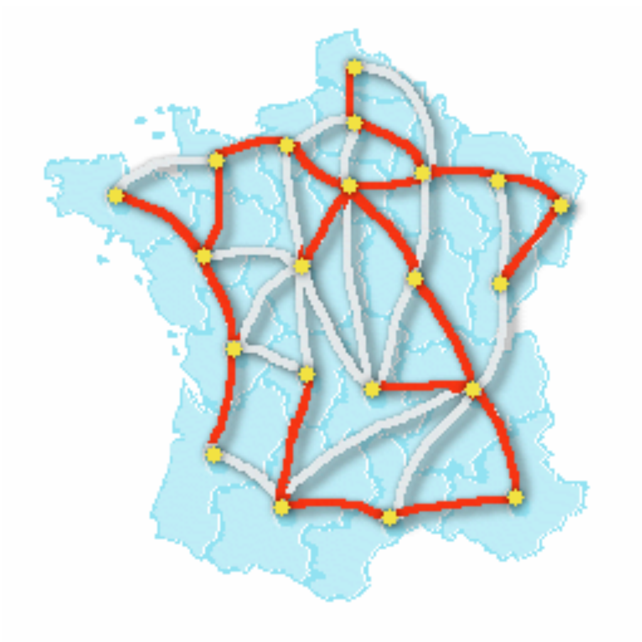
Algorithmique des graphes

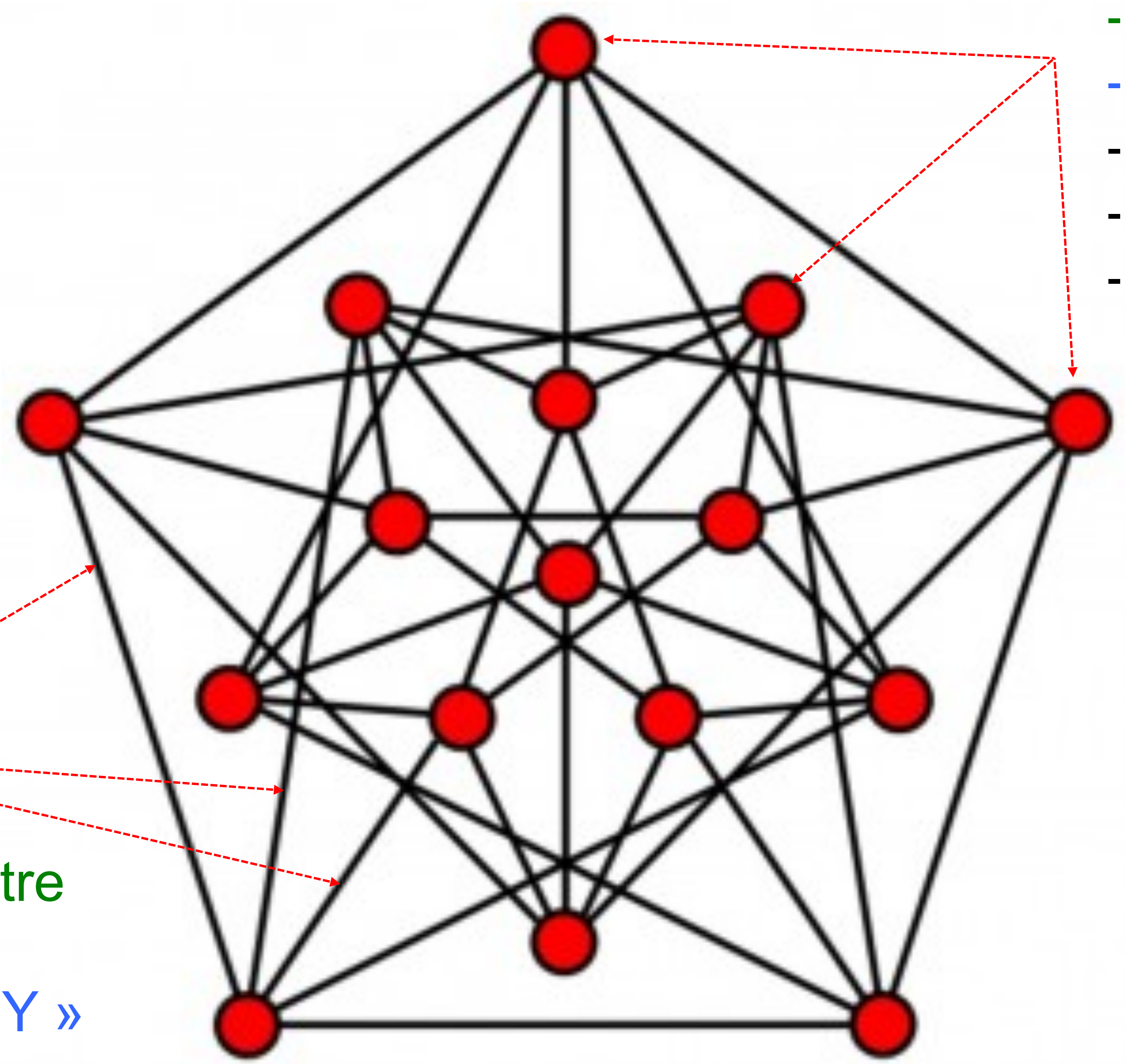
L2 S4 Informatique

Dominique Barth, UVSQ

Qu’est ce qu’un graphe :

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



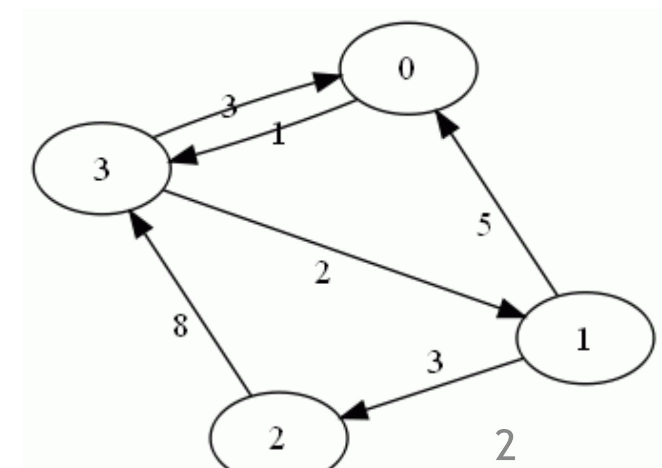


Elements

- Lieux
- Données
- Routeurs
- Insructions,
- ..

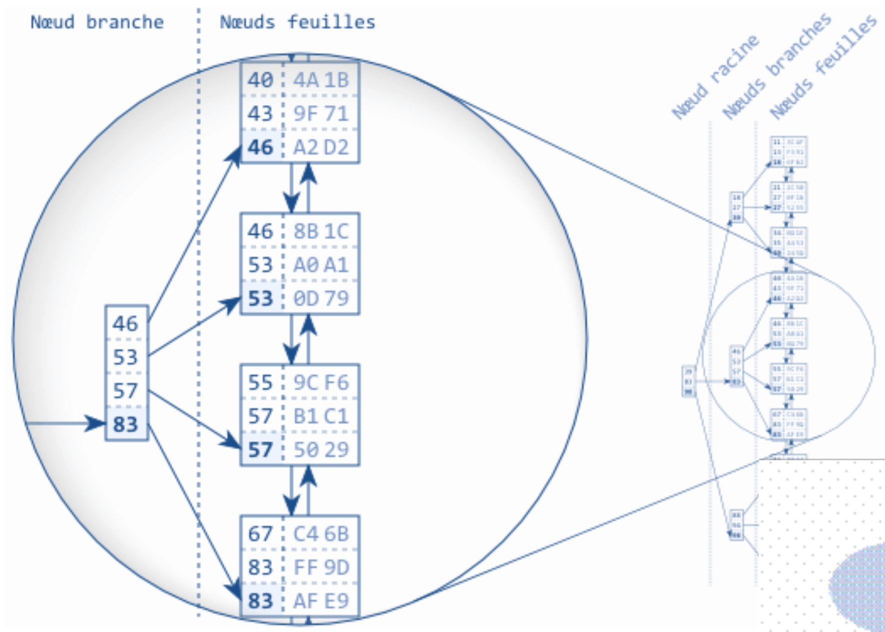
Relations

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

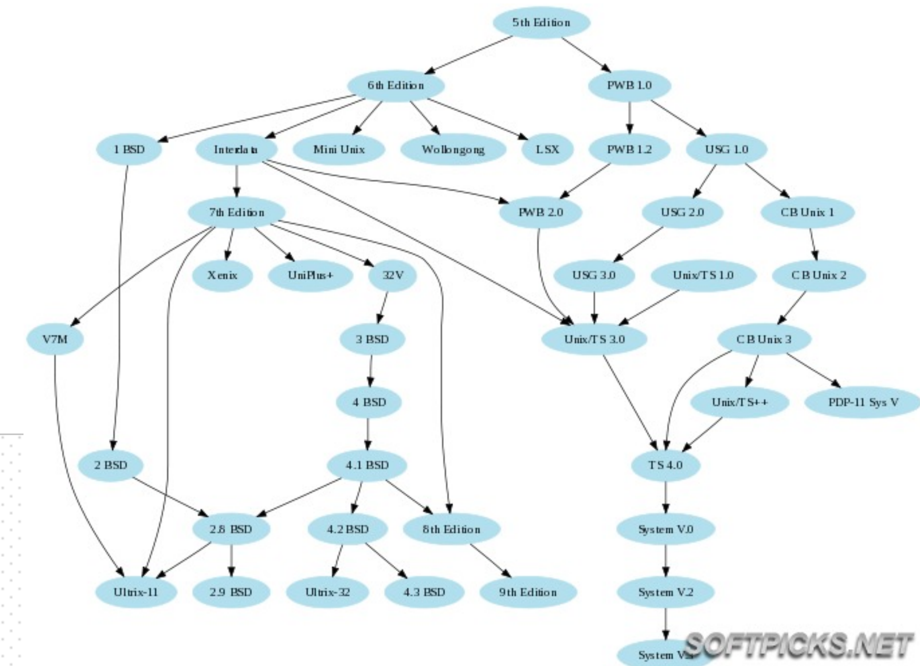


Dominique Barth, UVSQ

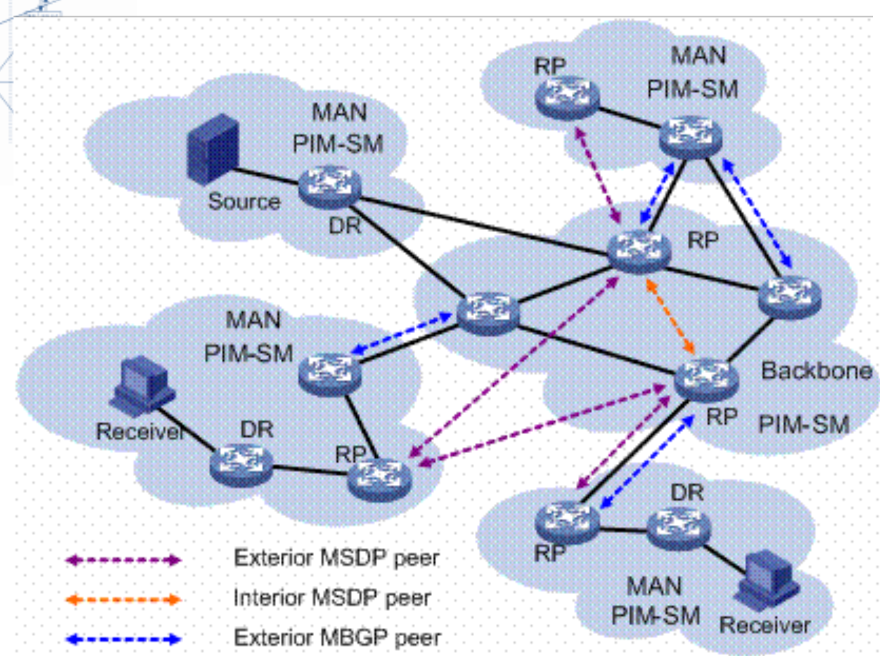
Au coeur de plusieurs disciplines en informatique …



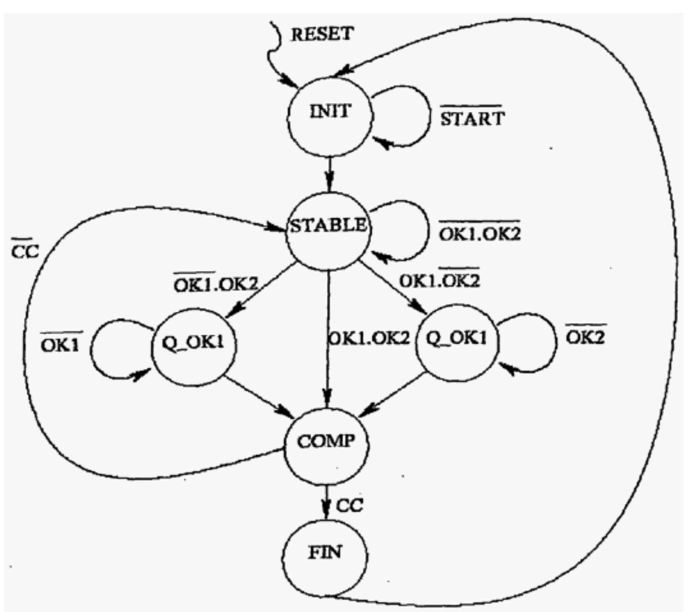
Bases de données



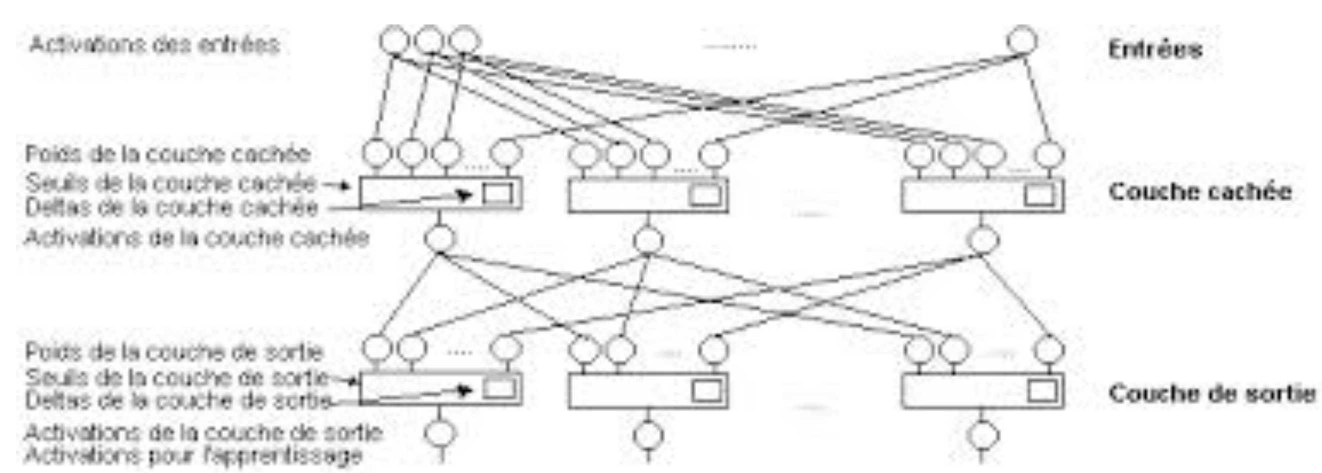
Génie logiciel



Réseaux et télécoms

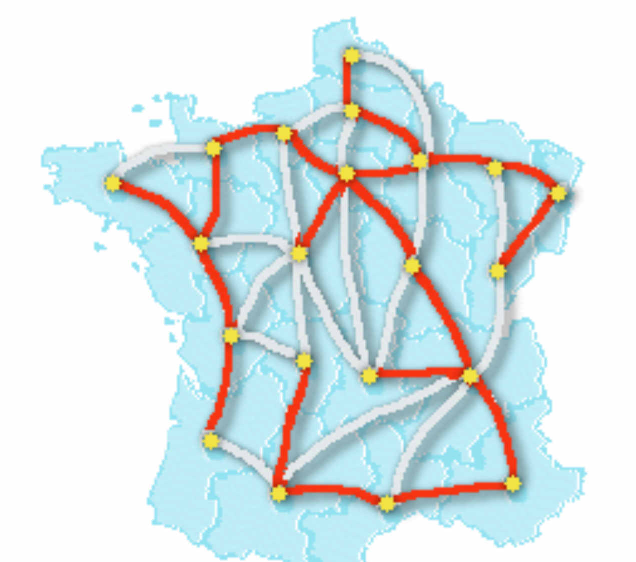


Cryptographie et sécurité

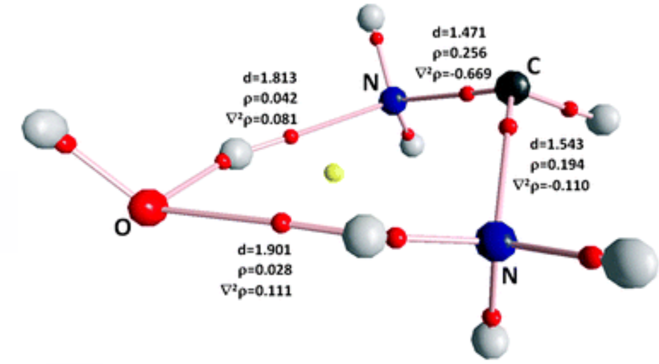


Architecture

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

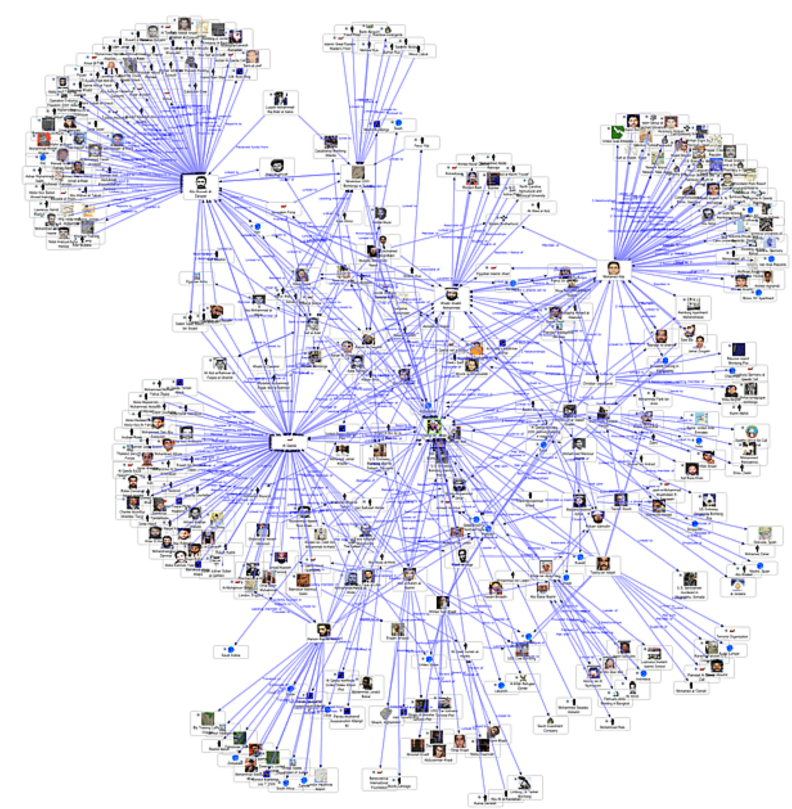


Transport/énergie

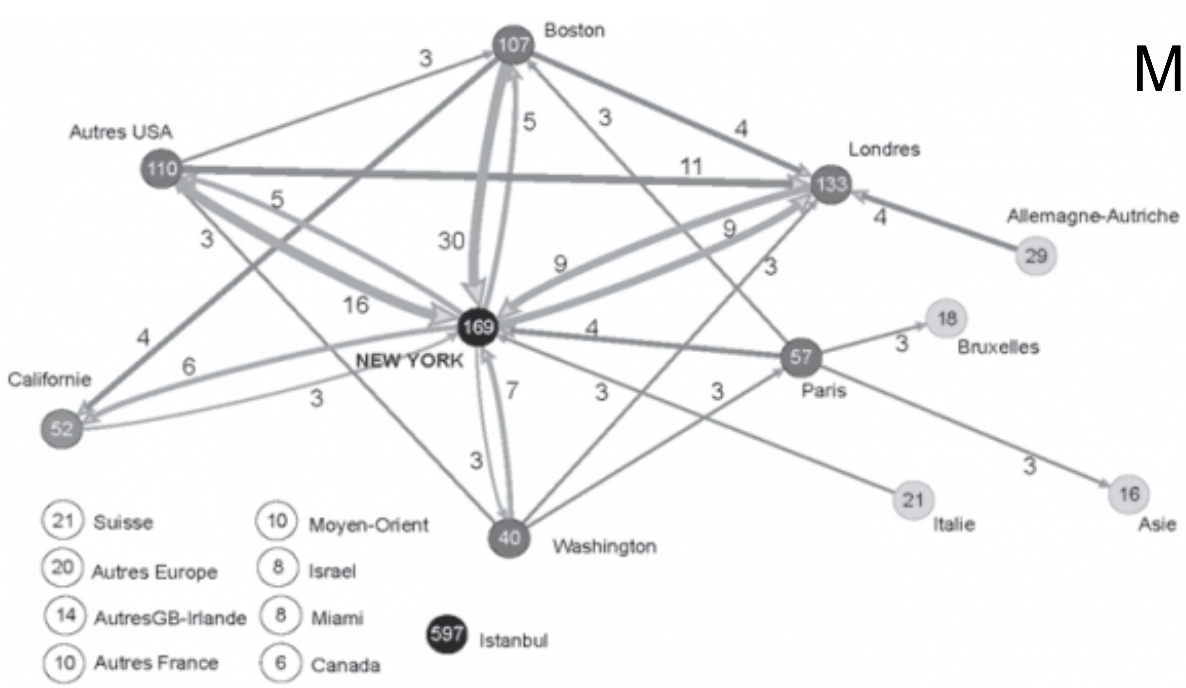


Biologie/Chimie

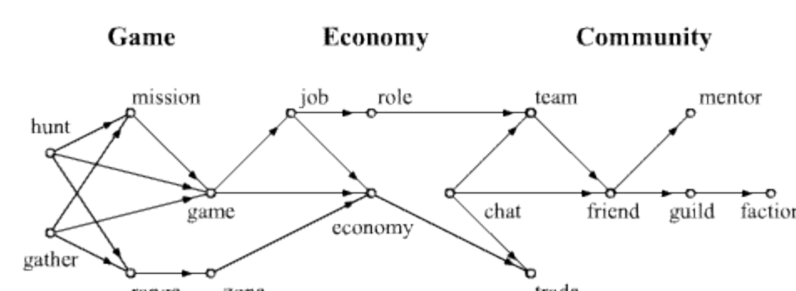
Médecine et santé



Réseaux sociaux



Sciences humaines et sociales



Finance

Dominique Barth, UVSQ

Petit rappel de complexité d'algorithmes

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

Une instance

1	2	3	4	5	6	7
3	2	5	7	6	9	2

8

Question : X est-il dans T?

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

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

Opérations : affectation, calcul, comparaison,..

Algorithme

```
i <- 1
Tant que (i<=N) et (T[i] = X)
        Faire i <- i+1
Retourner (i<=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
$aN+b$	Linéaire	1 million	x1000
$a_k N^k + .. + a_0$	Polynomiale (k=4)	4000	x 2
$c.\ a^N$	Exponentielle	150	+20
$c.\ N!$	Factorielle	12	+2

Les graphes : concepts de base

Claude Berge. Graphes. Ed. Dunod

Définitions

- Relation sur V : Application A de VxV dans {Vrai,Faux}

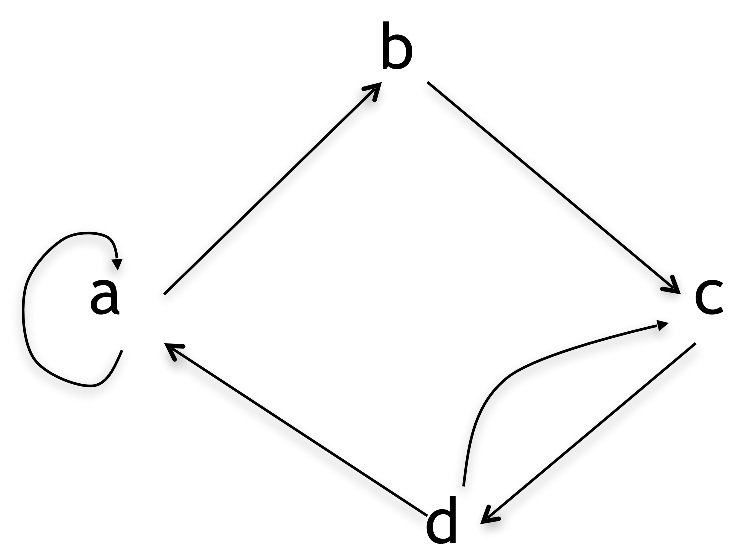
Exemple sur V={a,b,c,d}

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

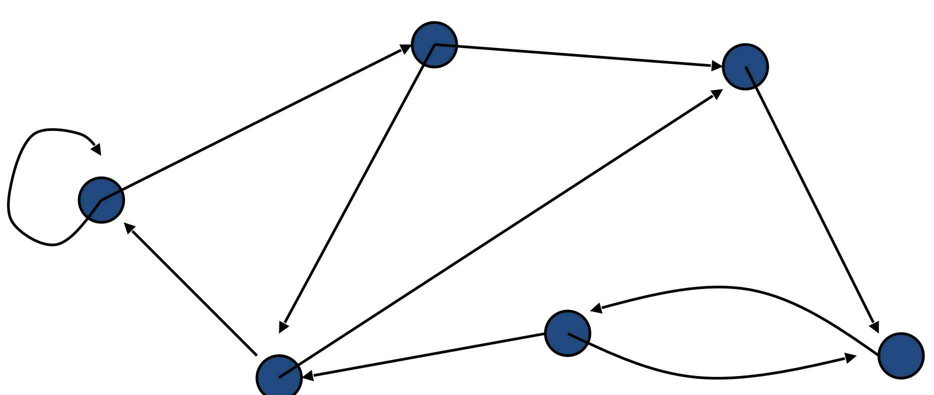
(Matrice d'adjacence)



(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 VxV, chaque élément de A est un **arc** (*arc* en anglais).

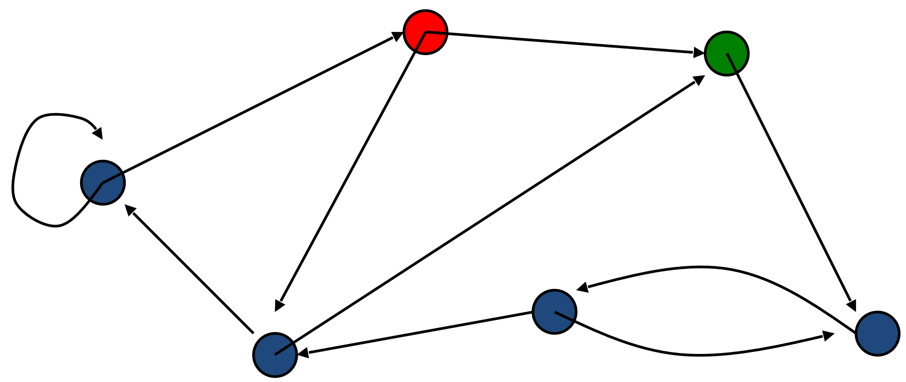


(vrai) graphe orienté

- Degrés
- Distances, diamètre
- Chaine, 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)

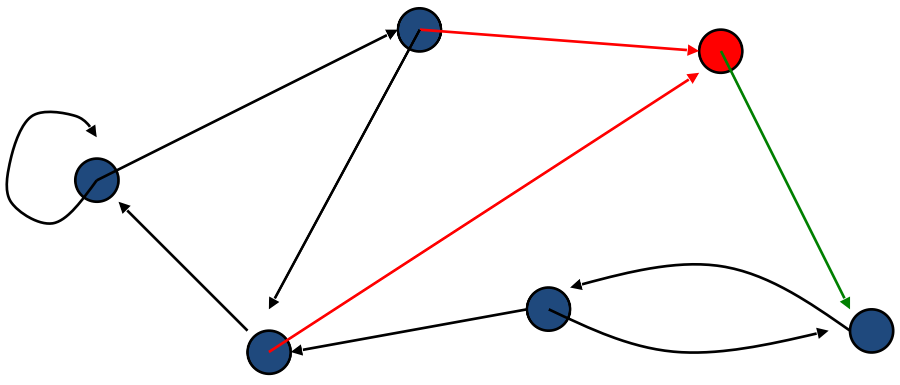


(vrai) graphe orienté

Dominique Barth, UVSQ

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.

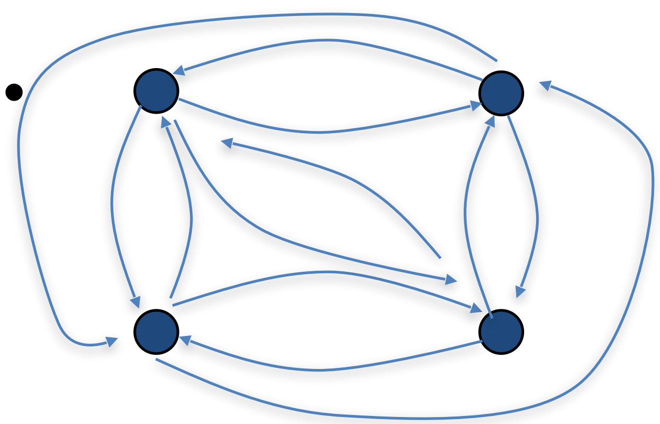


(vrai) graphe orienté

Dominique Barth, UVSQ

Définitions

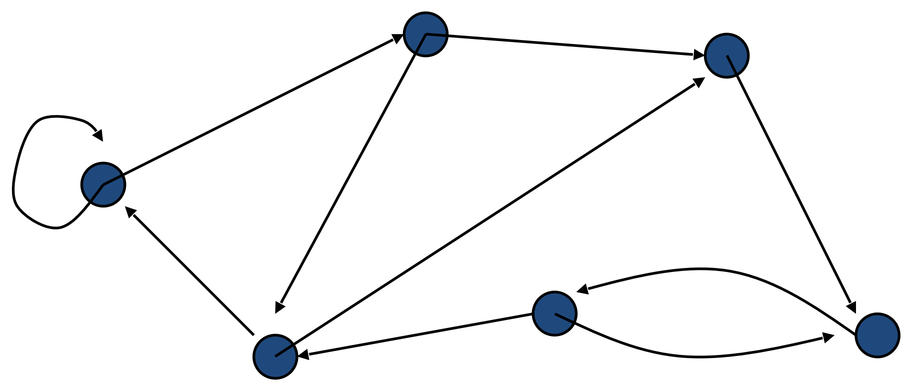
- Un **graphe orienté** G est **complet** si pour tout couple de sommets u et v, l'arc (u,v) existe ou l'arc (v,u) existe.



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

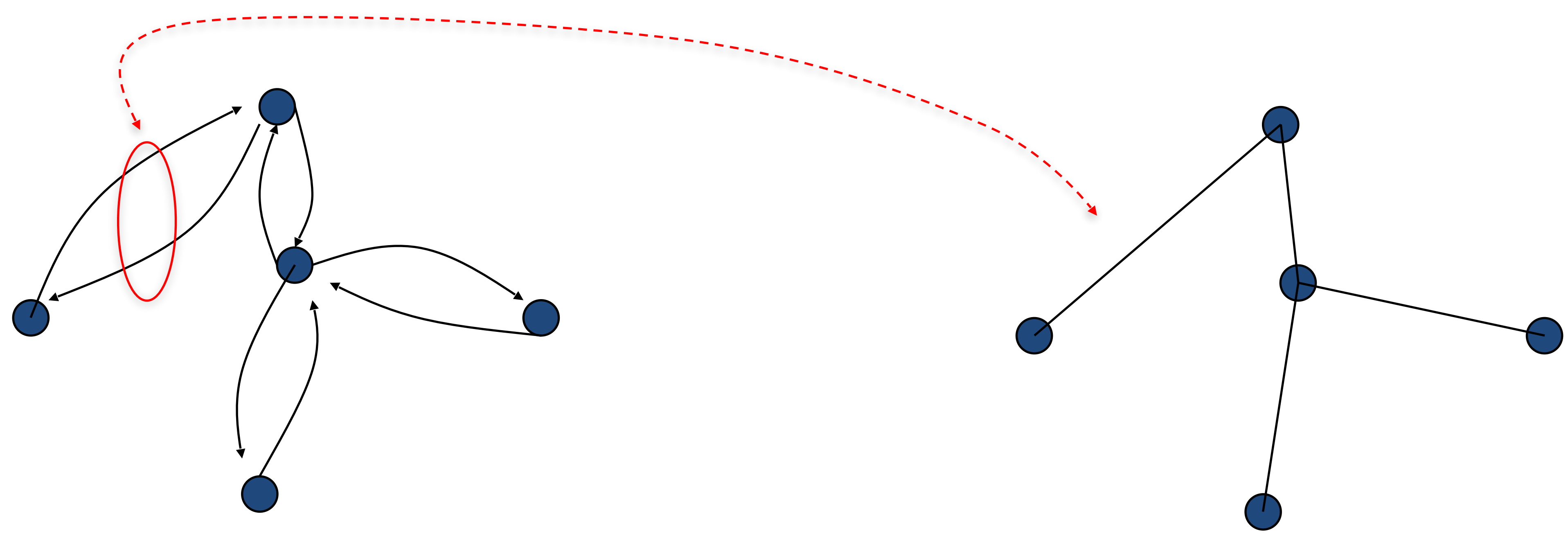
Graphes : relation (application de V*V dans {Vrai,Faux})

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



(vrai) graphe orienté

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



Graphe orienté symétrique

Graphe non-orienté

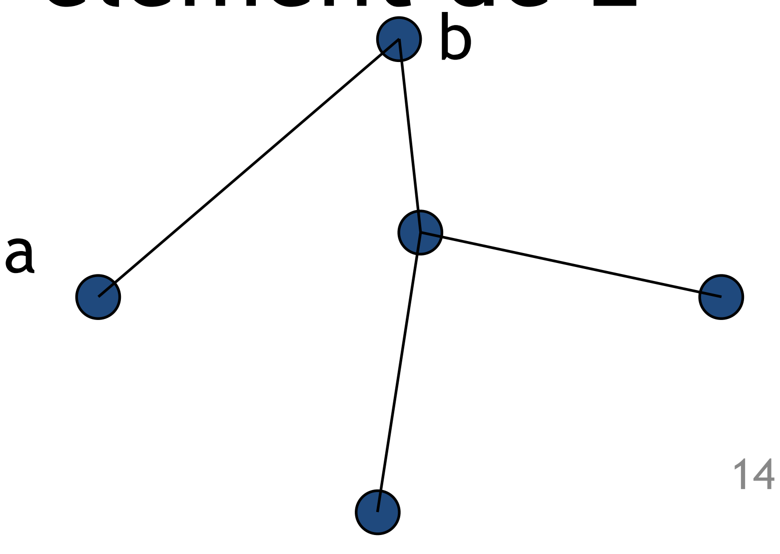
Dominique Barth, UVSQ

Définitions – cas non-orienté

Une vision paresseuse du graphe orienté symétrique…

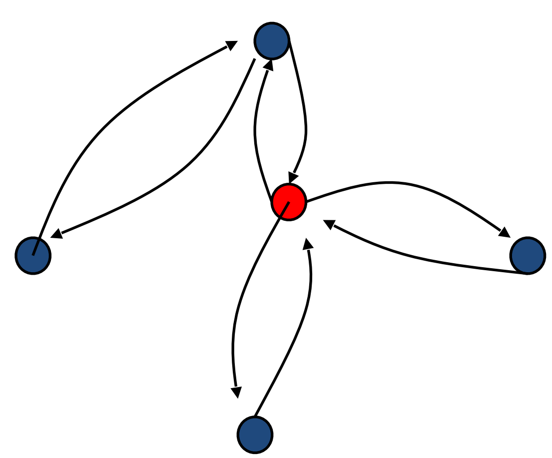
- Un **graphe non orienté** est un couple (V,E) où
  - V est un ensemble d'éléments appelés **sommets** ou **nœuds** (en anglais : *vertex*, *vertices*)
  - E est une partie de l'ensemble des paires (non ordonnées) de sommets. Chaque élément de E est une **arête** (*edge* en anglais).

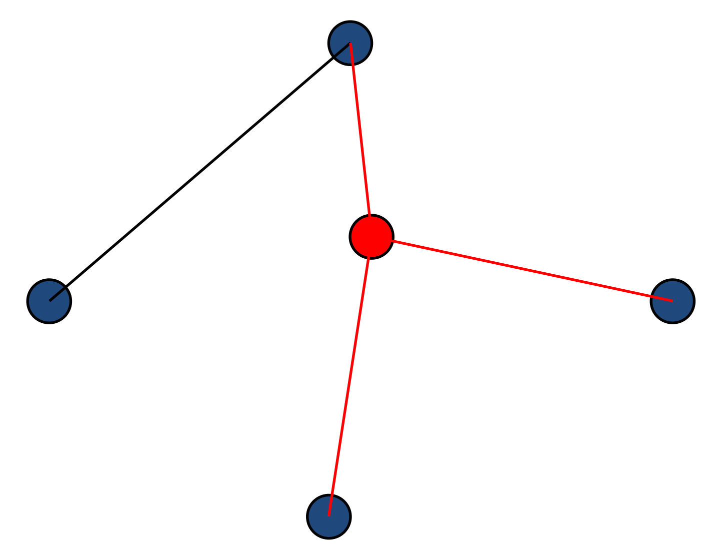
[a,b]=[b,a]



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.

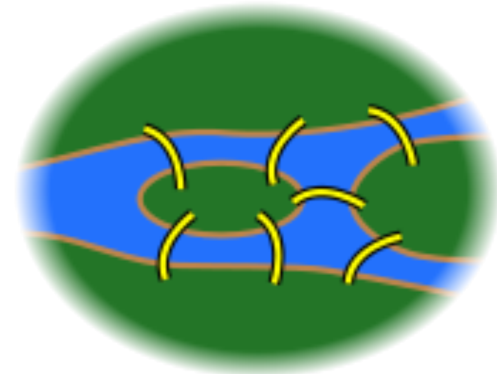


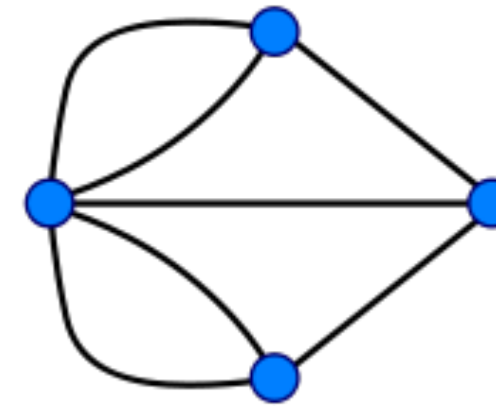


Graphe non-orienté

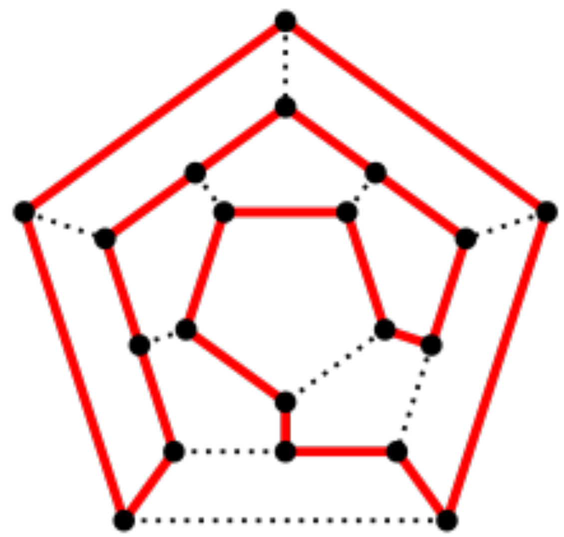
Une veille science en informatique…

L.P. Euler (1707 – 1783)

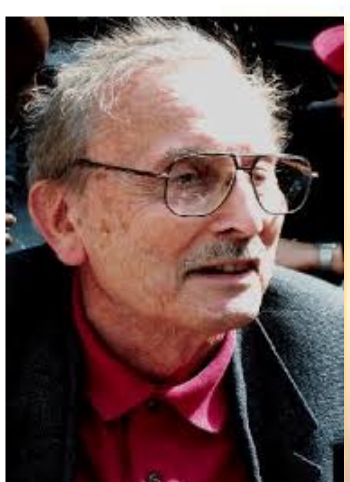


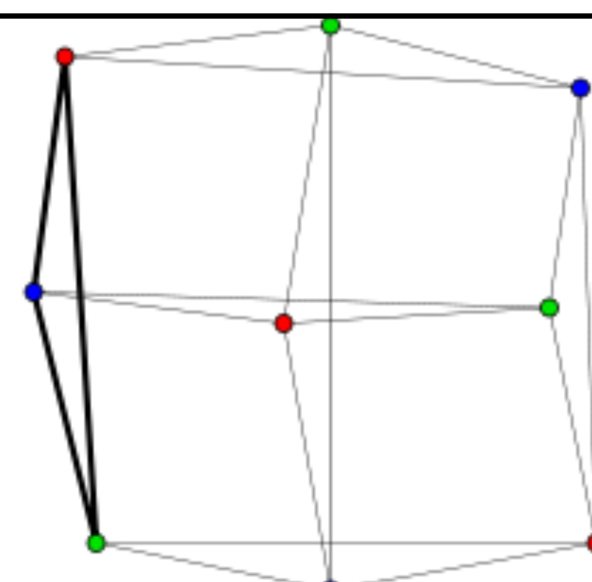


W.R. Hamilton (1805 – 1865)



Claude Berge (1926 – 2002)

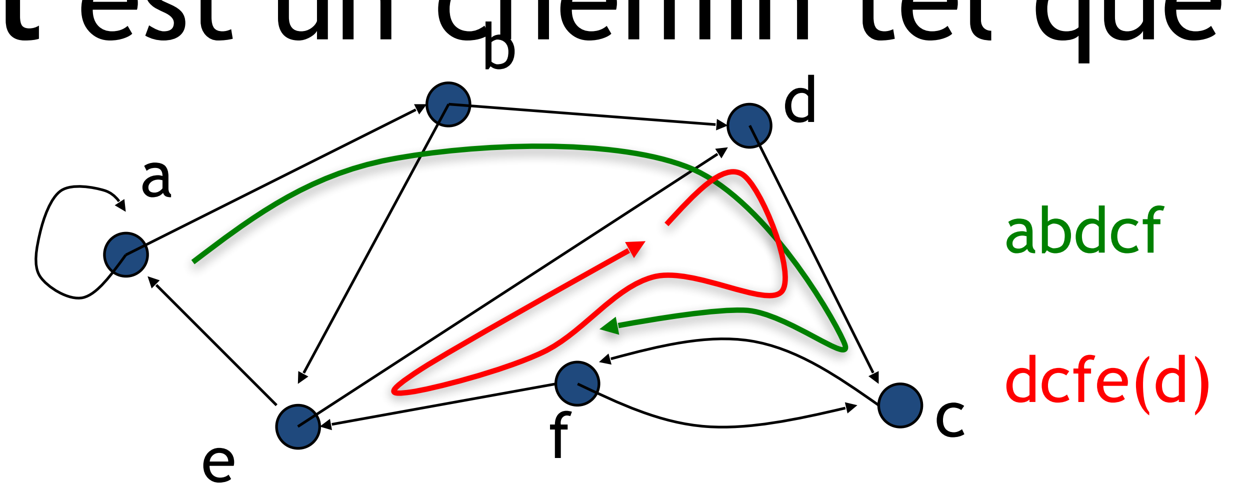




Chaines, chemins, cycles et circuits

Définitions – cas orienté

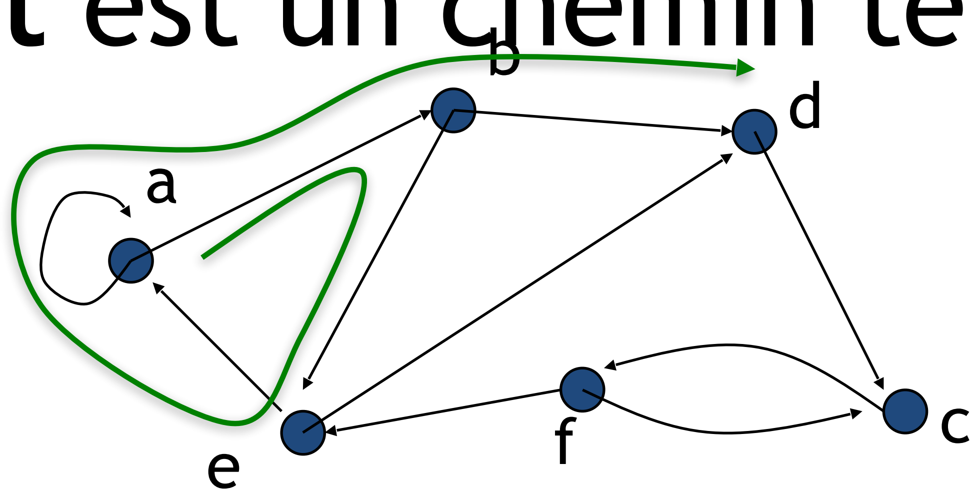
- Un **chemin** est une suite de sommets v_0v_1… v_n telle que chaque couple de sommets successifs est un arc. La **longueur** du chemin est n.
- Un **circuit** est un chemin tel que v_0=v_n.



Dominique Barth, UVSQ

Définitions – cas orienté

- Un **chemin** est une suite de sommets v_0v_1… 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.

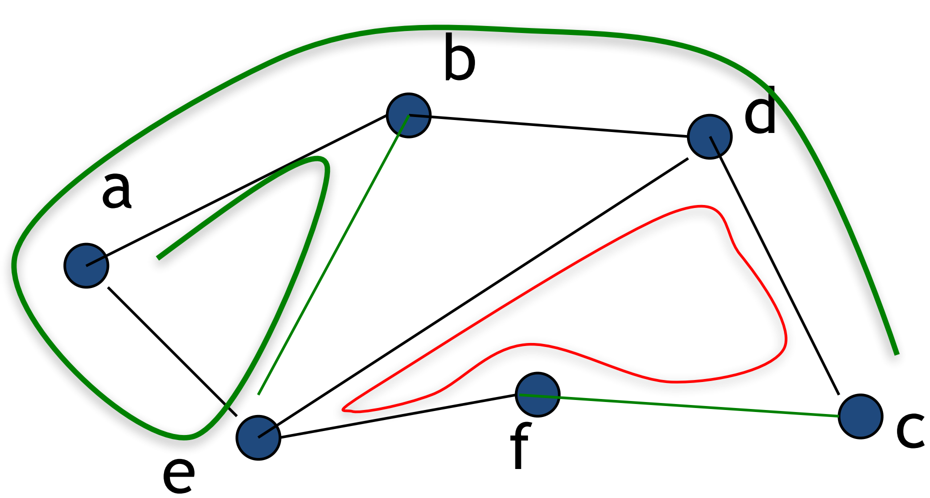


abeabd *(non élémentaire)*

Dominique Barth, UVSQ

Définitions – cas non-orienté

- Une **chaîne** est une suite de sommets v_0v_1… 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.



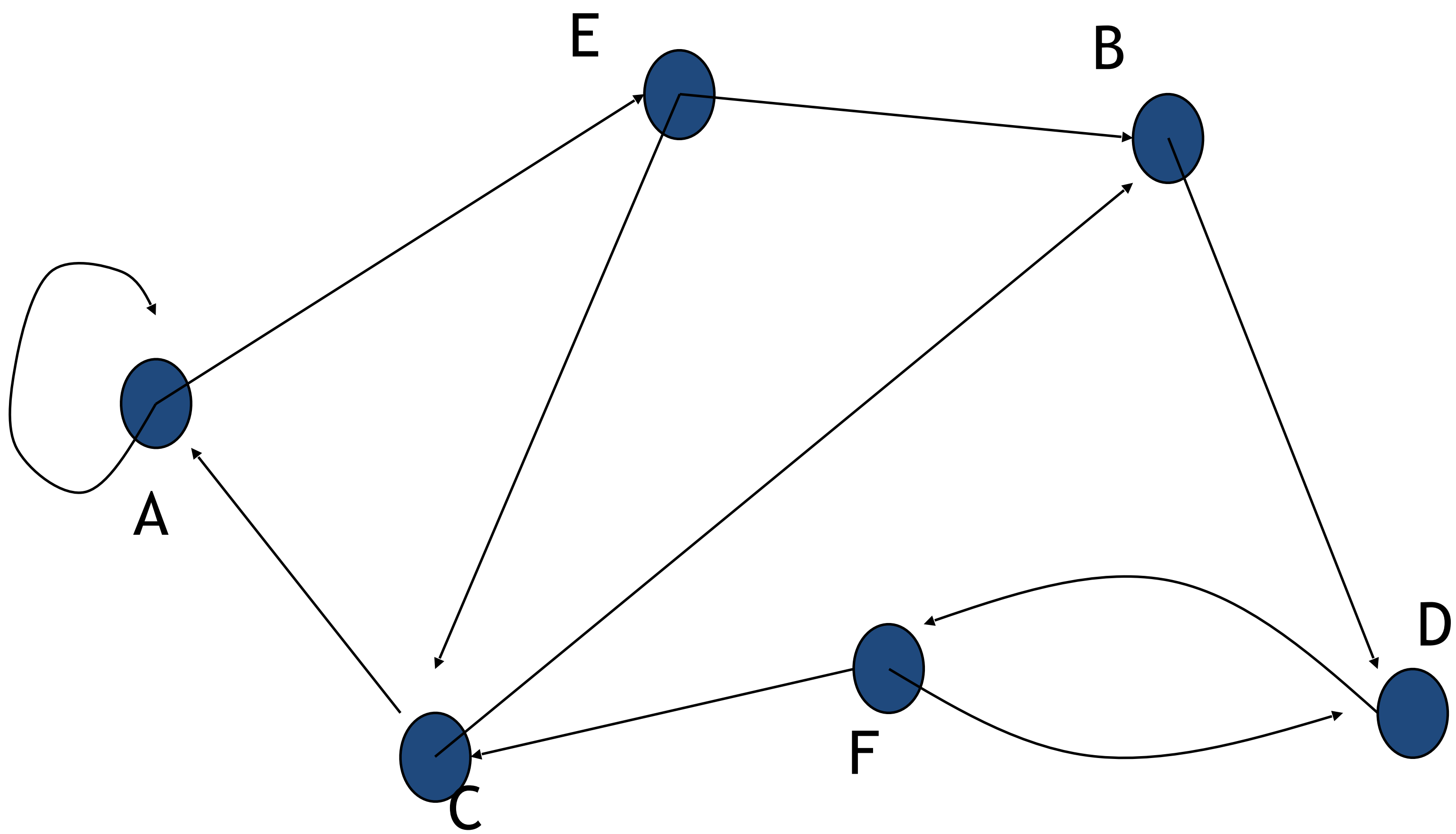
abeabdc

edcf(e)

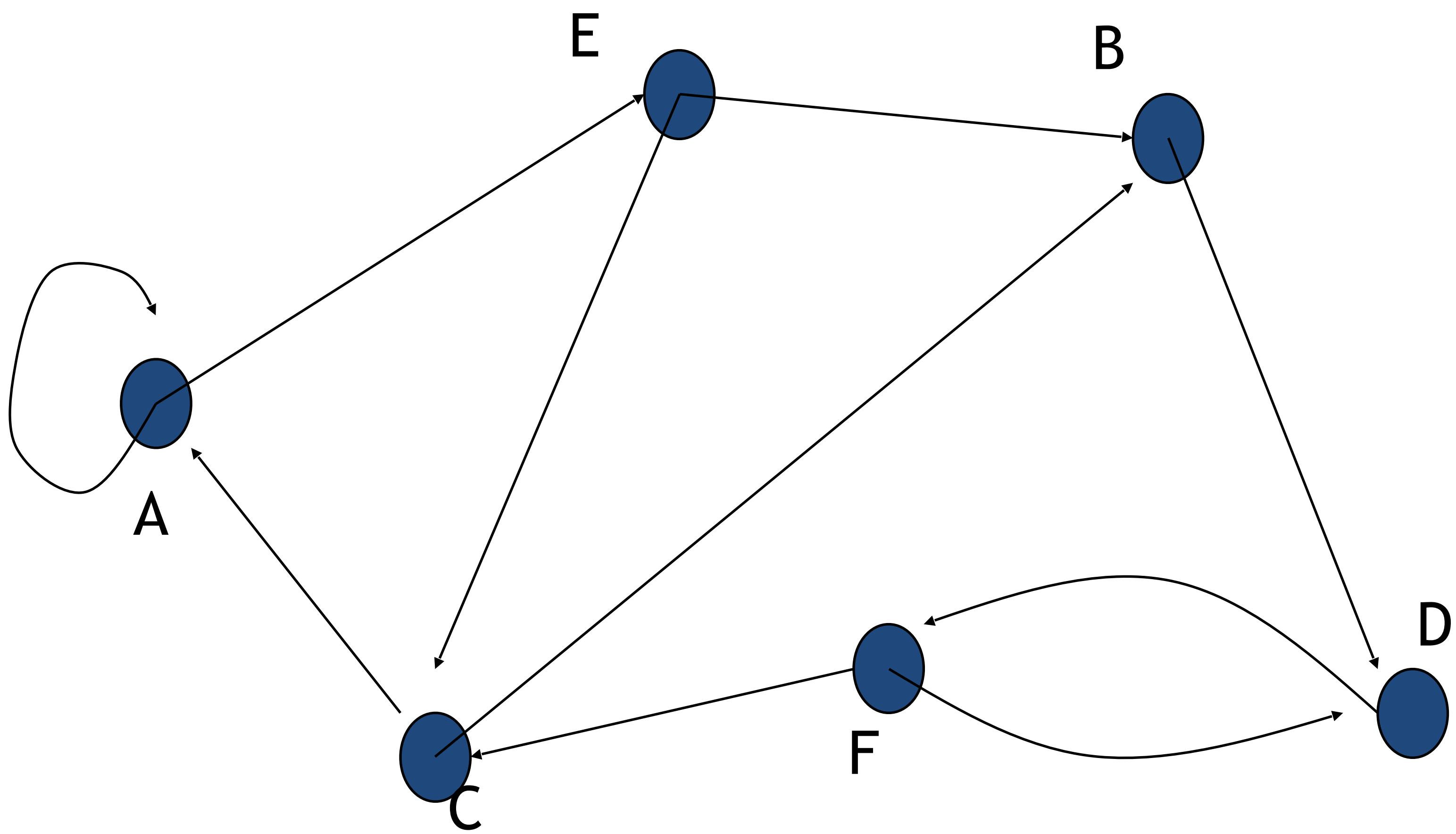
Pas si simple….

Il y a des chaine et des cycles dans les graphes orientés…

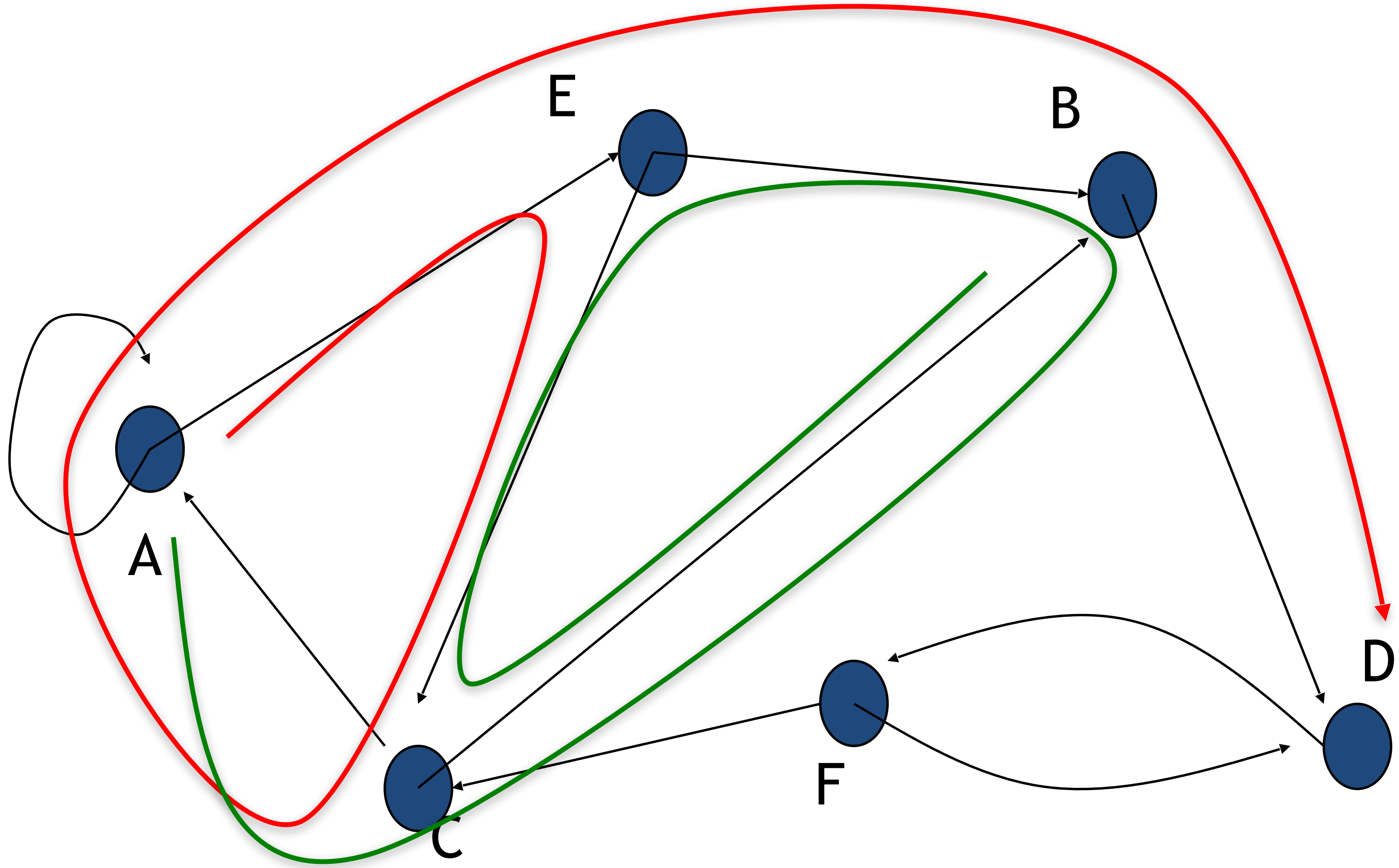
…et il y a des circuits et des chemins dans les graphes non-orientés (graphes orientés symétriques)



AECAEBD ?
ACBECB ?
ABCBDEA ?

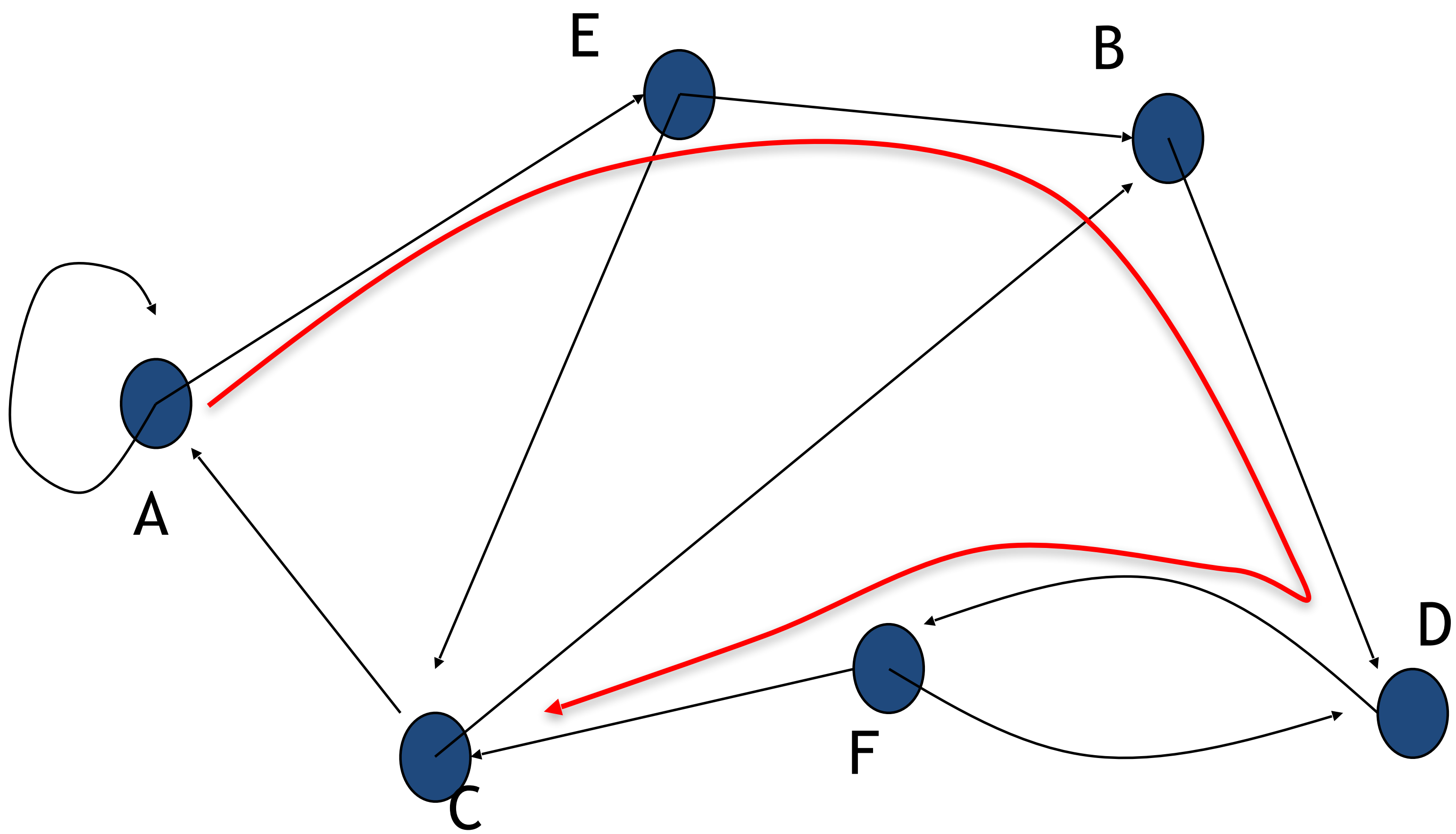


AECAEBD ?
ACBECB ?
ABCBDEA ?

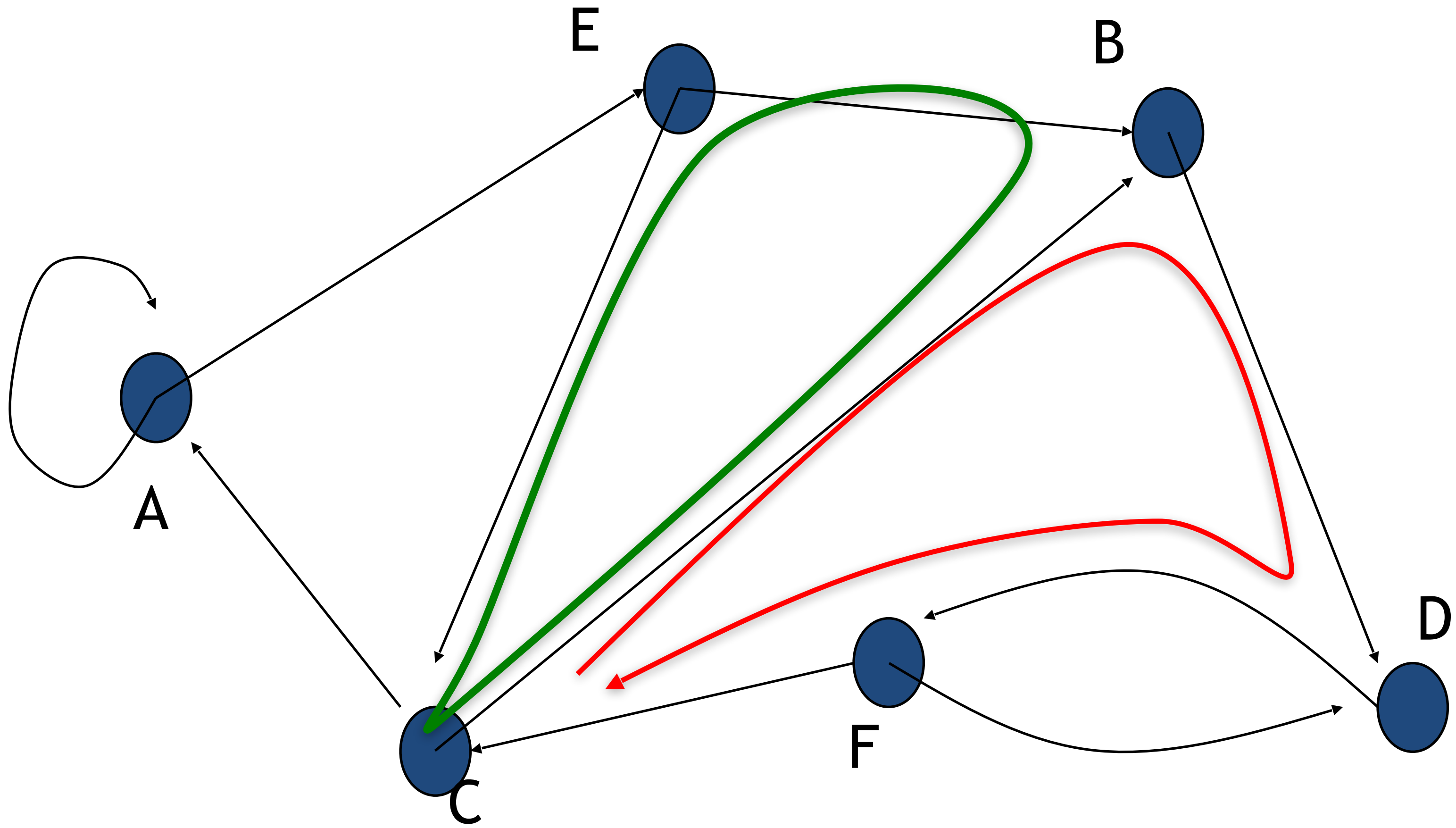


Chaine : ACBECB (=BCEBCA) => les deux orientations d'une même chaine.

Chemin : AECAEBD (*cas particulier de chaine*)

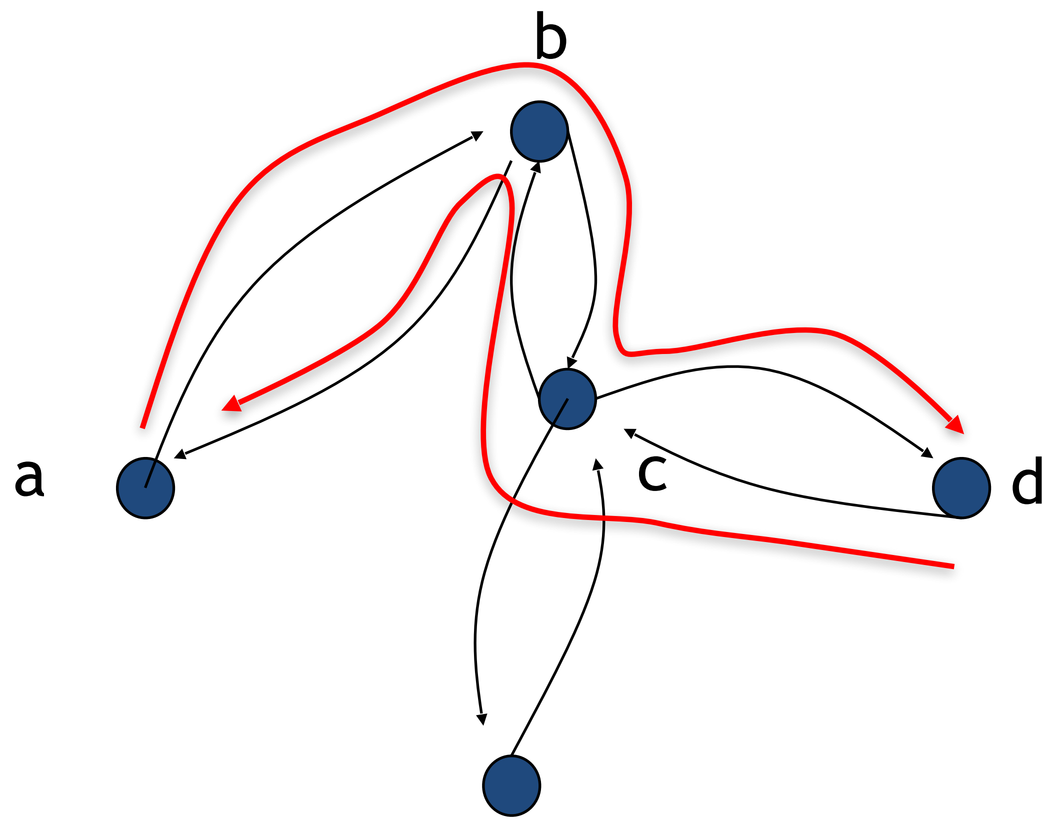


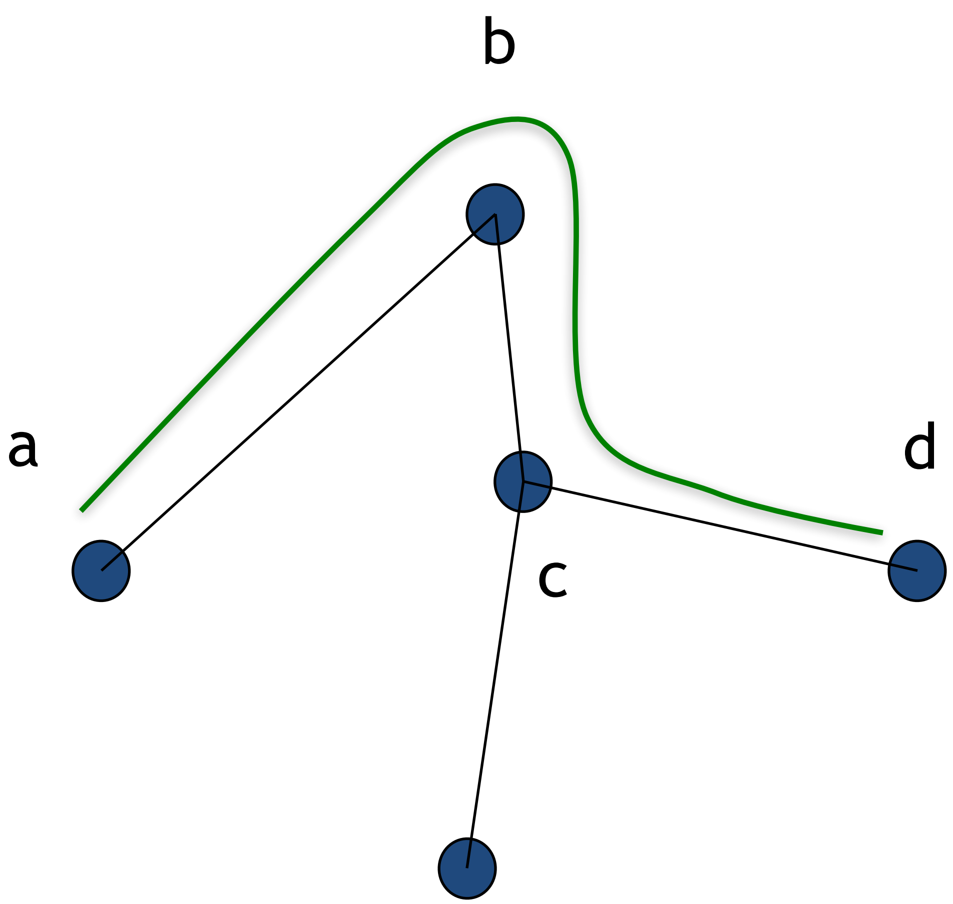
Chemin élémentaire : AEBDFC



Cycle : EBC(E)

Circuit : CBDF(C)

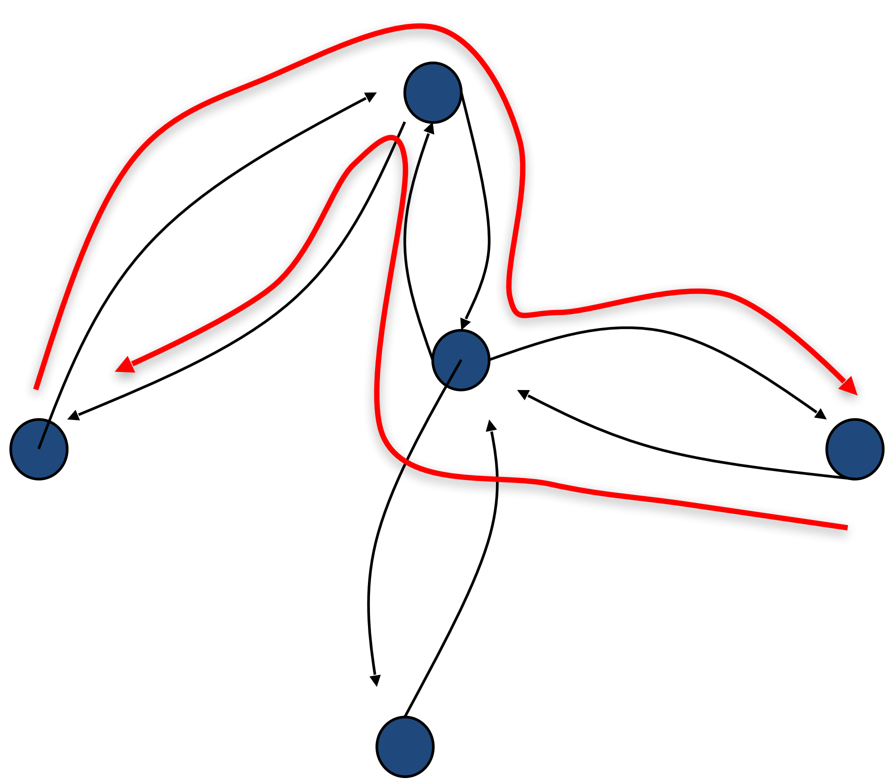


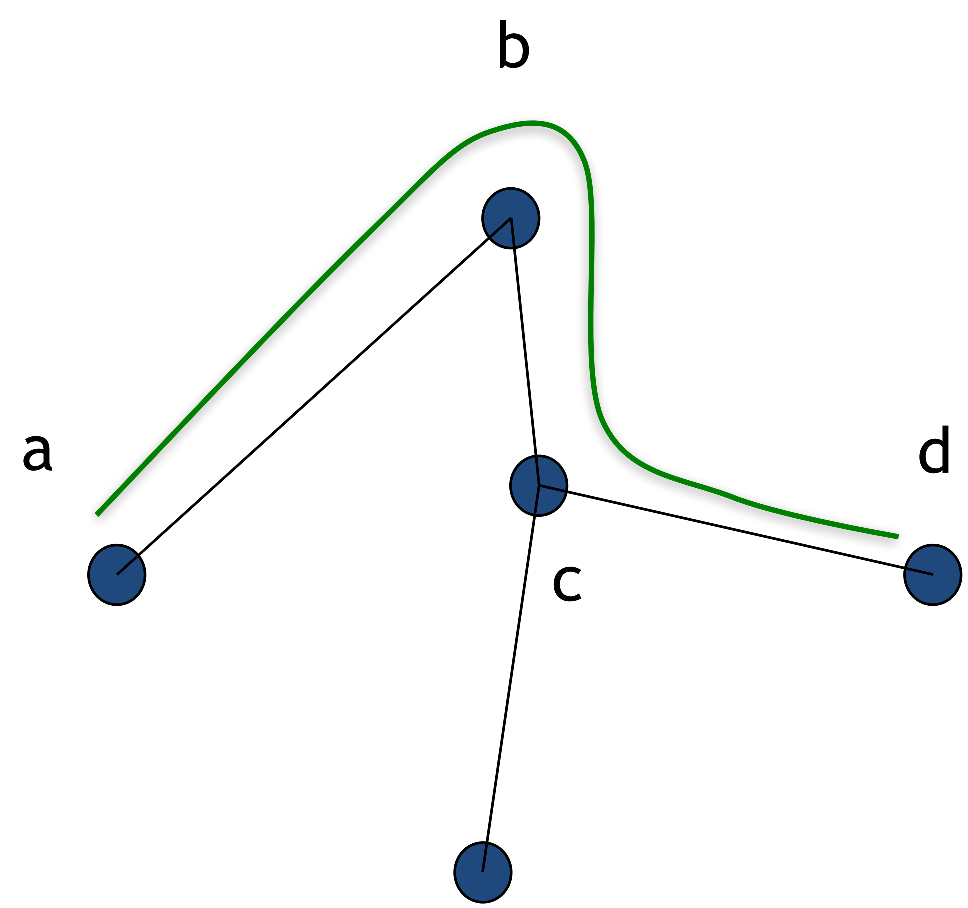


Deux chemins symétriques ……………….. Une chaine

Chemin dans un graphe non orienté = orientation d'une chaine
{abcd, dcba}

Bref,
- *Il y a des chaines et des cycles dans les graphes orientés*
- *Il y a des chemins et des circuits dans les graphes non-orien*





Dominique Barth, UVSQ

Définitions connexités – cas orienté

Un **chemin** est une suite de sommets $v_0v_1\ldots v_n$ telle que chaque couple de sommets successifs est un arc. La **longueur** du chemin est n (*nb. Un chemin est un cas particulier de chaines*)

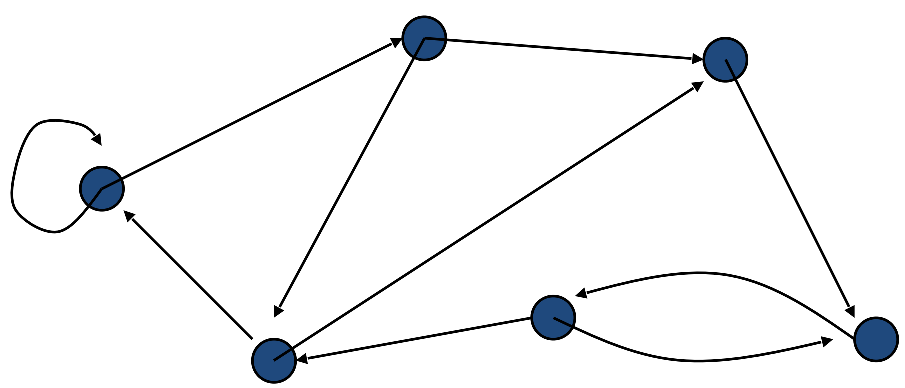
Faible

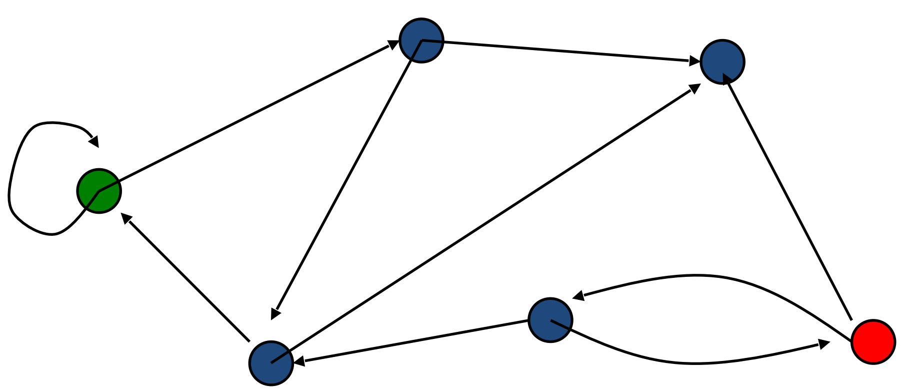
- Un graphe orienté G est **connexe** si et seulement si pour tout couple de sommets il existe une **chaîne** entre ces deux sommets .

- Un graphe orienté est **fortement connexe** si et seulement si pour tout couple de sommets il existe un **chemin** entre ces deux sommets.

Fort

Définitions connexités – cas orienté



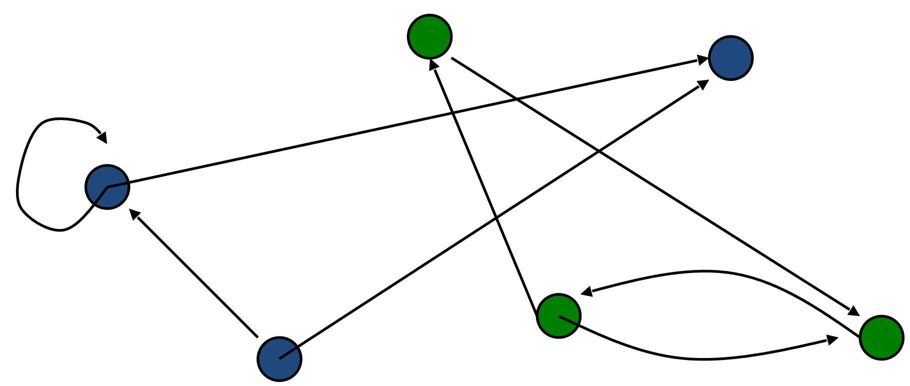


- Un graphe orienté G est **connexe** si et seulement si pour tout couple de sommets il existe une **chaîne** entre ces deux sommets dans le graphe non orienté obtenu en supprimant l'orientation des arcs de G.

- Un graphe orienté est **fortement connexe** si et seulement si pour tout couple de sommets il existe un chemin entre ces deux sommets.

Dominique Barth, UVSQ

Définitions connexités – cas orienté



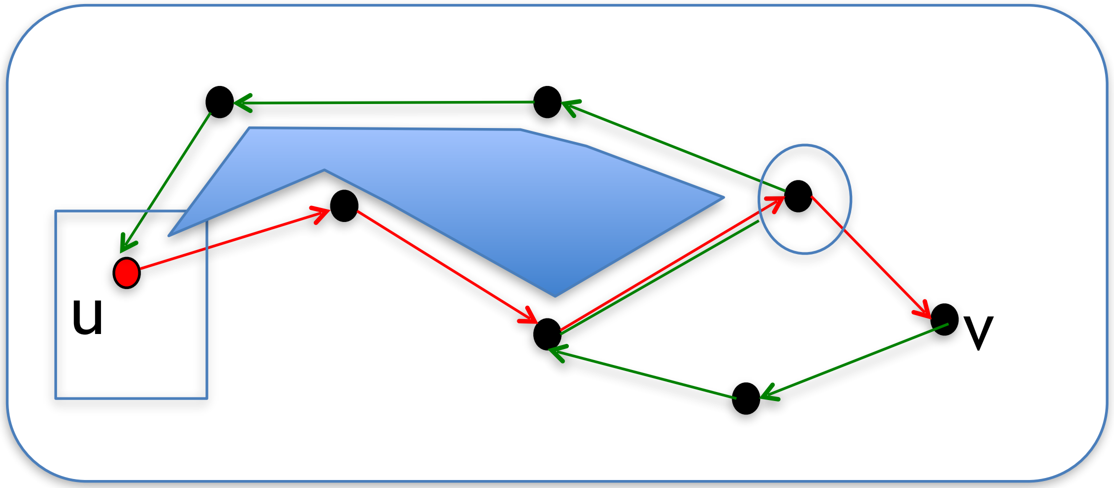
- Un graphe orienté G est **connexe** si et seulement si pour tout couple de sommets il existe une **chaîne** entre ces deux sommets dans le graphe non orienté obtenu en supprimant l'orientation des arcs de G.

- Un graphe orienté est **fortement connexe** si et seulement si pour tout couple de sommets il existe un chemin entre ces deux sommets.

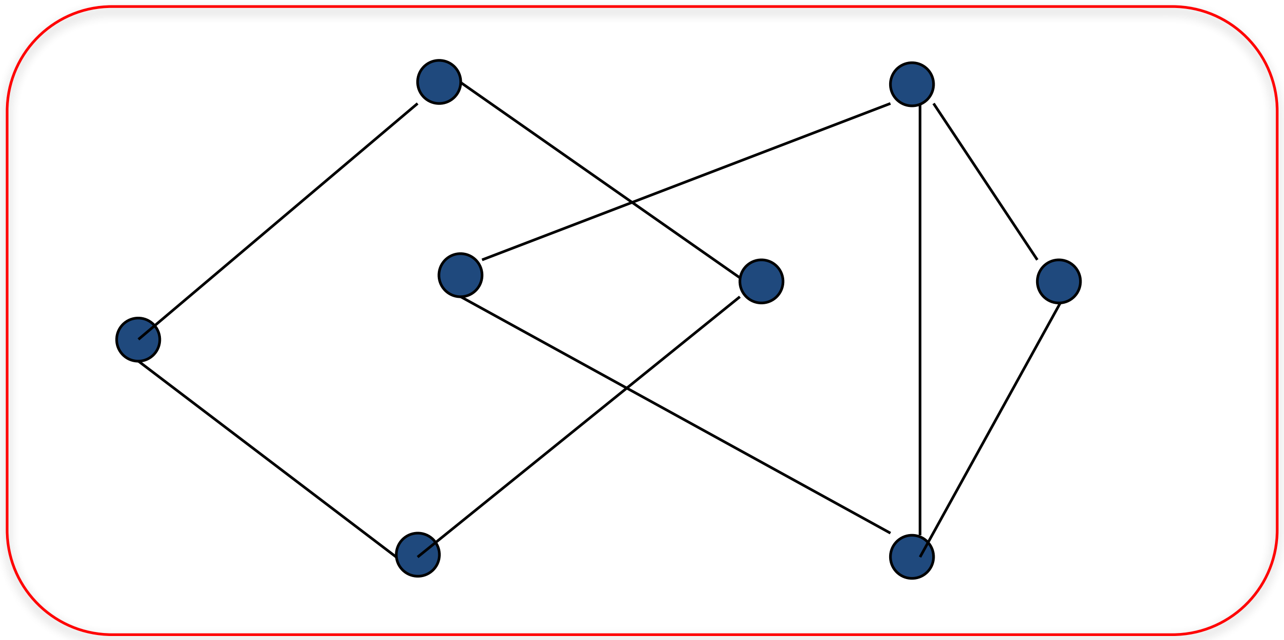
Définitions connexités – cas orienté

Définition : Un graphe orienté est **fortement connexe** si et seulement si pour tout couple de sommets il existe un chemin entre ces deux sommets.

Lemme : Si G fortement connexe alors tout sommet appartient à au moins un circuit



Définitions connexité – cas non orienté



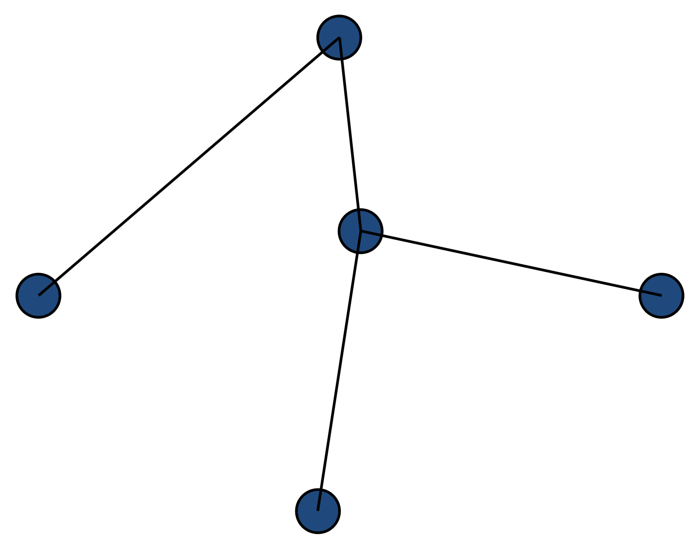
- Un graphe orienté G est **connexe** si et seulement si pour tout couple de sommets il existe une **chaîne** entre ces deux sommets dans le graphe non orienté obtenu en supprimant l'orientation des arcs de G.

Arbres et DAGs

Dominique Barth, UVSQ

Définitions – cas non-orienté

Un **arbre** est un graphe non-orienté connexe sans cycle (*élémentaire*).



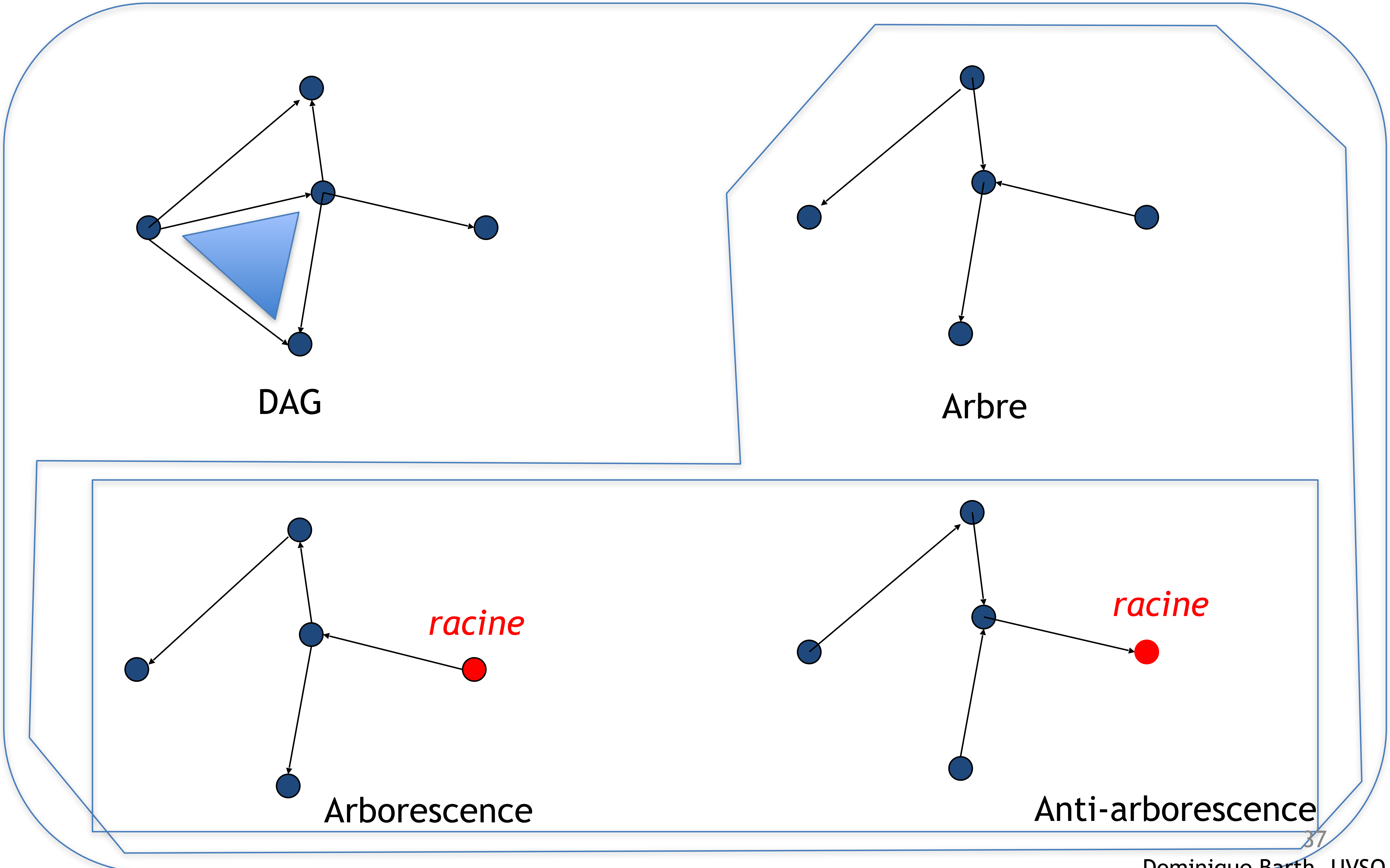
Dominique Barth, UVSQ

Définitions – cas orienté

- Un **DAG** est un graphe orienté connexe sans circuit
- Un **arbre** est un DAG sans cycle
- Une **arborescence** est un arbre avec un seul sommet de degré entrant nul *(la racine)*
- Une anti-**arborescence** est un arbre avec un seul sommet de degré sortant nul *(la racine)*

Dominique Barth, UVSQ

Définitions – cas orienté



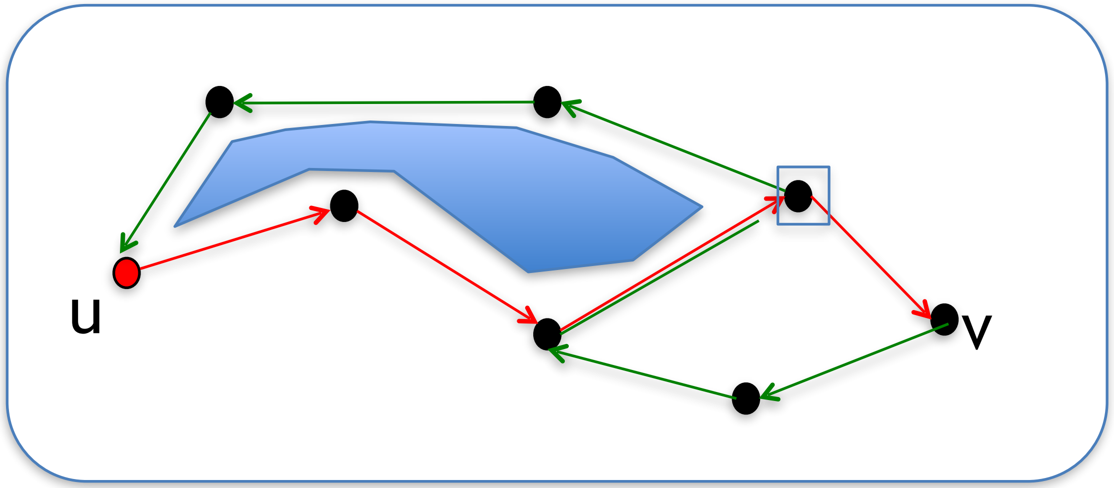
Définitions – cas orienté

- Un **DAG** est un graphe orienté connexe sans circuit
- Un **arbre** est un DAG sans cycle
- Une **arborescence** est un arbre avec un seul sommet de degré entrant nul *(la racine)*
- Une anti-**arborescence** est un arbre avec un seul sommet de degré sortant nul *(la racine)*

Questions :

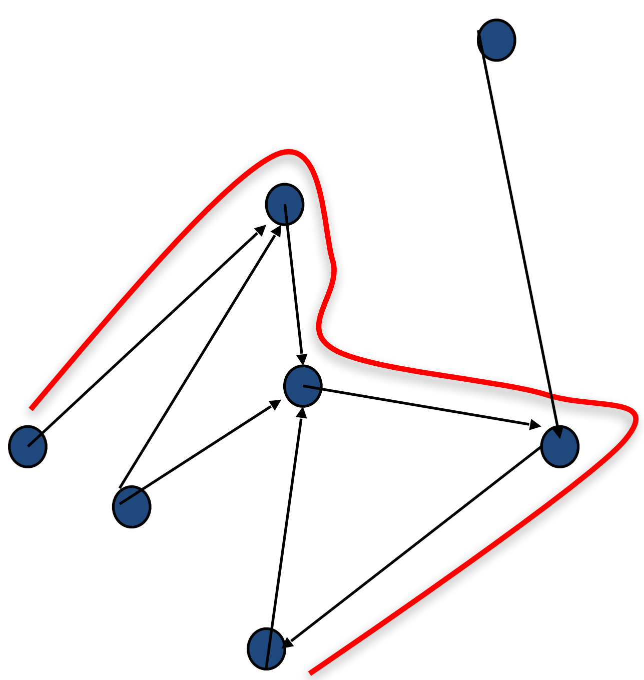
- *un DAG peut-il être fortement connexe?*

Si G fortement connexe, alors pour tout couple de sommets u et v :



- *Comment reconnaître qu'un graphe orienté connexe est un DAG?*

Chemin maximal (au sens de l'inclusion)
(« *qu'on ne peut pas agrandir sur les bords* »

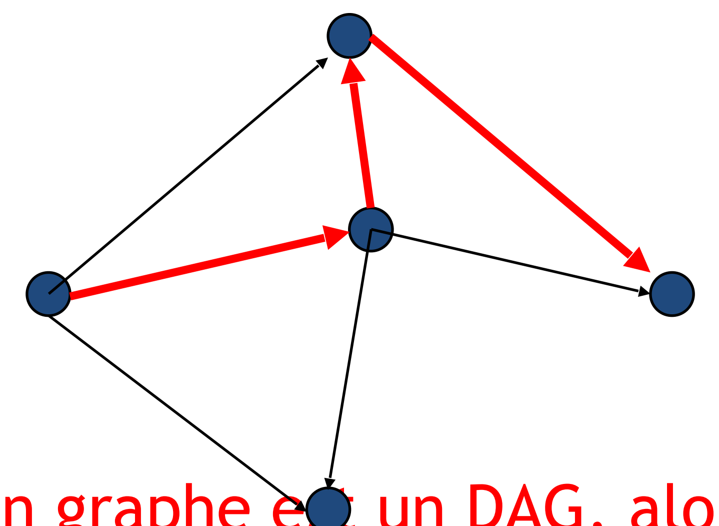


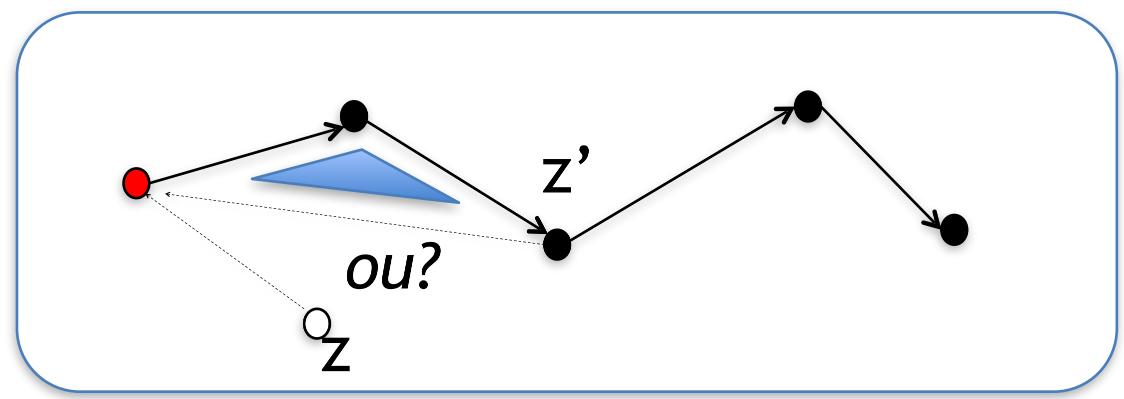
Comment reconnaître qu'un graphe orienté connexe est un DAG?

Deux propriétés

- Si un graphe est un DAG, alors il contient au moins un sommet de degré entrant nul

Idée de preuve : Un chemin maximal dans un graphe (tout graphe en contient un) :





DAG quelconque

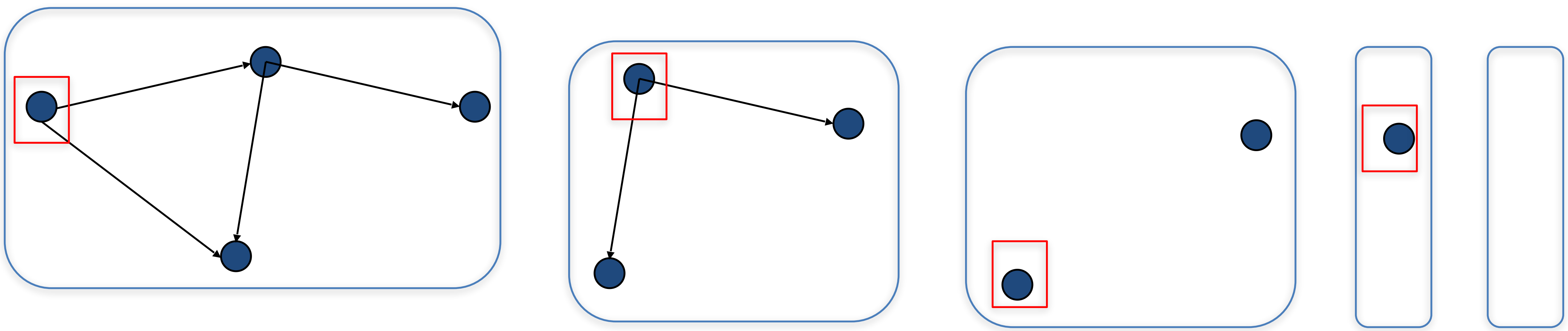
- Si un graphe est un DAG, alors le graphe obtenu en supprimant un de ses sommets est aussi un DAG

On ne peut pas créer un circuit en enlevant un sommet.

Algorithme

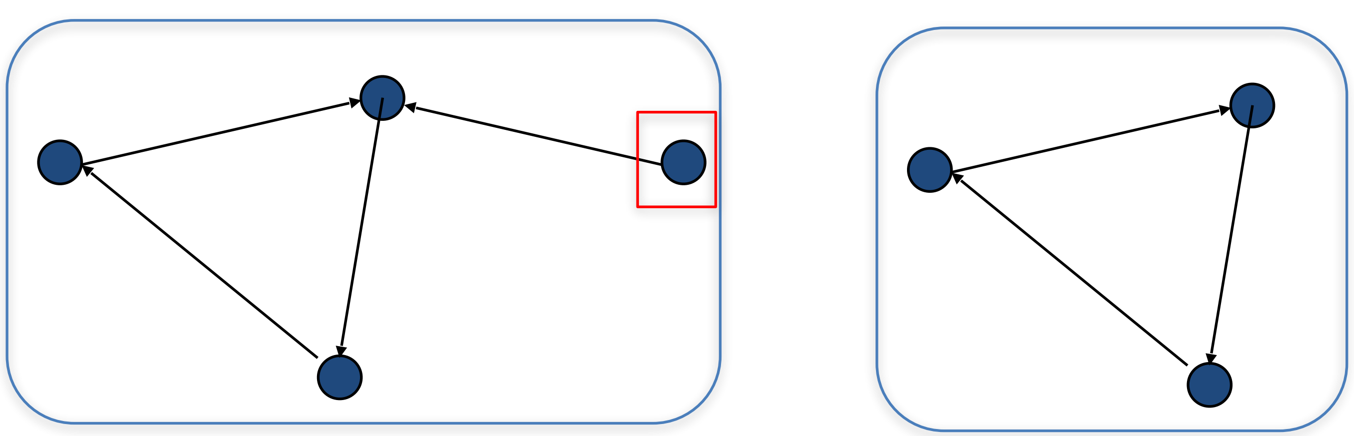
- Si G est un DAG, alors il contient un sommet de degré entrant nul v. On supprime v.
- Si G est un DAG, alors G-{v} est un DAG.
- On itère le processus jusqu'au graphe vide.
- Si à une étape, le graphe ne contient pas de tel sommet v, alors G n'était pas un DAG

Exemple 1 :



Graphe vide, donc DAG

Exemple 2 :



Blocage, donc pas un DAG