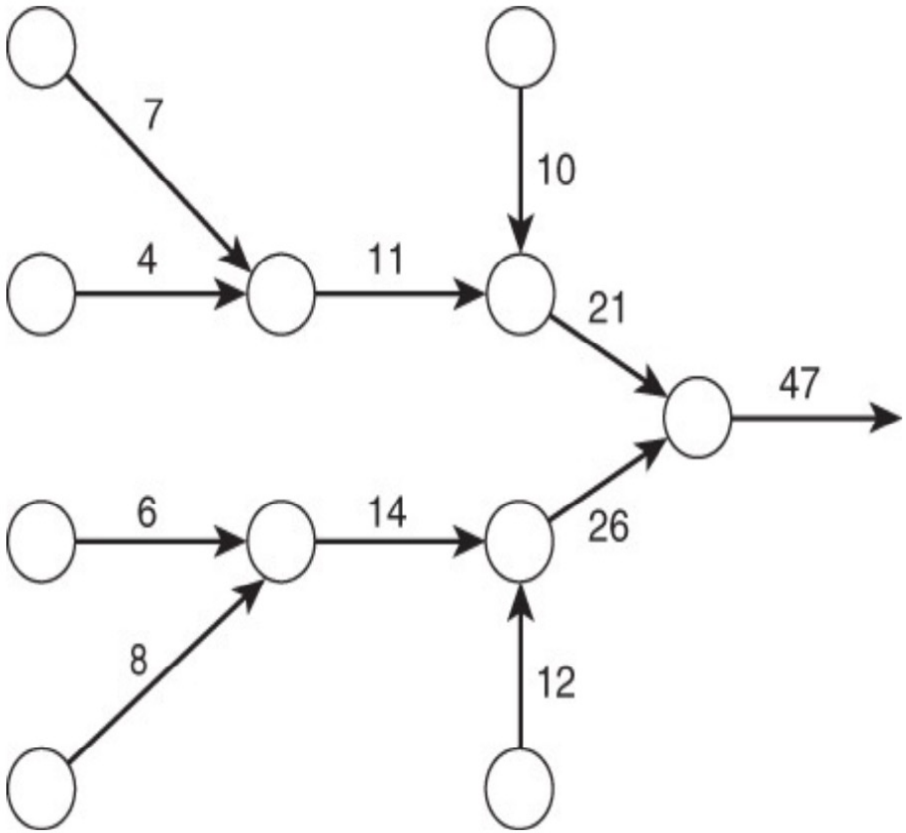
Betweenness of a Vertex

Betweenness is a measure of how much of a ***bottleneck*** a given vertex is.



In the example above, both vertices 1 and 2 will have ***high betweenness*** levels. Indeed, they form a bottleneck in the graph and if one of them were removed, it would leave the graph ***disconnected***.

Flow Networks



A flow network is a ***direct graph***.

It adopts ***capacitated*** edges.

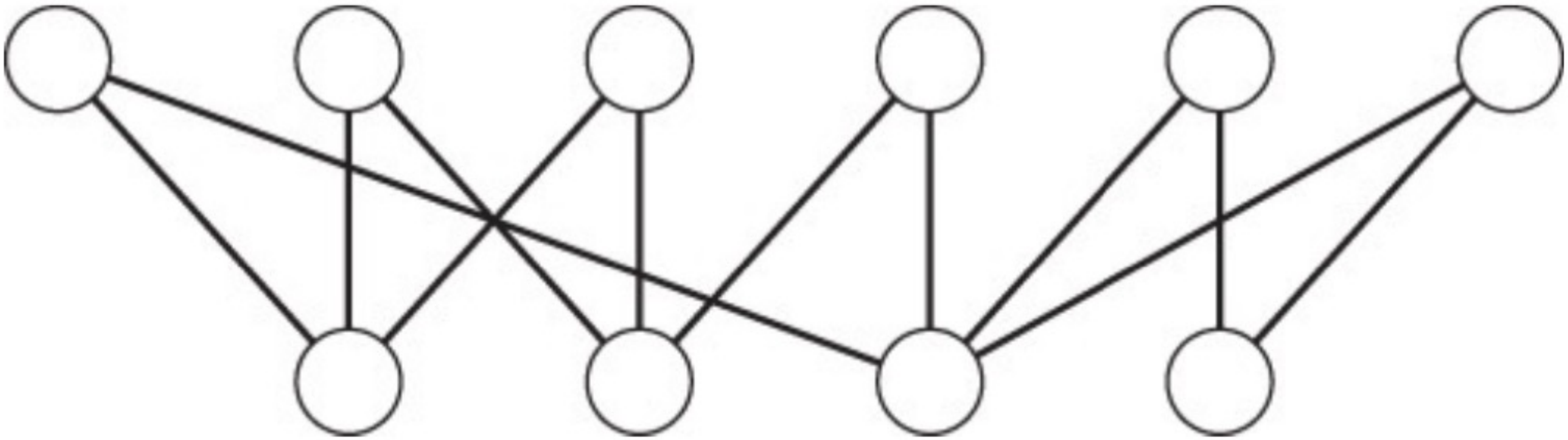
Each vertex has a set of ***incoming*** and ***outcoming*** edges.

In each vertex the sum of the capacities of incoming edges ***cannot be greater*** than the sum of the capacities of outcoming edges.

The unique exception to the previous rule regards the ***source*** and the ***sink*** nodes.

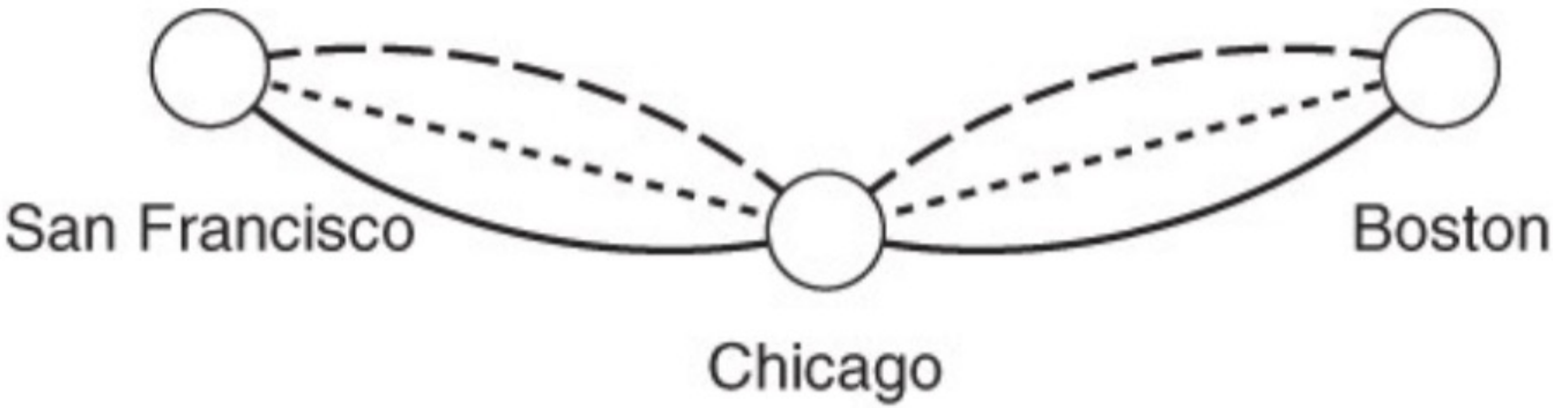
Applications: road systems, transportation networks, networks of storms drains

Bipartite Graph



- There are two ***distinct sets*** of vertices.
- Each vertex is connected only with vertices of ***the other set***.
- It can be used for modeling relationships between students and teachers.

Multigraph



- It is characterized by the presence of ***multiple edges*** between vertices.
- In the example above, each edge may represents a ***different way*** (by plane, by truck, etc.) to shipping item between the cities.
- Each edge can be equipped with ***different sets of properties***.

Suggested Readings

Chapter 13 of the book "*Dan Sullivan, NoSQL For Mere Mortals, Addison-Wesley, 2015*".

Images

If not specified, the images shown in this lecture have been extracted from:

"Dan Sullivan, NoSQL For Mere Mortals, Addison-Wesley, 2015"