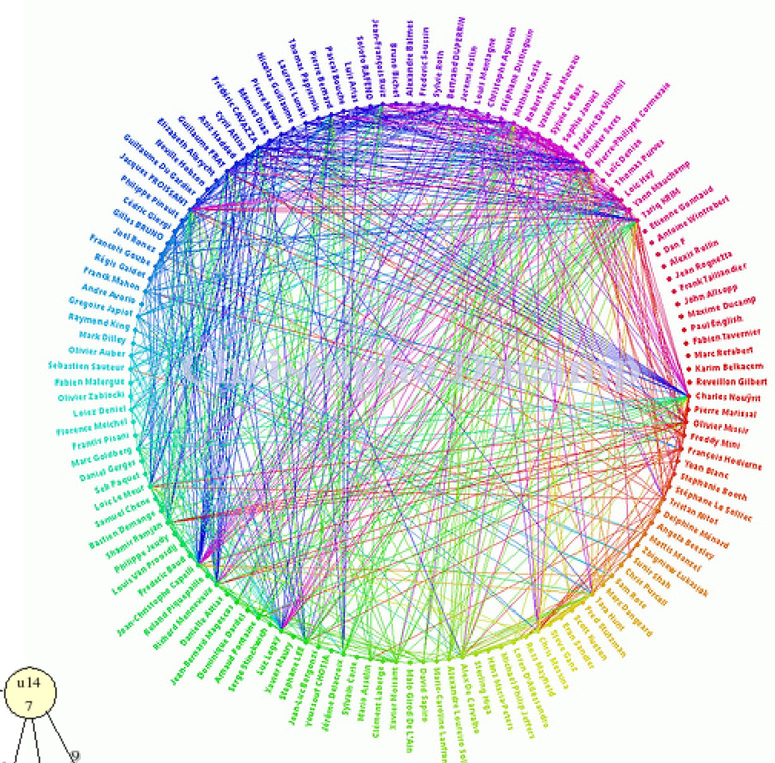
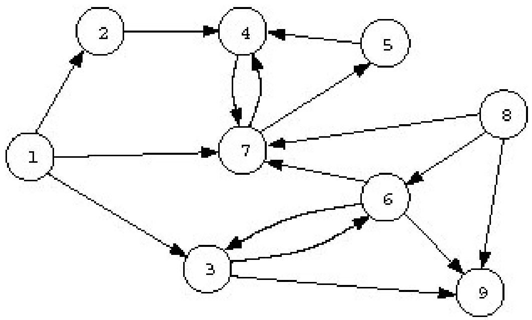
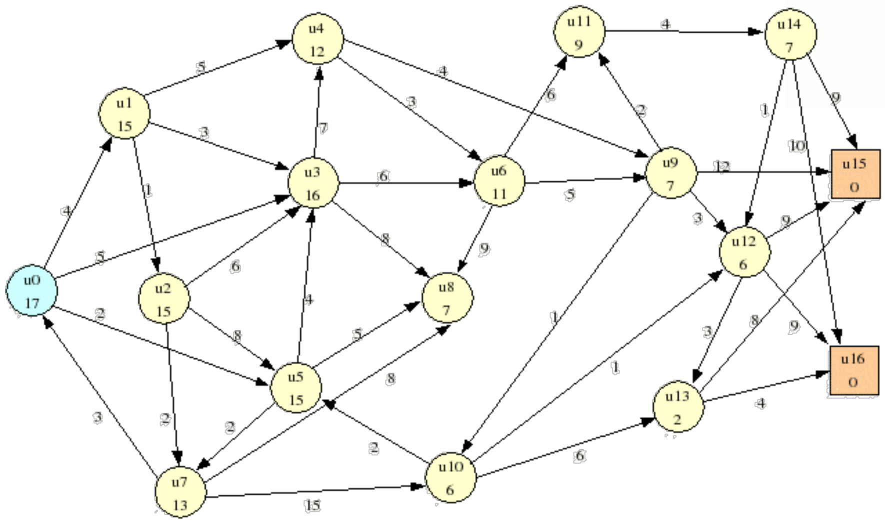
Les Graphes







Heike Ripphausen-Lipa, Beuth Hochschule für Technik Berlin
Jean-Michel Adam, Université Grenoble Alpes

Objectif pédagogique

- Comprendre l’intérêt de la modélisation
- Utilité des graphes pour modéliser le monde réel
- Connaître différentes structures de données pour représenter les graphes

Modèle

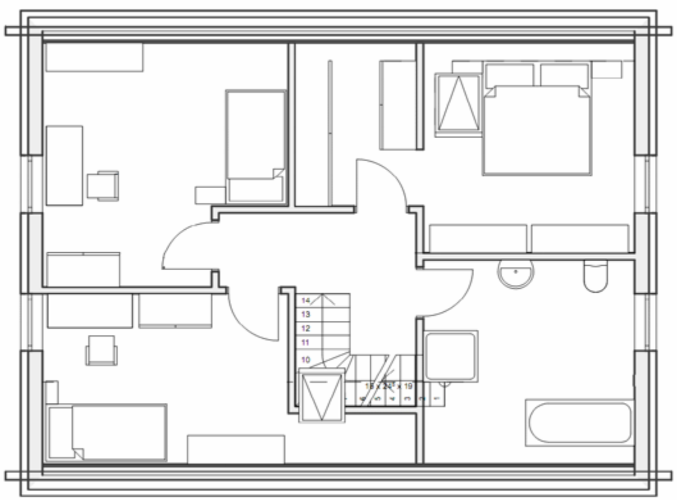
Pourquoi utiliser des modèles ?

Qu‘est-ce qui caractérise un modèle ?

- Un modèle est une simplification de la réalité
- La simplification se concentre sur l'essentiel selon un certain but
- Nous utilisons des modèles pour trouver des solutions simples à des problèmes complexes

Exemples de modèles

Maison: modèle 2D

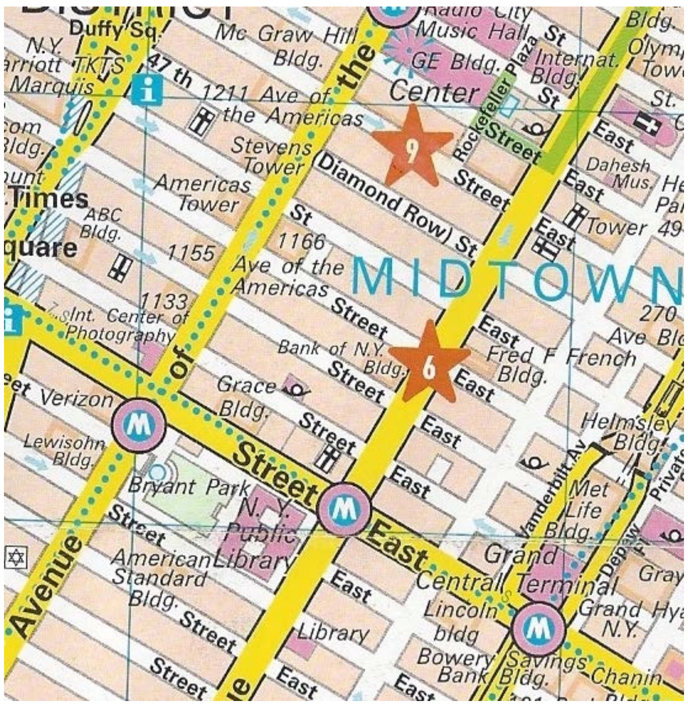


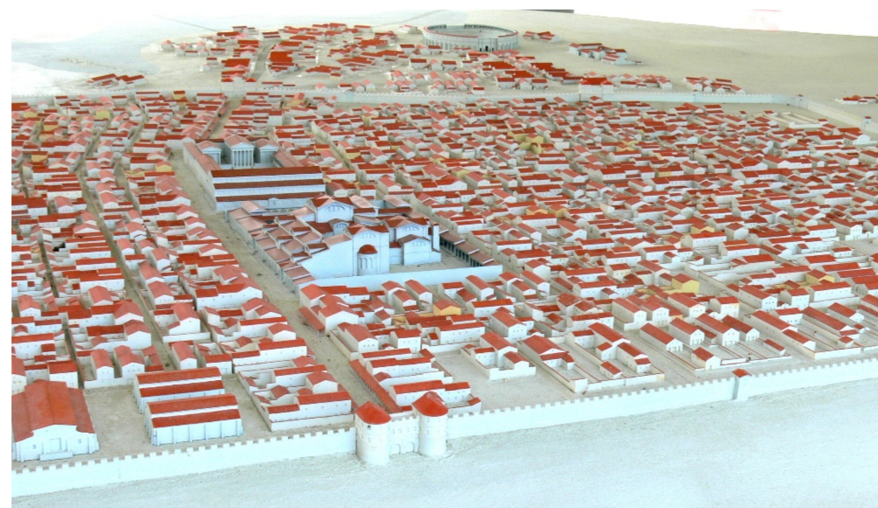
Maison: modèle 3D

Example models

Ville: plan

Ville: Modèle 3D





Modèle mathématique: équations

Equations pour modéliser certaines contraintes ou situations

Problème du monde réel :

Je dois louer une voiture pour une journée

Offre 1: location: 35 €; coût au km: 45 cts

Offre 2: location: 45 €; coût au km: 35 cts

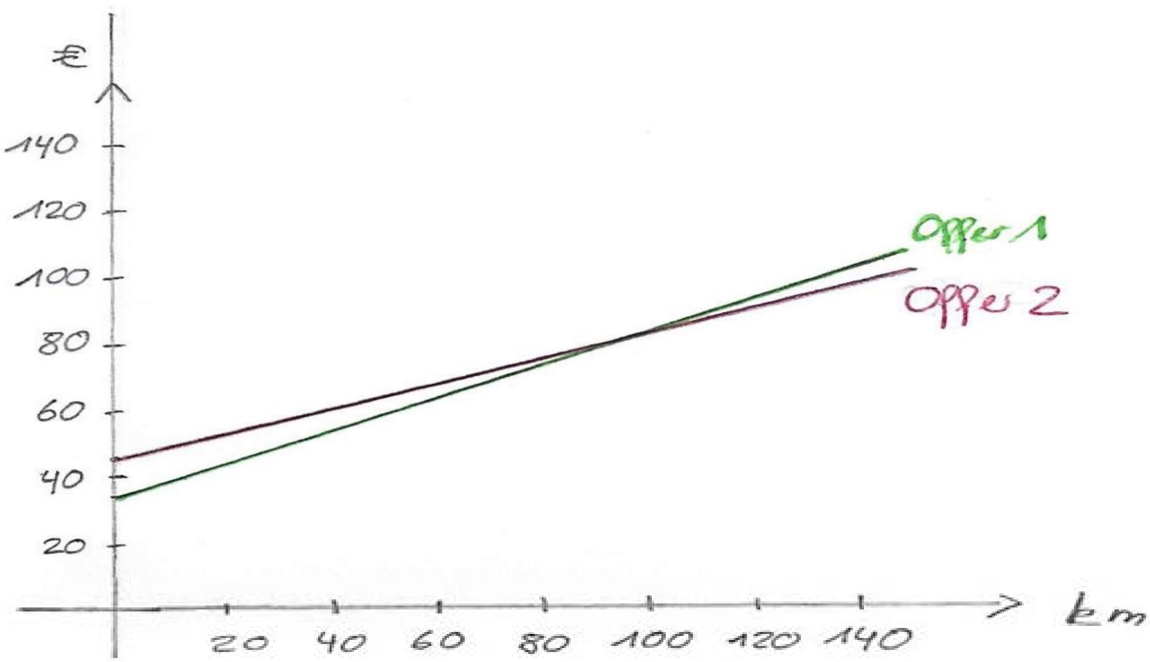
Modèle mathématique: équations

Transformation du monde réel en équations:

Coût Offre1: 35€ + 0,45€ * **n**

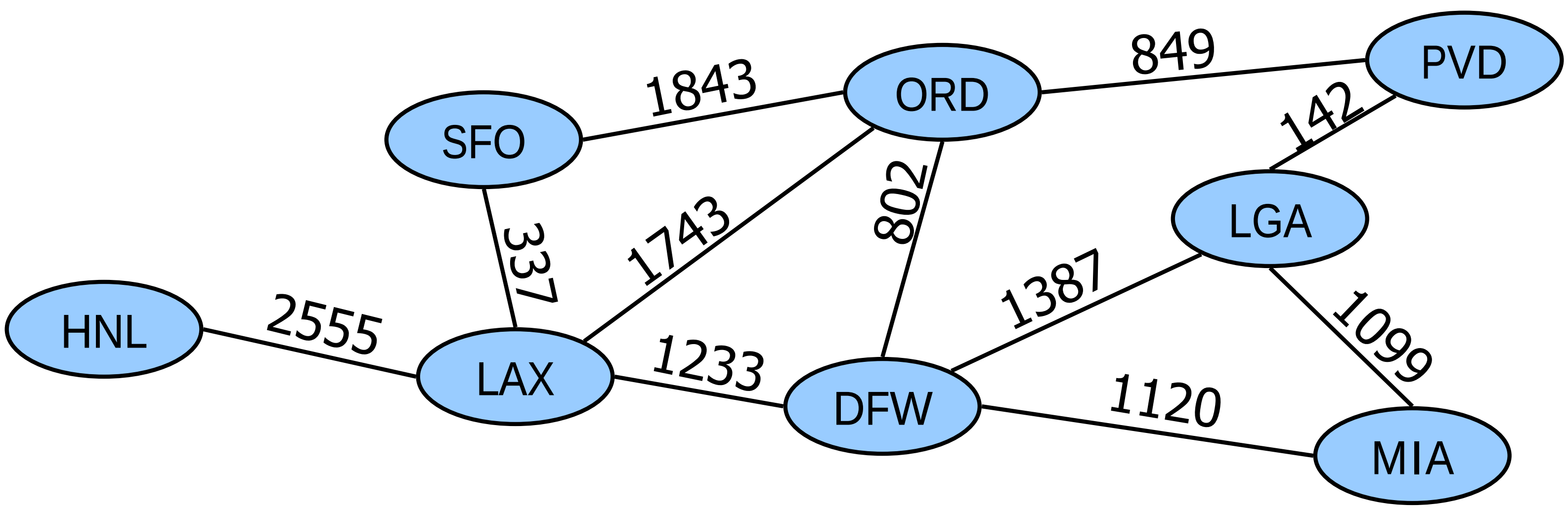
Coût Offre2: 45€ + 0,35€ * **n**

où n est le nombre de km à parcourir



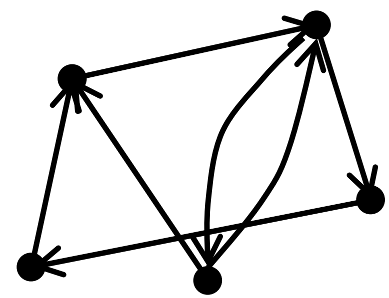
Définition

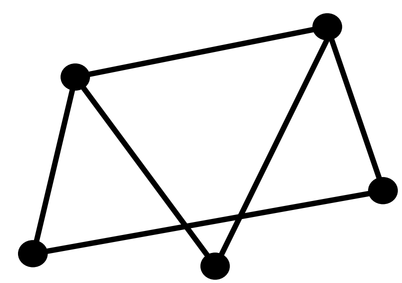
- Un graphe G = (***V, E***) est composé de :
  - ***V*** : ensemble de sommets
  - ***E*** : ensemble d'arêtes reliant les sommets de V
  - sommets et arêtes contiennent de l'information
- Exemple (réseaux aériens):
  - Chaque sommet représente un aéroport et contient le code de 3 lettres représentant cet aéroport
  - Chaque arête représente une route aérienne entre deux aéroports et contient la longueur de cette route



Graphes - Types d’arêtes

- Arête orientée (Arc) :
  - Une paire ordonnée de sommets (u,v)
  - Le premier sommet u est appelé l’origine
  - Le deuxième sommet v est appelé la destination
  - Par ex, un vol d’avion
- Arête non-orientée :
  - Une paire non-ordonnée de sommets (u,v)
  - Par ex, une route piétonne
- Graphe orienté :
  - Toutes les arêtes sont orientées
  - Par ex. réseau de vols
- Graphe non-orienté :
  - Toutes les arêtes sont non-orientées
  - Par ex. la carte routière





Graphes - Applications

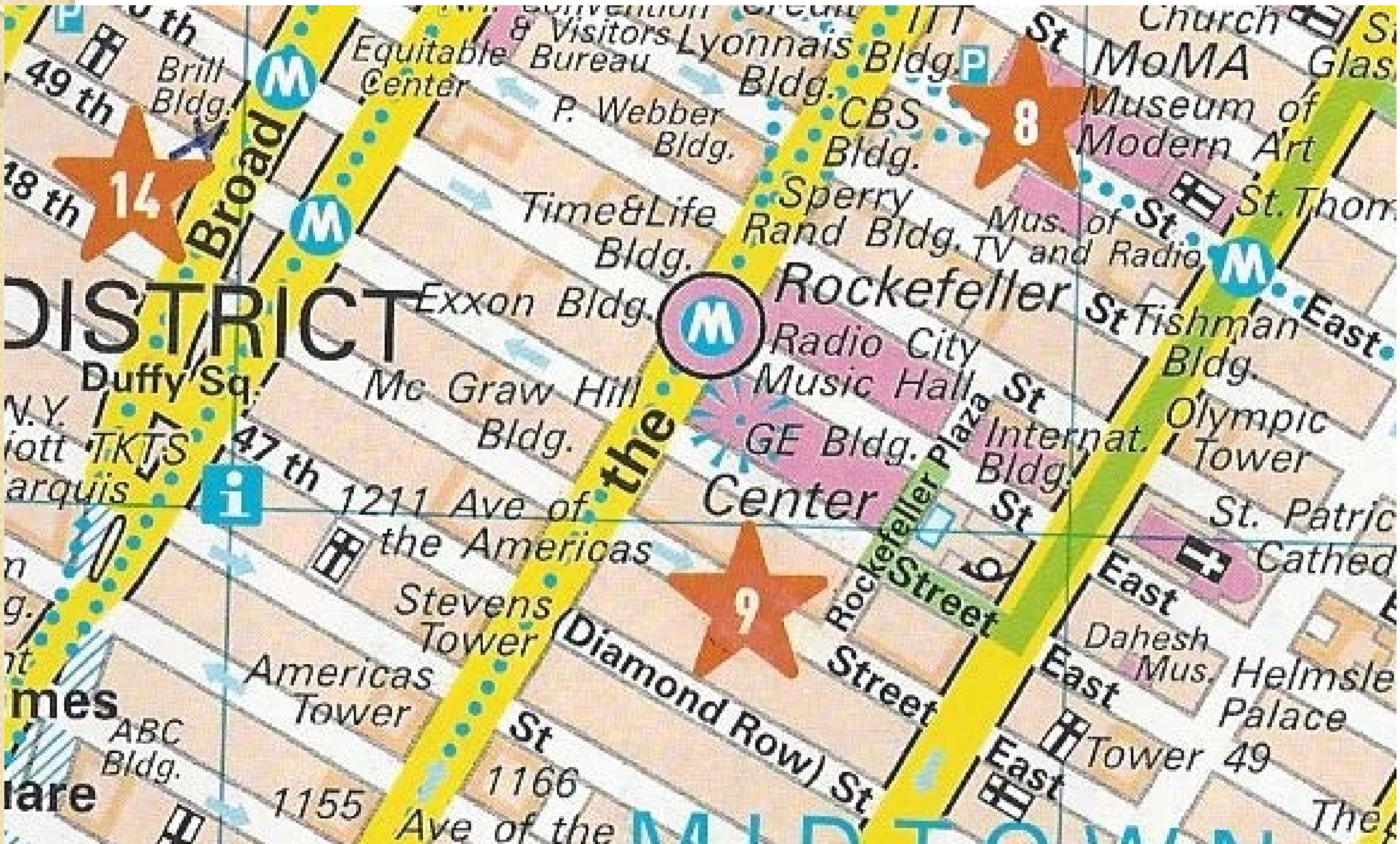
- Circuits électroniques
  - Circuit imprimé
  - Circuit intégré
- Réseaux routiers
  - Réseau routier urbain
  - Réseau de vol aérien
- Réseaux informatiques
  - connexion par réseau local (LAN)
  - Internet
  - Web
- Bases de données
  - Diagramme Entité-Association

Exemple de modélisation par graphe orienté

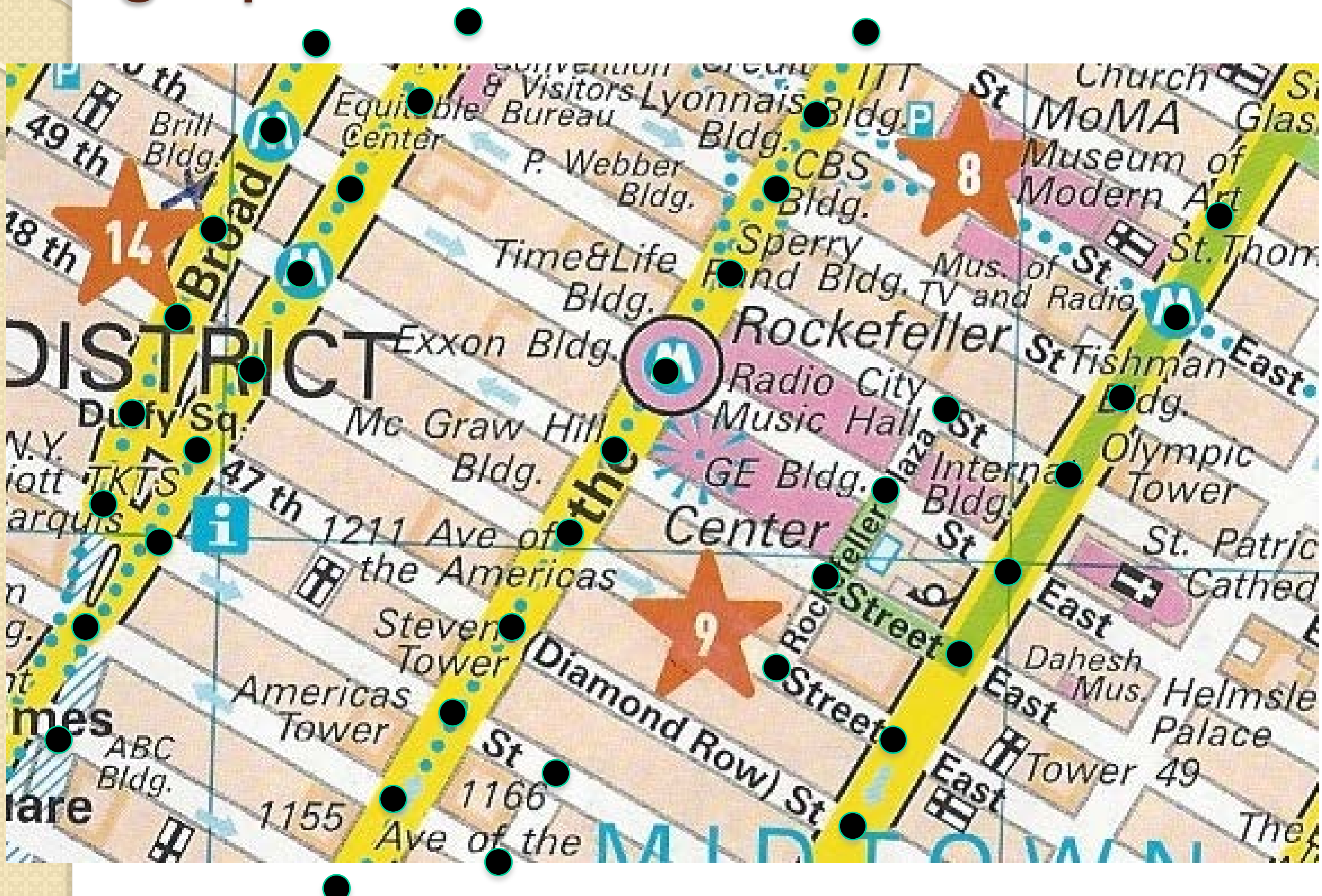
Trouver un chemin entre deux emplacements dans une ville.

- Les croisements et embranchements sont représentés par des noeuds
- Les rues doivent être modélisées par des arcs, puisque certaines rues sont à sens unique.

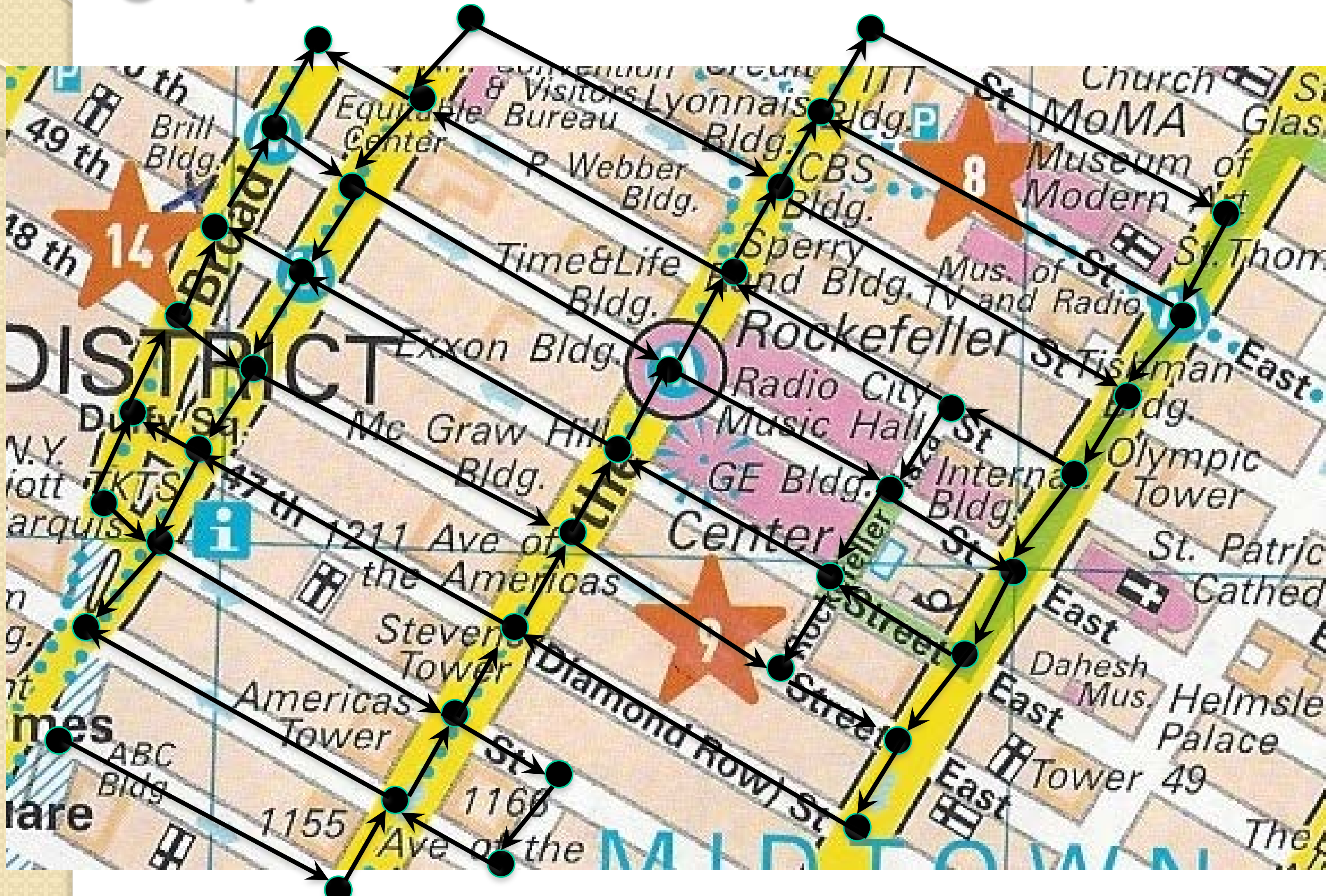
Modélisation d‘une carte par un graphe



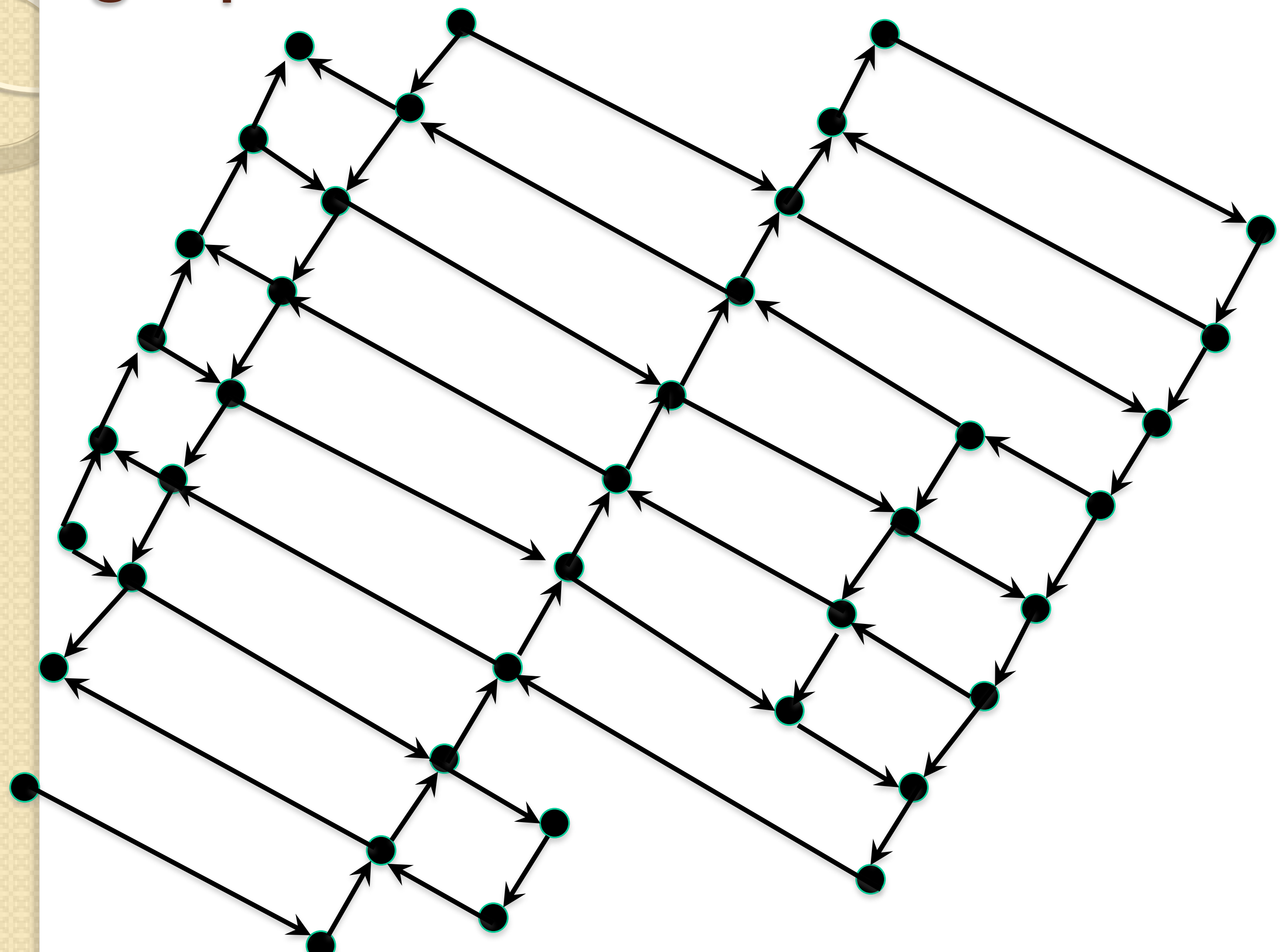
Modélisation d‘une carte par un graphe



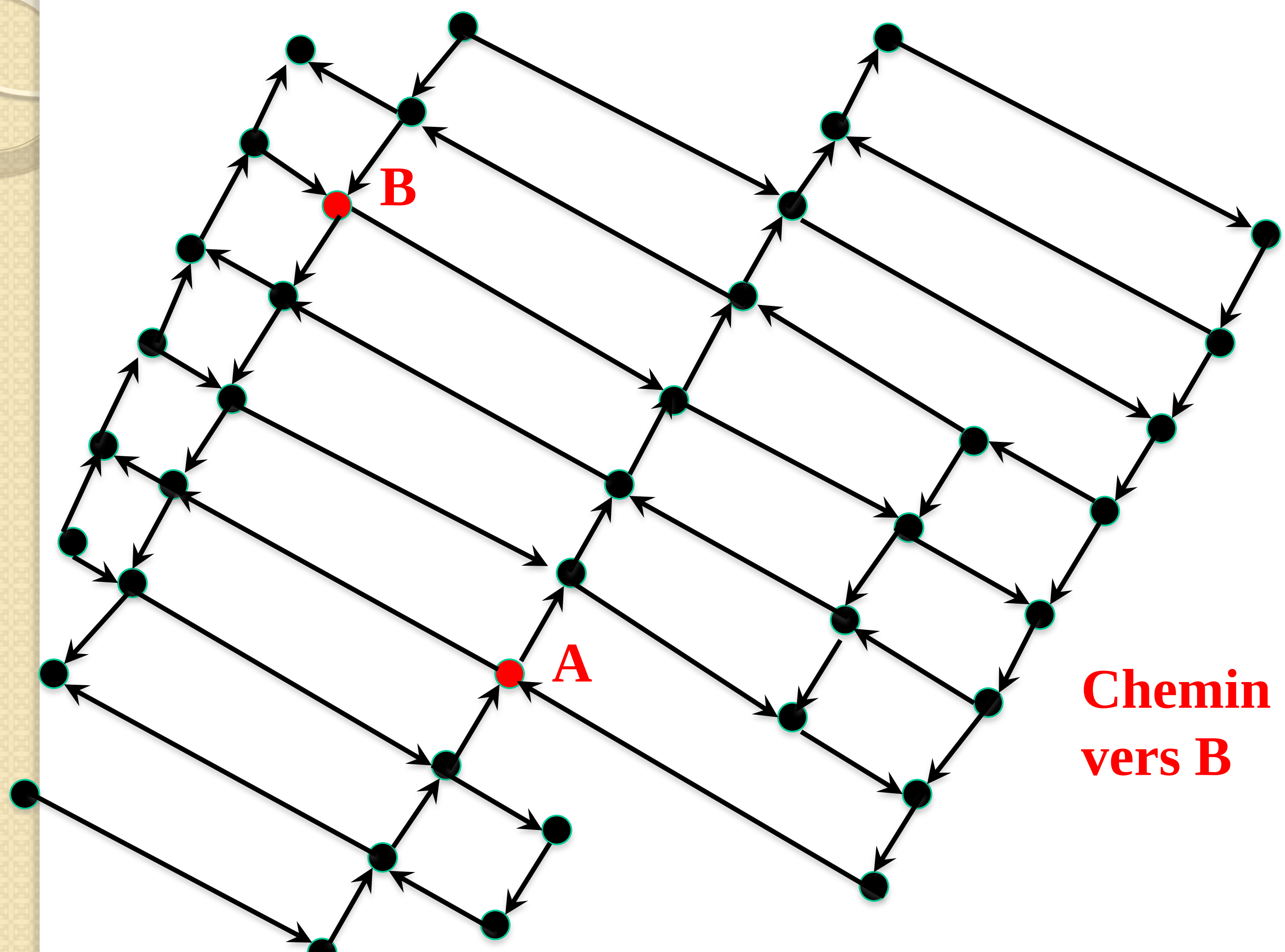
Modélisation d‘une carte par un graphe



Modélisation d‘une carte par un graphe

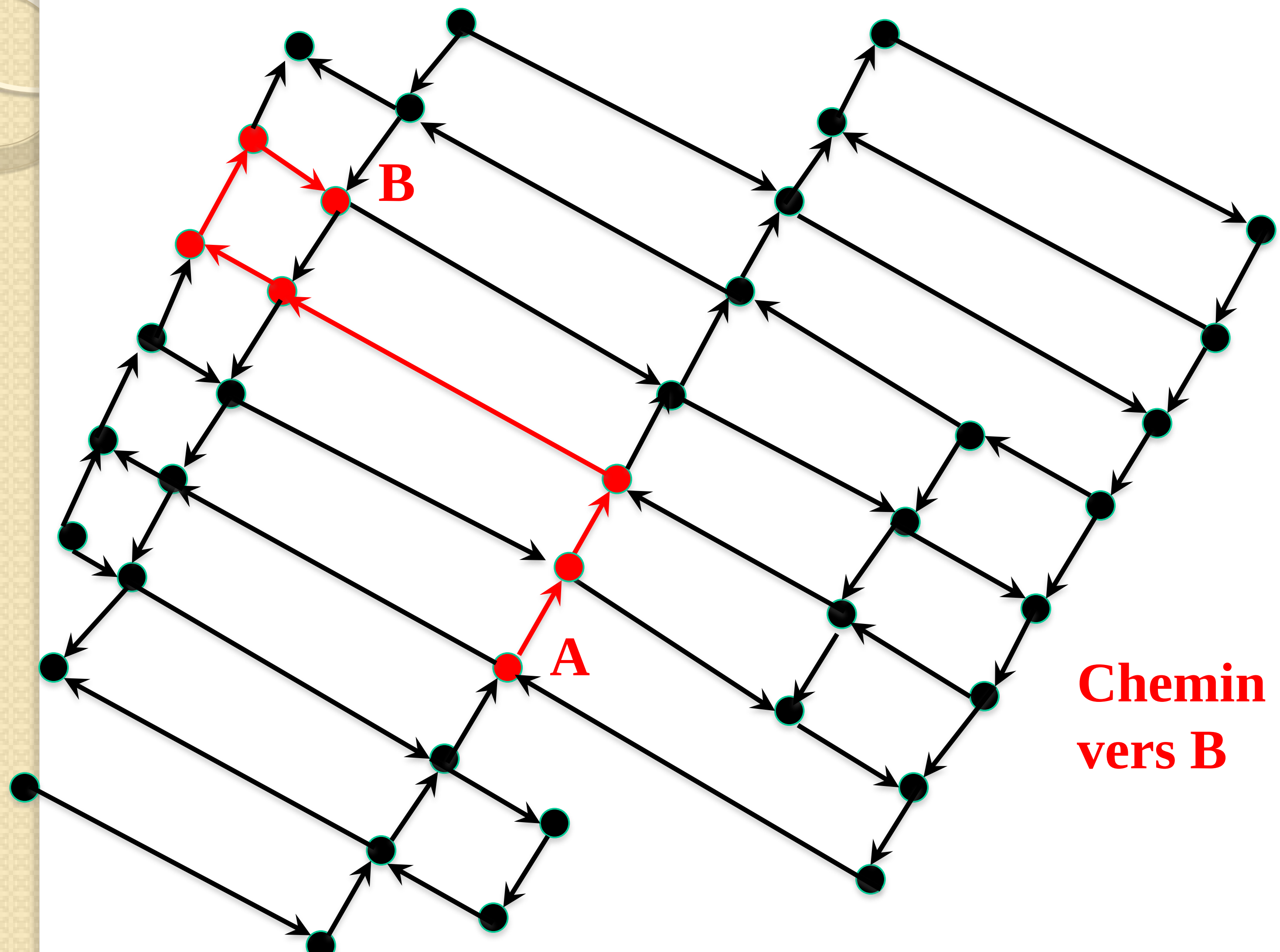


Chemin



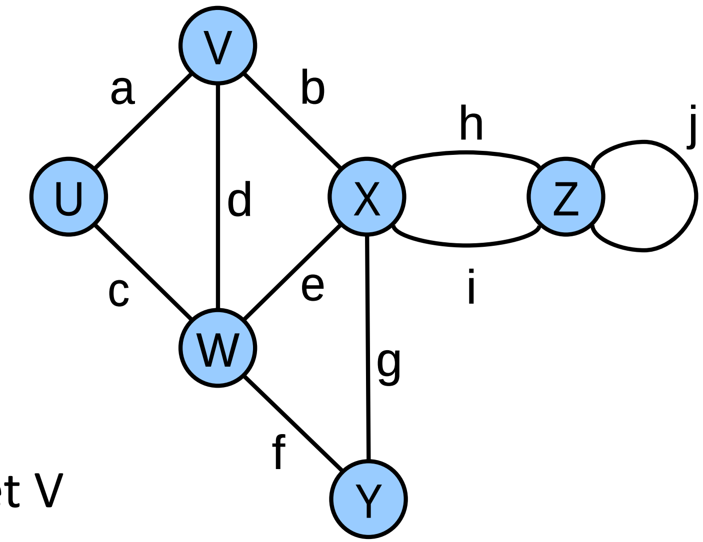
Chemin de A vers B

Chemin



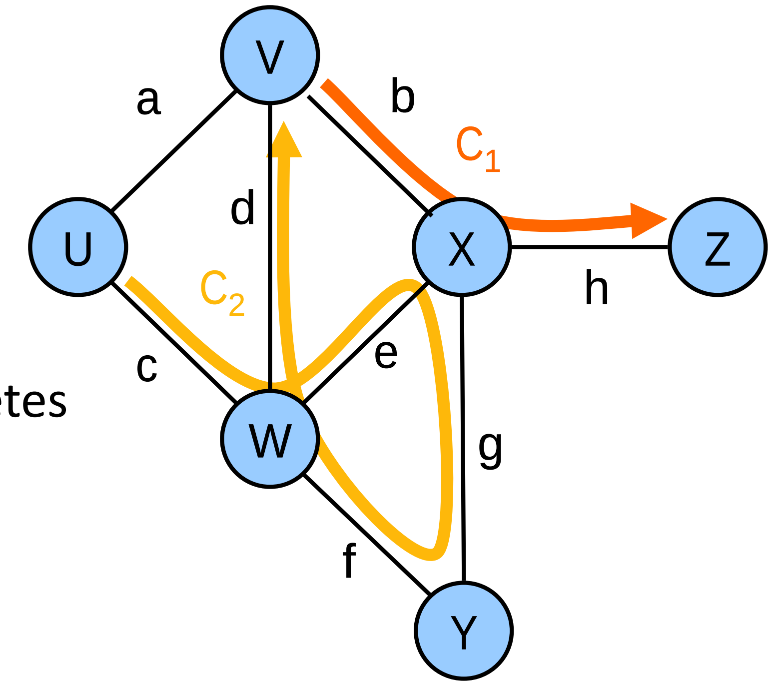
Chemin de A vers B

Graphes - Terminologie



- Extrémités d'une arête :
  - U et V sont les extrémités de a
- Arêtes incidentes à un sommet :
  - a, d, et b sont incidentes au sommet V
- Sommets adjacents :
  - Deux sommets sont adjacents ou voisins s'ils sont reliés par une arête. Ex: U et V sont adjacents
- Degré d'un sommet :
  - Nb. d' arêtes ayant comme extrémité le sommet. Ex: Le degré du sommet X est 5
  - Dans un graphe orienté on distingue également le degré entrant et le degré sortant
- Arêtes multiples (parallèles) :
  - Arêtes reliant les mêmes sommets. Ex.: h et i sont multiples
- Boucle :
  - Une boucle est une arête ayant deux fois le même sommet comme extrémité. Ex: j est une boucle

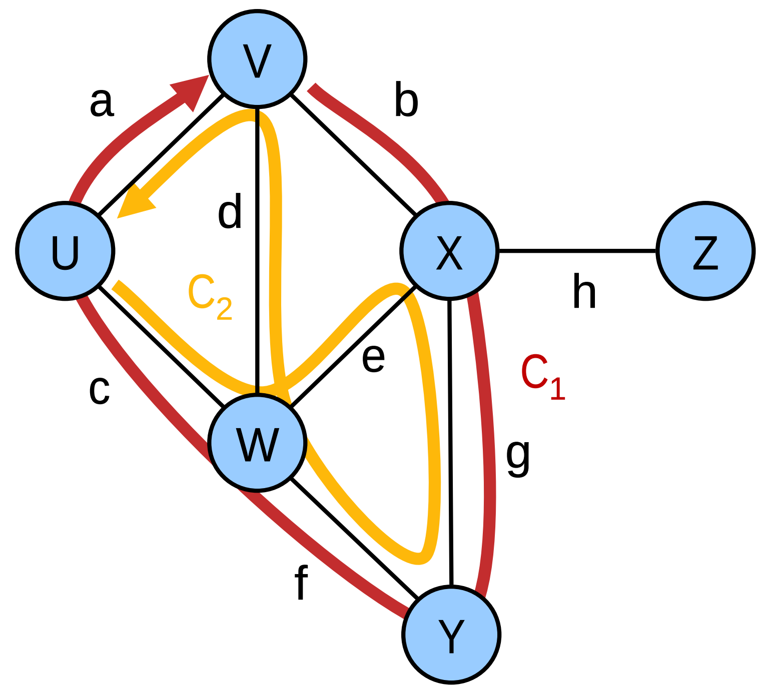
Graphes - Terminologie



- Chemin :
  - séquence alternant sommets et arêtes
  - commence dans un sommet
  - se termine dans un sommet
  - chaque arête se trouve entre ses extrémités
- Chemin simple :
  - chemin dans lequel tous les sommets et toutes les arêtes visités sont distincts
- Exemples :
  - C_1=(V,b,X,h,Z) est un chemin simple
  - C_2=(U,c,W,e,X,g,Y,f,W,d,V) n'est pas un chemin simple

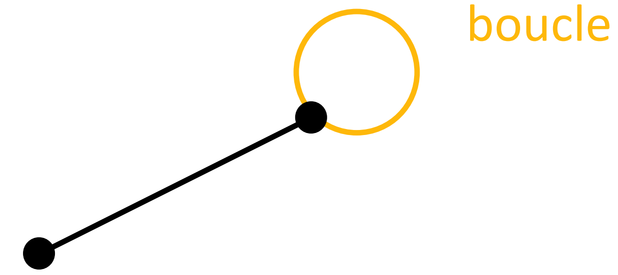
Graphes - Terminologie

- Cycle :
  - Une séquence circulaire de sommets et arêtes
  - chaque arête se trouve entre deux sommets
- Cycle simple :
  - cycle dans lequel tous les sommets et les arêtes visités sont distincts
- Exemples :
  - C_1=(V,b,X,g,Y,f,W,c,U,a,↵) est un cycle simple
  - C_2=(U,c,W,e,X,g,Y,f,W,d,V,a,↵) n'est pas un cycle simple



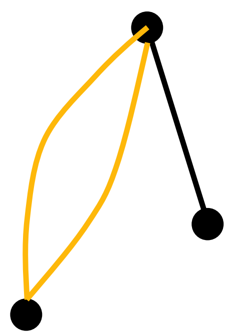
Boucle

Une arête **`e = {u, v}`** avec **`u = v`** est appelée une ***boucle***.



Arêtes parallèles

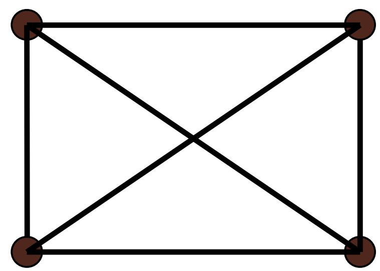
Des arêtes qui connectent la même paire de sommets sont appelées ***arêtes parallèles***



Un ***graphe simple*** est un graphe sans boucle ni arêtes parallèles

Graphe complet

Un graphe est un ***graphe complet***, s’il existe une arête entre toutes les paires de sommets

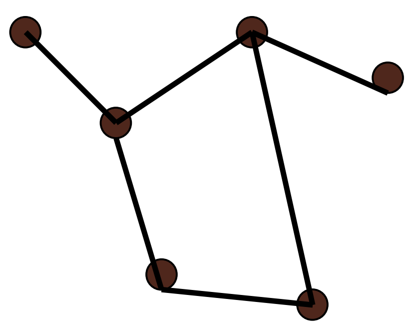


Graphe complet à 4 sommets

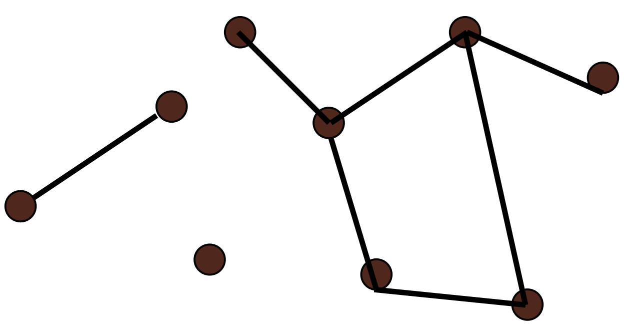
Graphe connexe

Un graphe **`G = (V, E)`** est appelé ***connexe***, si pour chaque paire **`{u, v}`** il existe au-moins un chemin entre **`u`** et **`v`**.

Graphe connexe

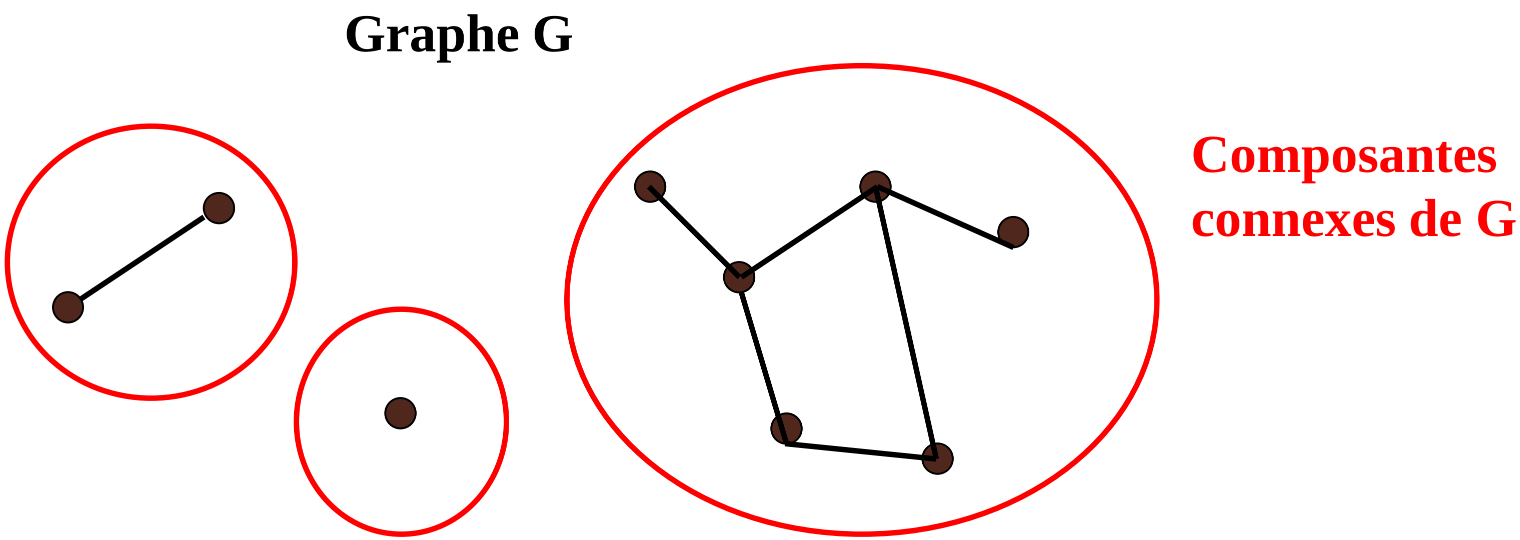


Graphe non connexe



Composantes connexes

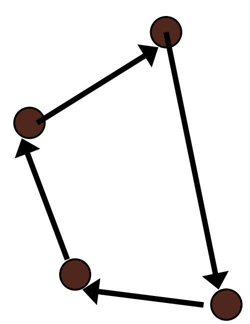
Les sous-graghes connexes de **`G = (V,E)`** sont appelés ***composantes connexes*** de G.



Graphe fortement connexe

Un graphe orienté `G = (V, E)` est dit ***fortement conexe***, si pour chaque paire `{u, v}` il y a au-moins un chemin de **`u`** vers **`v`**.

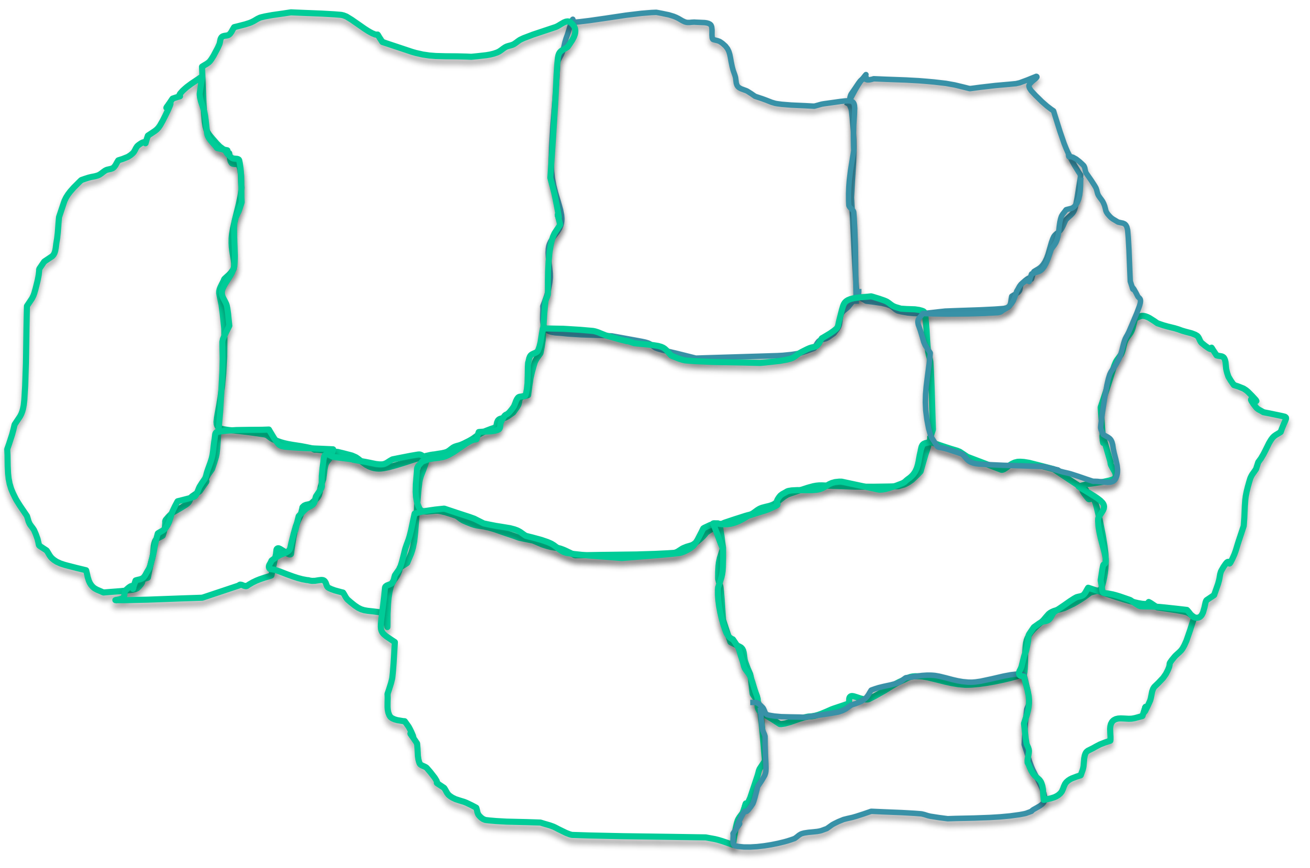
Graphe fortement conexe



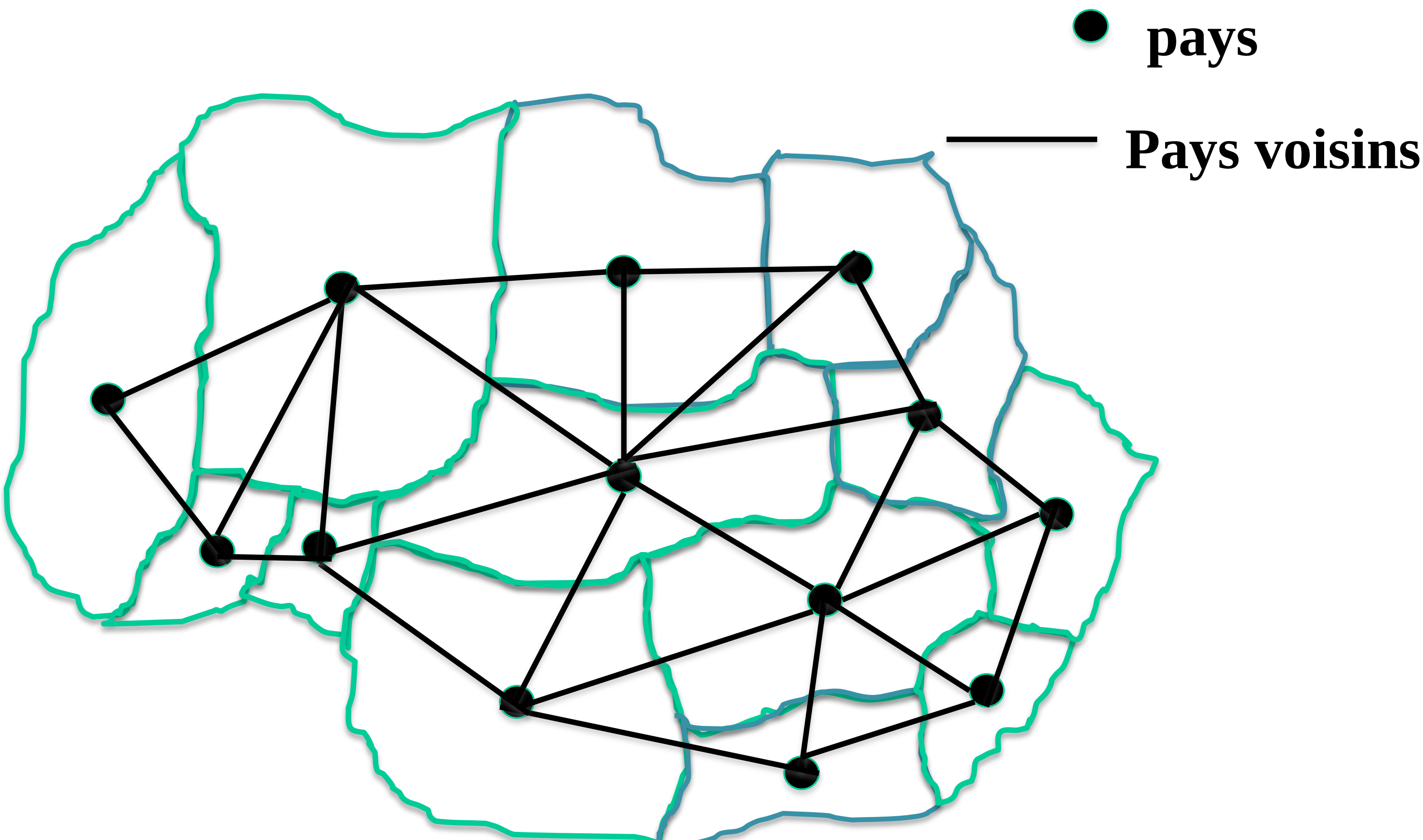
Des graphes pour modéliser le monde réel

Coloration de carte

Colorer une carte de sorte que deux pays voisins aient des couleurs différentes

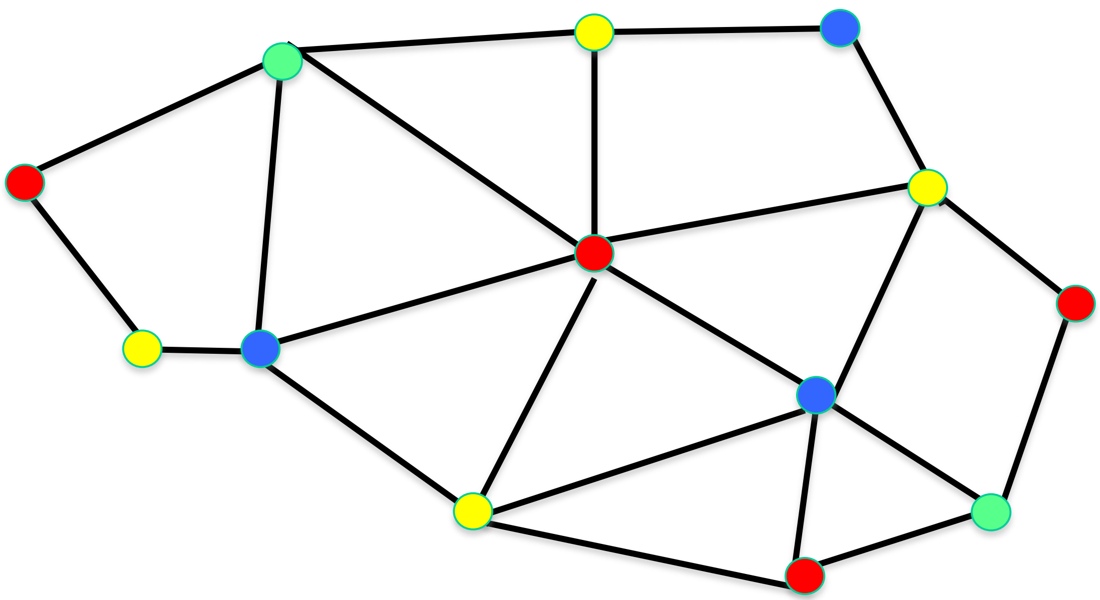


Modélisation

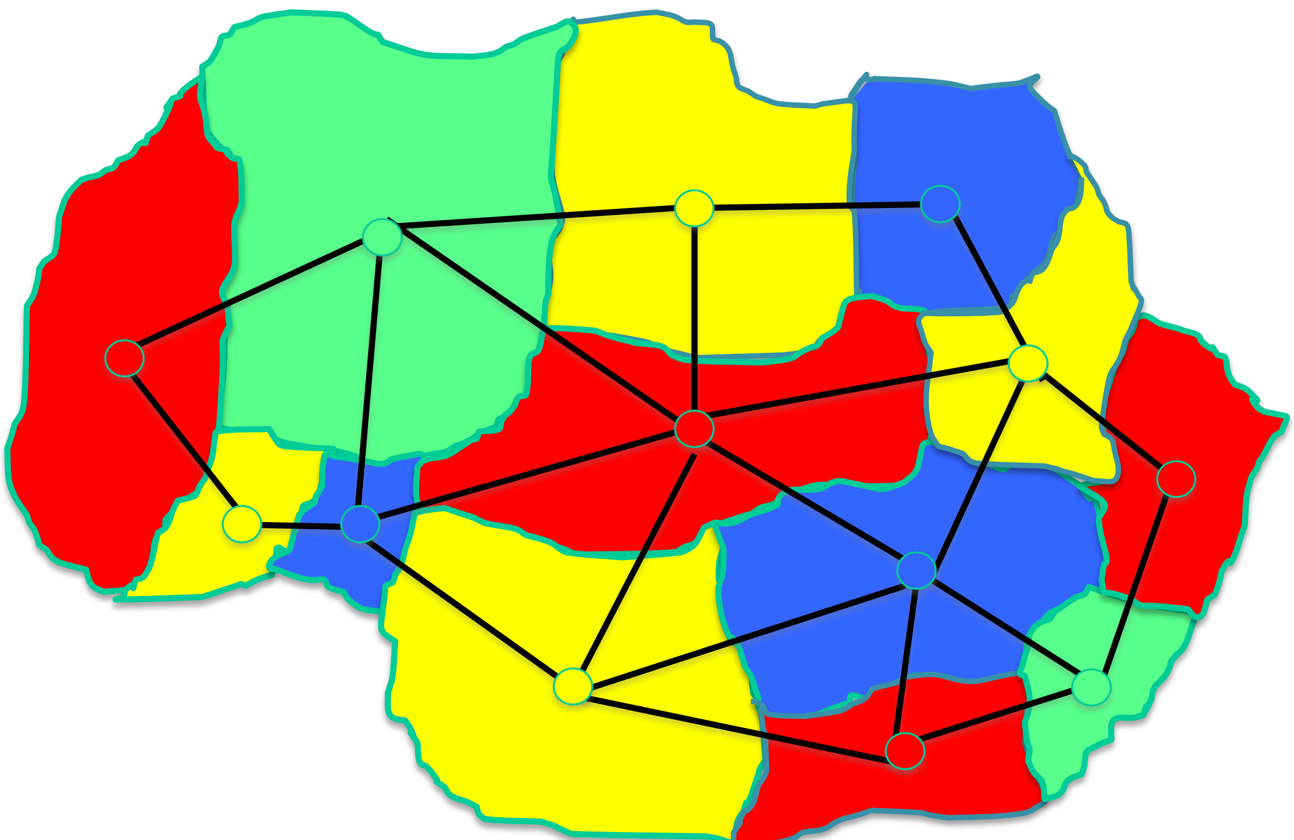


Coloration des sommets

Colorer les sommets de sorte que les noeuds voisins aient des couleurs différentes !



Coloration des pays



Problème d'appariement

Données:

- **x** travailleurs veulent effectuer un travail aujourd‘hui
- **y** travaux proposés, chacun prend la journée entière
- Chaque travailleur n‘est capable de faire que certains travaux

Problème:

Trouver une affectation de travaux aux travailleurs, de sorte qu‘un nombre maximum de travaux soient faits.

Décrire le problème comme un problème de théorie des graphes !

Problème d'appariement

Graphe G

- Les sommets sont les travailleurs et les travaux
- Une arête relie un travailleur à un travail si le travailleur est capable de faire le travail.

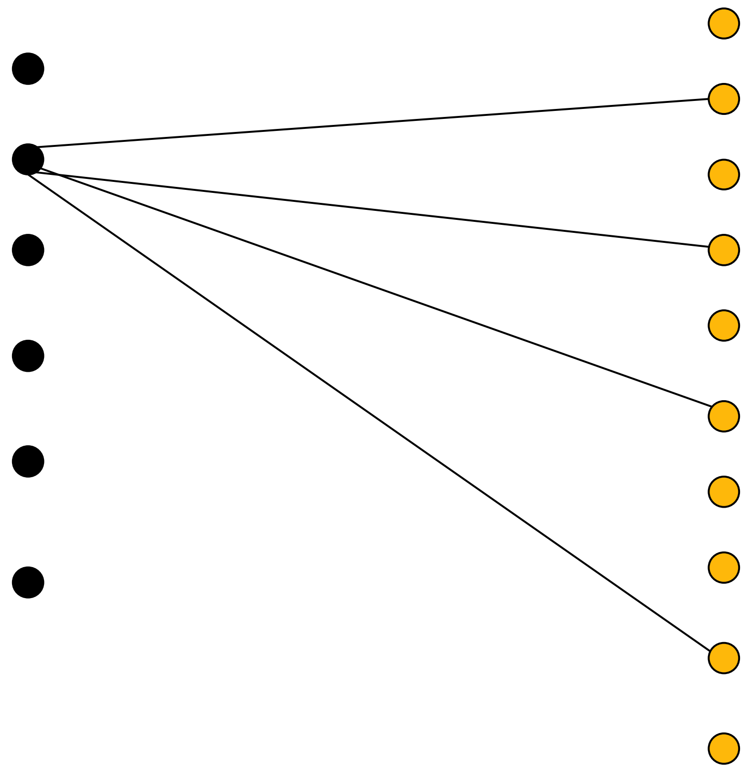
Problème

Trouver un sous-ensemble maximum d‘arêtes sans avoir deux arêtes qui aient un sommet commun.

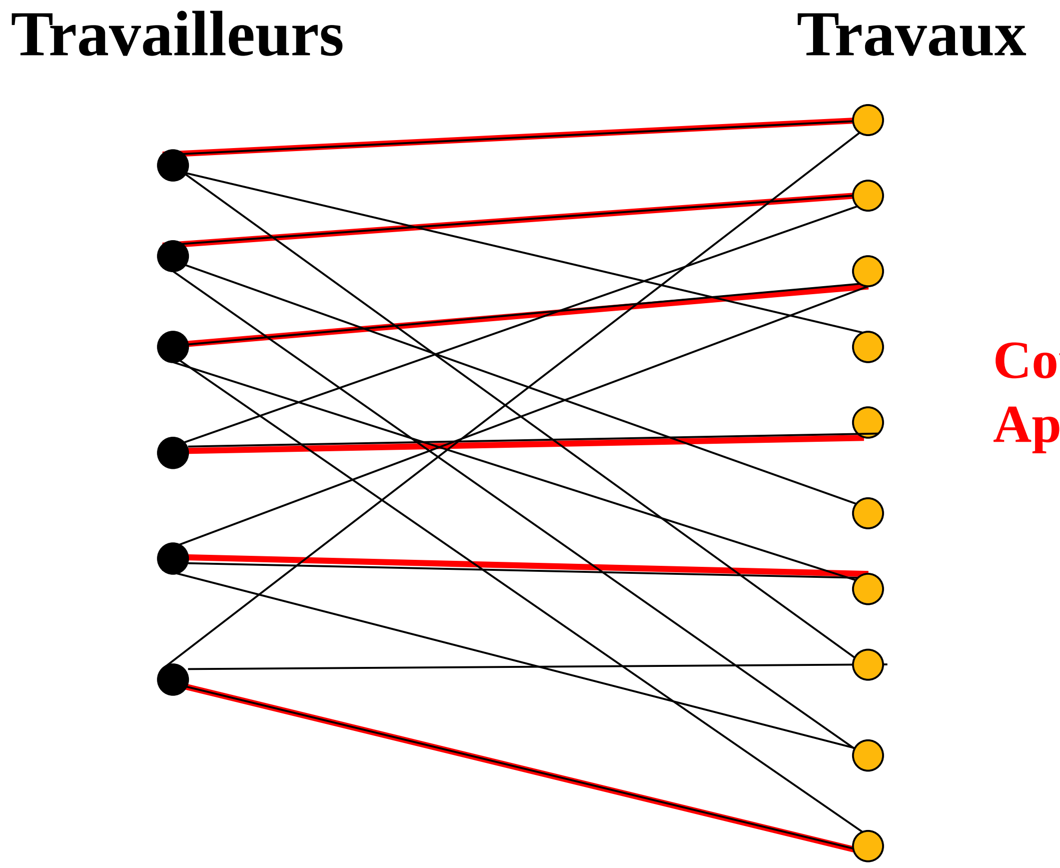
Problème d'appariement

Travailleurs **Travaux**

Les arêtes relient le travailleur aux travaux qu‘il peut faire



Problème d'appariement



Couplage maximal / Appariement maximal

Couplage maximal

Le problème décrit précédemment est connu sous le nom de ***problème de couplage maximal*** :

Pour un graphe **`G = (V,E)`**

Un ***couplage*** **`M`** est un sous-ensemble de **`E`**, tel que pour chaque sommet **`v`** au plus une arête de **`M`** est incidente à **`v`**

Un ***couplage maximal*** est un couplage avec un nombre maximum d‘arêtes

Couplage biparti maximal

Le problème décrit précédemment est un cas particulier du problème de couplage maximal d'un graphe biparti.

Un graph est biparti, si les sommets **V** peuvent être partitionnés en deux sous-ensembles **V_1** et **V_2**, de sorte que chaque arête relie un sommet de **V_1** à un sommet de **V_2**

Dand le problème d'appariement **V_1** correspond aux travailleurs et **V_2** aux travaux.

La tournée des villes

Un touriste veut faire une tournée à travers **n** villes d'Allemagne (Berlin, Cologne, Munich, ...)

Données

La distance entre les villes

Problème

Trouver la tournée la plus courte à travers toutes les villes

Décrire le problème comme un problème de théorie des graphes !

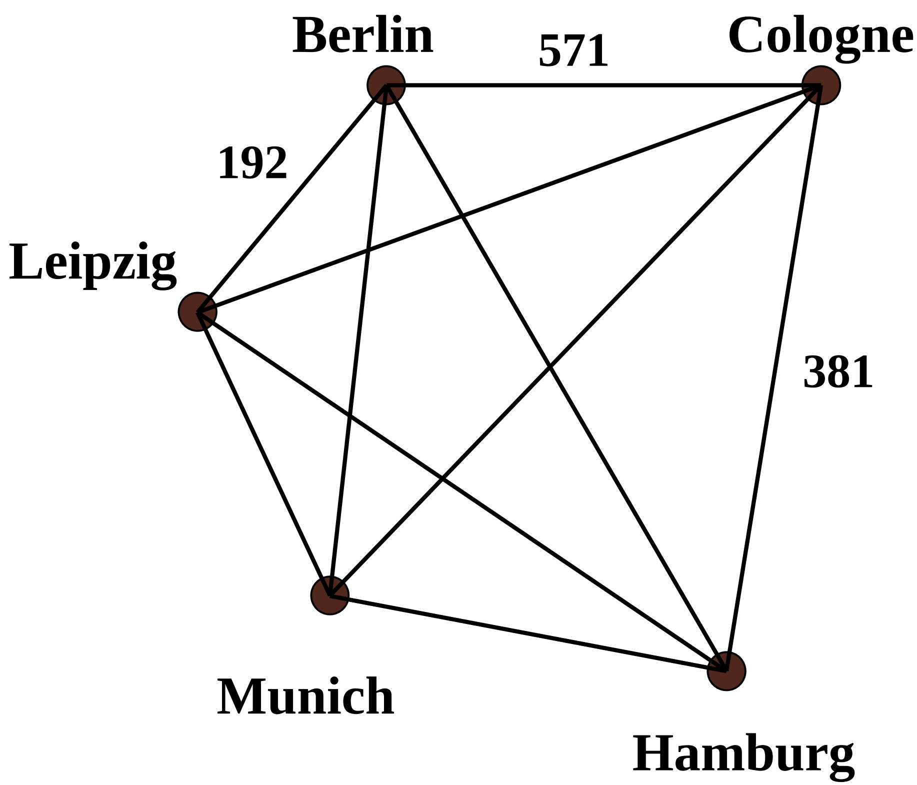
La tournée des villes

Graph G=(V,E) avec

- Les sommets sont les villes
- Des arêtes relient les villes (il y a une arête entre chaque paire de villes)
- Chaque arête porte un coût : la distance entre les deux villes reliées.

Trouver une cycle passant par toutes les sommets avec coût minimum

La tournée des villes



Problème du voyageur de commerce

Données : **n** sommets (villes), **m** arêtes (distances) entre **tous** les couples de sommets.

Problème : trouver le parcours circulaire (cycle) qui passe par tous les sommets (villes)

Arbre recouvrant minimal

(Minimum Spanning Tree)

Données:

- Sommets (ordinateurs),
- Arêtes (connexions entre les ordinateurs)
- Poids des connexions (coût des connexions).

Problème:

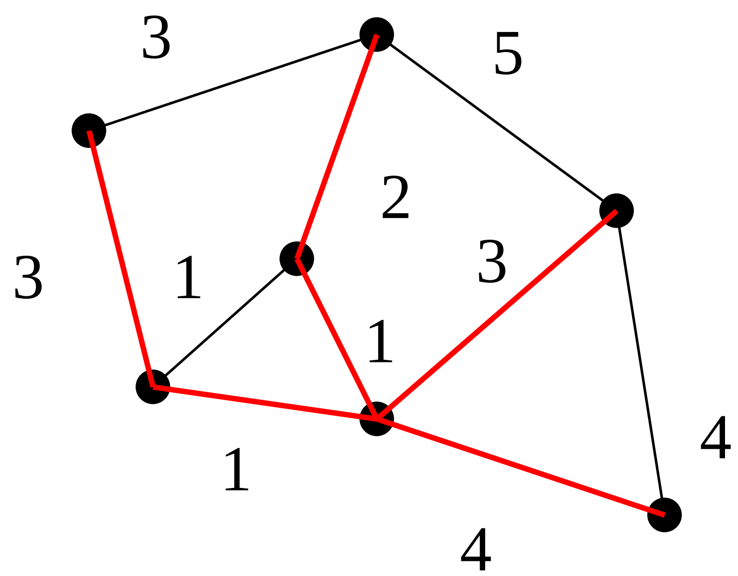
Trouver l'arbre recouvrant minimal : sous-ensemble d'arêtes tel que:

- un chemin entre chaque sommet existe

- la somme des poids des arêtes sélectionnées est minimale

Arbre recouvrant minimal

Trouver l'arbre recouvrant minimal



Arêtes de l'arbre recouvrant minimal

Représentation des graphes

Structures de données pour représenter les graphes

Il y a différentes possibilités pour représenter les graphes

- Matrice d’adjacence
- Liste d’adjacence

La représentation à utiliser dépend des algorithmes et du type de graphe manipulé

Nous n’allons considérer que des graphes simples ; les concepts décrits peuvent facilement être appliqués aux autres graphes

Matrice d'adjacence

Une matrice d'adjacence **A** d'un graphe **G =(V, E)** avec **n** noeuds est une matrice **n x n** dont les éléments comportent les valeurs suivantes :

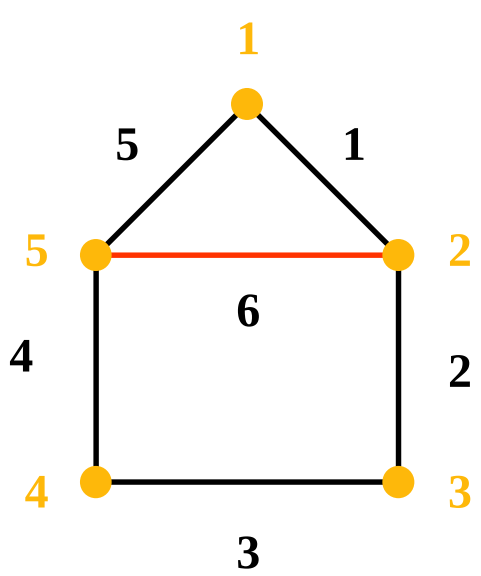
$$A_{ij} = \begin{cases} \mathbf{1}, \text{ le noeud } \mathbf{i} \text{ est voisin du noeud } \mathbf{j}, \text{ c.-à-d. une arête } \mathbf{\{i,j\}} \in \mathbf{E} \text{ ;} \\ \mathbf{0}, \text{ sinon} \end{cases}$$

Remarques:

- Dans un graphe orienté c'est l'arc **(i,j)** qui doit être un élément de **E**
- Dans un graphe non orienté la matrice d'adjacence est symétrique

Matrice d’adjacence

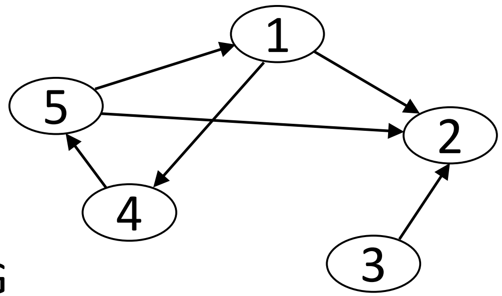
Exemple



Matrice d’adjacence

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	0	1
3	0	1	0	1	0
4	0	0	1	0	1
5	1	1	0	1	0

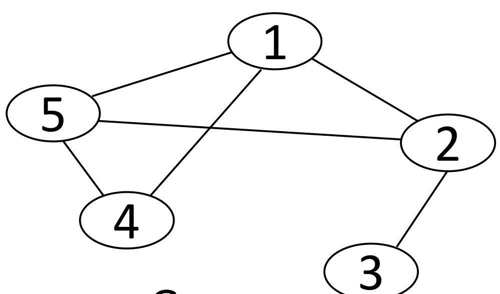
Matrice d'adjacence - Exemples



G

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 \\
3 & 0 & 1 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 1 \\
5 & 1 & 1 & 0 & 0 & 0 \\
\hline
\end{array}$$

Si G est non-orienté



G

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & 1 & 0 & 1 & 1 \\
2 & 1 & 0 & 1 & 0 & 1 \\
3 & 0 & 1 & 0 & 0 & 0 \\
4 & 1 & 0 & 0 & 0 & 1 \\
5 & 1 & 1 & 0 & 1 & 0 \\
\hline
\end{array}$$

matrice symétrique

Matrice d'adjacence (observation)

Espace occupé: $\boxed{n \times n}$

On gaspille beaucoup d'espace si la matrice est éparse

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

Liste d’adjacence

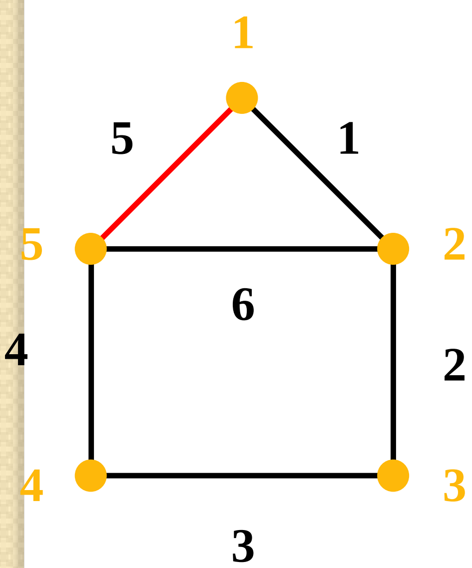
A chaque sommet on associe la liste des sommets reliés par des arêtes/arcs; on y intègre l’information portée par les arêtes.

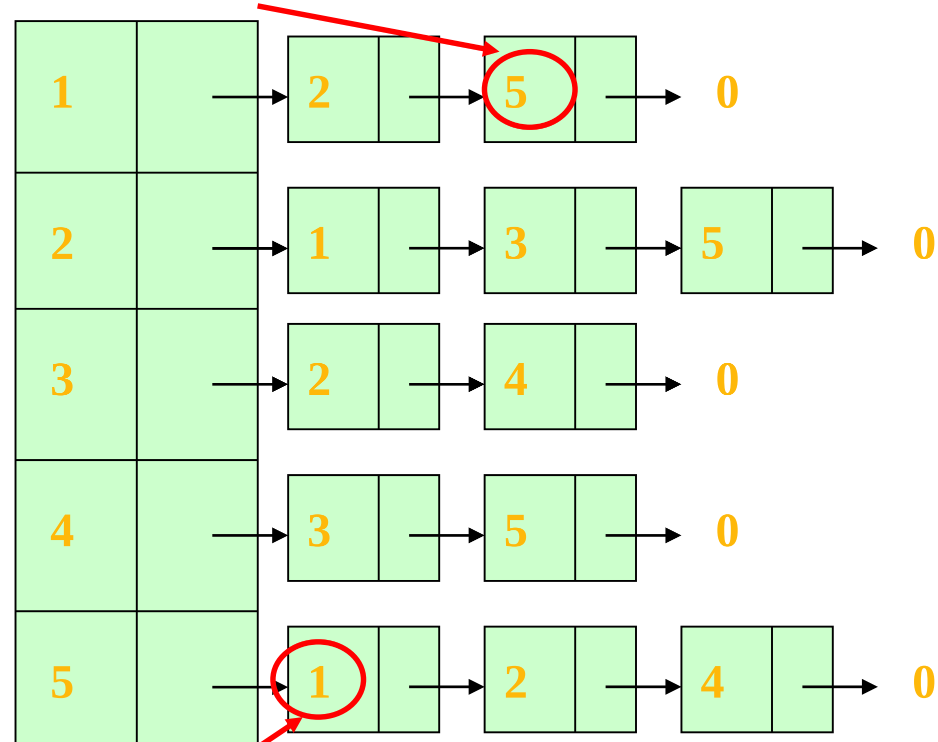
Ce peut être une liste chainée simple ou doublement chaînée

Liste d'adjacence

Représente l'arête entre le sommet 1 et le sommet 5

Exemple





Représente l'arête entre le sommet 5 et le sommet 1

Comparaison entre Matrice d‘adjacence et liste d‘adjacence

Avantages de la liste d’adjacence :

- Dans le cas des graphes avec un petit nombre d'arêtes, les listes d'adjacence sont plus efficaces en occupation mémoire
- Les listes d'adjacence permettent un parcours efficace de tous les sommets adjacents à un sommet donné

Avantages de la matrice d’adjacence :

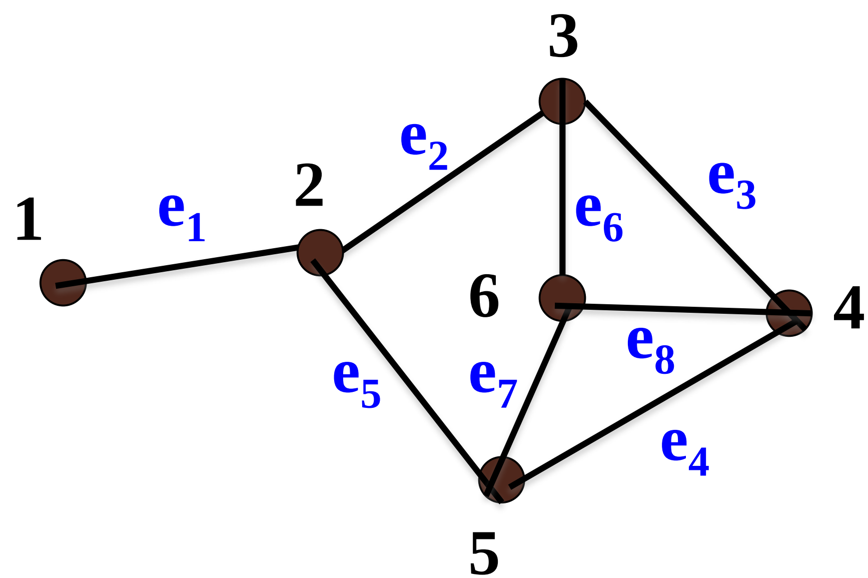
- Elle permet de determiner en un temps uniforme si une arête entre deux sommets donnés existe

Exercice 1 : Représentation d‘un graphe

On considère le graphe suivant :

Representer le graphe

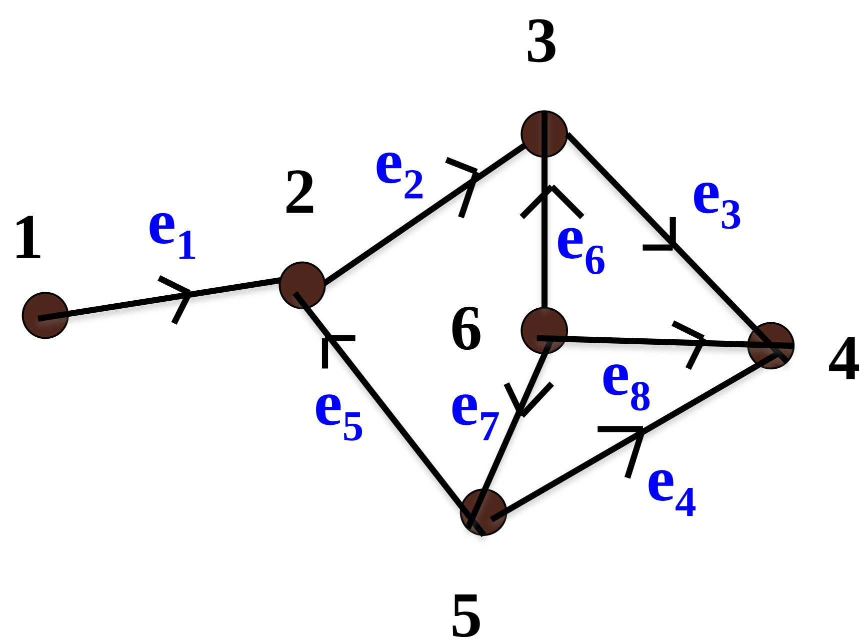
- Par une matrice d‘adjacence
- Par des listes d‘adjacence



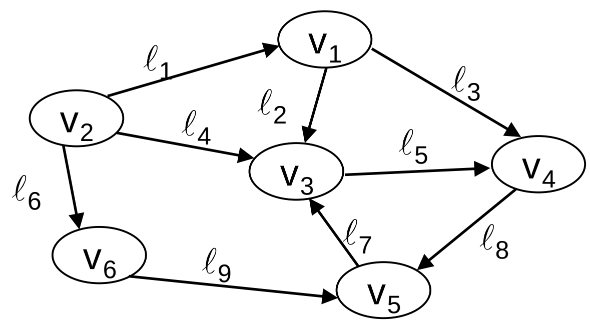
Exercise 2 : Représentation d'un graphe orienté

On considère le graphe orienté suivant :

Representer le graphe
- Par une matrice d'adjacence
- Par des listes d'adjacence



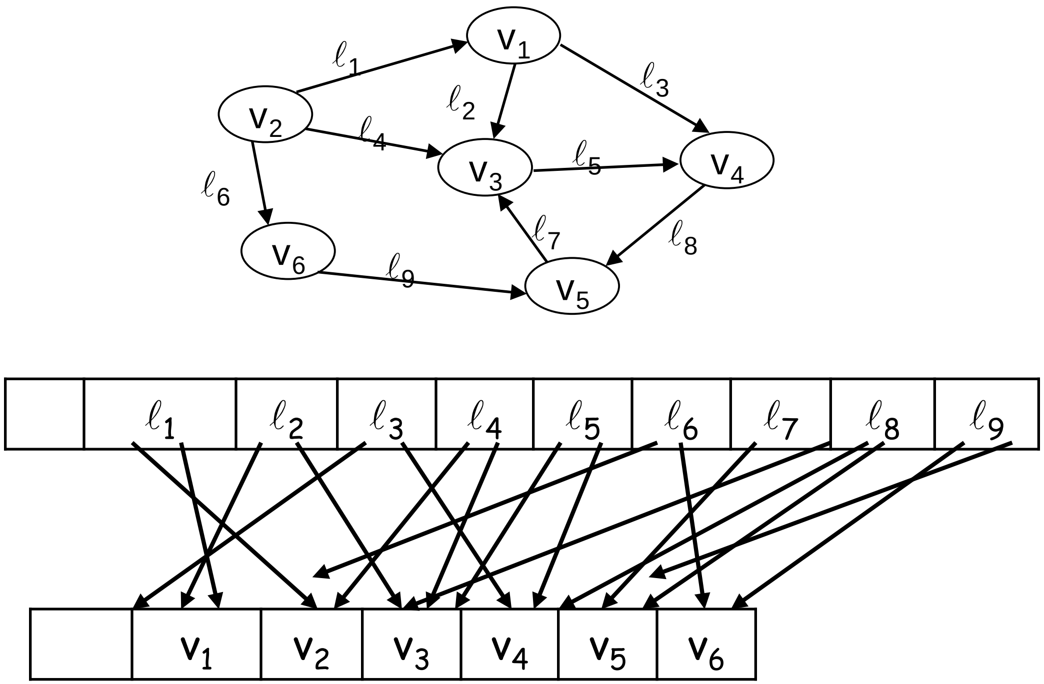
Autre représentation: Matrice d'incidence



	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ_7	ℓ_8	ℓ_9
v_1	-1	1	1	0	0	0	0	0	0
v_2	1	0	0	1	0	1	0	0	0
v_3	0	-1	0	-1	1	0	-1	0	0
v_4	0	0	-1	0	-1	0	0	1	0
v_5	0	0	0	0	0	0	1	-1	-1
v_6	0	0	0	0	0	-1	0	0	1

Espace occupé: $\boxed{n \times m}$

Structure liste d'arêtes - Exemple



Espace: $\boxed{n + m}$

La structure ‘liste d’arêtes’

- Objet sommet :
  - élément
  - référence à la position dans la séquence de sommets
- Objet arête :
  - élément
  - référence à l’origine de l’arête
  - référence à la destination de l’arête
  - référence à la position dans la séquence d’arêtes
- Séquence de sommets :
  - séquence d’objets sommets
- Séquence d’arêtes :
  - séquence d’objets arêtes