Graph Theory

2^{nd} year computer science L2

Pr. Cherif Foudil

Computer Science Department

Biskra University

2023

Course program

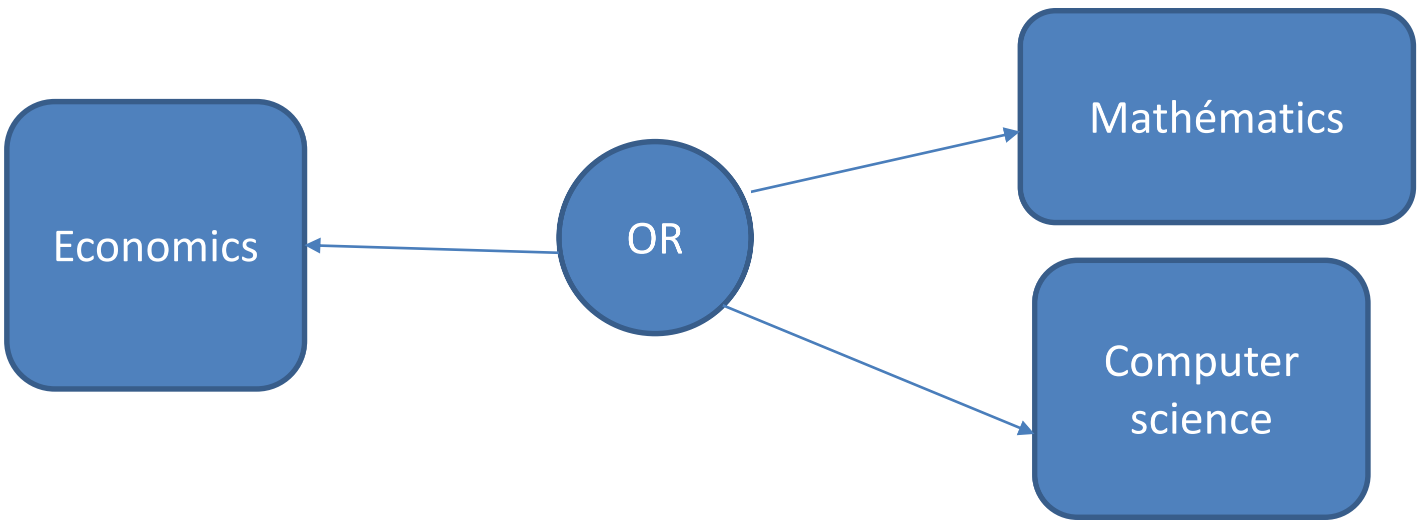
- Basic definitions
- Connectivity in a graph
- Notable paths
  - Hamiltonien et Eulérien
  - Hamiltonian and Eulerian
- Trees and arborescences
- Pathfinding
- Maximum flow problem

Chapter 1

Basic definitions

1.1 Motivation

- **Graph theory** (GT) falls within the field of **operations research** (OR), which serves as a crossroads where **economics**, **mathematics**, and **computer science** intersect.

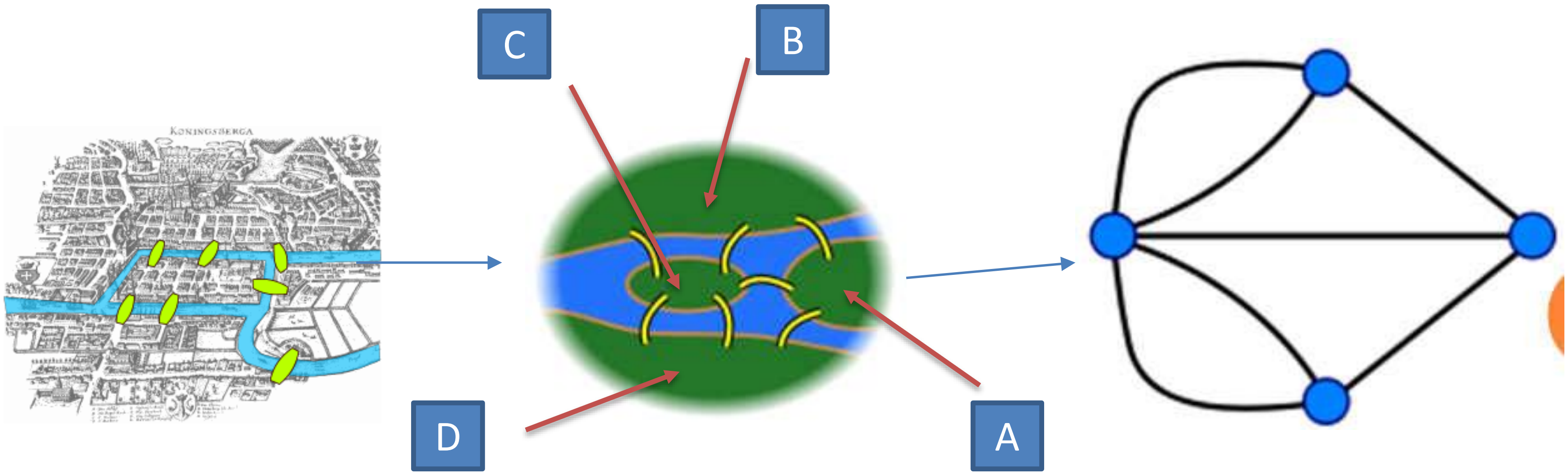


- The objective of **OR** is to find **optimal solutions** for economic problems using mathematical methods that can be programmed by computer.

1.2 History

Question:

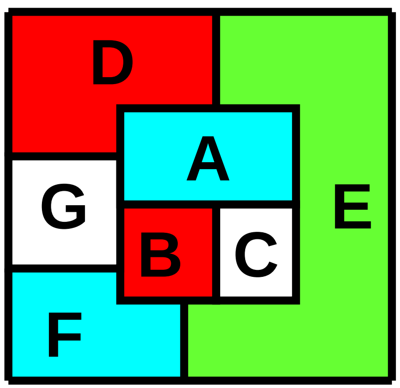
- Is it possible to walk through the city of Königsberg in such a way as to cross each bridge exactly one time and return to the starting district?

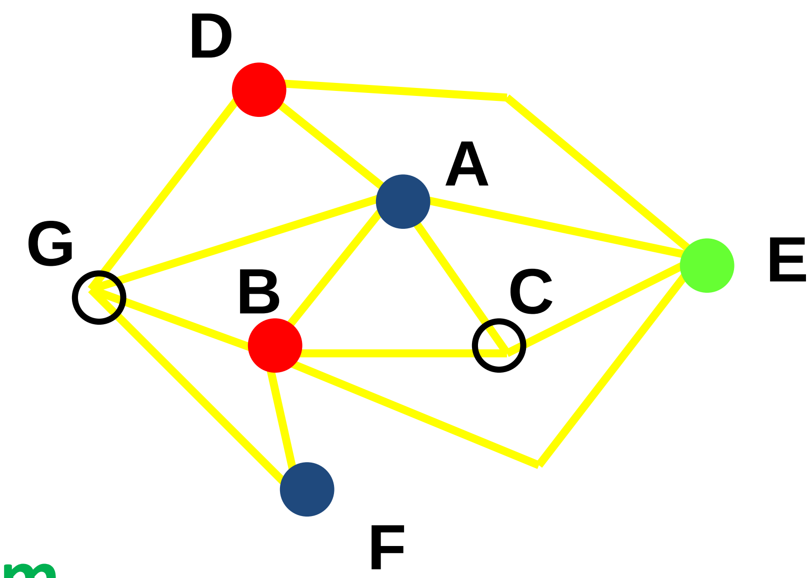


- The German mathematician **L. Euler** (1736) provided an answer to the problem faced by the residents of the city of Königsberg: how to cross the seven bridges of this city without ever crossing the same one twice. → **This marked the birth of graph theory**.

1.2 History

- In 1852, graph theory gained popularity thanks to the "**Four-Color Theorem.**"
- It was demonstrated that only four different colors are needed to color any geographical map in such a way that two adjacent regions (sharing a common boundary) always receive distinct colors.





- **The four-color problem...**

1.2 History

- Starting in 1946, GT (Graph Theory) experienced significant development thanks to researchers motivated by solving practical problems. Among them:
- **Edsger Dijkstra (1959**) for the pathfinding problem,
- **Ford and Fulkerson (1956)** for the maximum flow problem,
- **Bernard Roy (1958)** developed the MPM method for the scheduling problem.

1.2 History

- Graph theory has become essential in our daily lives for various networks:
- **Transportation networks**: road, air, rail, maritime, water, gas, electricity...
- **Data communication networks**: landline, mobile, WIFI...
- **Information networks**: databases, the web, social networks...

1.3 Graph concepts

1.3.1 definitions

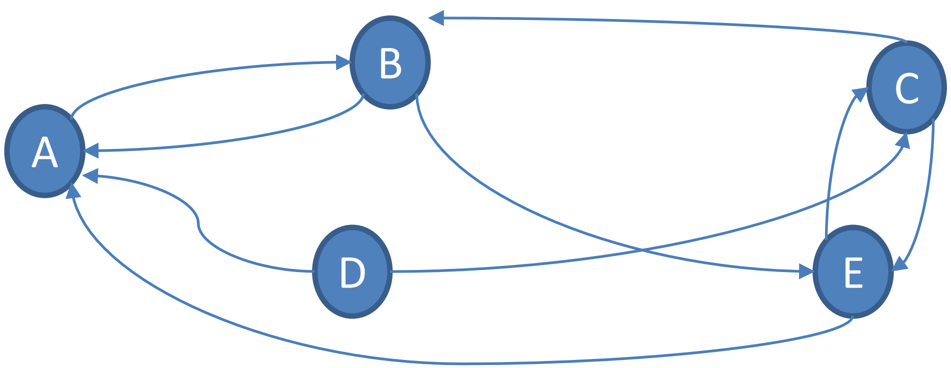
-A graph G, denoted as G=(V, E), is defined as:

- A finite set of vertices V = { v1,v2,…vn }
- A finite set of edges (or arcs) E, where an edge (or arc) connects a pair of vertices from V, $E = \{ e_1, e_2, \ldots e_m \}$

Each edge has two endpoints

It is not easy to visualize a given graph in this form → drawing it is more helpful.

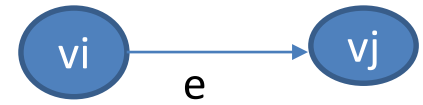
Example: A one-way traffic plan of a city, where each locality (vertex) and each road are represented by a directed arc indicating the direction of traffic.



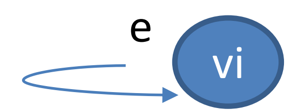
1.3 Graph concepts

1.3.2 Directed graph

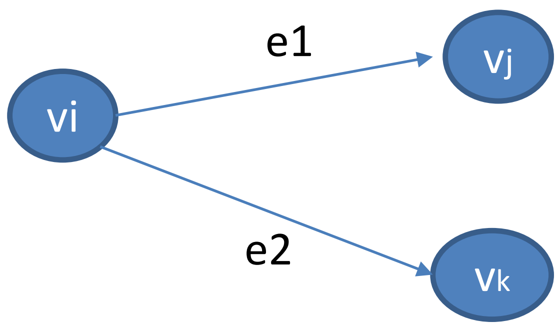
A directed graph is a graph in which if e = (v_i, v_j), v_i is the origin of the arc e, and v_j is the destination of e.



A loop, denoted as b = (v_i, v_i), is an arc where the origin coincides with the destination.



Two arcs are adjacent if they have at least one common endpoint.

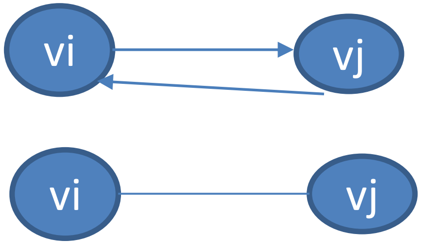


e_1 and e_2 are adjacent; they have v_i in common

1.3 Graph concepts

1.3.2 Undirected graph

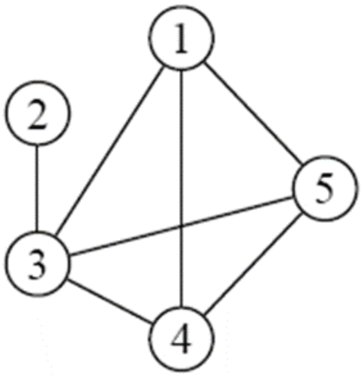
An undirected graph is a graph in which for all v_i, $v_j \in V$, if $(v_i, v_j) \in E$, then $(v_j, v_i) \in E$.



Example: $G= (V, E)$,

$V= (1, 2, 3, 4, 5)$ $E= \{(1, 5), (1, 4), (1, 3), (2, 3), (3, 4) ; (4, 5), (3, 5)\}$.

The graph G is represented in a schematic way as follows:



1.3 Graph concepts

1.3.2 Undirected graph

An edge e in E is defined by an unordered pair (v_1, v_2) of vertices called the endpoints of e.

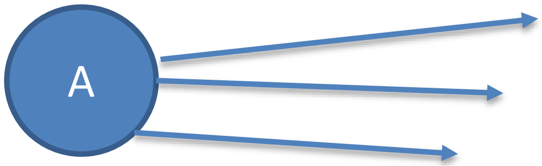
If the edge e connects vertices v_1 and v_2, we can say that these vertices are adjacent or incident to e, or that the edge e is incident to vertices v1 and v2.

The order of a graph is the number of vertices **n** in that graph.
$|V| = n$

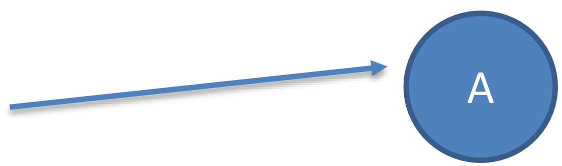
1.3 Graph concepts

1.3.3 The degree of a vertex:

- The vertex A is the initial endpoint of 3 arcs; we say that the positive degree of A is equal to 3, denoted by: $\mathbf{d^{+}_{G}(A) = 3}$



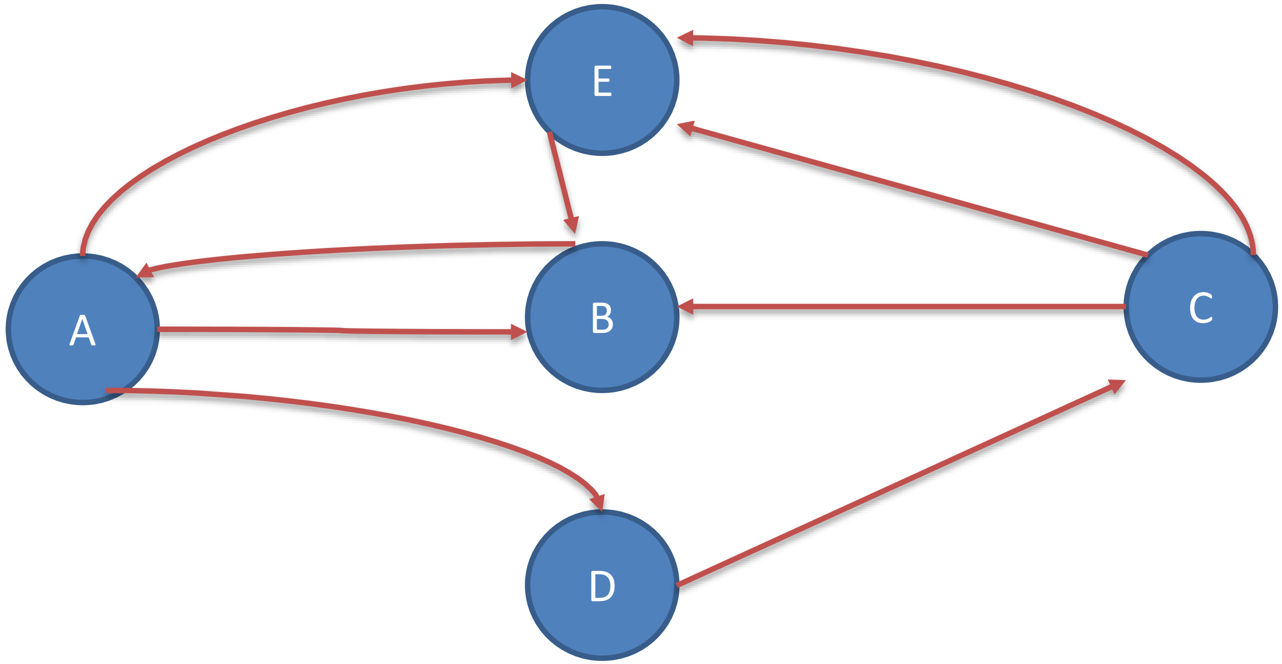
- The vertex A is the terminal endpoint of one arc; we say that the negative degree of A is equal to 1, denoted by: $\mathbf{d^{-}_{G}(A) = 1}$



- The degree $\mathbf{d_G(A) = d^{+}_{G}(A) + d^{-}_{G}(A)}$

1.3 Graph concepts

1.3.3 The degree of a vertex : Example



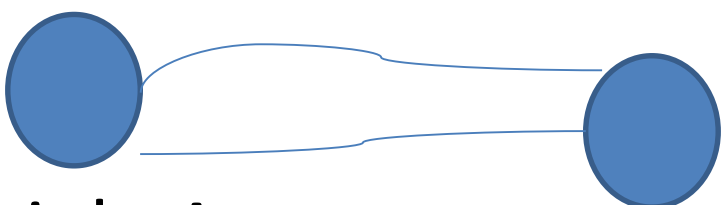
	A	B	C	D	E	
$d^+_G(x)$	3	1	3	1	1	9
$d^-_G(x)$	1	3	1	1	3	9
$d_G(x)$	4	4	4	2	4	18

1.3 Graph concepts

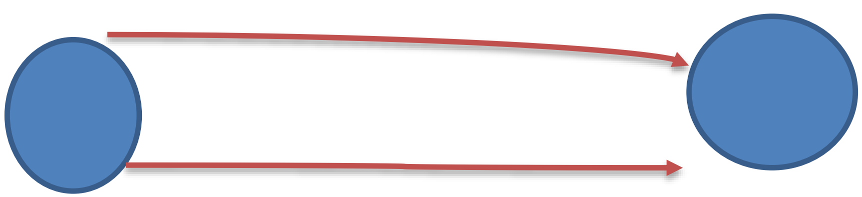
1.3.4 Types of graphs

a) Multiple Graph: G = (V, E) is a graph in which the set E of edges may contain more than one edge connecting two given vertices.

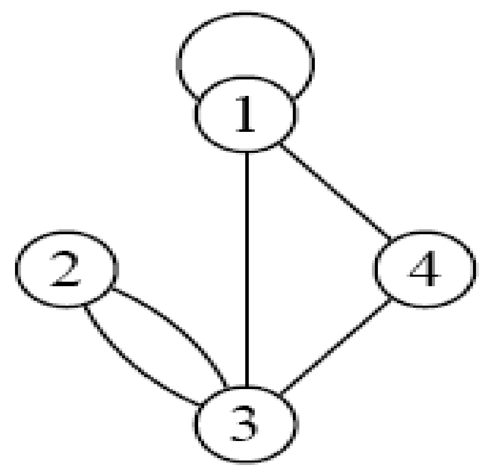
Multiple Edges



Multiple Arcs



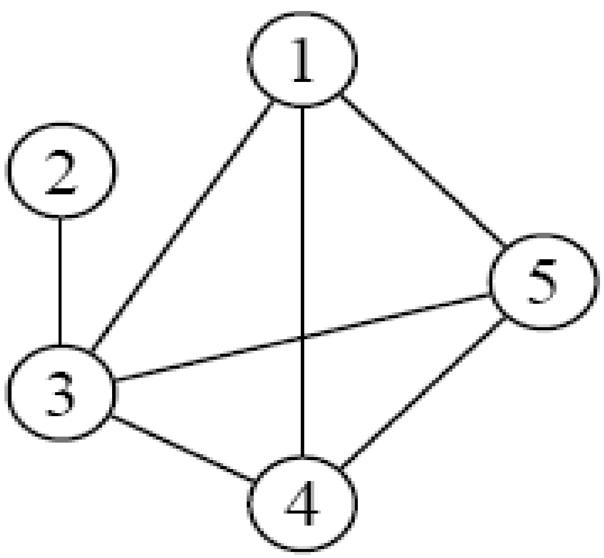
multiple graph



1.3 Graph concepts

1.3.4 Types of graphs

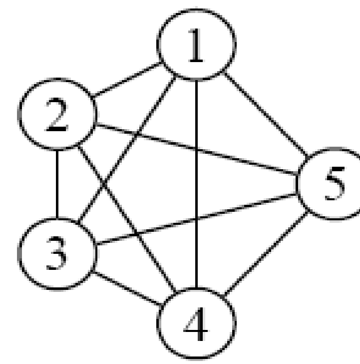
b) Simple Graph: A graph without loops or multiple edges (arcs).

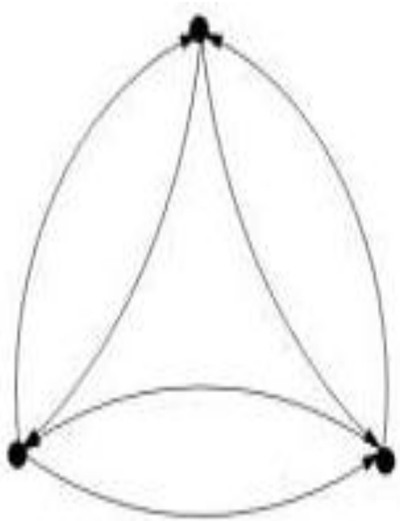


1.3 Graph concepts

1.3.4 Types of graphs

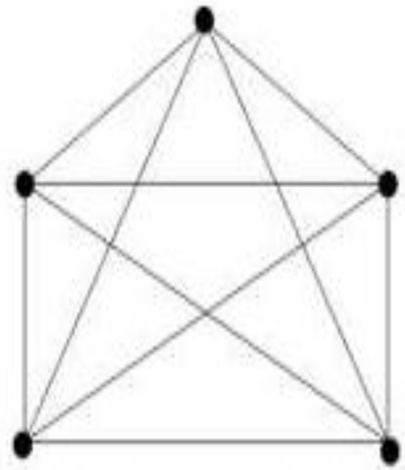
c) Complete Graph: A graph in which every vertex is directly connected to every other vertex.





Graphe complet
(orienté)
sur trois sommets

Graphe complet
(non-orienté)
sur trois sommets



Graphe complet
(non-orienté)
sur cinq sommets

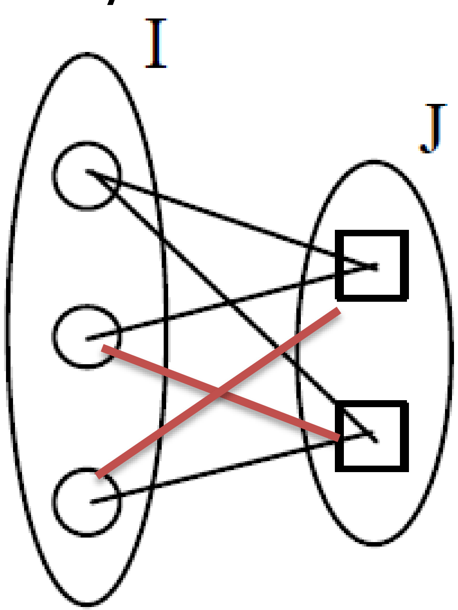
1.3 Graph concepts

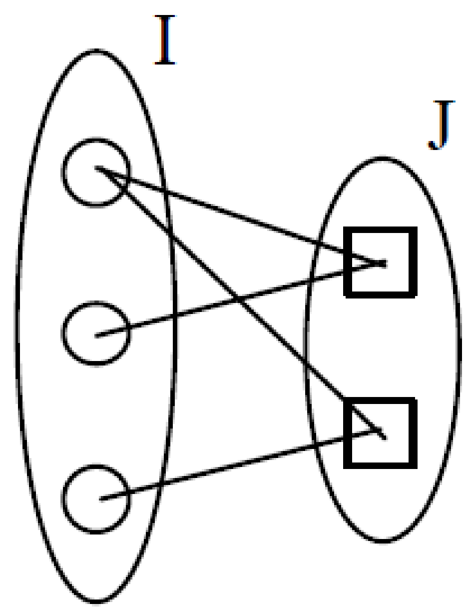
1.3.4 Types of graphs

d) Bipartite Graph: If its set of vertices can be divided into two distinct subsets I and J such that each edge has one endpoint in I and the other in J.

e) Complete Bipartite Graph :

If every vertex in set I is connected to every vertex in set J.

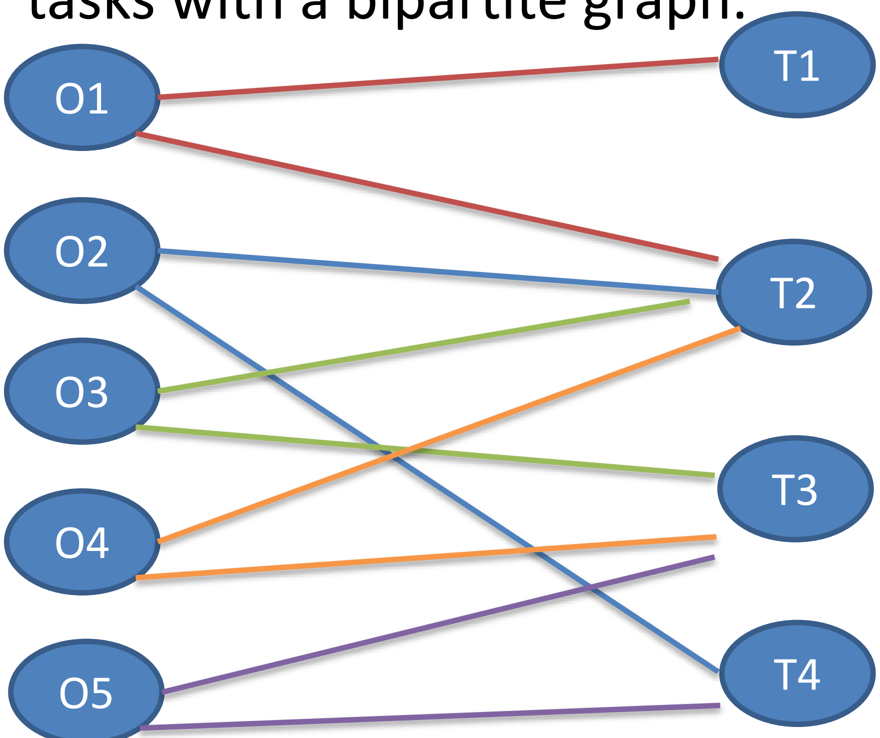




1.3 Graph concepts

1.3.4 Types of graphs

d) Example of a Bipartite Graph: In a workshop with 5 workers, each capable of performing 1 to 4 tasks, we represent the possibilities of assigning workers to different tasks with a bipartite graph.



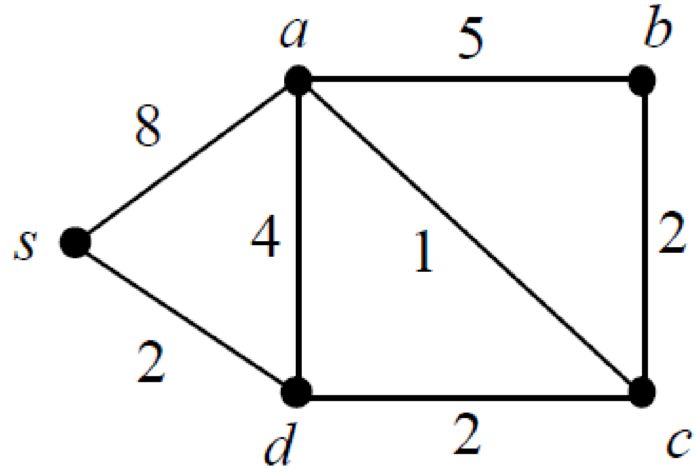
Note: If each worker can perform all tasks, a **complete bipartite graph** is obtained.

1.3 Graph concepts

1.3.4 Types of graphs

f) Weighted Graph:

When the edges represent a cost, they are assigned a number, creating a **weighted graph**.

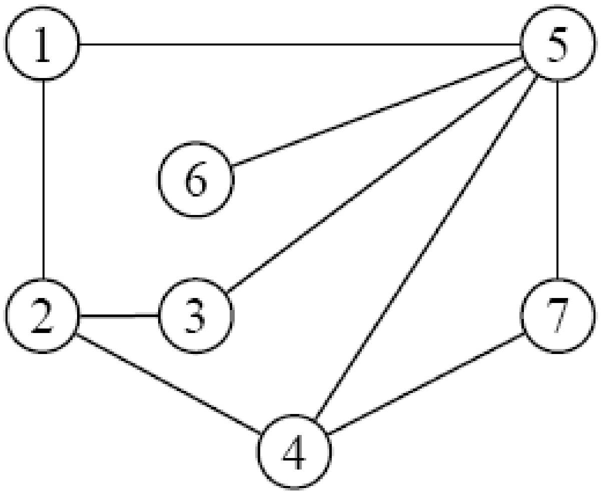


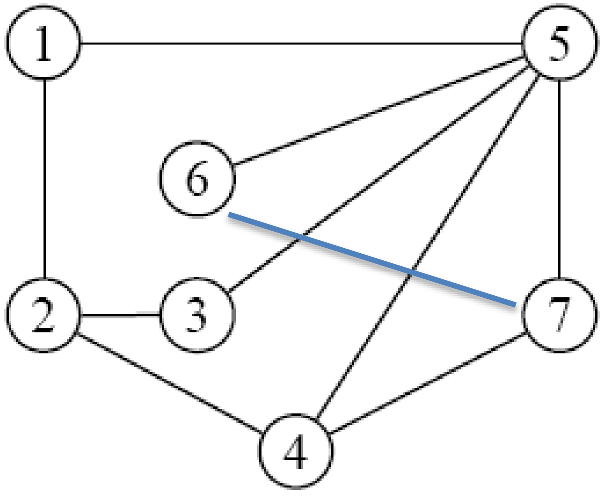
1.3 Graph concepts

1.3.4 Types of graphs

g) Planar Graph:

If its edges do not intersect, in other words, if it can be drawn in a plane in such a way that its edges do not cross.

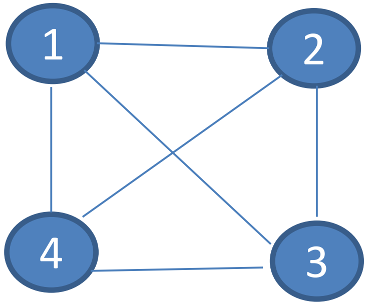


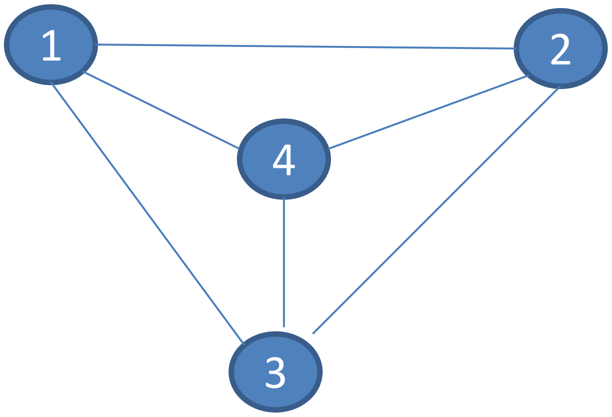


1.3 Graph concepts

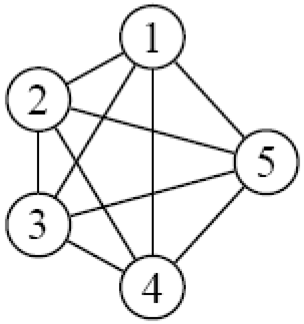
1.3.4 Types of graphs

Example 1: Complete graph with 4 vertices, it is planar if it can be transformed.



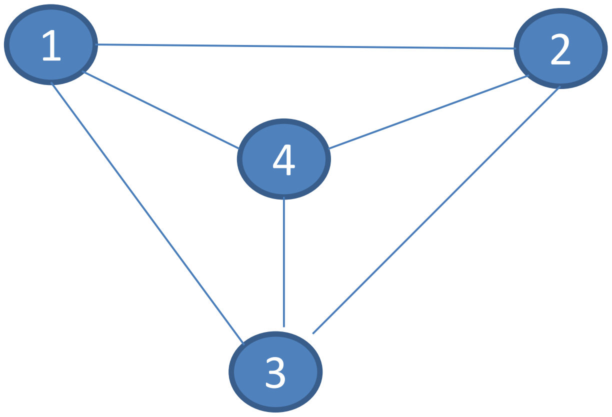


Example 2: Complete graph with 5 vertices, it is not planar, it cannot be transformed.



1.3 Graph concepts

1.3.4 Types of graphs G=(V,E)



Note: Planar graphs satisfy the formula(Euler formula):

$|V| + F = |E| + 2$ **4 + 4 = 6 + 2**

$|V|$ number of vertices

$|E|$ number of edges

F number of faces or regions

1.3 Graph concepts

1.3.4 Types of graphs

h) Other graphs:

- A **reflexive** graph is a graph having a loop on each vertex
- A graph G = (V, E) is **symmetric** if,

  $\forall$ arc $e_1 = (v_i, v_j) \in E$, the arc $e_2 = (v_j, v_i) \in E$.

- A graph G = (V, E) is **antisymmetric** if,

  $\forall$ (For every) arc $e_1 = (v_i, v_j) \in E$, the arc $e_2 = (v_j, v_i) \notin E$.

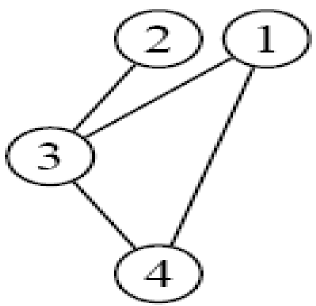
- A graph G = (V, E) is **transitive** if,

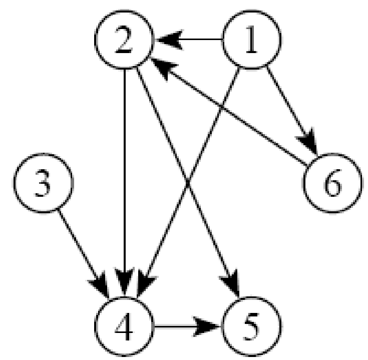
  $\forall$ arc $e_1 = (v_i, v_j) \in E$ and arc $e_2 = (v_j, v_k) \in E$ then the arc $e_3 = (v_i, v_k) \in E$.

1.4 Representation of a graph

a) Sagittal representation (drawing):

Vertices are represented by circles, and the relationships are represented by lines or arrows,





b) **Matrix representation.**

For a given graph G=(V,E) with |V|=n and |E|=m (n vertices and m edges), we associate 4 types of matrices:

Adjacency matrix, Incidence matrix, Arc matrix, Associated matrix

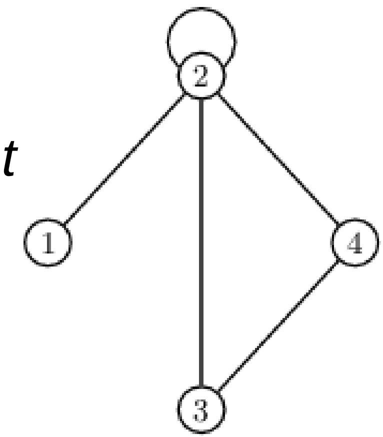
1.4 Representation of a graph

1- Adjacency Matrix: (undirected graph):

Let G be an undirected graph with n vertices numbered from 1 to n. The adjacency matrix of the graph is called matrix A=(ai,aj), where $a_{i,j}$ is the number of edges connecting vertex i to vertex j.

Example: a graph and its corresponding adjacency matrix:

*Other definition: A square matrix where each row and column represent a vertex, and the entries indicate whether there is an edge between the vertices. It's often used for **simple graphs.***



$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

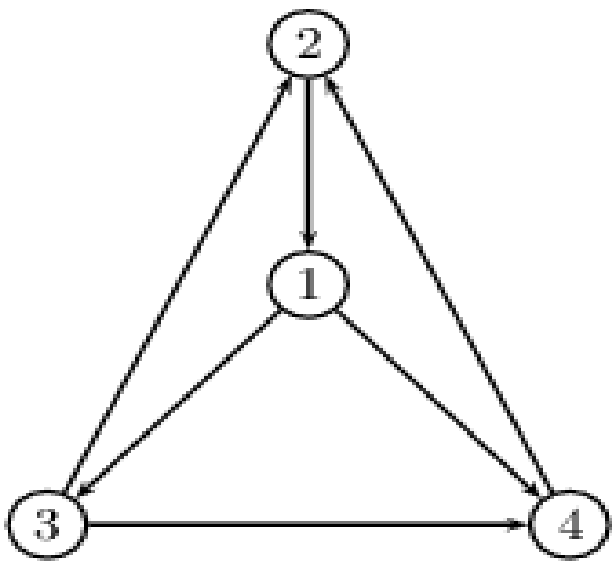
It is a symmetric matrix

1.4 Representation of a graph

Adjacency Matrix: (directed graph)

We can also define the adjacency matrix of a directed graph. This time, the coefficient $a_{i,j}$ represents the number of arcs originating from vertex i and ending at vertex j.

Example : For the following graph:



$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

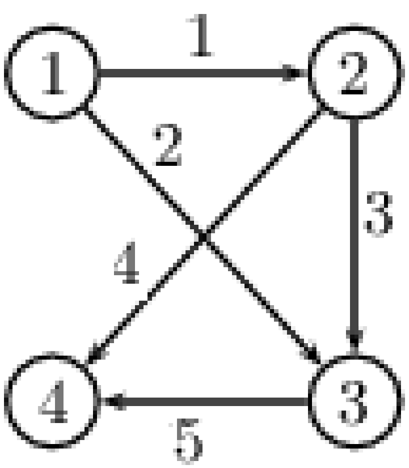
This matrix is no longer symmetric.

1.4 Representation of a graph

2- Incidence matrix: (directed graph)

Let G be a directed graph with n vertices numbered from 1 to n, and m arcs numbered from 1 to m. The **incidence matrix** of the graph is called matrix $A=(a_{i,j})$ consisting of n rows and m columns such that:

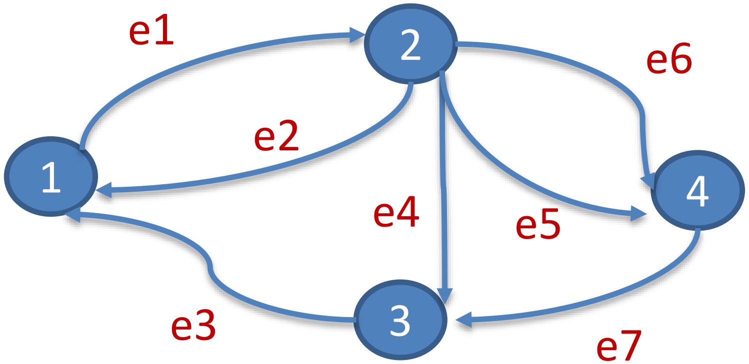
- $a_{i,j}$ equals +1 if the arc numbered j has vertex i as its origin;
- $a_{i,j}$ equals -1 if the arc numbered j has vertex i as its destination;
- $a_{i,j}$ equals 0 in all other cases. **A(vertice, arc)**



$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 \end{pmatrix}.$$

1.4 Representation of a graph

Incidence matrix : Example: G=(V,E) of order 4 with 7 arcs.



Note: *You should start filling the matrix by beginning with the arcs.*

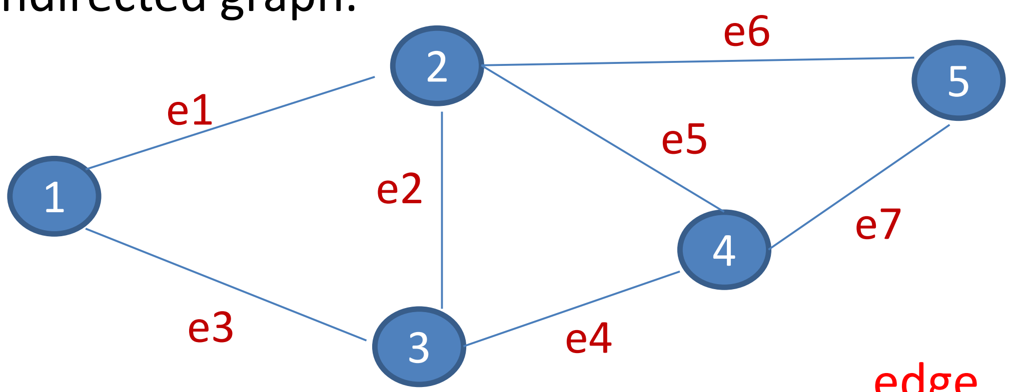
Arc

Vertice

	e1	e2	e3	e4	e5	e6	e7
1	1	-1	-1	0	0	0	0
2	-1	1	0	1	1	1	0
3	0	0	1	-1	0	0	-1
4	0	0	0	0	-1	-1	1

1.4 Representation of a graph

Incidence matrix : One can also define the incidence matrix (vertex/edge) for an undirected graph.



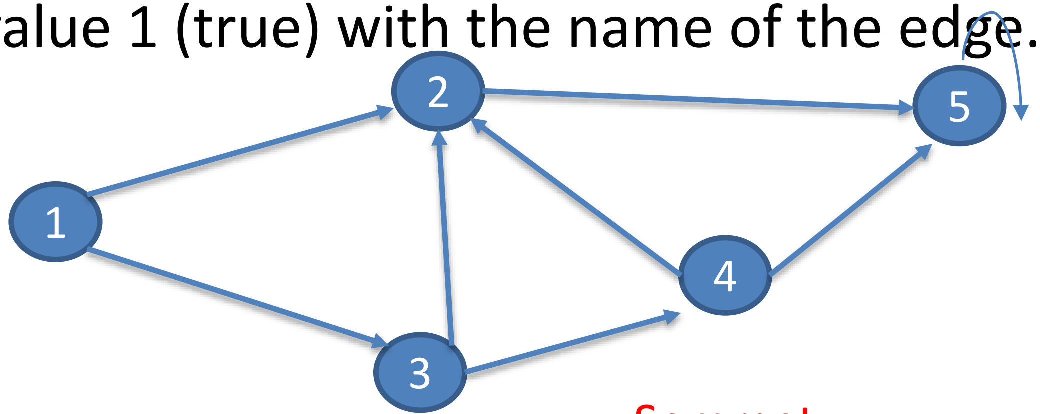
edge

Vertice

	e1	e2	e3	e4	e5	e6	e7
1	1	0	1	0	0	0	0
2	1	1	0	0	1	1	0
3	0	1	1	1	0	0	0
4	0	0	0	1	1	0	1
5	0	0	0	0	0	1	1

1.4 Representation of a graph

3- Edge Matrix: It is defined from the adjacency matrix by replacing the value 1 (true) with the name of the edge.



Sommet

Sommet

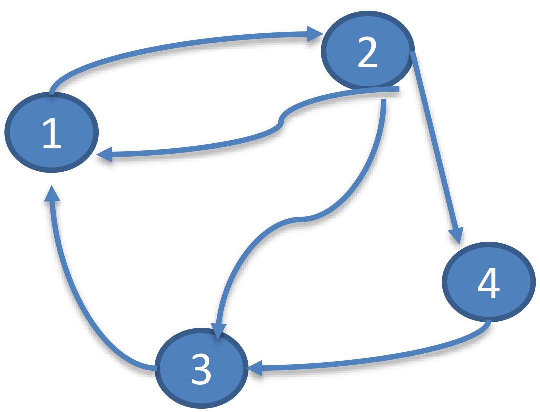
	1	2	3	4	5
1	0	12	13	0	0
2	0	0	0	0	25
3	0	32	0	34	0
4	0	42	0	0	45
5	0	0	0	0	55

1.4 Representation of a graph

4- Associated Matrix: We replace the name with the number of edges or arcs.

C) Representation by dictionaries:

The graph is represented by a table or dictionary of vertices. Each vertex has a list of **successor** (**predecessor**) vertices. We use a function Γ such that $\Gamma^{+}(x)$ is the list of successors and $\Gamma^{-}(x)$ is the list of predecessors.



Vertice	Successor $\Gamma^{+}(x)$	Predecessor $\Gamma^{-}(x)$
1	2	2,3
2	1,3,4	1
3	1	2,4
4	3	2