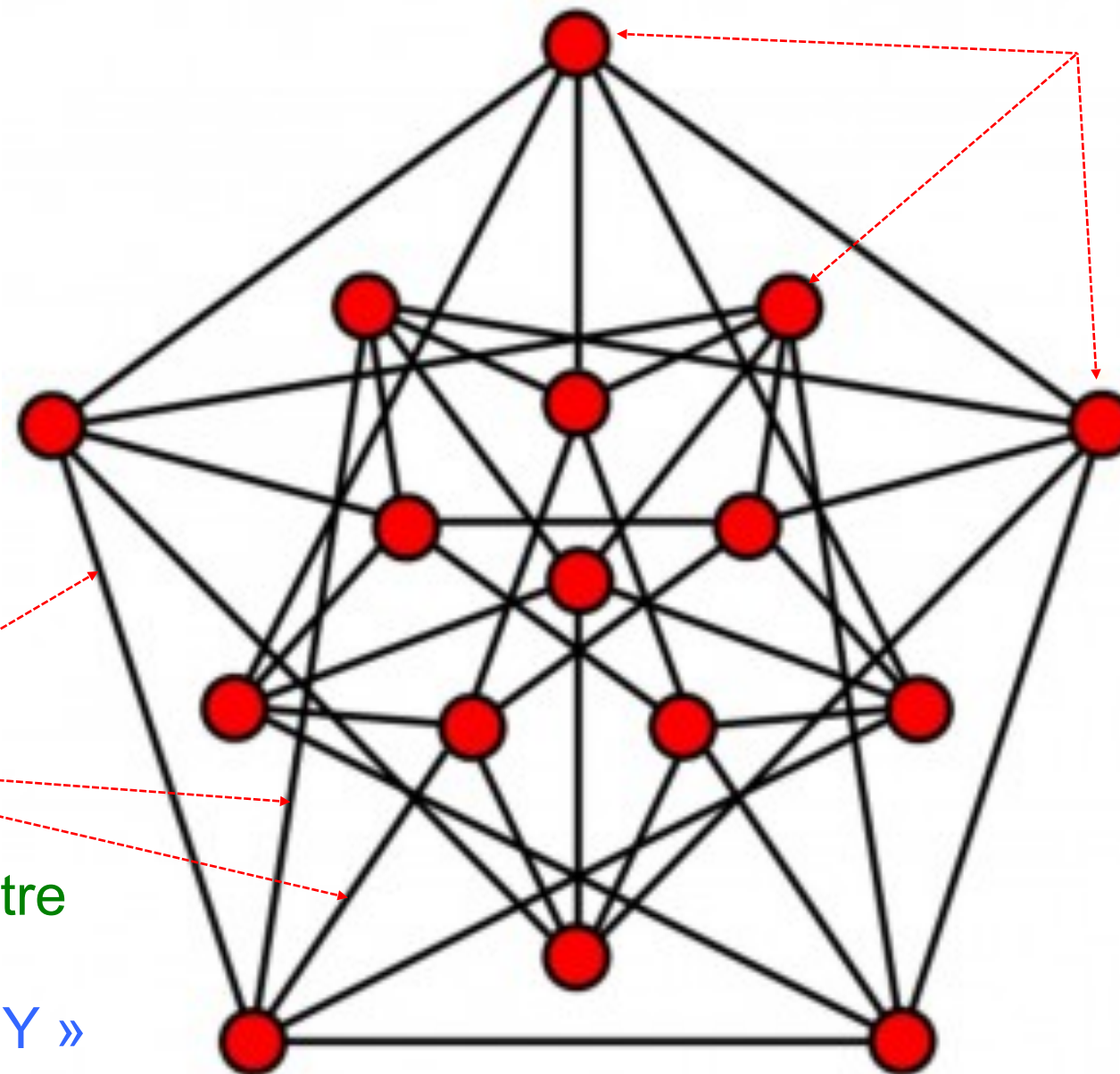
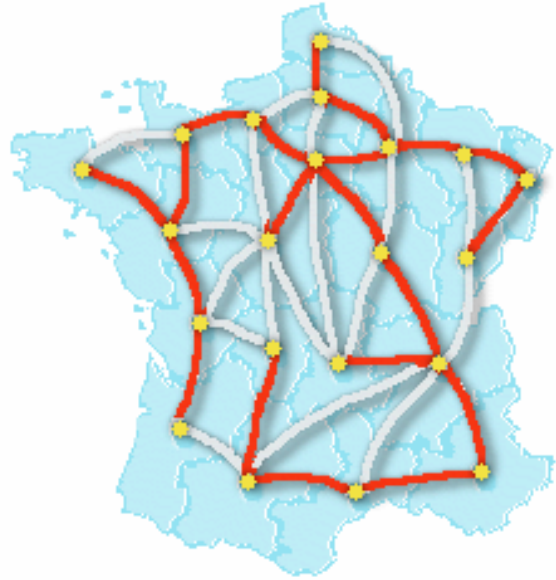


Algorithmique des graphes

L2 S4 Informatique

Qu'est ce qu'un graphe :

un objet mathématique et une structure de donnée modélisant des éléments d'un ensemble en relation

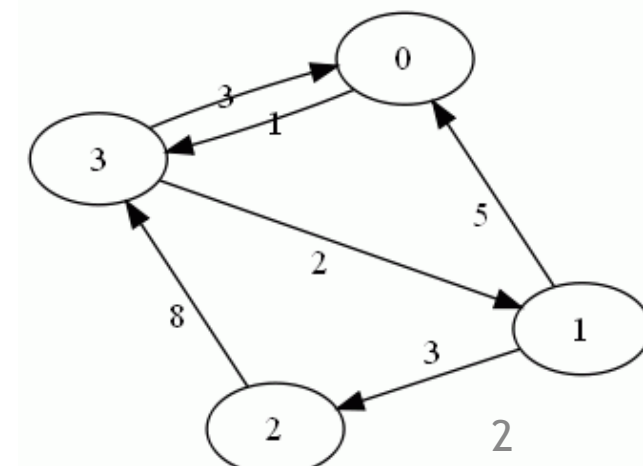


Elements

- Lieux
- Données
- Routeurs
- Instructions,
- ..

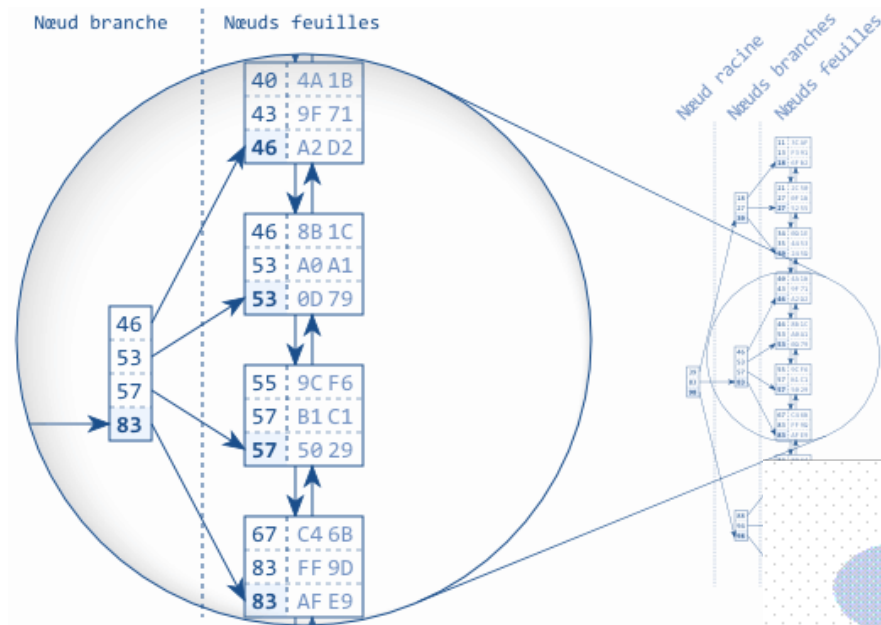
Relations

- « Il existe un route entre deux villes »
- « X est plus petit que Y »

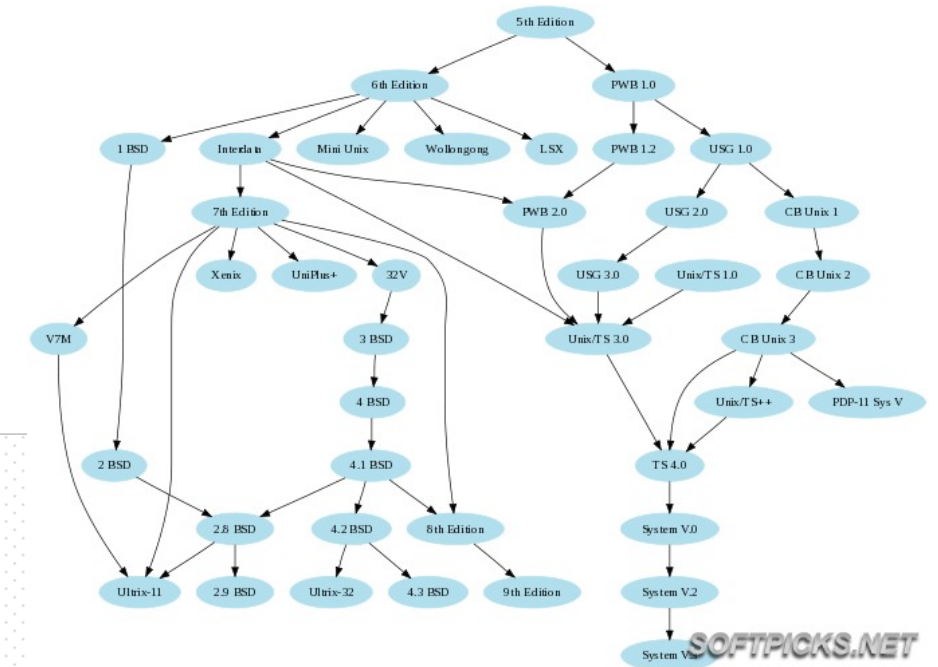


Dominique Barth, UVSQ

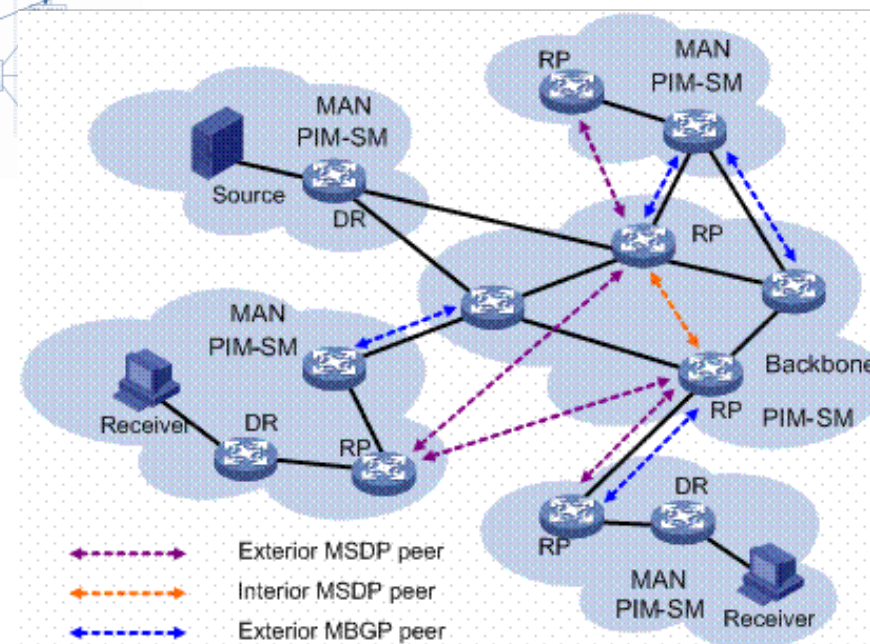
Au coeur de plusieurs disciplines en informatique ...



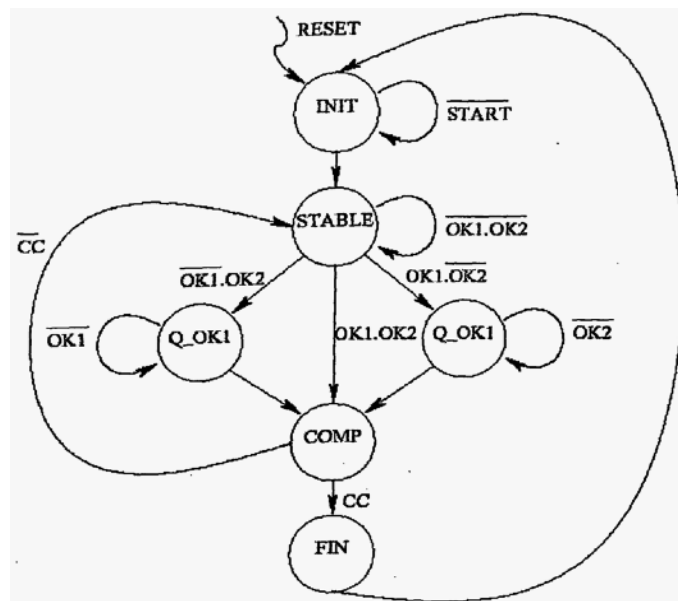
Bases de données



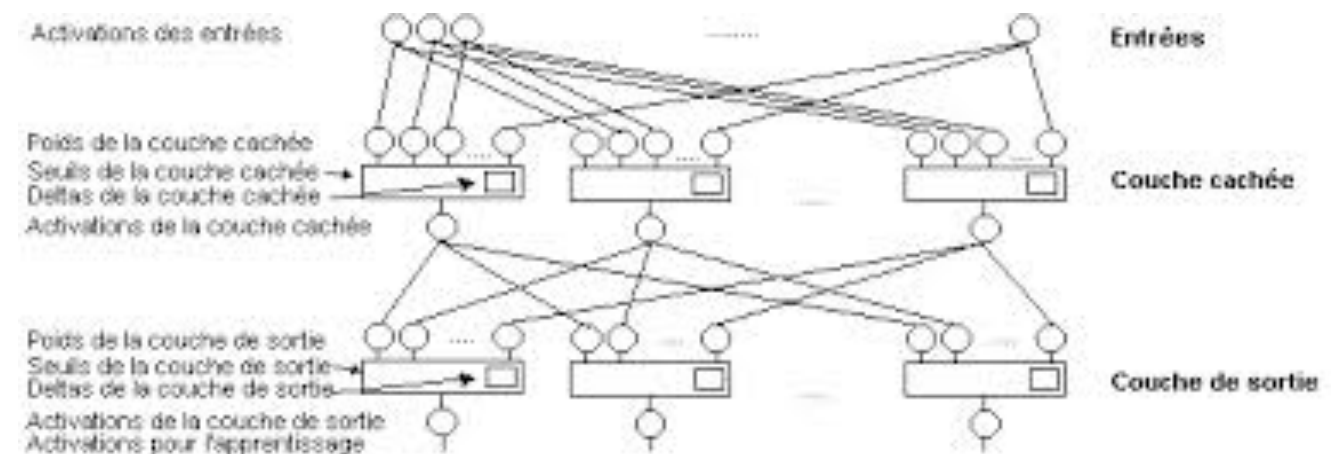
Génie logiciel



Réseaux et télécoms

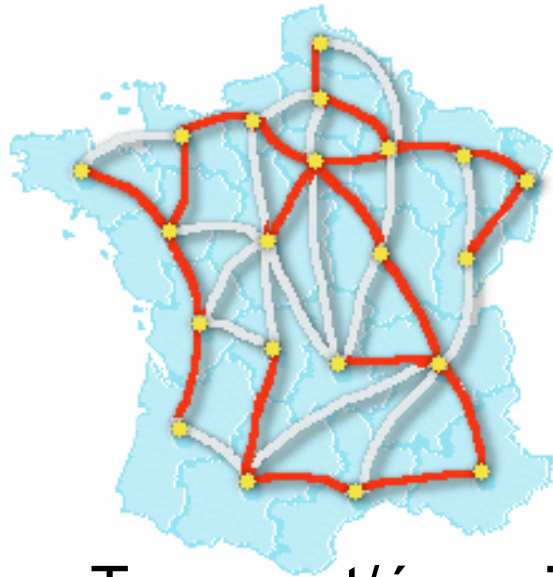


Cryptographie et sécurité



Architecture

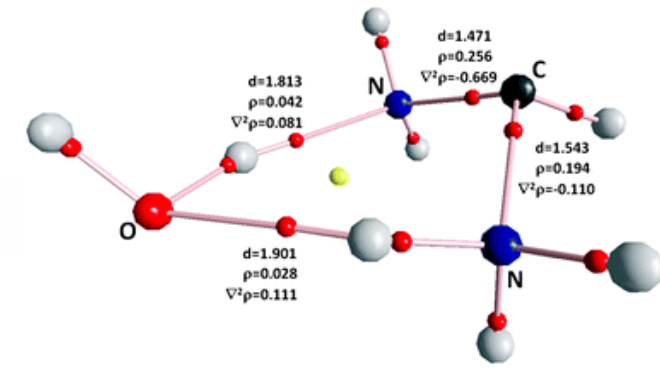
.. Et au coeur de plusieurs domaines d'applications ...



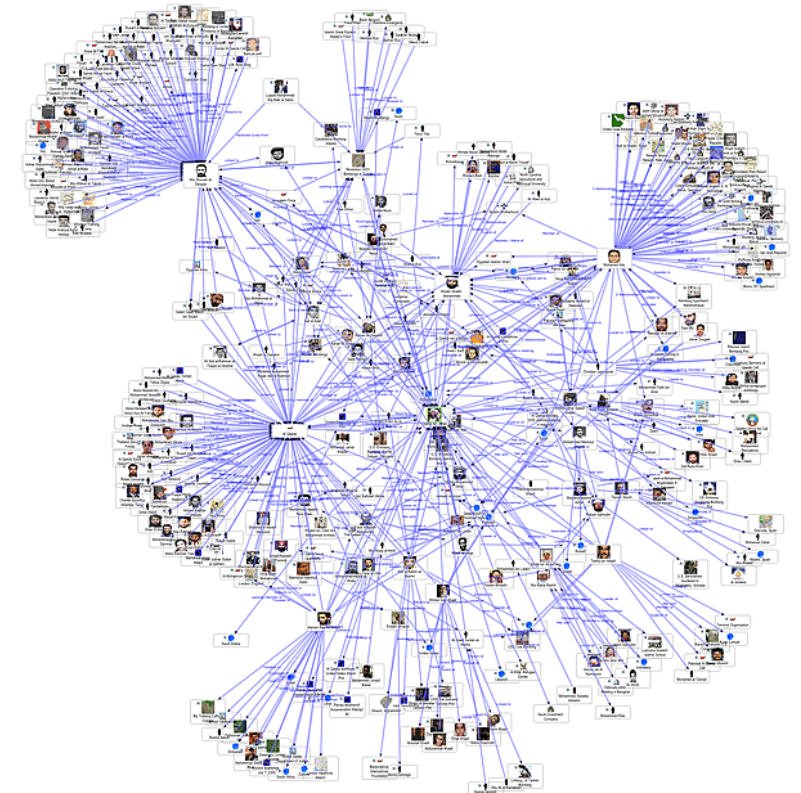
Transport/énergie



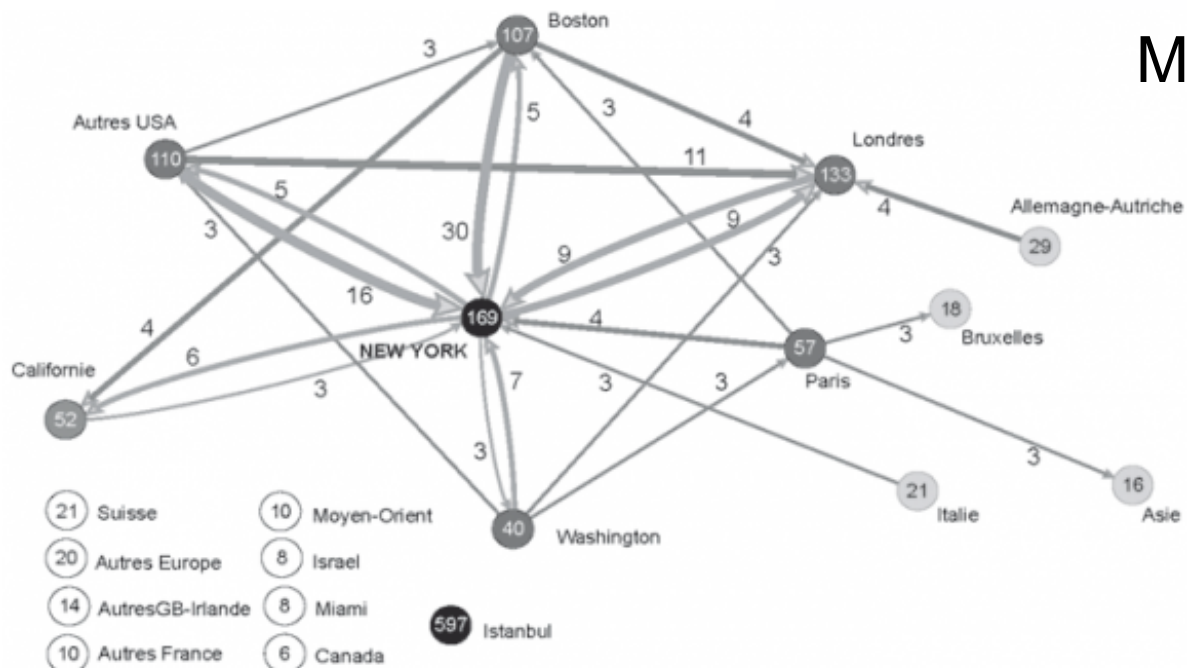
Médecine et santé



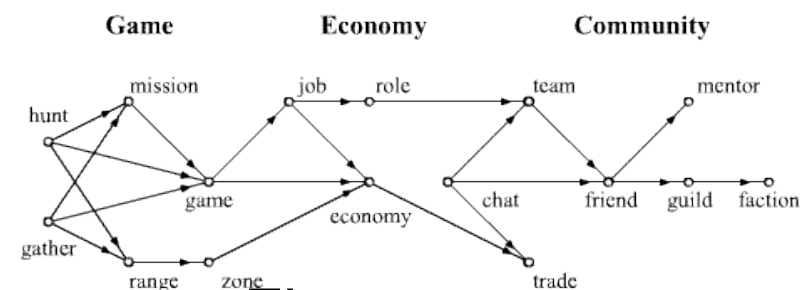
Biologie/Chimie



Réseaux sociaux



Sciences humaines et sociales

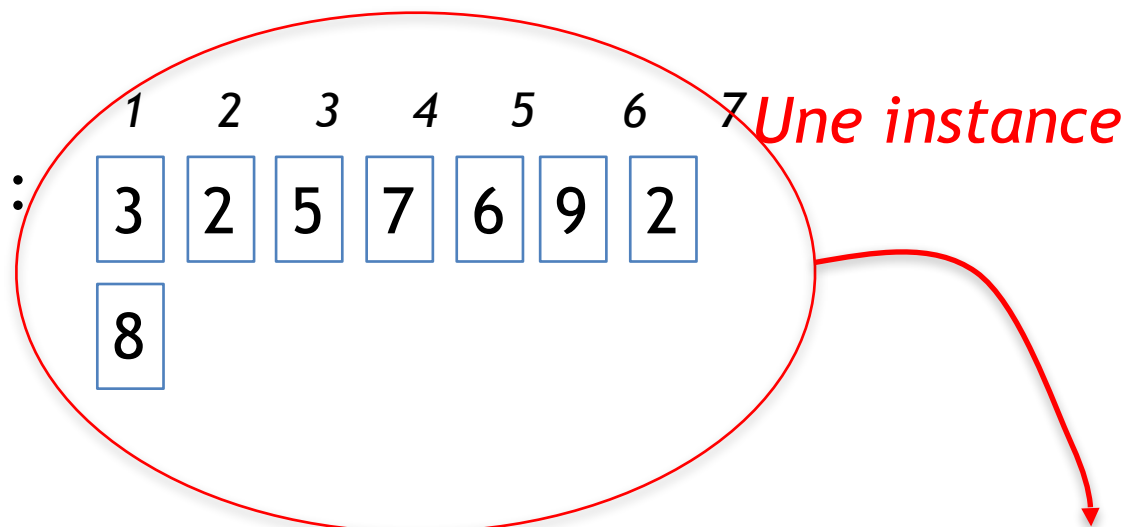


Finance

Petit rappel de complexité d'algorithmes

Données :

- Un tableau d'entier de N cases T :
- Un entier X :



Question : X est-il dans T?

Taille du problème : nombre de variables élémentaires pour coder les données (ici $N+1$)

(attention, parfois descendre au niveau du nombre de bits.)

Opérations : affectation, calcul, comparaison,...

Algorithme

$i \leftarrow 1$

Tant que $(i \leq N)$ et $(T[i] = X)$

Faire $i \leftarrow i+1$

Retourner $(i \leq N)$

← 1 opération

← 3 opérations, au plus $N+1$ fois

← 2 opérations, au plus N fois

← 1 opération

Complexité dans le pire des cas : $1 + 3(N+1) + 2N + 1 = 5N+5$ (type $aN+B$, fonction de $O(N)$)

Vraiment utile de s'embêter?

Difficulté d'un problème : plus petite complexité d'un algorithme le résolvant

Taille d'un problème : nombre de sommets, de liens

| | Complexité | Nombre de données traitées / 24h | processeur x 1000 |
|-------------------------|-------------------|-------------------------------------|-------------------|
| $a_k N^k + \dots + a_0$ | Linéaire | 1 million | x1000 |
| | Polynomiale (k=4) | 4000 | x 2 |
| $c \cdot a^N$ | Exponentielle | 150 | +20 |
| $c \cdot N!$ | Factorielle | 12 | +2 |

Les graphes : concepts de base

Définitions

- Relation sur V : Application A de $V \times V$ dans $\{\text{Vrai}, \text{Faux}\}$

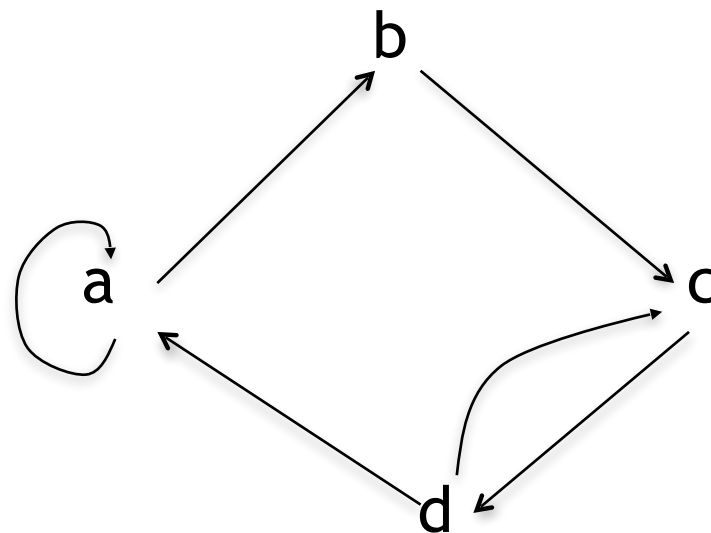
Exemple sur $V = \{a, b, c, d\}$

$$A = \{(a, a), (a, b), (b, c), (c, d), (d, c), (d, a)\}$$

(*Extension*)

| | a | b | c | d |
|---|---|---|---|---|
| a | 1 | 1 | 0 | 0 |
| b | 0 | 0 | 1 | 0 |
| c | 0 | 0 | 0 | 1 |
| d | 1 | 0 | 1 | 0 |

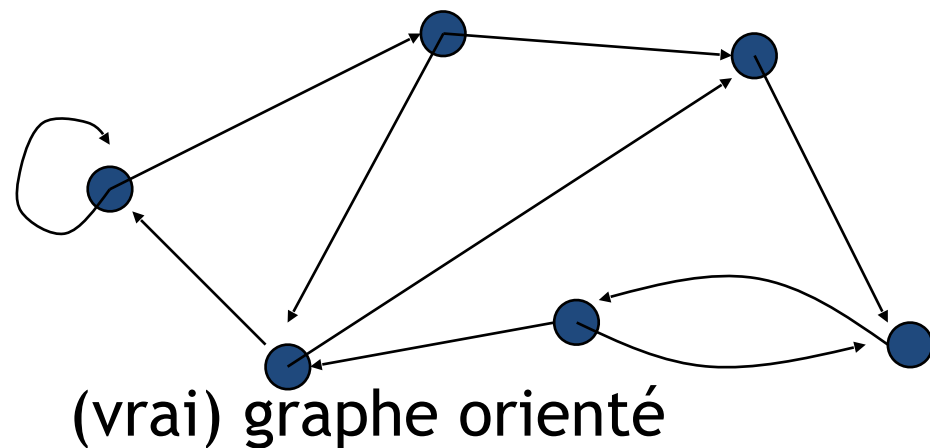
(*Matrice d'adjacence*)



(*Graphe de la relation*)

Définitions

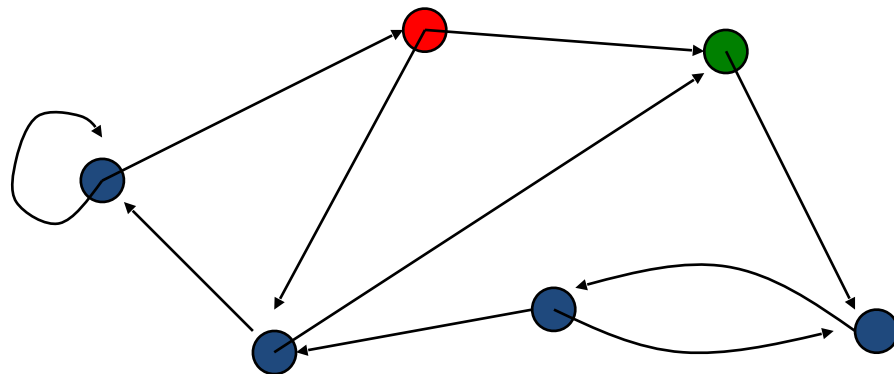
- Un **graphe orienté** G est un couple (V,A) où
 - V est un ensemble d'éléments appelés **sommets** ou **nœuds** (en anglais : *vertex*, *vertices*)
 - A est une partie de $V \times V$, chaque élément de A est un **arc** (*arc* en anglais).



- Degrés
- Distances, diamètre
- Chaîne, chemin, cycle, circuits
- Connexité, forte-connexité, k-connexité
- pondération, étiquetage

Définitions

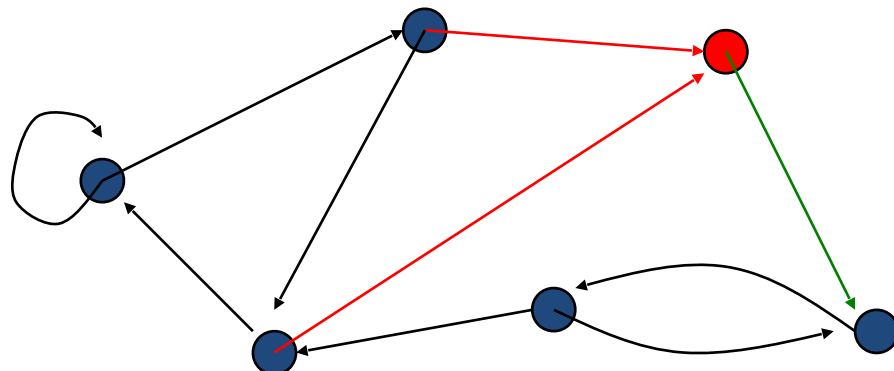
- Pour un arc (v,w) d'un graphe orienté, on dit que w est un **successeur** de v , et que v est un **prédécesseur** de w .
- Deux arcs sont consécutifs si l'extrémité finale de l'un est l'extrémité initiale de l'autre $((u,v),(v,z))$.
- Une **boucle** est un arc (u,u)



(vrai) graphe orienté

Définitions

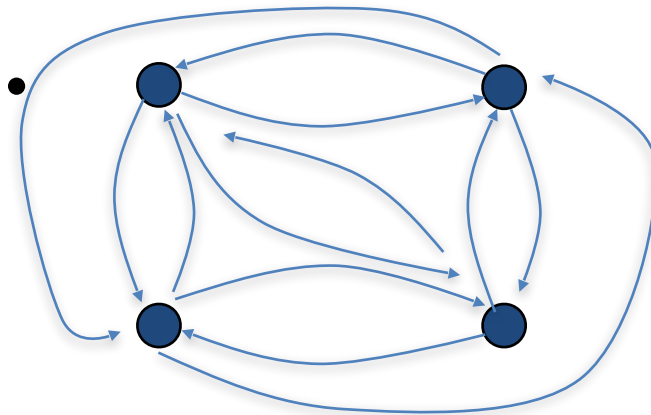
- Un arc est **entrant (sortant)** d'un sommet si ce sommet est son extrémité initiale (finale).
- Degrés d'un sommet
 - Le **degré sortant** de v , noté $d^+(v)$, est le nombre d'arcs dont v est origine.
 - Le **degré entrant** de v , noté $d^-(v)$, est le nombre d'arcs dont v est extrémité finale
- Le **degré entrant (sortant)** d'un graphe est le degré entrant(sortant) maximum de ses sommets.



(vrai) graphe orienté

Définitions

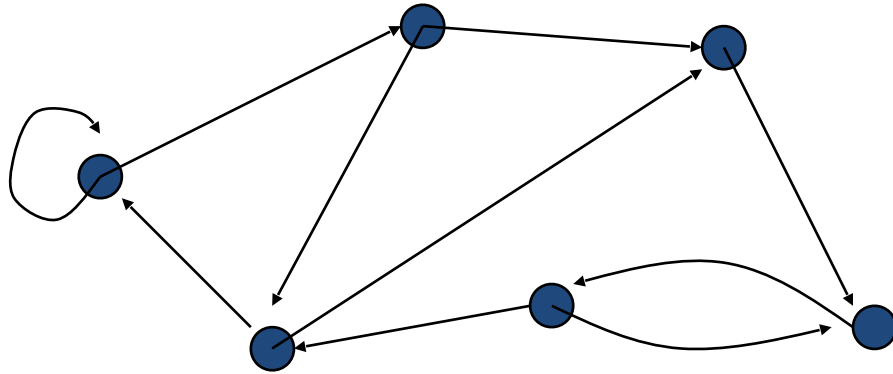
- Un graphe orienté G est complet si pour tout couple de sommets u et v , l'arc (u,v) existe ou l'arc (v,u) existe.



- Un graphe orienté $G=(V,A)$ est orienté-symétrique ssi (u,v) est un arc implique que (v,u) est un arc

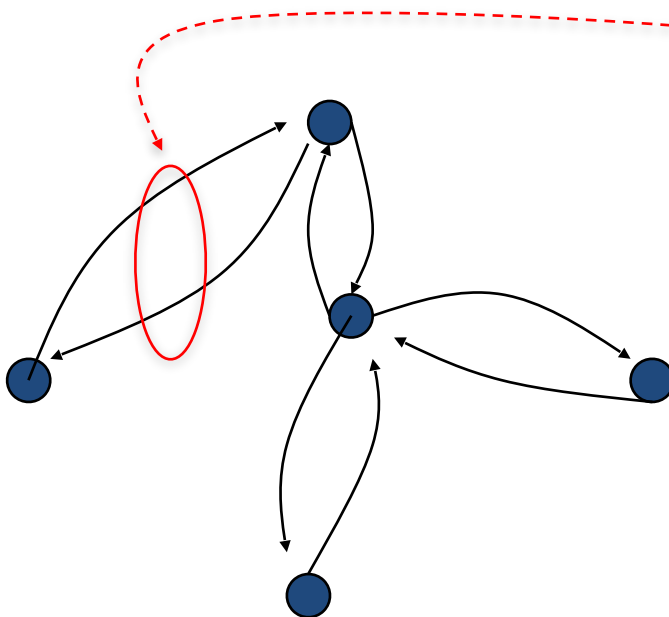
Graphes : relation (application de $V \times V$ dans $\{\text{Vrai}, \text{Faux}\}$)

Graphe de la relation, matrice d'adjacence, listes par extension

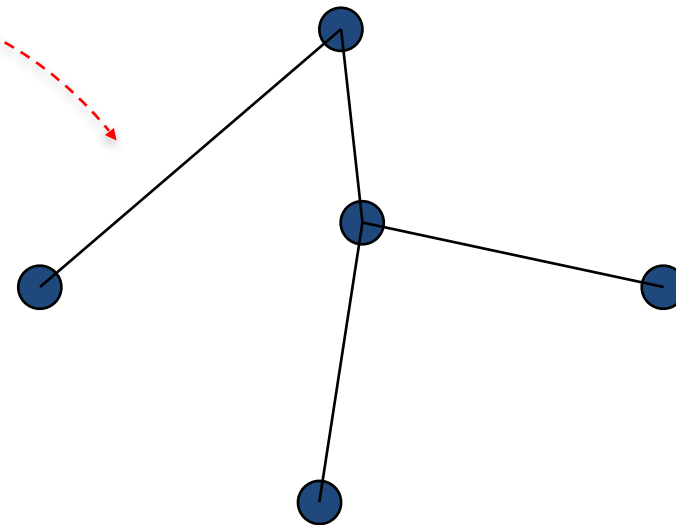


(vrai) graphe orienté

- Degrés
- Distances, diamètre
- Chaîne, chemin, cycle, circuits
- Connexité, forte-connexité, k-connexité
- pondération, étiquetage



Graphe orienté symétrique



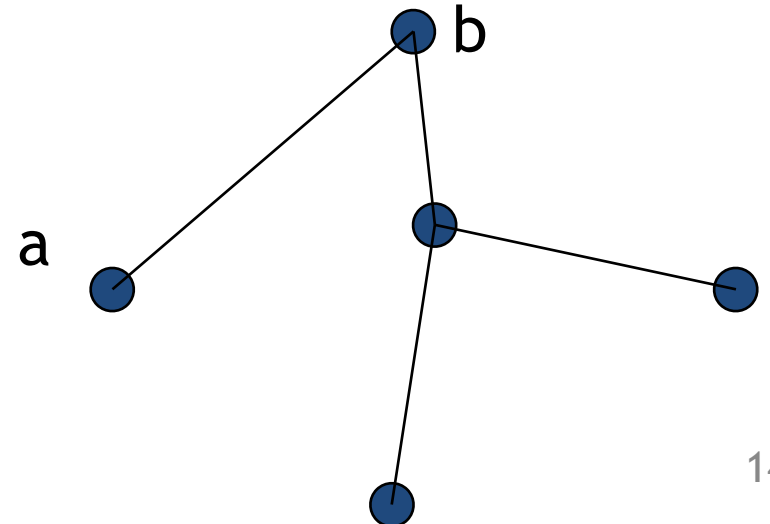
Graphe non-orienté

Définitions - cas non-orienté

Une vision paresseuse du graphe orienté symétrique...

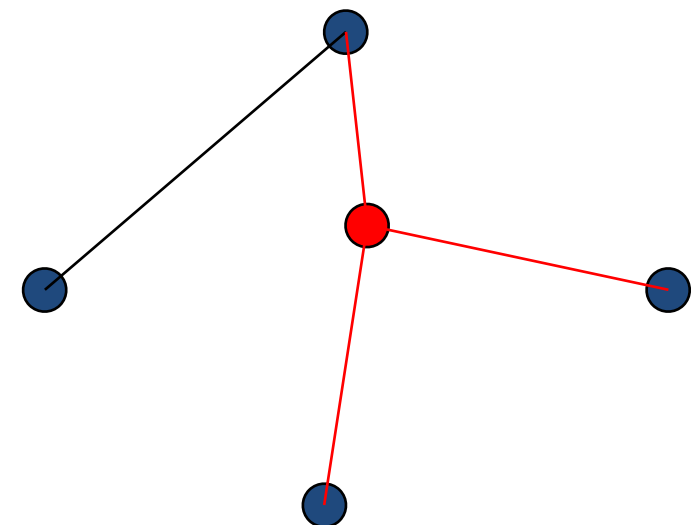
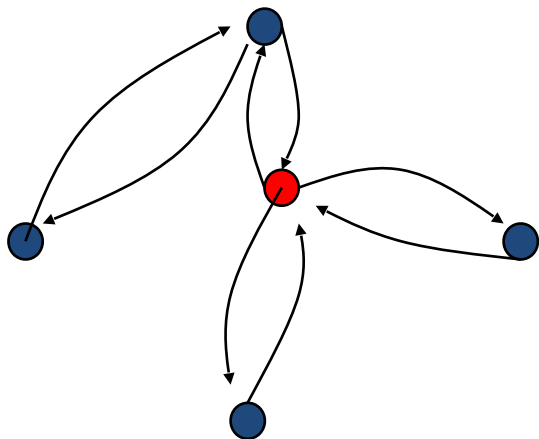
- Un **graphe non orienté** est un couple (V,E) où
 - V est un ensemble d'éléments appelés **sommets** ou **nœuds** (en anglais : *vertex*, *vertices*)
 - E est une partie de l'ensemble des paires (non ordonnées) de sommets. Chaque élément de E est une **arête** (*edge* en anglais).

$$[a,b]=[b,a]$$



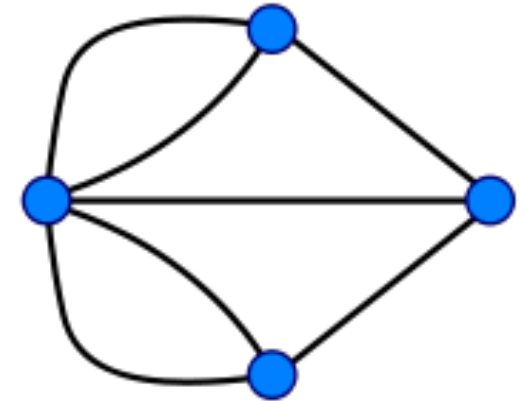
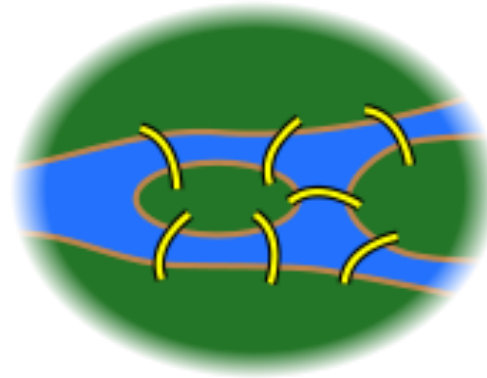
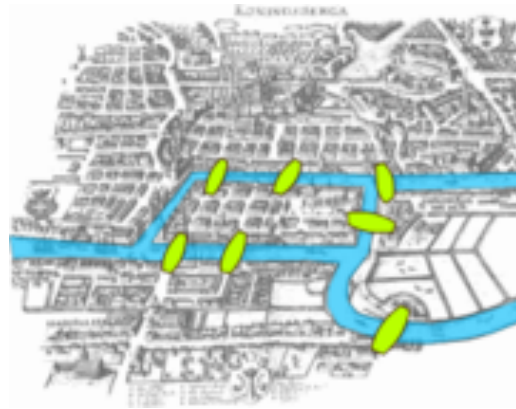
Définitions - cas non-orienté

- Une arête est **incidente** à un sommet si ce sommet est l'une de ses extrémités.
- Le **degré** d'un sommet v , noté $d(v)$, est le nombre d'arêtes qui lui sont incidentes.
- Le **degré** d'un graphe est le degré maximum de ses sommets.

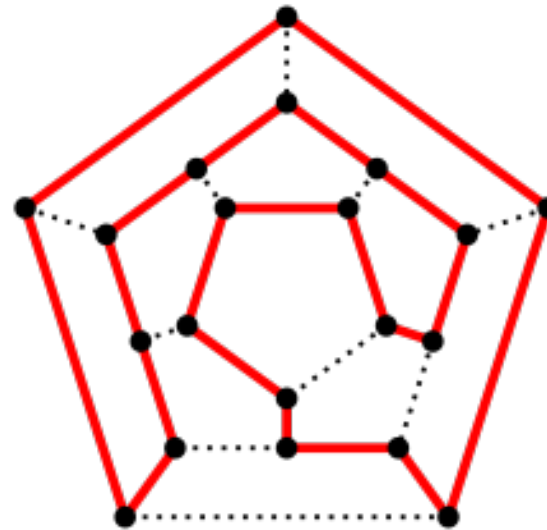


Graphe non-orienté

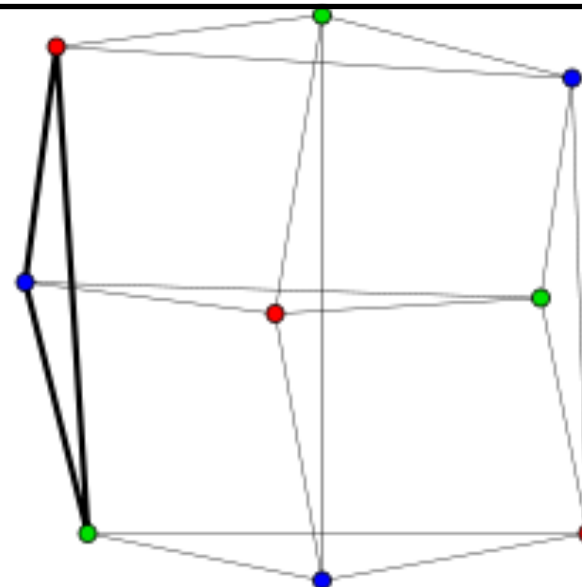
L.P. Euler (1707 – 1783)



W.R. Hamilton (1805 – 1865)



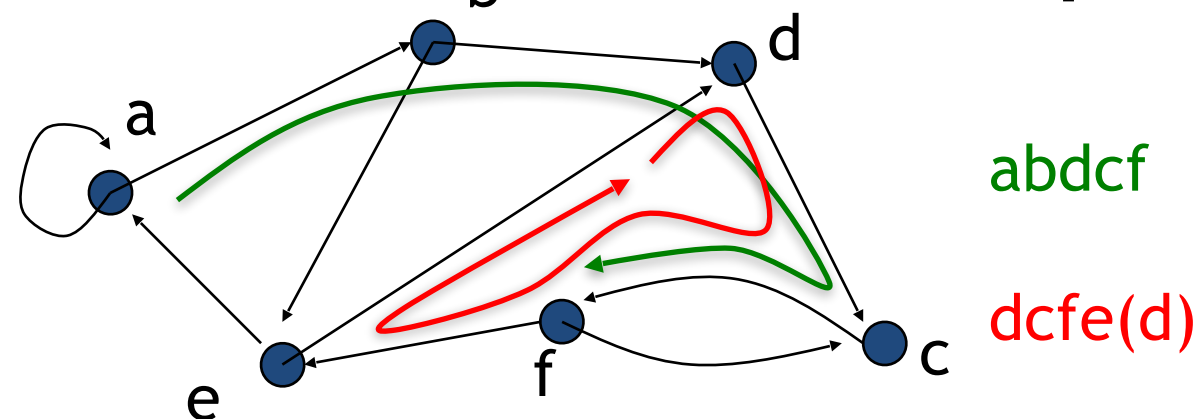
Claude Berge (1926 – 2002)



Chaines, chemins, cycles et circuits

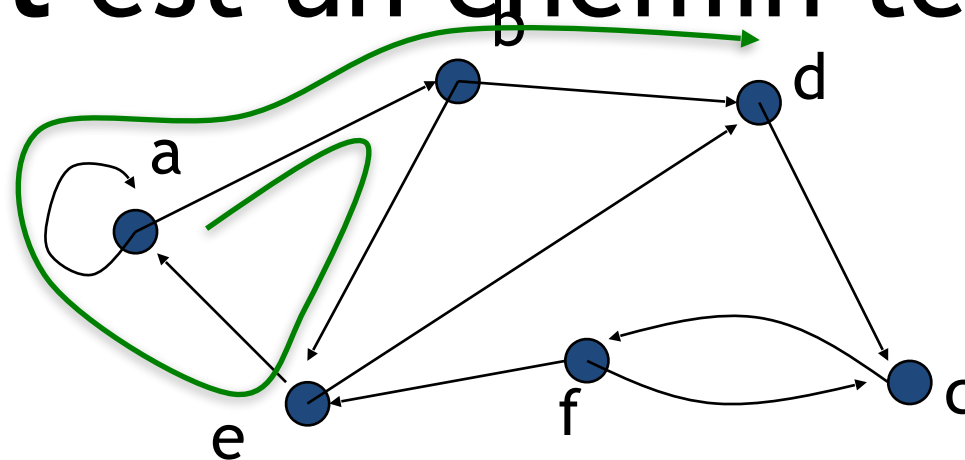
Définitions - cas orienté

- Un **chemin** est une suite de sommets $v_0 v_1 \dots v_n$ telle que chaque couple de sommets successifs est un arc. La **longueur** du chemin est n .
- Un **circuit** est un chemin tel que $v_0 = v_n$.



Définitions - cas orienté

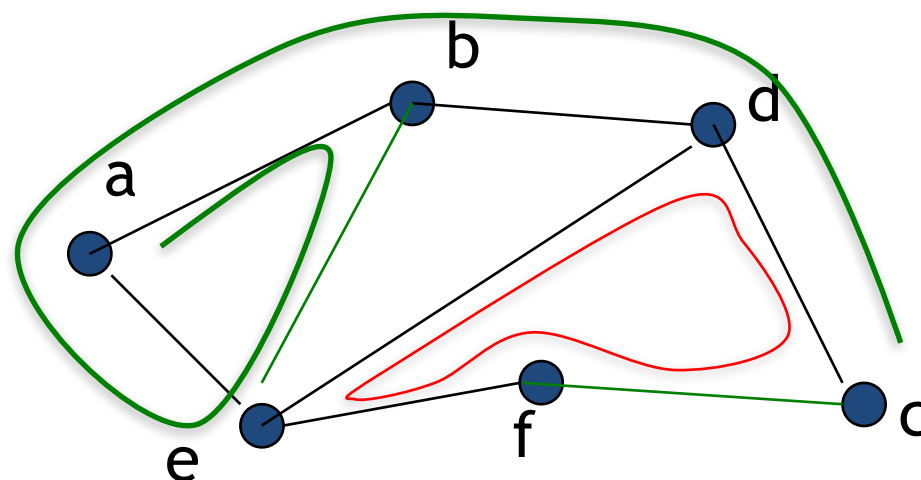
- Un **chemin** est une suite de sommets $v_0 v_1 \dots v_n$ telle que chaque couple de sommets successifs est un arc. La **longueur** du chemin est n .
- Un **circuit** est un chemin tel que $v_0 = v_n$.



abeabd (non élémentaire)

Définitions - cas non-orienté

- Une **chaîne** est une suite de sommets $v_0v_1\dots v_n$ telle que chaque paire de sommets successifs v_iv_{i+1} est soit un arc (v_iv_{i+1}) , soit un arc $(v_{i+1}v_i)$
- La **longueur** de la chaîne est n .
- Un **cycle** est une chaîne telle que $v_0=v_n$.



abeabdc

edcf(e)