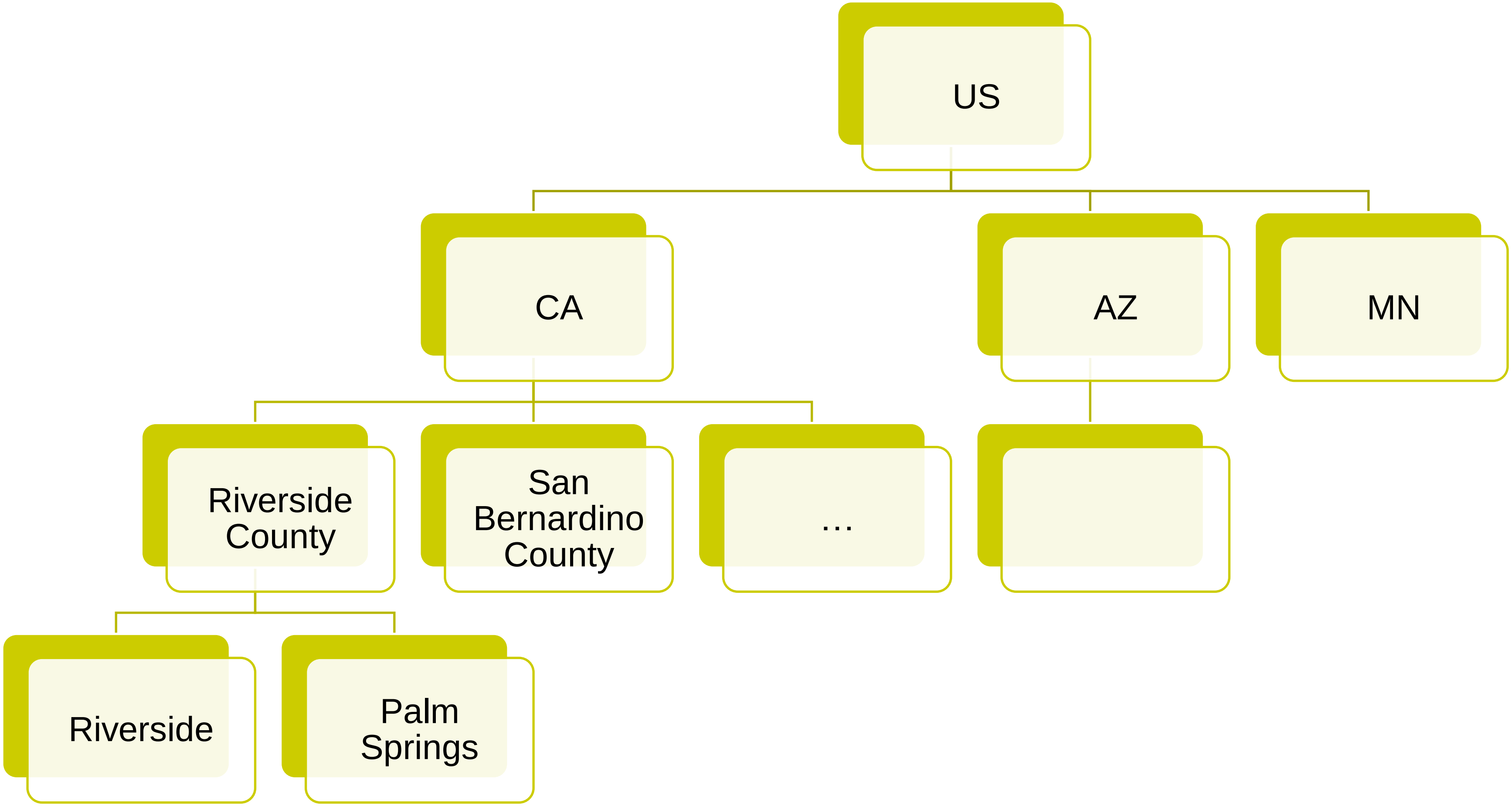
Graphs

Chapter 9

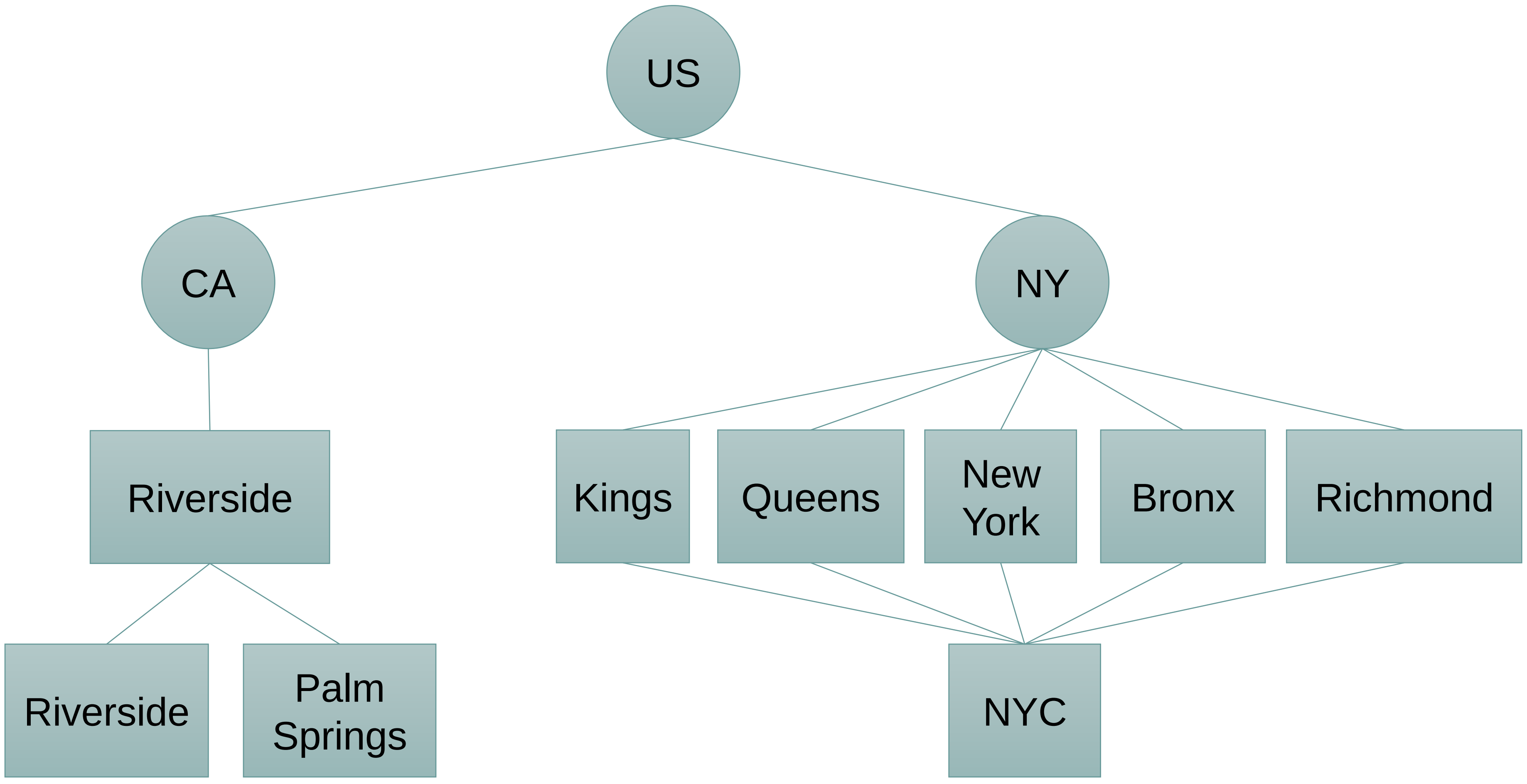
Objectives

- Getting familiar with the graph model
- Understand the basic terminology of a graph
- Recognize the different types of graph
- Understand the graph ADT
- Understand the two common graph representations

Flashback (Trees)



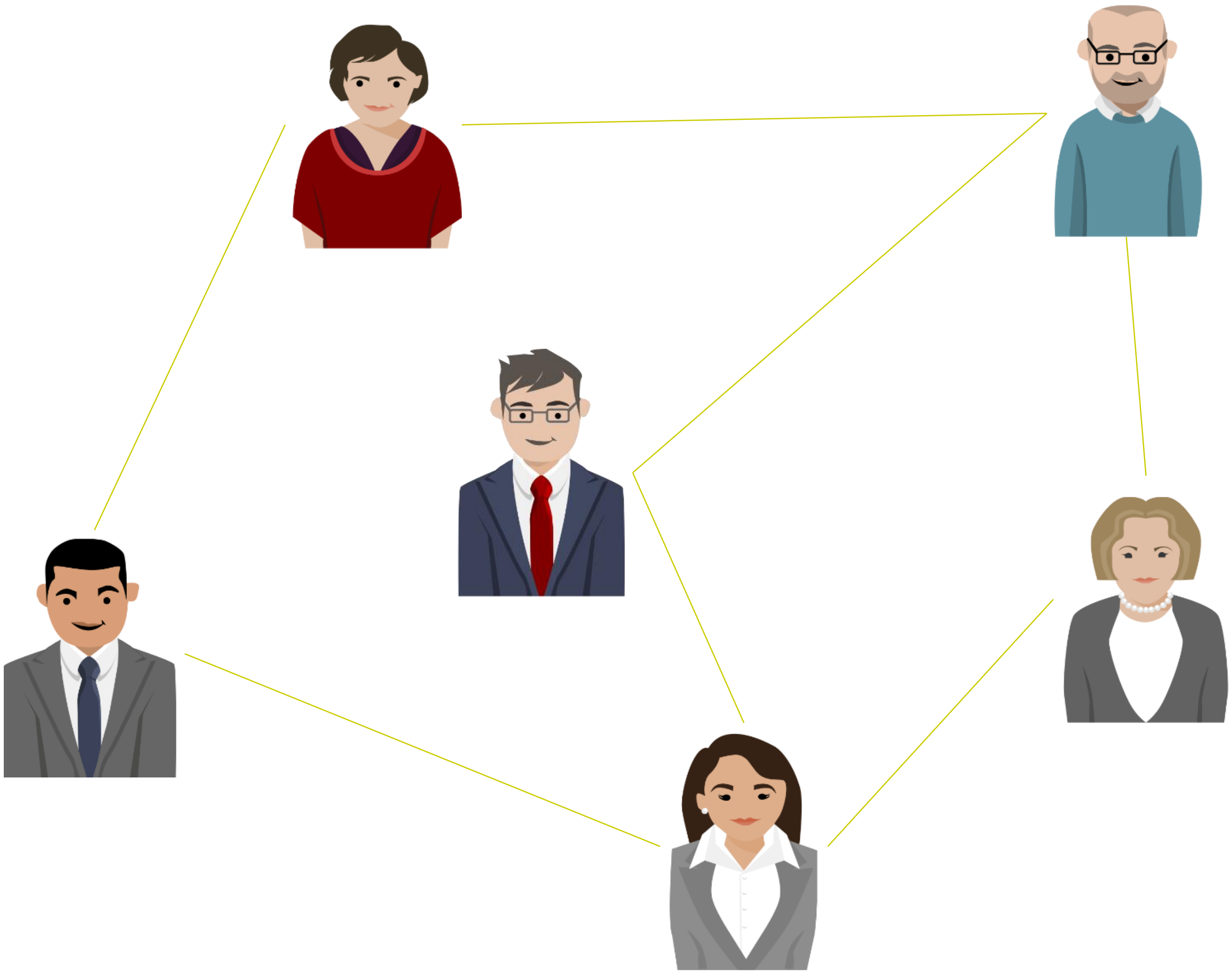
Flash Forward (Graphs)



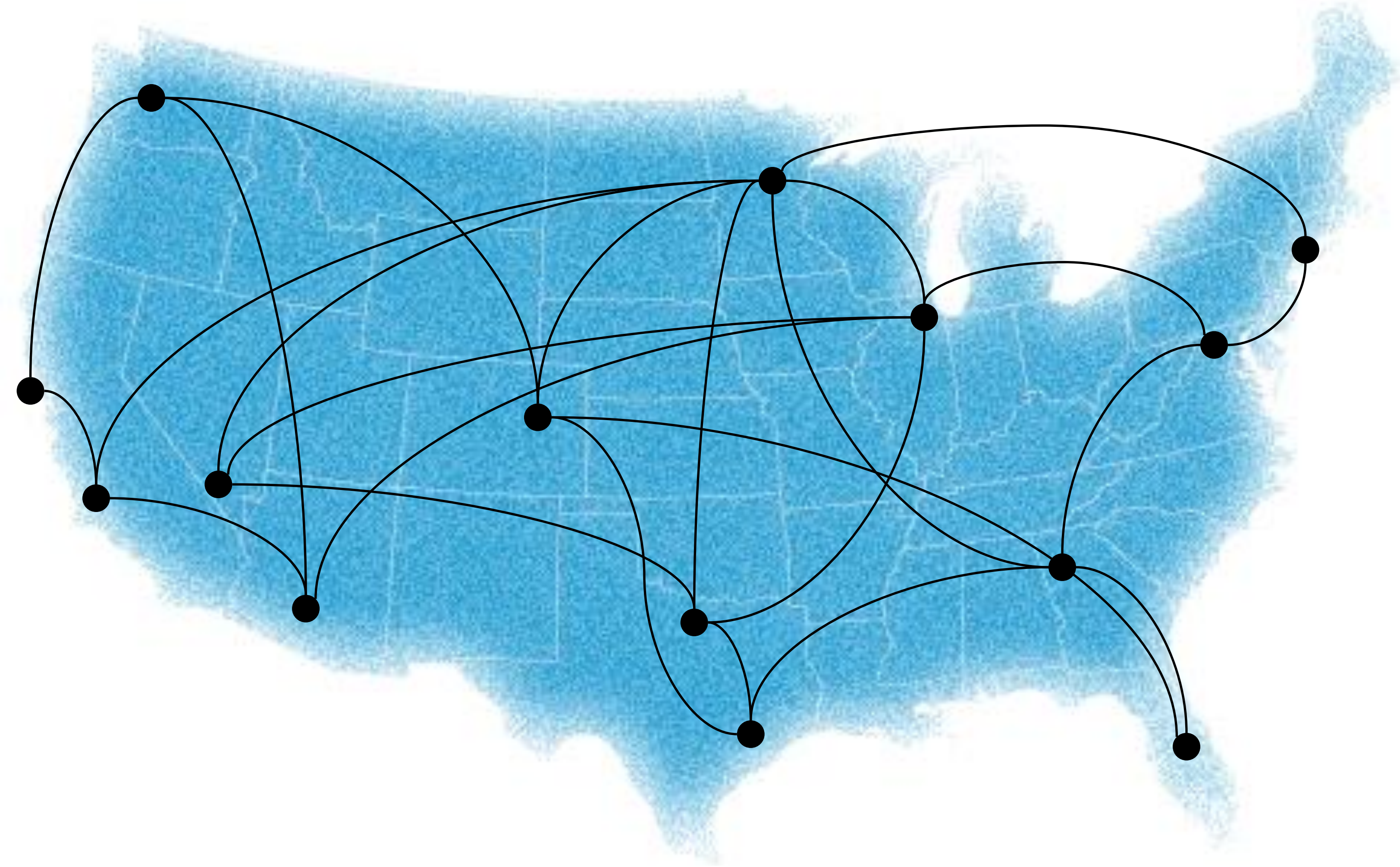
Applications of Graphs

- Networks
  - Social networks
  - Business network
  - Computer networks (even wireless networks)
  - Road networks
- Many-to-many relationships
  - Students and courses
  - Students and departments

Example: Social Network



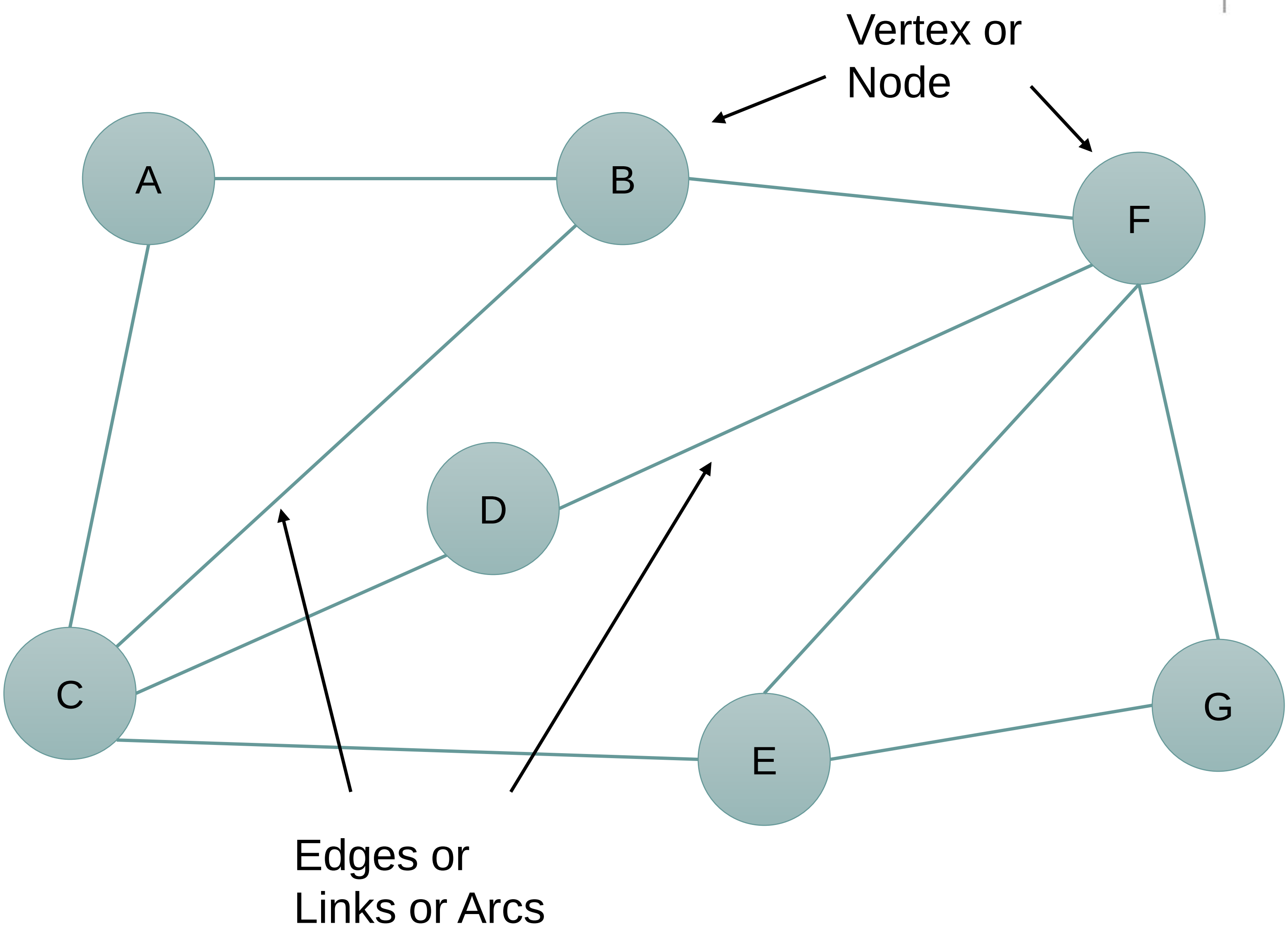
Example: Airport Network



Graph Model

- A Graph (G) consists of a set of Vertices (V) and Edges (E). $G = (V, E)$
- $V = \{v_1, v_2, \dots, v_{|V|}\}$
- $E = \{e_1, e_2, \dots, e_{|E|}\}$
- $e = (v, w), e \in E, v \in V, w \in V$

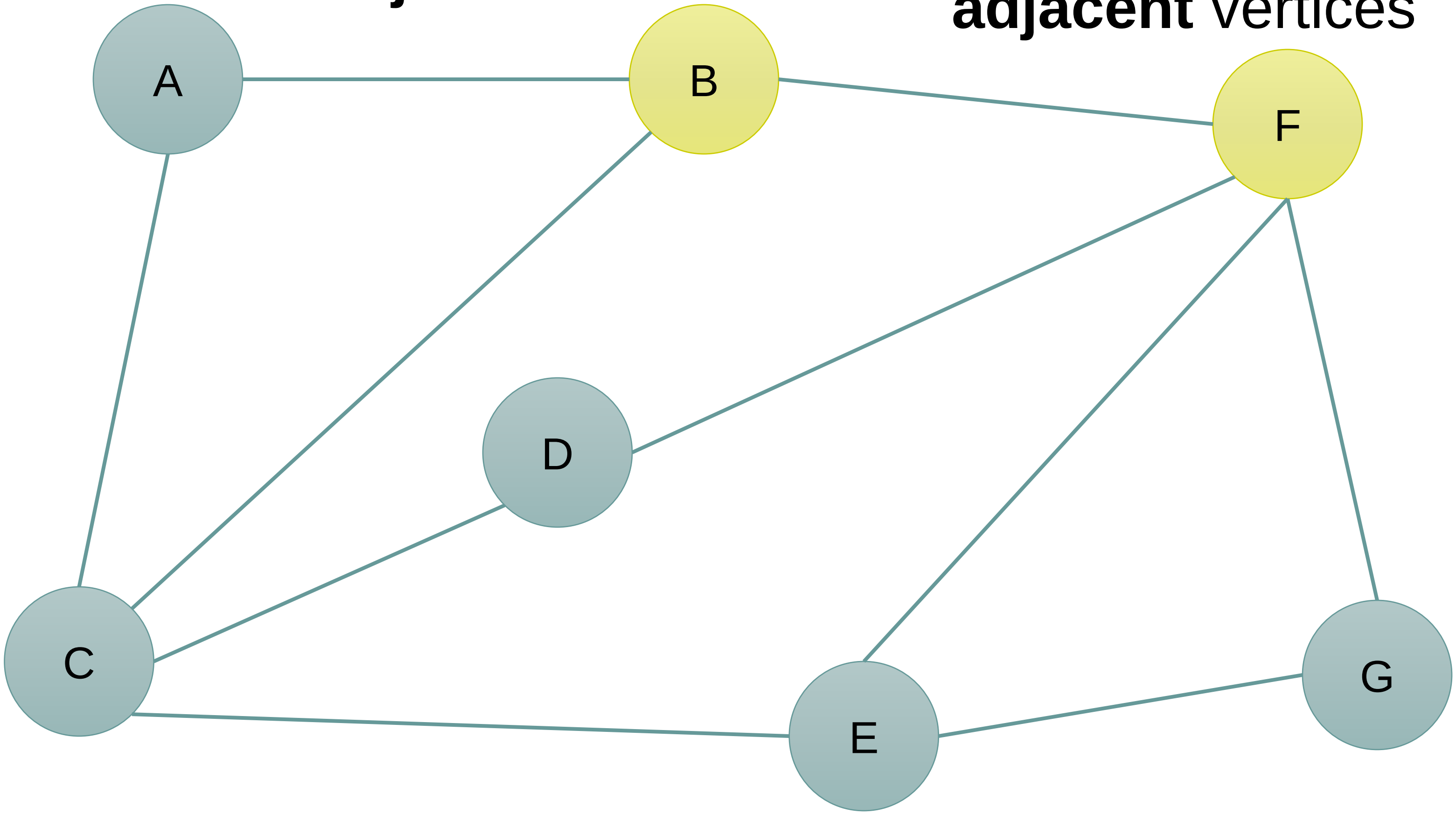
Graph Terminology



Adjacency

Two vertices with an edge connecting them are called **adjacent** vertices

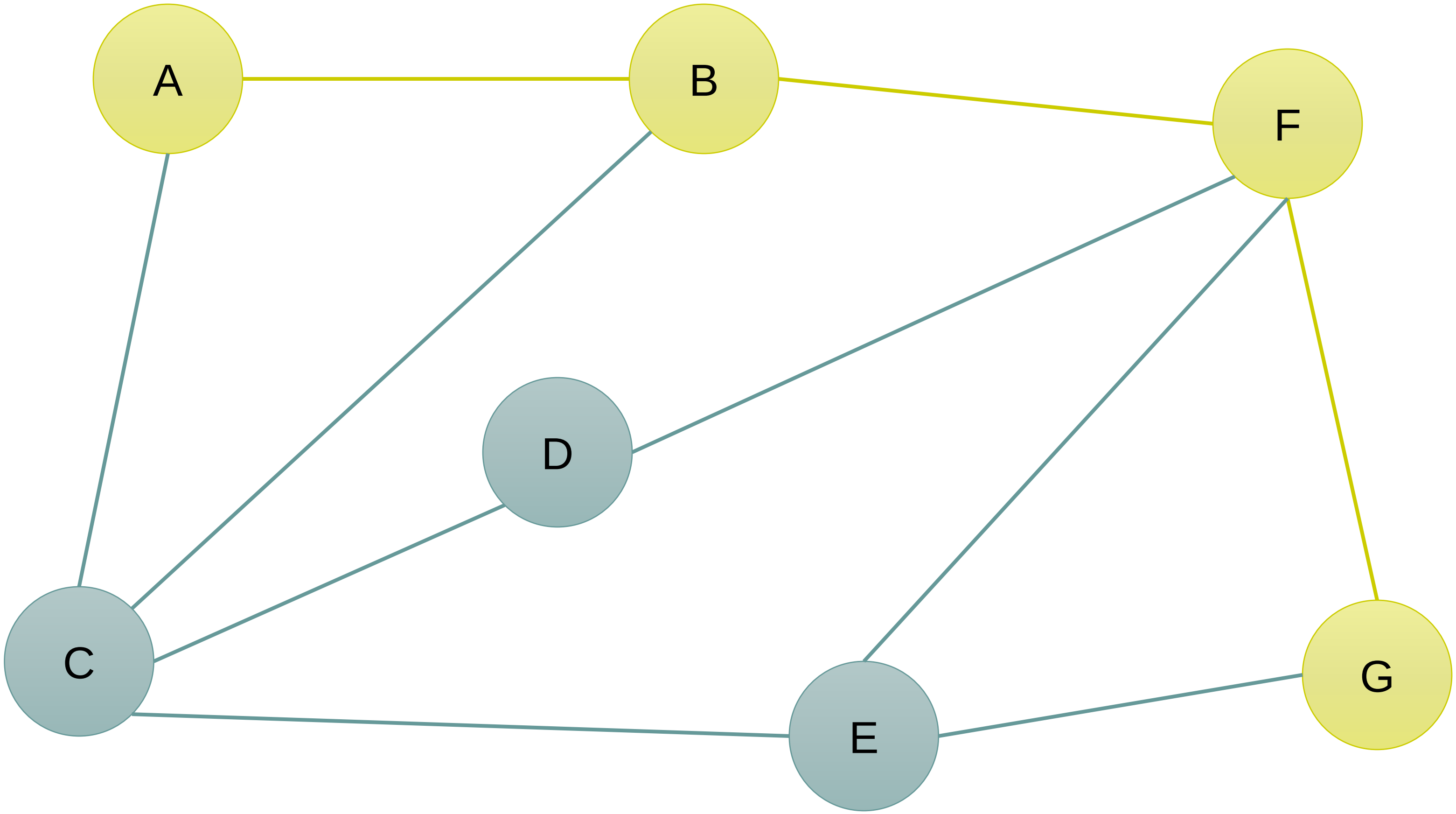
B and F are **adjacent** vertices



All adjacent vertices of a vertex are called **neighbors**

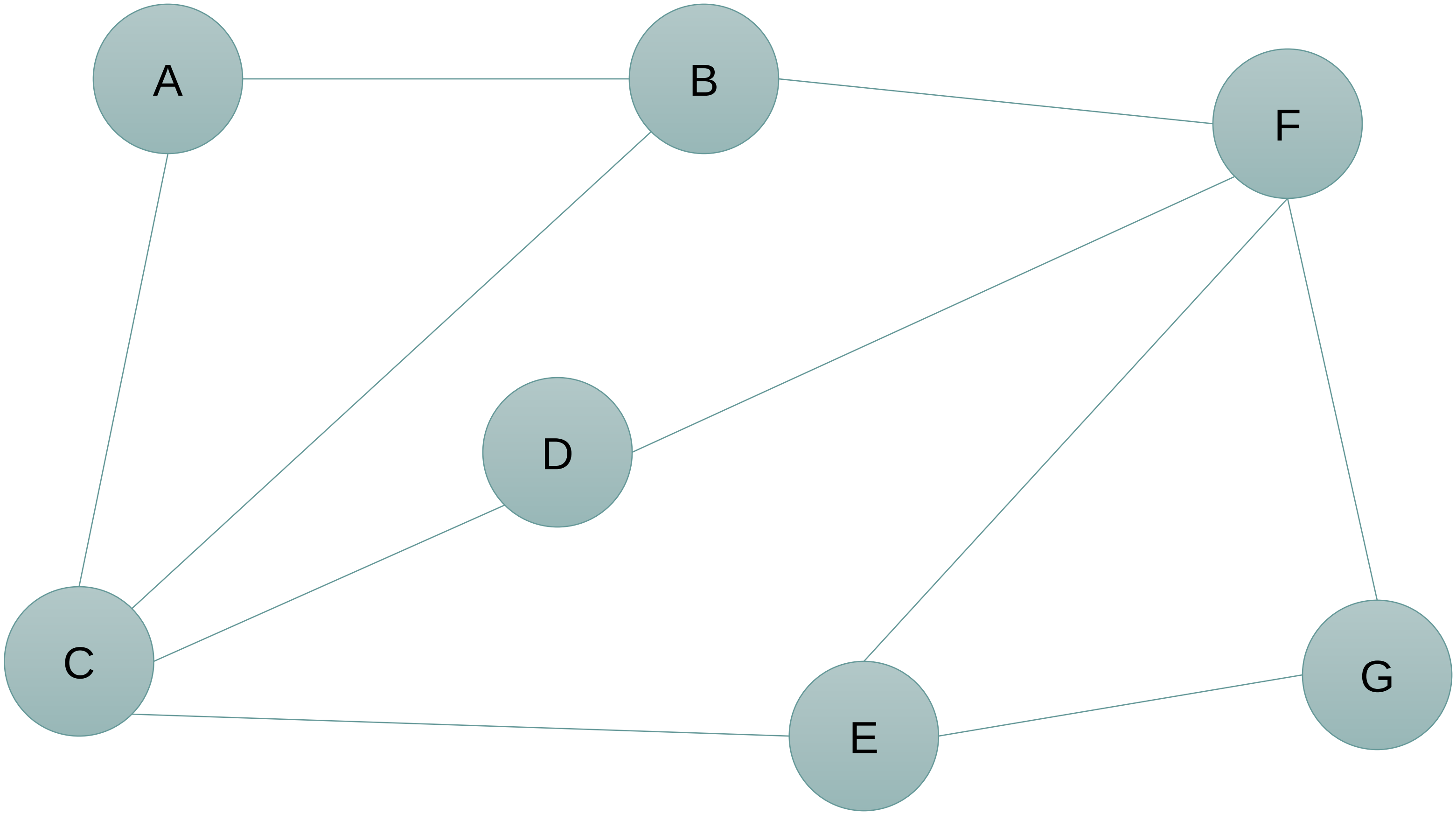
Path

A, B, F, G is a **path** on the graph



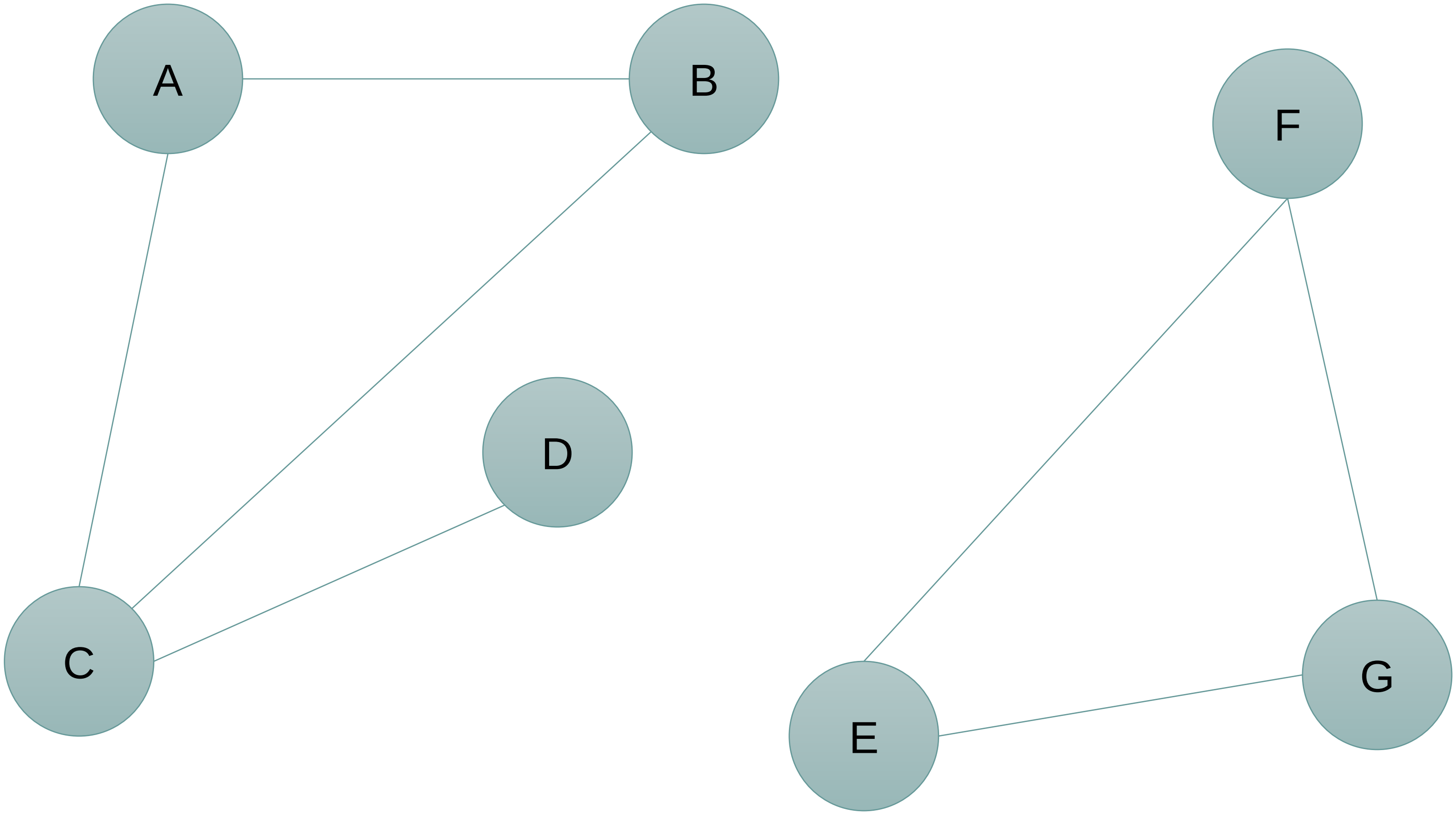
A and G are said to be **connected**

Connected Graphs



A graph is **connected** if every pair of vertices are connected

Unconnected Graphs

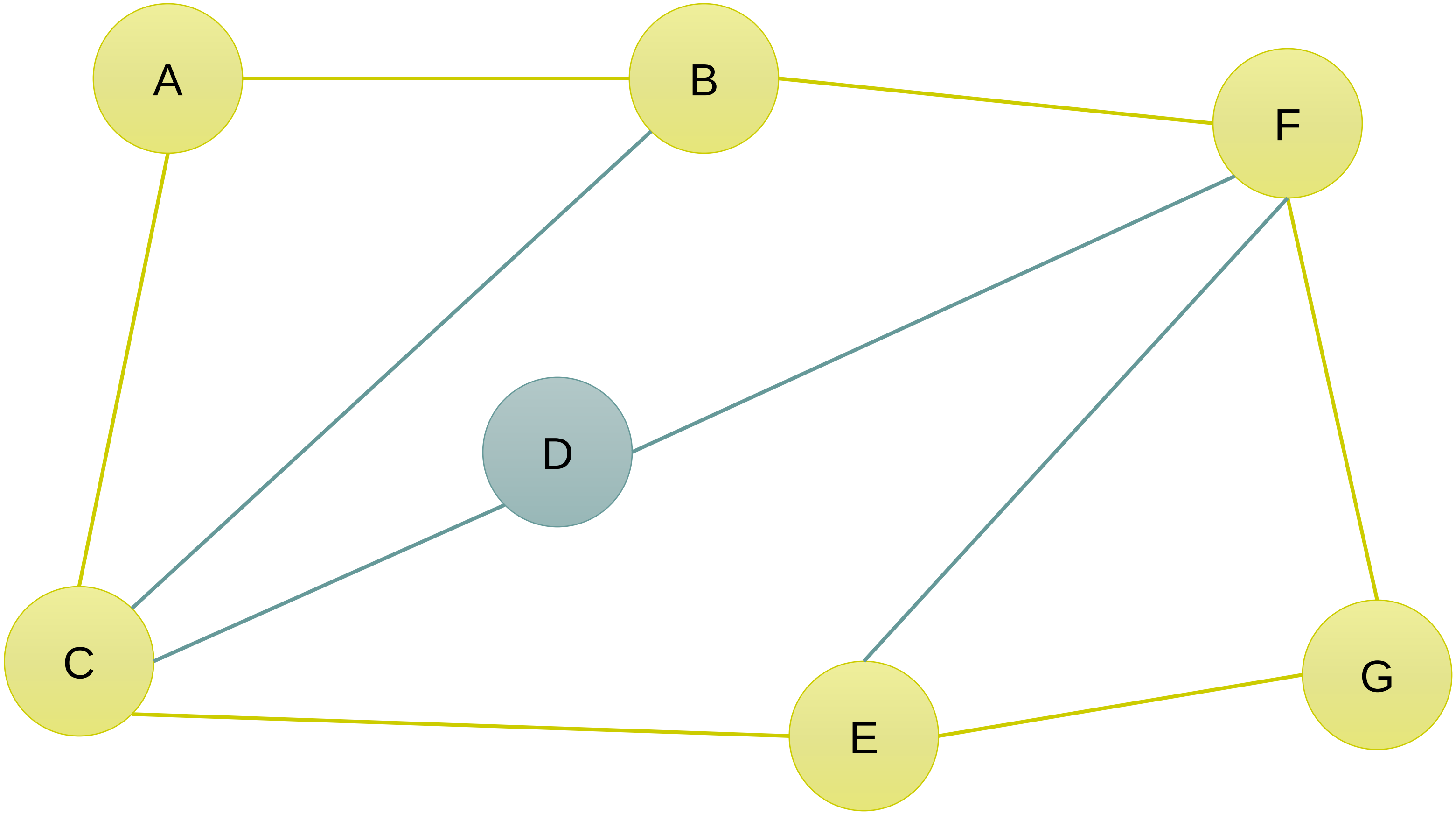


A graph is **unconnected** if there is at least one pair of vertices that are not connected

Cycles

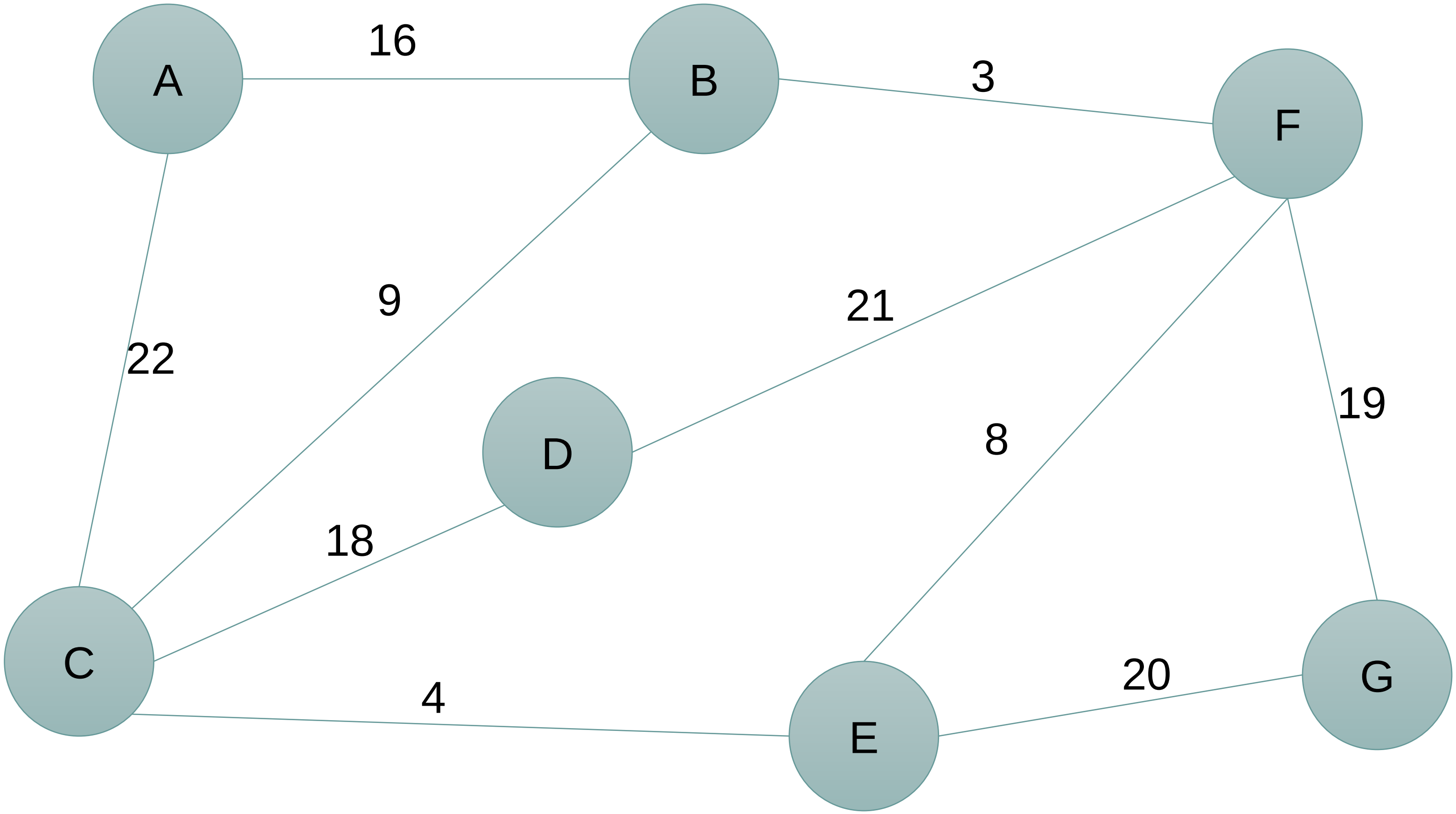
A, B, F, G, E, C, A is a **cycle**

A cycle is a path where the first and last vertices are the same



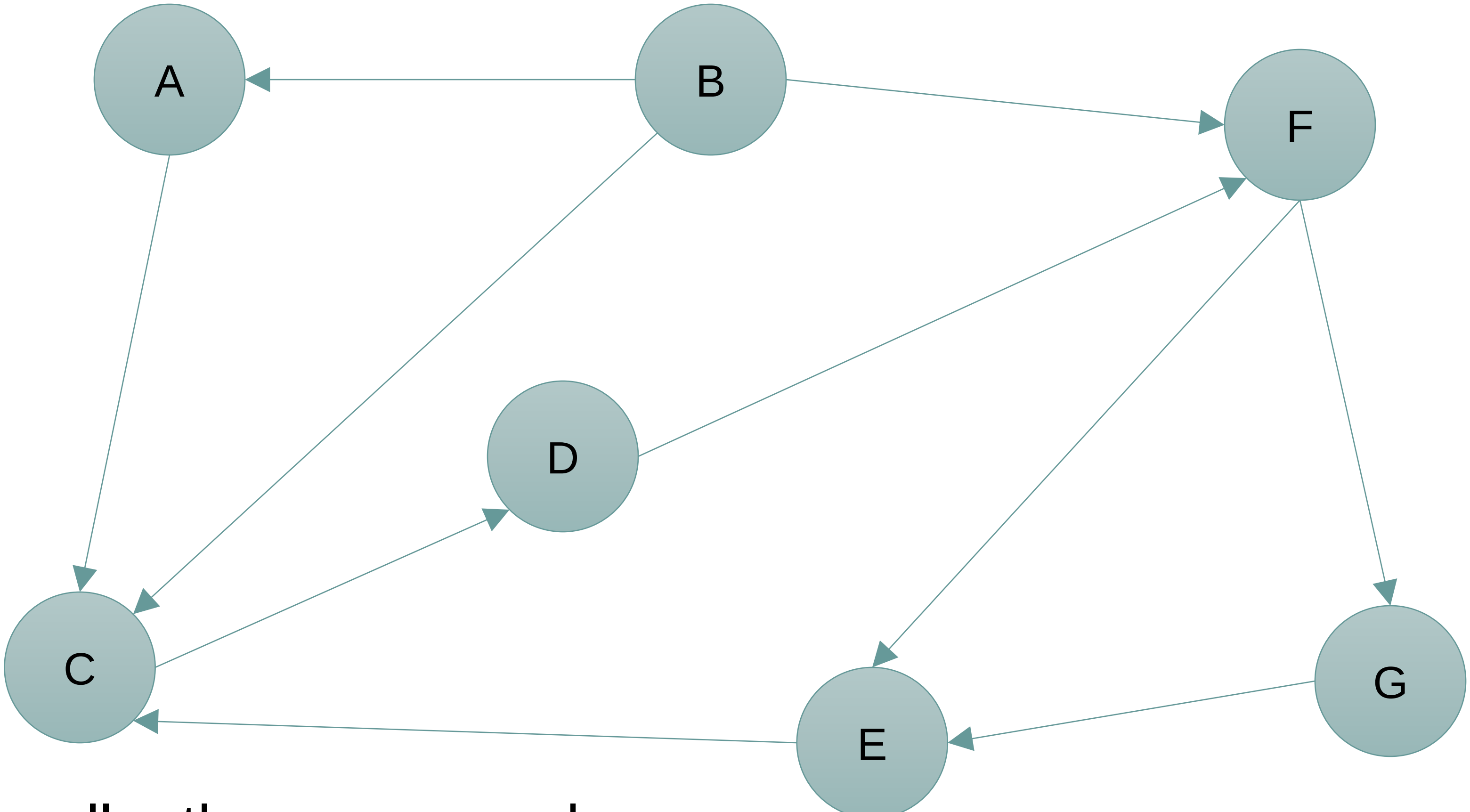
Weighted Graphs

A vertex and/or edge might have an associated **weight** or **cost**



Directed Graphs

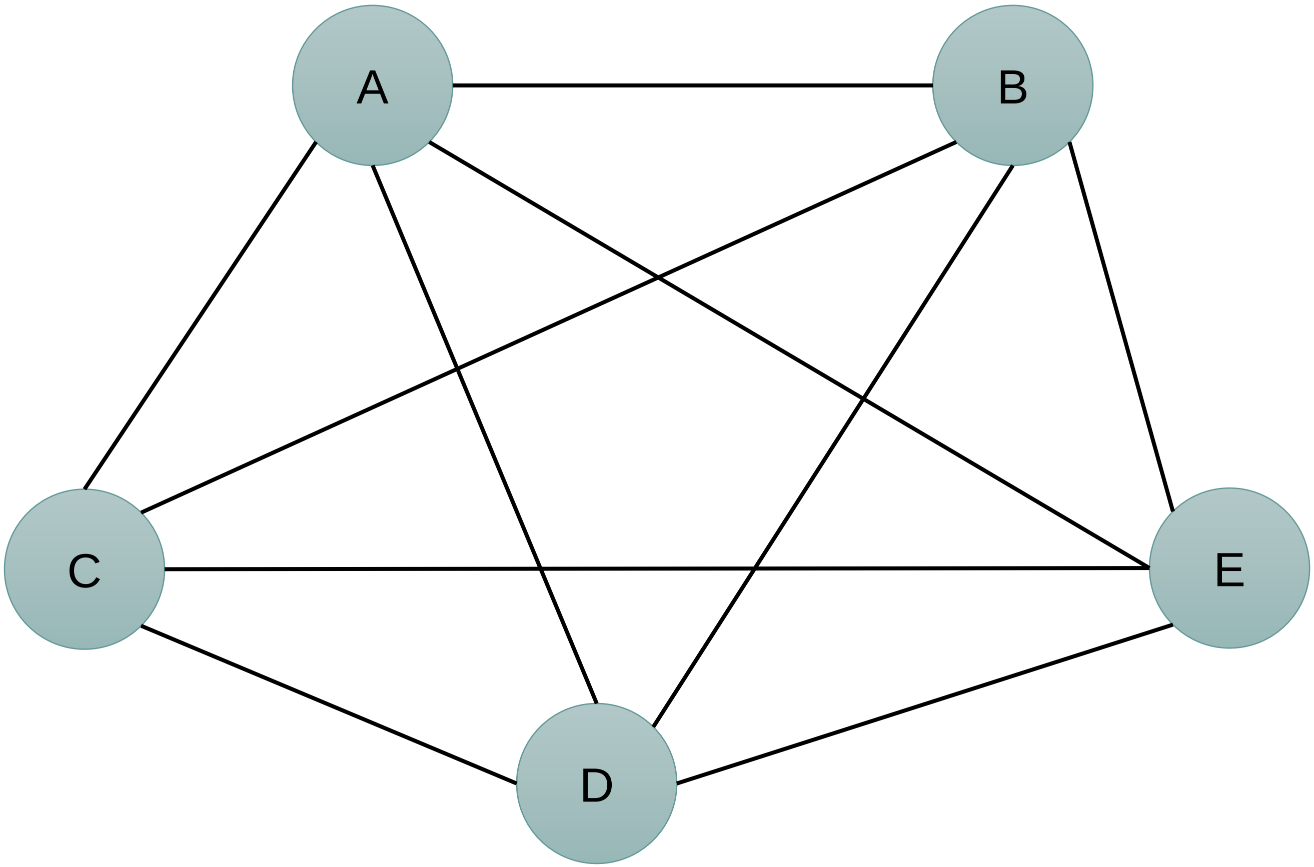
$e = (v, w)$ is an ordered pair



We call v the source and w the destination

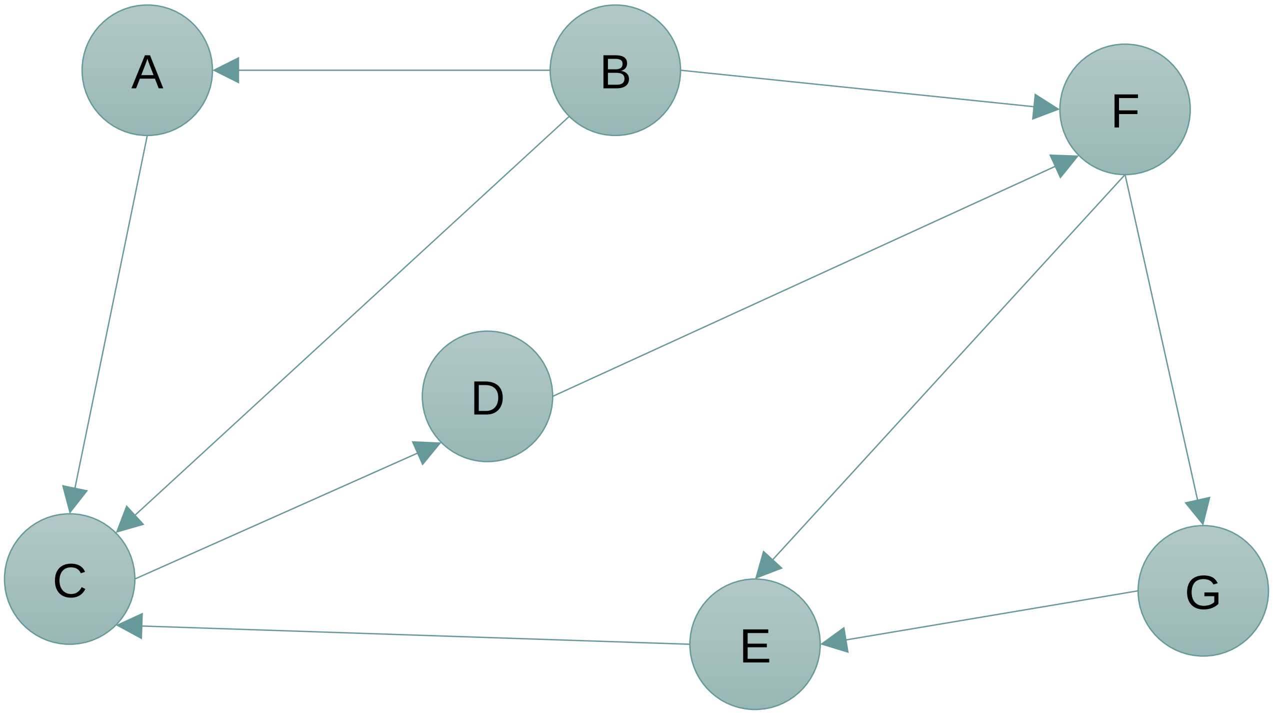
Complete Graph

In a complete graph, there is a direct edge between every pair of vertices



Graph Representation

- Adjacency matrix
- Adjacency list



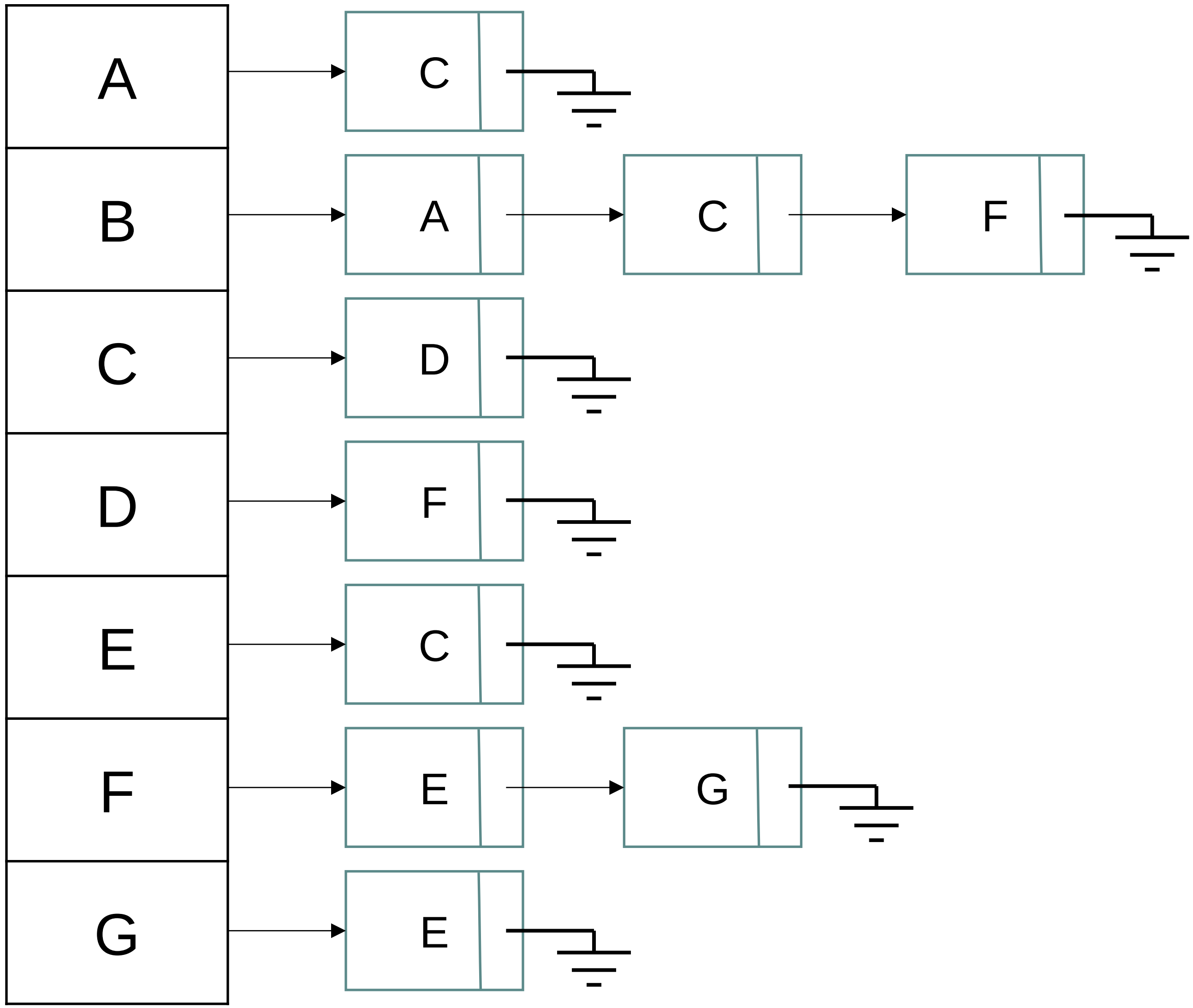
Adjacency Matrix

Destination

Source

	A	B	C	D	E	F	G
A	0	0	1	0	0	0	0
B	1	0	1	0	0	1	0
C	0	0	0	1	0	0	0
D	0	0	0	0	0	1	0
E	0	0	1	0	0	0	0
F	0	0	0	0	1	0	1
G	0	0	0	0	1	0	0

Adjacency List



Undirected Graph

- For undirected graphs, we usually store an undirected edge $e = (v, w)$ as two directed edges $e_1 = (v, w)$ and $e_2 = (w, v)$