

THÉORIE DES GRAPHES

PLAN

- 1 **Généralités et définitions**
- 2 **Problèmes et complexité**
- 3 **Sous-graphes particuliers**
- 4 **Arbres et arbre couvrant**
- 5 **Centre , diamètre**
- 6 **Couplages**

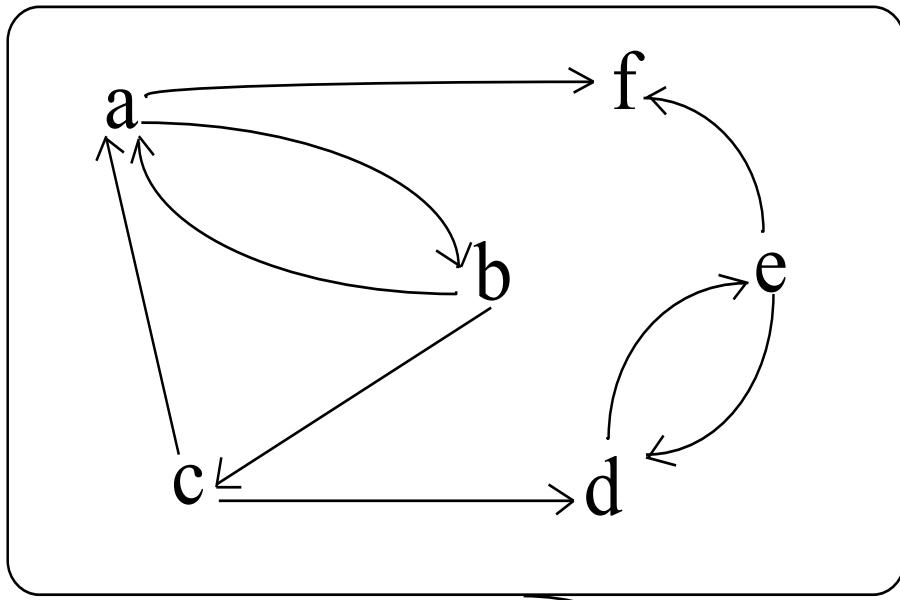
1

GÉNÉRALITÉS

ET

DÉFINITIONS

GRAPHES ORIENTÉS



$$U_1 \subseteq X_1 \times X_1$$

EXAMPLE

$$G_1 = (X_1, U_1)$$

$$X_1 = \{\text{sommets}\} = \{a, b, c, d, e, f\}$$

$$U_1 = \{\text{arcs}\} =$$

$$\{(a, b), (b, a), (b, c), (c, a), (c, d), (a, f), (e, f), (d, e), (e, d)\}$$

$\Gamma : X \rightarrow P(X)$

$x \rightarrow \Gamma(x)$ ou $\Gamma^+(x) = \{\text{successeurs de } x\}$

$\Gamma^-(x) = \{\text{prédécesseurs de } x\}$

EXAMPLE

$$\Gamma_1(b) = \{a, c\}$$

$$\Gamma^-_1(b) = \{a\}$$

$$\Gamma_1(f) = \{\emptyset\}$$

$$\Gamma^-_1(d) = \{c, e\}$$

Chemin : suite d'arcs telle que
l'extrémité terminale d'un arc coïncide avec
l'extrémité initiale de l'arc suivant

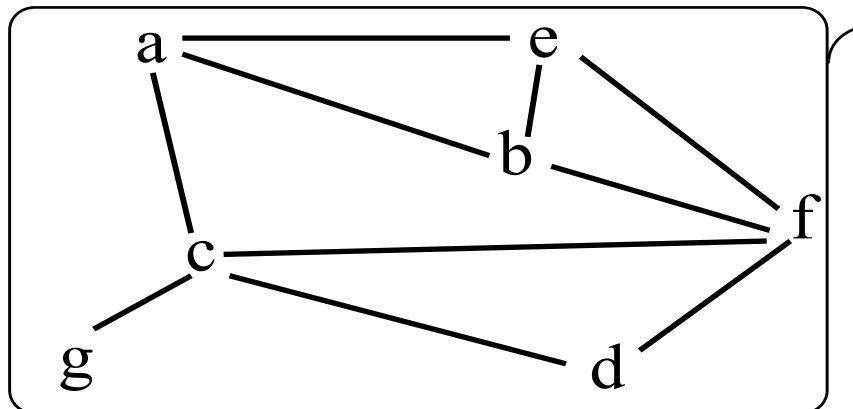
EXAMPLE $((a,b),(b,c),(c,d))$ ou (a,b,c,d)

boucle : arc du type (x,x)



GRAPHES NON ORIENTÉS

$G = (X, A)$ A est un ensemble d'arêtes
arête : arc "sans orientation"



$G3 = (X3, Y3)$

$A3 \subseteq X3 \times X3$

$A3 = \{[a,e], [a,b], [e,f], \dots\}$

Chaîne $[a,e,f,d]$ ou
 $([a,e], [e,f], [f,d])$

Chaîne : suite d'arêtes telle que toute arête a une extrémité commune avec l'arête précédente (sauf la première) et l'autre avec l'arête suivante (sauf la dernière)

Soit $G=(X,A)$

y est **adjacent** ou **voisin** de x si $[x,y] \in A$

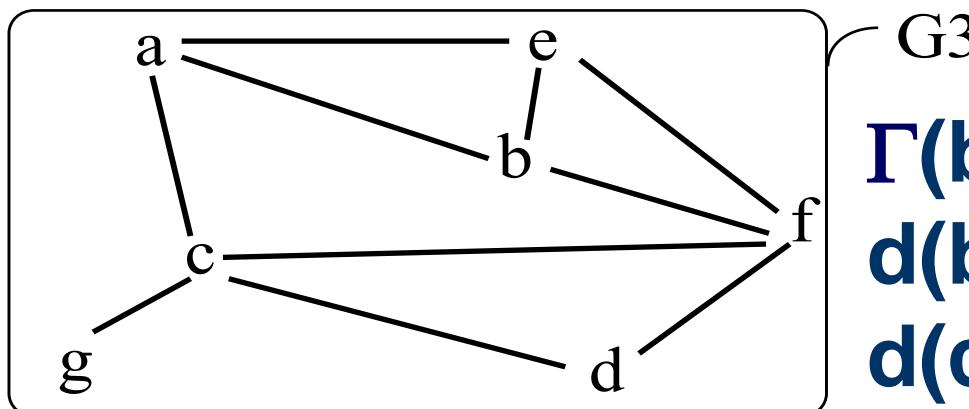
$\Gamma(x)=\{\text{voisins de } x\}$

degré

$d(x) = \text{nombre de voisins de } x \in X$
 $\text{cardinal}(\Gamma(x))$

EXAMPLE

b est voisin
de a, e et f



G3

$\Gamma(b)=\{a,e,f\}$

$d(b) = 3$

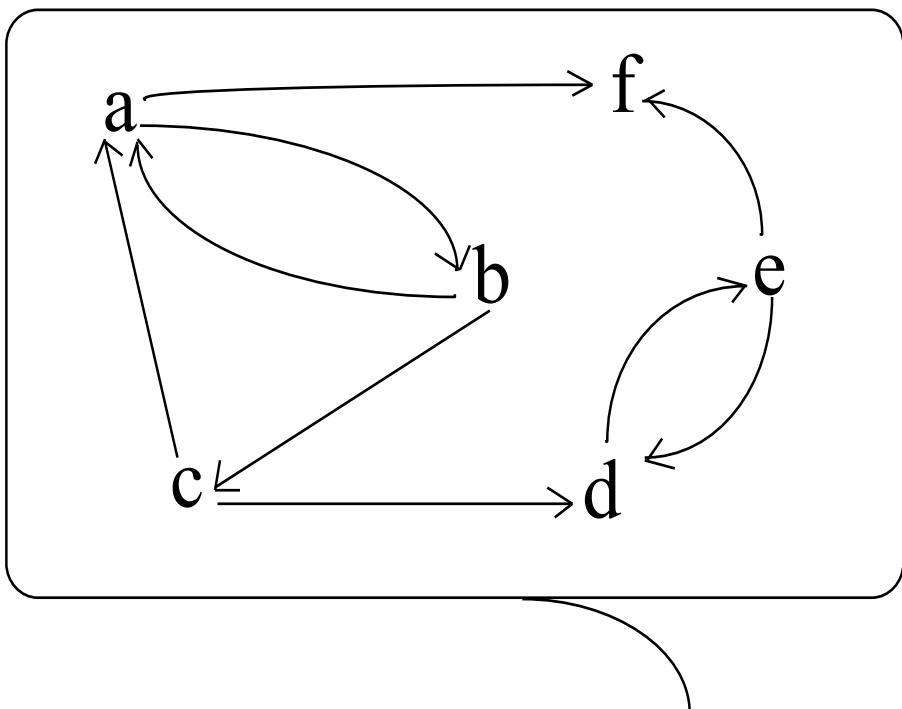
$d(c) = 4$

demi-degré intérieur

$d^-(x) = \text{nombre de prédécesseurs de } x = \text{Card}(\Gamma^-(x))$

demi-degré extérieur

$d^+(x) = \text{nombre de successeurs de } x = \text{Card}(\Gamma^+(x))$



EXAMPLE

$$\begin{array}{ll} d^+(b)=2 & d^-(b)=1 \\ d(b)=d^+(b)+d^-(b)=3 & \\ d^-(f)=2 & d^+(f)=0 \end{array}$$

UTILISATIONS DES GRAPHES

- **Modélisation, représentation de problèmes**

Exemple: plan de ville, arbre généalogique, jeux, états d'un système..

- **Résolution de problèmes**

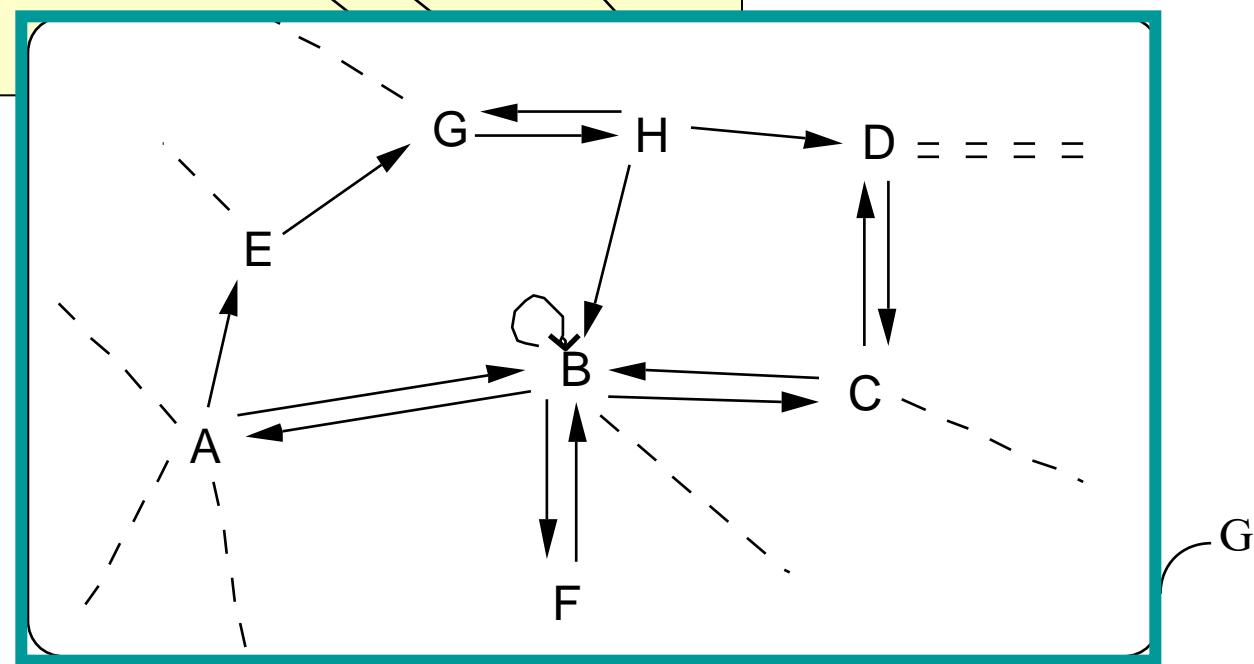
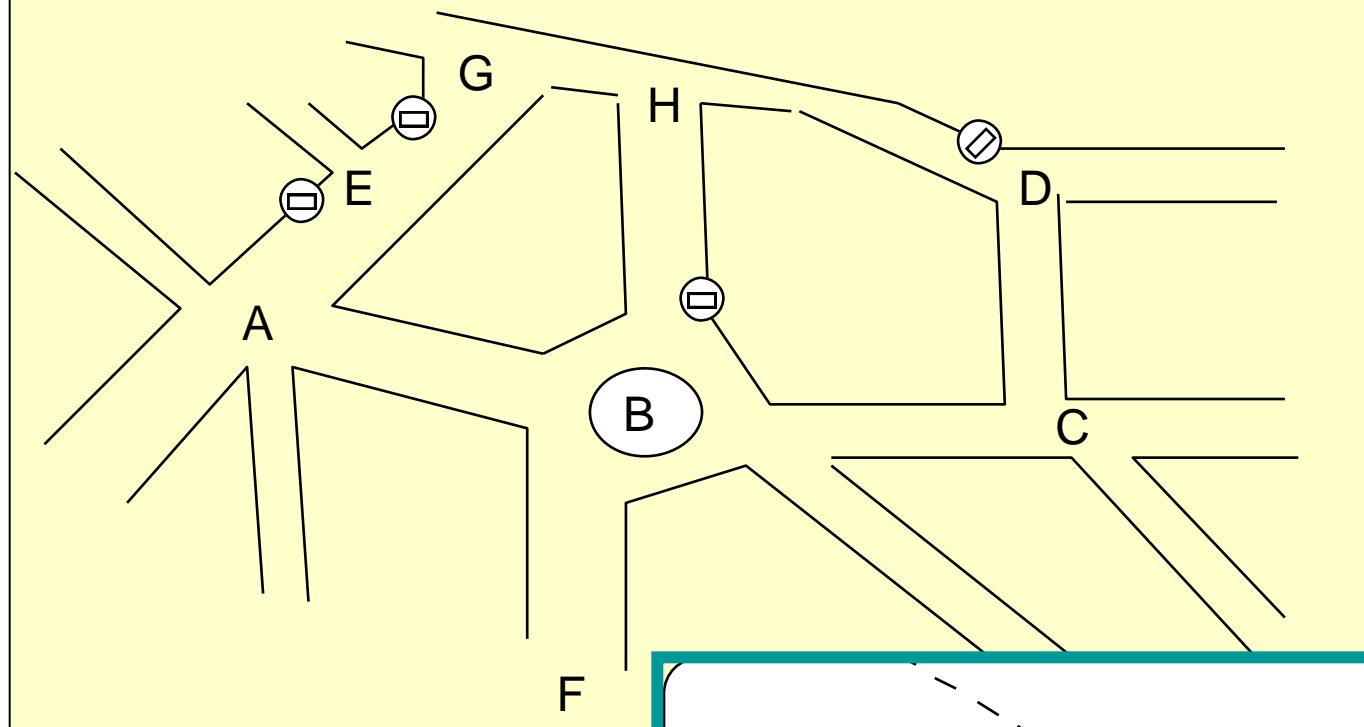
Exemple: plus court chemin, ordonnancement, flots, ...

- **Outils**

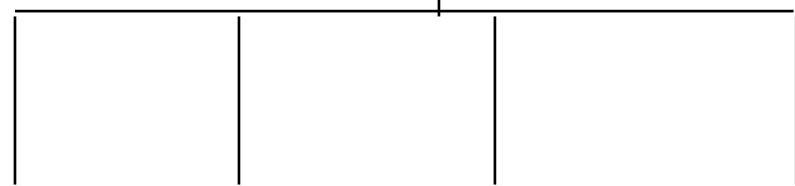
Exemple: structures de données,..

- **...**

○ Sens Interdit



Amélie et
Jules



Sophie Fabien et Y David et Z Elodie et X



Mappymail Rechercher

Messagerie Accueil Radio My Netscape Search Signets

Mappy - Votre itinéraire

mappy

français

Itinéraires Plans Adresses utiles Info Route Mobilité Adresses utiles • Hôtels • Restaurants • Sélection Mappy Tourisme • RATP Ile de France E-mail ***** Mot de passe ? S'inscrire

Itinéraire [Trajet RATP]

départ N° et rue, métro, quartier, gare
Ville, lieu-dit, aéroport Code postal
paris Pays France - DOM

arrivée N° et rue, métro, quartier, gare
Ville, lieu-dit, aéroport Code postal
marseille Pays France - DOM

Voiture: Express Sans péage Piéton **ok**

Vos options

Votre véhicule Routière
Votre carburant Essence 1.25 €/l
Vos indemnités 0 €/Km

Votre devise €
Les distances en Km
Cols d'altitude Autoriser

Vos étapes

1 N° et rue, métro, quartier, gare Ville, lieu-dit, aéroport Code postal Pays France - DOM

2 N° et rue, métro, quartier, gare Ville, lieu-dit, aéroport Code postal Pays

Critère 1: le temps

Critère 2 : le coût

Calcul du chemin de temps (ou coût) minimal



Itinéraires en Europe • RATP Ile de France • Itinéraires Piéton • Calcul de note de frais

Votre itinéraire

De Paris (75, France)

A Marseille (13, France)

Itinéraire Express

Essence 66.38 € (66 l)

Distance 774.40 km

Indemnités 0.00 €

Voies express 766.98 km

Durée (1) 07H30

Véhicule Routière

Péages (2) FRA : 48.80 EUR

(1) Hors pause, conditions normales de météo et trafic

(2) Hors ponts et tunnels payants hors de France

[Modifier les options](#)

08 99 70



La carte



Carte générale

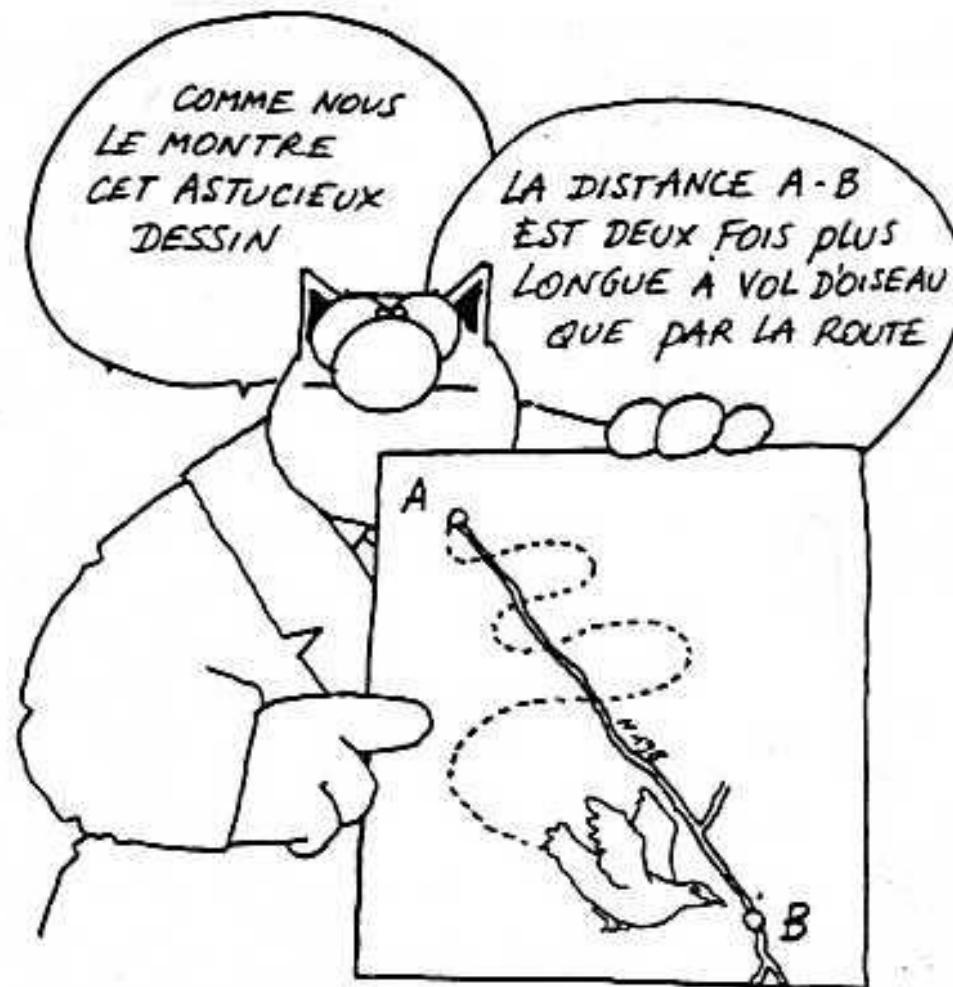
Carte détaillée

[Sauvegarder](#) [Personnaliser](#) [Itinéraire retour](#) [Imprimer](#) [Nouvelle recherche](#)

Cumul	Temps	Feuille de route
0 m	0H00	Paris (75, France)
5 Km	0H09	Sortir de Paris [5.00km] Prendre l'A6a/E05/E15/E50 [2.59km] en suivant le panneau
6 Km	0H12	Radar : 90 Km/h - A6a - Pt km 0,9 - Arcueil
8 Km	0H12	Rejoindre l'A6b/E05/E15/E50 [2.01km]
12 Km	0H14	Rejoindre l'A6a/E15 [3.29km] en suivant le panneau



Optimum
trouvé
facilement
par un
algorithme
de
graphe



Philippe Geluck - Le chat, 2

Domaines d'applications

Olivier Nocent

- **Cartographie**

Réseau routier, réseau internet, ...

- **Économie – Gestion**

Planning de livraisons, gestion de flots, ordonnancement...

- **Chimie – Biologie**

Modélisation de molécules, ADN, ...

- **Sciences Sociales**

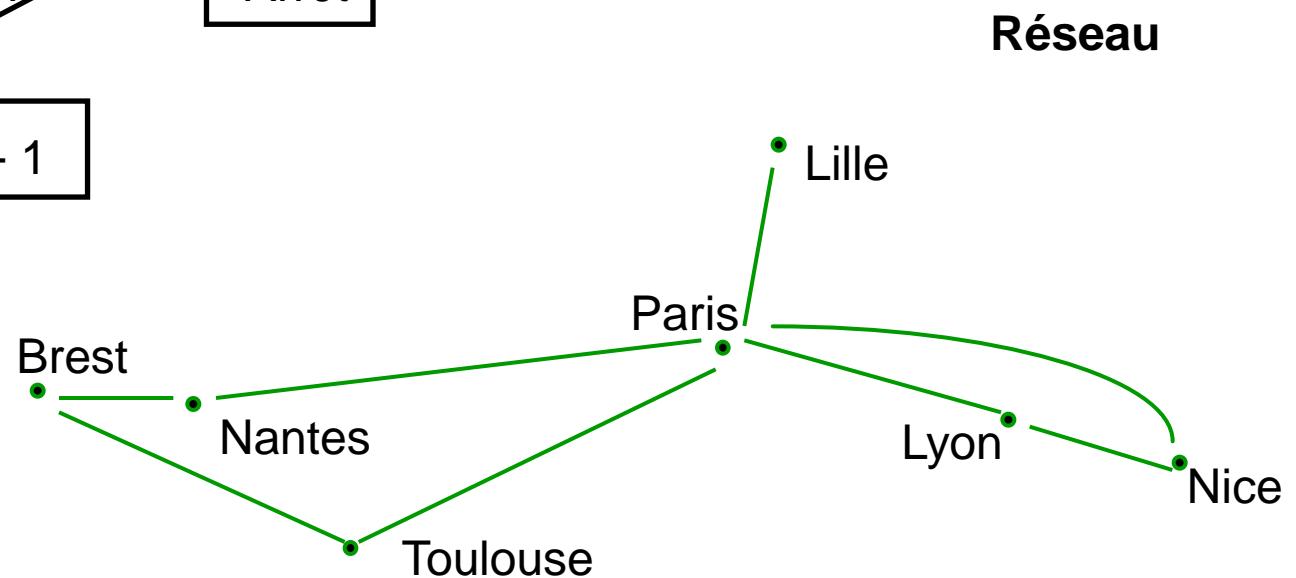
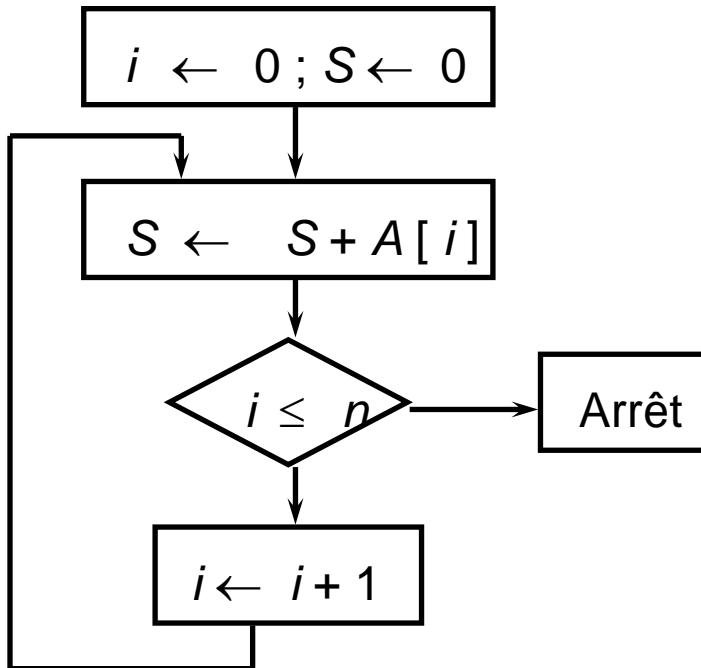
Généalogie, phénomènes de masse, conflits, ...

- **Linguistique**

Grammaire, Compilation, ...

-

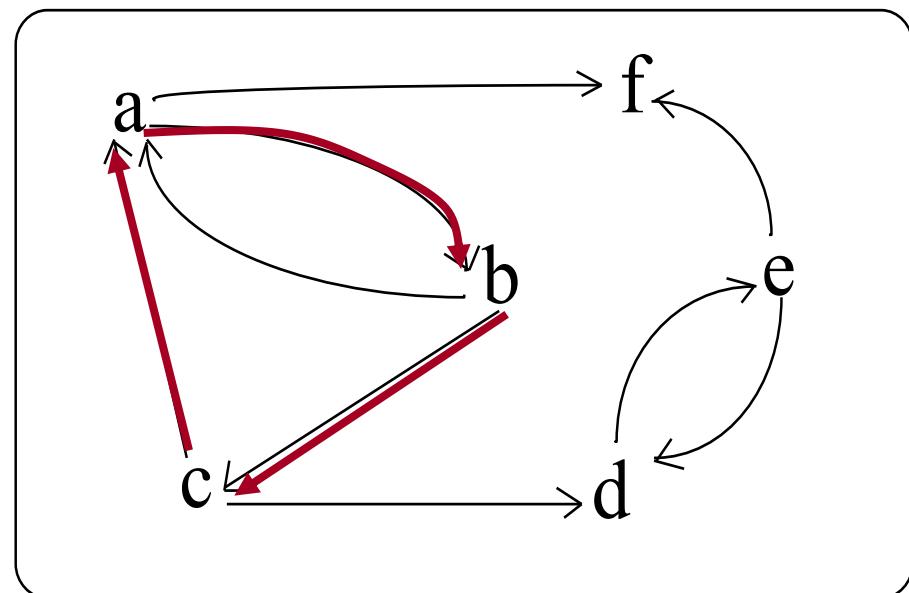
Flot de contrôle d'un programme



QUELQUES DEFINITIONS

circuit : chemin dont le premier sommet coïncide avec le dernier

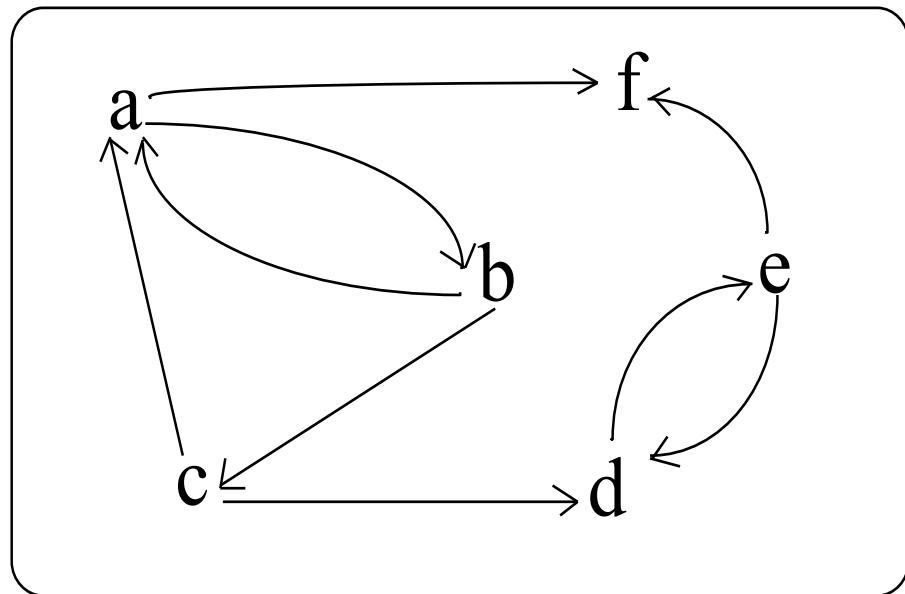
EXAMPLE
 (a,b,c,a)
circuit de G1



G1

chemin hamiltonien :
chemin qui passe une
fois et une seule par
chaque sommet

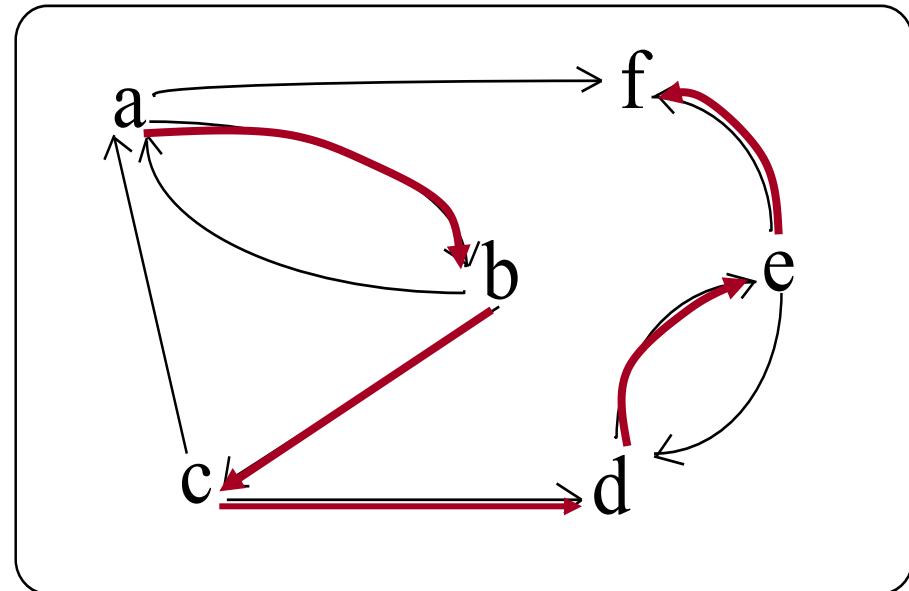
*Existe-t-il un chemin
hamiltonien ?*



G_1

chemin hamiltonien :
chemin qui passe une
fois et une seule par
chaque sommet

OUI (suite)
(a,b,c,d,e,f)



G1

Exemple d'application

ADN constitué de 4 nucléotides A,C,G,T

ADN constitué de 2 brins identiques

On casse les brins et ensuite on élimine les morceaux contenus dans d'autre morceaux . Par exemple AC contenu dans TACGA

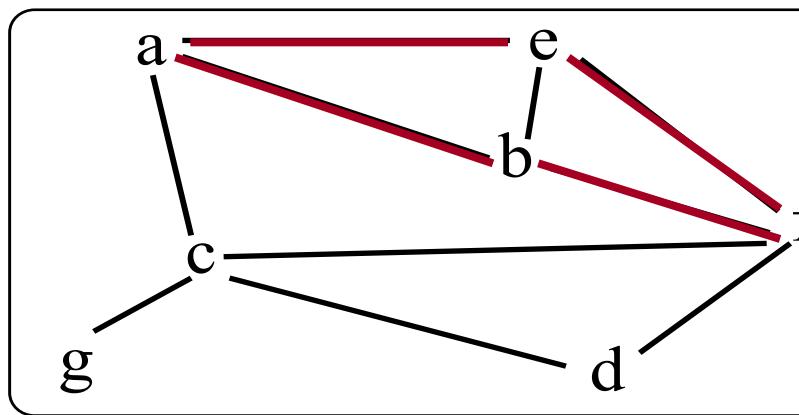
On se retrouve avec un ensemble de morceaux dits libres

Exemple:

TACGA, ACCC, CTAAAG, GACA

Comment reconstituer un brin d'ADN ?

cycle : chaîne dont les deux extrémités coïncident (et qui ne passe pas 2 fois par la même arête)



G3

EXAMPLE

cycle de G3:
[aefba]

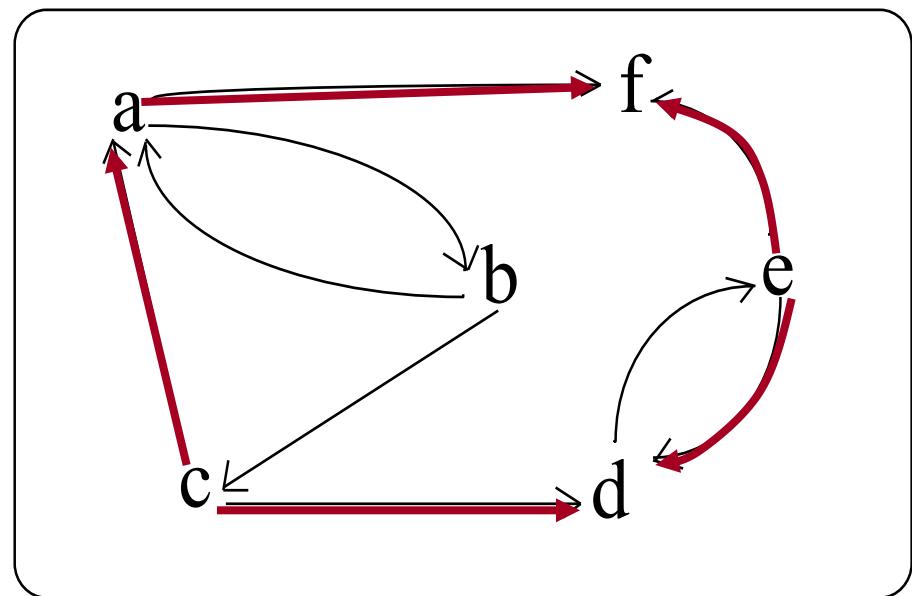
Récapitulatif du vocabulaire

Correspondances orienté - non orienté

Graphe non orienté	Graphe orienté
sommet, noeud	sommet, noeud
arête	arc
degré	$\frac{1}{2}$ -degré intérieur, extérieur
voisin, adjacent	successeur, prédécesseur
chaine	chemin
cycle	circuit
connexité	forte-connexité

chaîne et cycle sont définis aussi dans un graphe orienté (on ne tient plus compte de l'orientation)

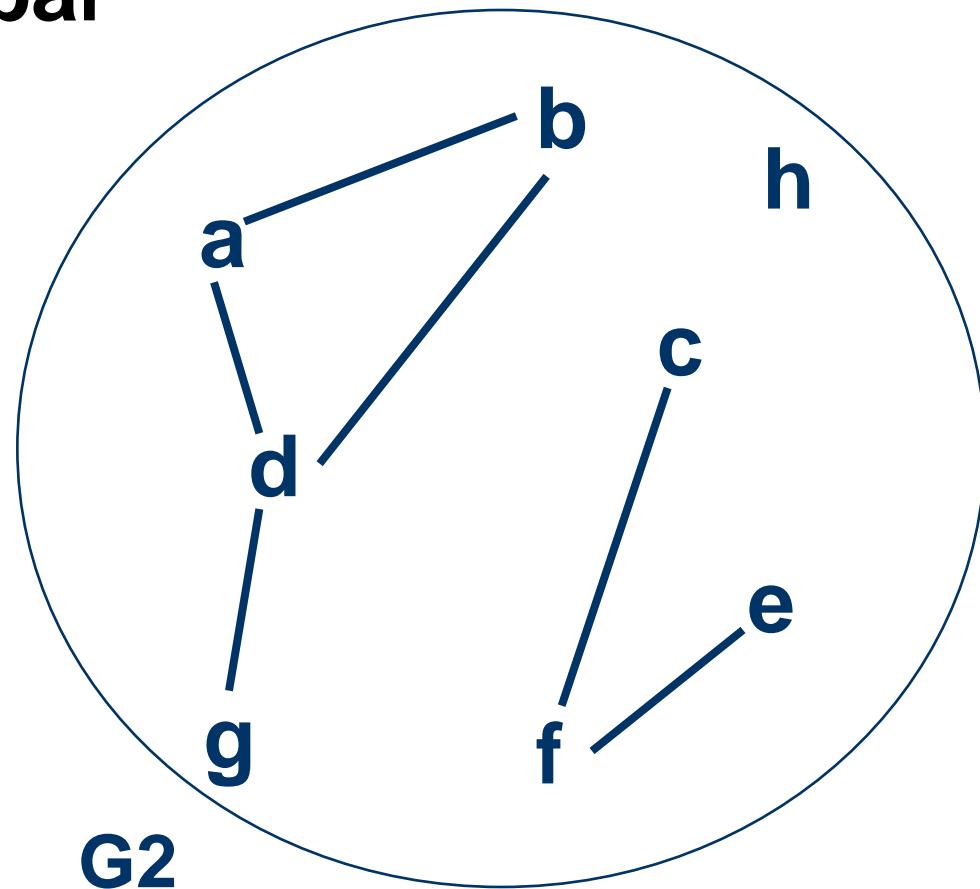
EXAMPLE
un cycle
de G1:
(afedca)



G1

connexité : un graphe est connexe si toute paire de sommet est reliée par une chaîne

EXAMPLE
G1 et G3 connexes
G2 non connexe

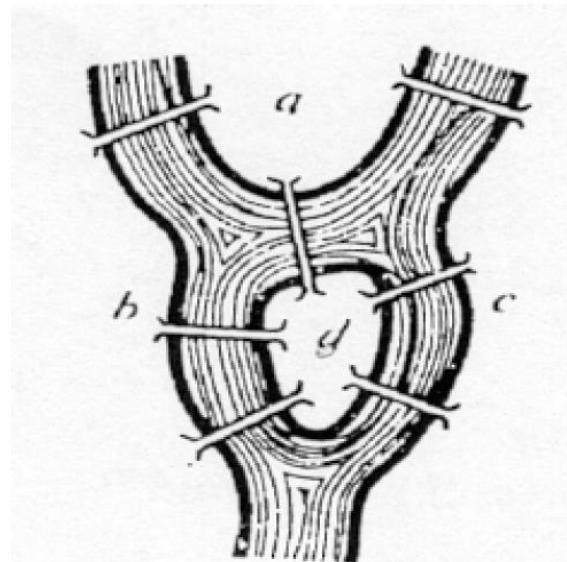


Das Königsberger Brückenproblem

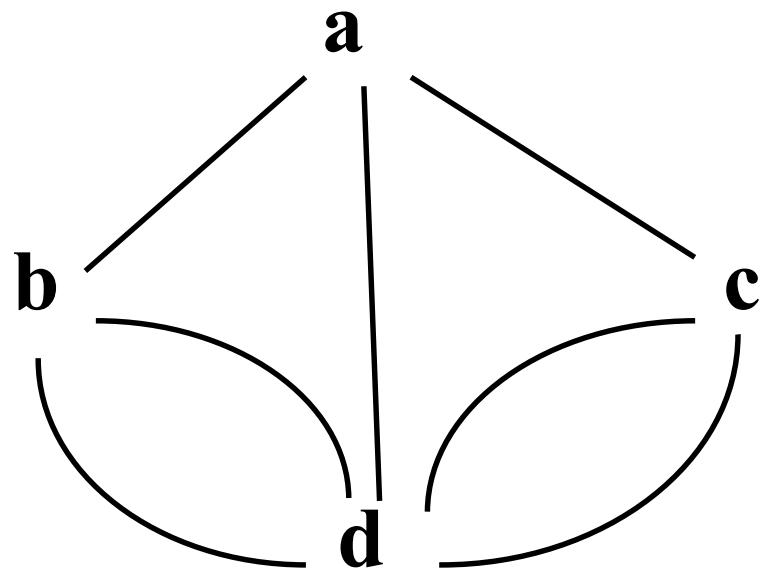
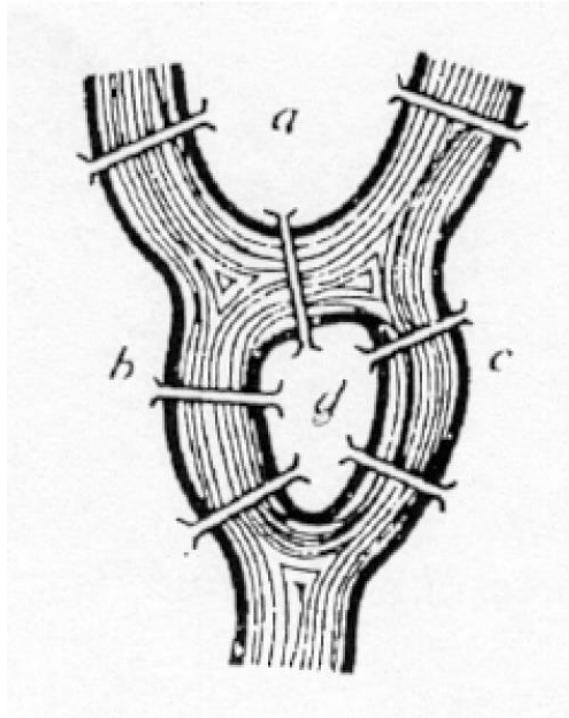


Un peu d'Histoire...

La ville de Koeninsberg est traversée par la Pregel, qui coule de part et d'autre de l'île de Kneiphof, et possède sept ponts.



Euler (1736) : Peut-on se promener dans la ville en traversant chaque pont une et une seule fois ?



1-graphe ou **graphe simple**:

une arête au plus entre 2 sommets

multigraphe ou **k-graphe**:

k arêtes au plus entre 2 sommets

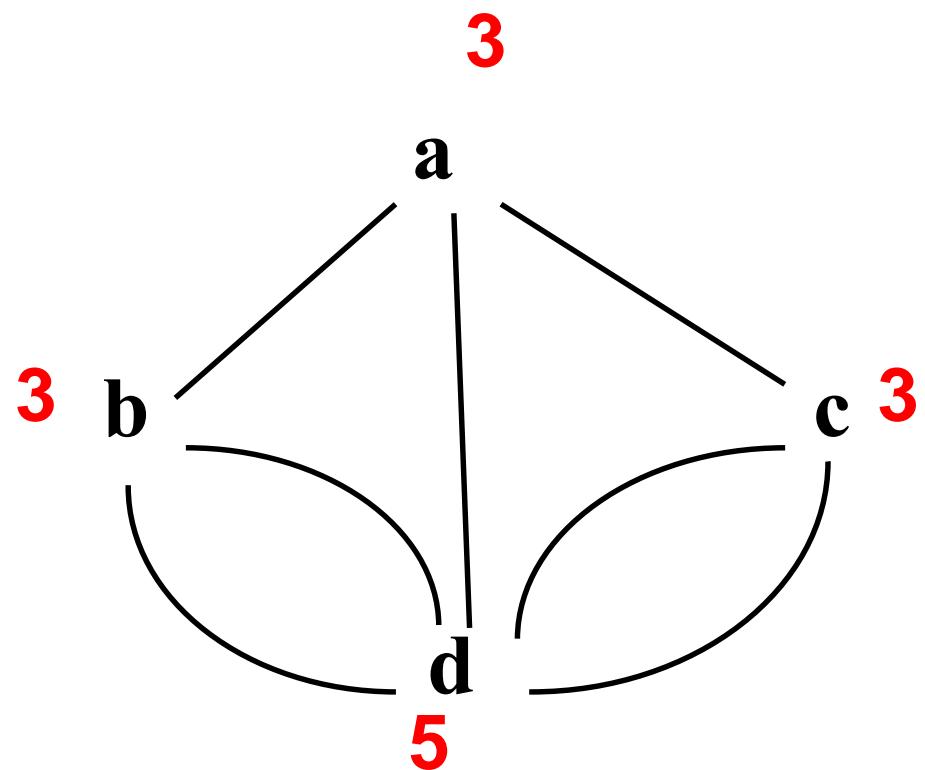
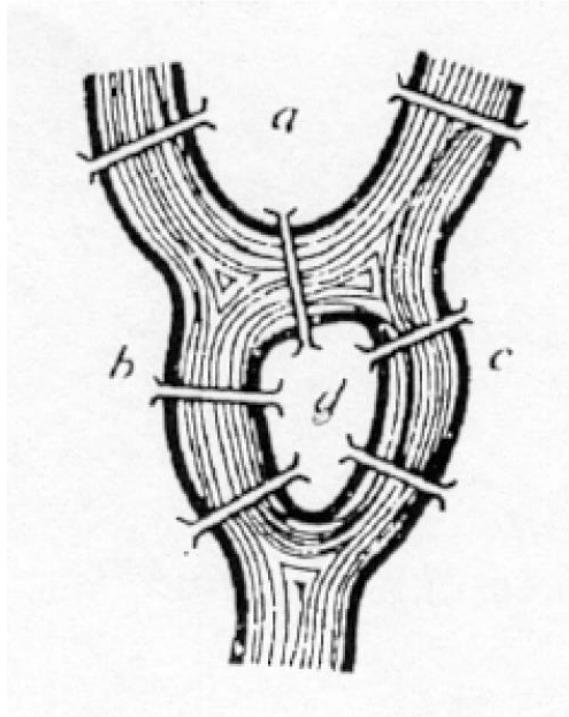
chaîne eulérienne : chaîne qui passe une fois et une seule par chaque arête

Théorème d'Euler

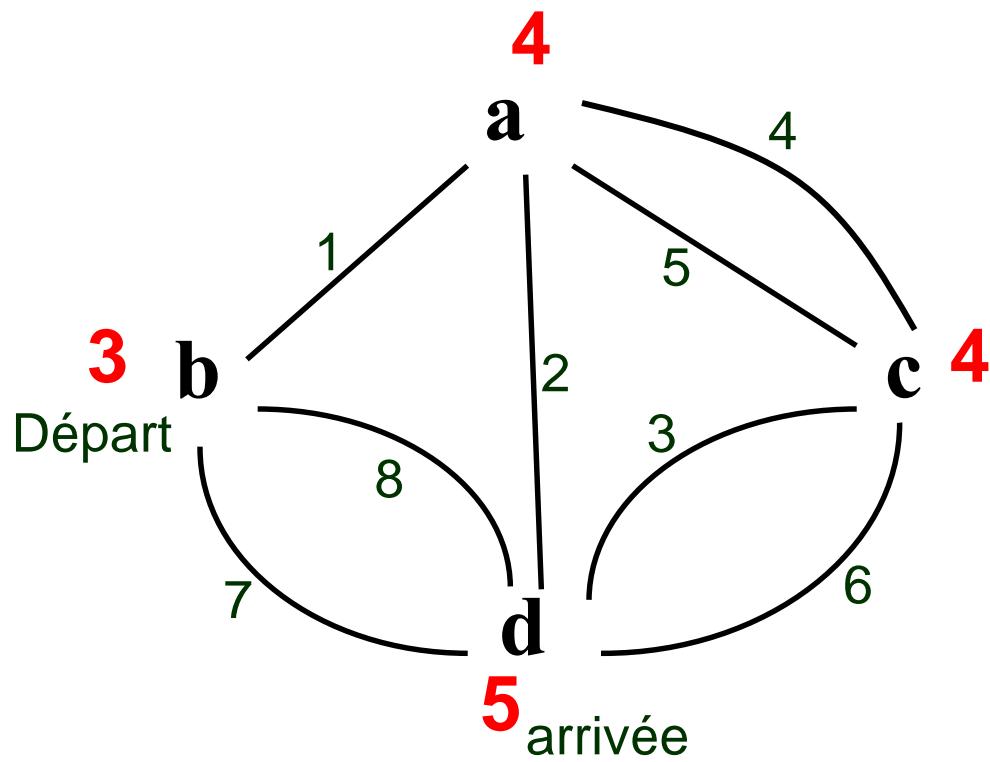
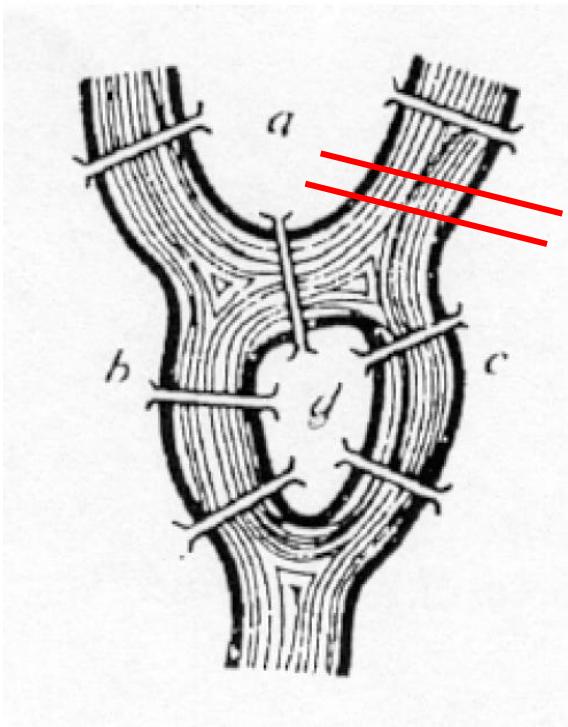
Un multigraphe connexe admet une chaîne eulérienne si et seulement si le nombre de sommets de degré impair est 0 ou 2

Peut-on se promener dans la ville en traversant chaque pont une et une seule fois ?

NON !



En ajoutant un pont: OUI !



Théorème d'Euler

Un multigraphe connexe admet une chaîne eulérienne si et seulement si le nombre de sommets de degré impair est 0 ou 2

C'est nécessaire car si on suit une chaîne eulérienne

- quand on traverse un sommet (on compte 2 arêtes)
 - quand on ne traverse pas un sommet (il est de degré impair)
- et c'est soit le début soit la fin de la chaîne

C'est suffisant pour un graphe à 1 arête ou 2 arêtes

Supposons vrai pour un nombre arêtes $< m$.

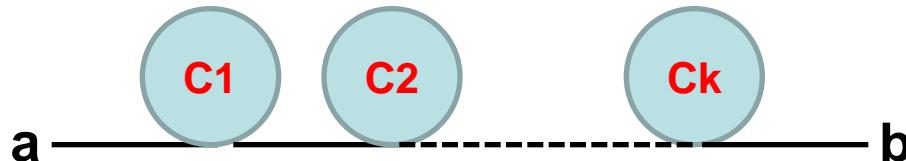
Soit un graphe connexe m arêtes et sans perte de généralité avec 2 sommets (uniquement) a et b de degré impair

Preuve th. Euler (condition suffisante suite)

Le graphe est connexe donc il existe chaîne de a à b.
Si la chaîne est eulérienne c'est fini.

Sinon, retirons cette chaîne. Tous les sommets sont alors de degré pair.
Soit C_1, \dots, C_k les composantes connexes du graphe obtenu après retrait.
 C_1, \dots, C_k ont moins de m arêtes donc par récurrence il existe cycle eulérien dans chacune.

Finalement, en faisant l'union de tous ces cycles eulériens et de la chaîne initiale de a à b, on obtient une chaîne eulérienne d'extrémités a et b.



Un algorithme de recherche de chaîne eulérienne

Supposons (sans perte de généralité) G a exactement 2 degrés impairs

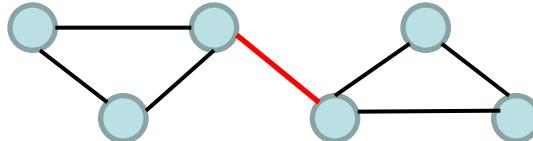
$G' = G$

Partir d'un sommet de degré impair

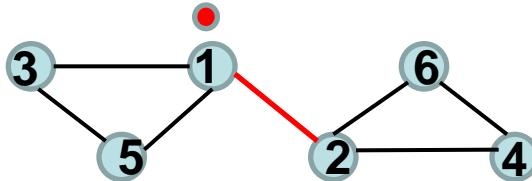
Tant qu'il y a des arêtes dans G'

1. Traverser une arête e qui n'est pas un isthme pour G'
sauf si on est sur un sommet de degré 1 dans G'
2. $G' = G' \setminus \{e\}$

Un isthme est une arête qui déconnecte le graphe



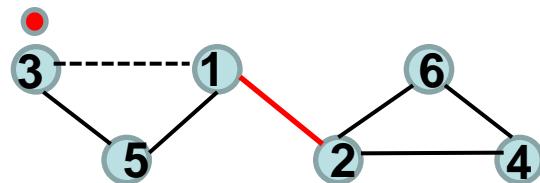
Exemple



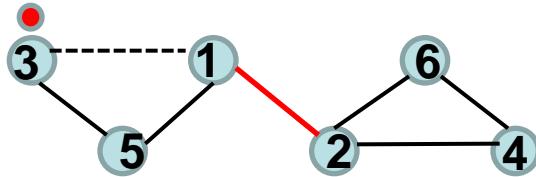
On part de 1

On ne va pas en 2 car on traverse l'isthme rouge et $d_G(1) \neq 1$

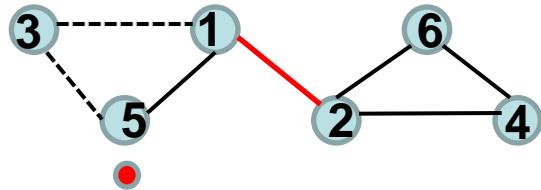
On va en 3



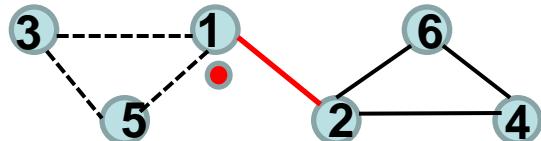
On retire $e=[1,3]$



On va en 5 On retire $e=[3,5]$

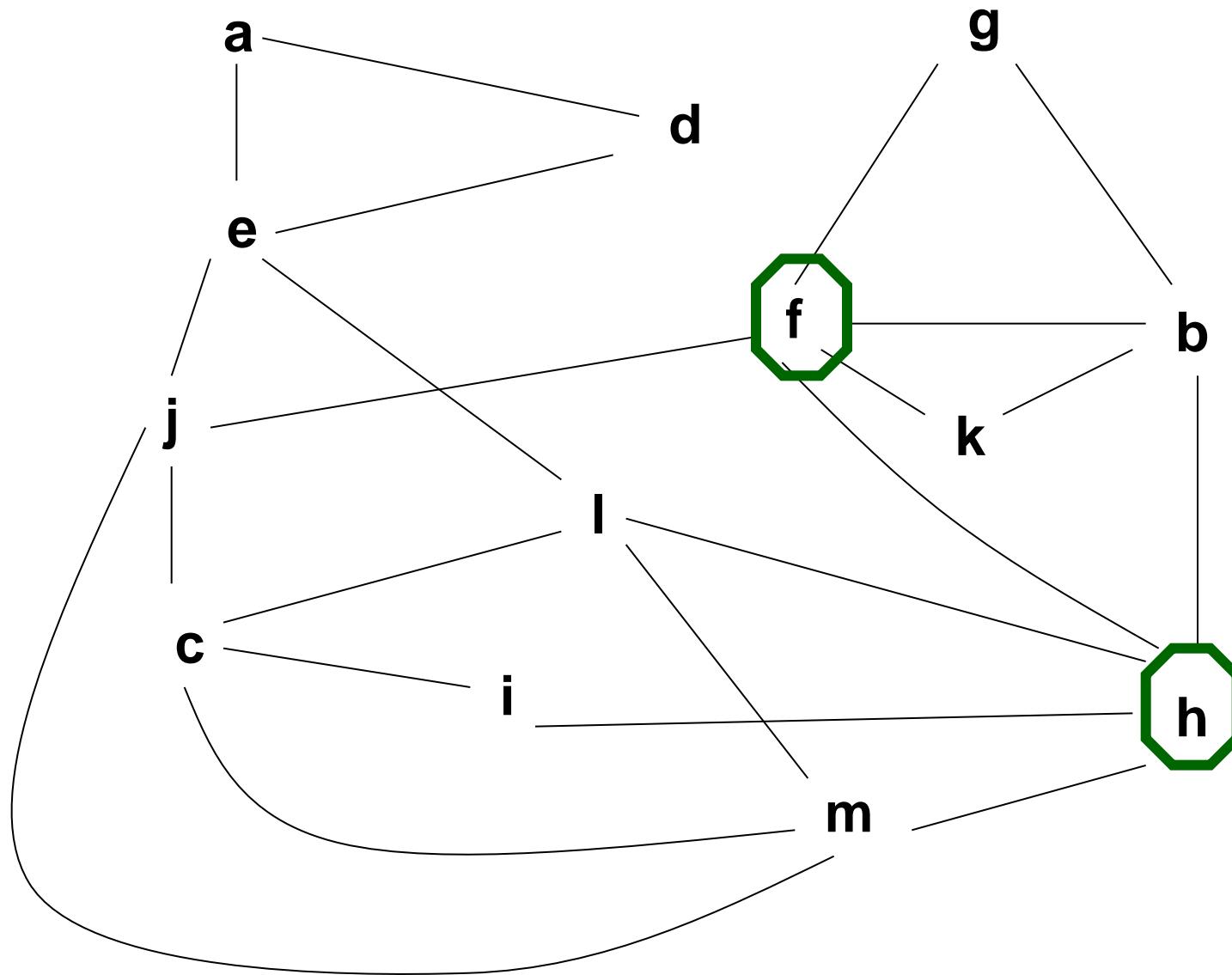


On va en 1 On retire $e=[5,1]$



Cette fois, $d_G(1)=1$ on peut traverser $e=[1,2]$

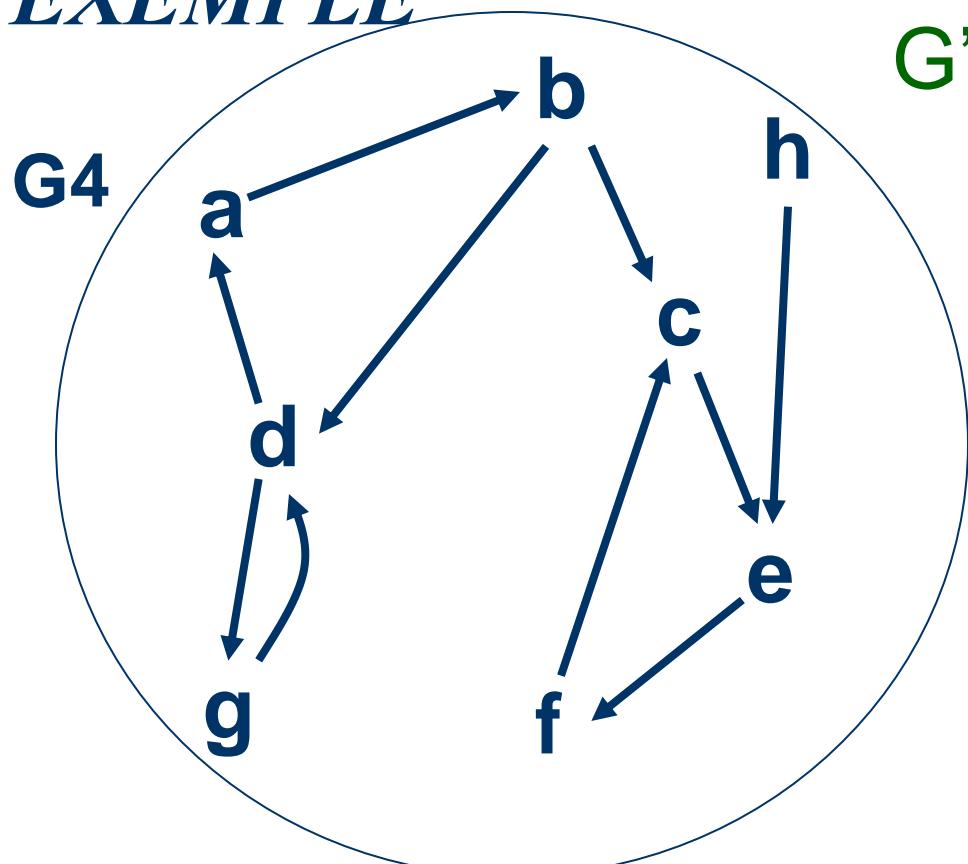
Trouver chaîne eulérienne dans ce graphe



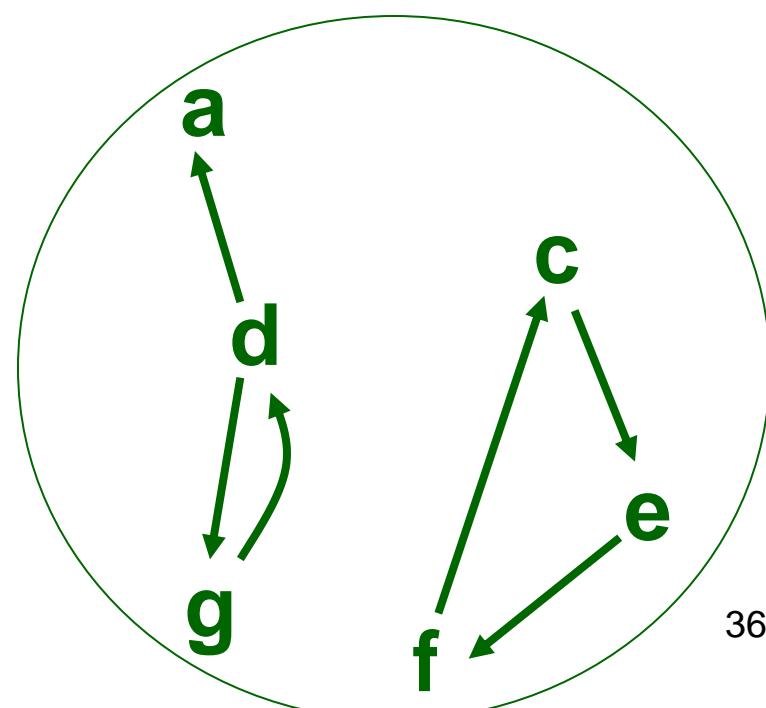
CONNEXITE ET FORTE CONNEXITE

sous-graphe $G'=(X',A')$ de G (orienté ou non):
 $X' \subseteq X$ $A' \subseteq A$ et $\forall x,y \in X', [x,y] \in A \Leftrightarrow [x,y] \in A'$

EXAMPLE



G'_4 sous-graphe de G_4



sous-ensemble maximal pour une propriété \wp : sous-ensemble tel que l'ajout d'un élément lui fait perdre la propriété \wp

Composante connexe de G : sous-graphe de G connexe maximal

EXAMPLE

G' a 2 composantes connexes

Algorithme de recherche des composantes connexes d'un graphe

entrée: graphe $G=(X,A)$

sortie: liste des composantes connexes

principe:

- déterminer une composante connexe C en partant d'un sommet quelconque
- retirer C du graphe et recommencer

Procédure Comp_connexes

$J \leftarrow 1;$

Tant que tous les sommets n'ont pas été classés faire

 Choisir un sommet de départ non classé;

$S \leftarrow$ sommet de départ;

 Classer S dans la composante numéro J ;

 Tant que S non fini faire

 Tant que c'est possible faire

$S \leftarrow$ un voisin de S non classé;

 Classer S dans J ;

 Fin tant que;

 Marquer S par fini;

$S \leftarrow$ sommet à partir duquel S a été classé **si** S n'est pas
 le sommet de départ;

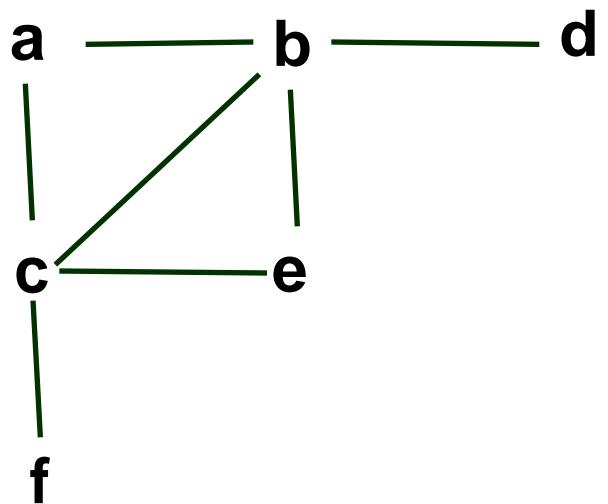
Fin tant que;

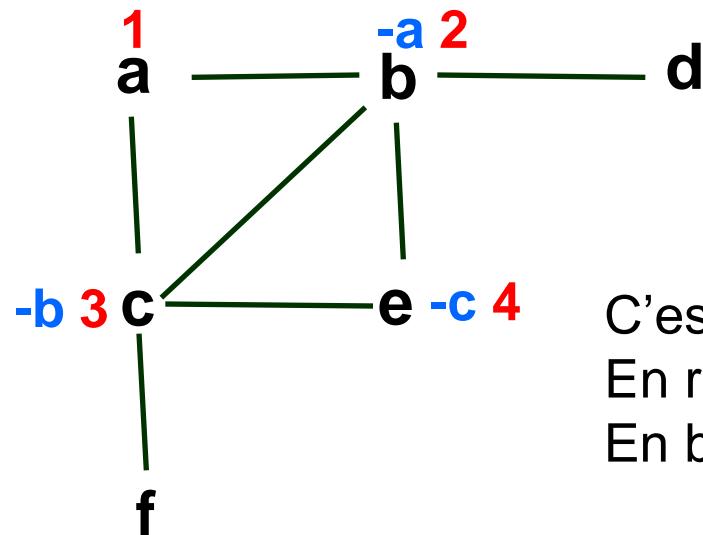
$J \leftarrow J+1;$

Fin tant que;

Fin;

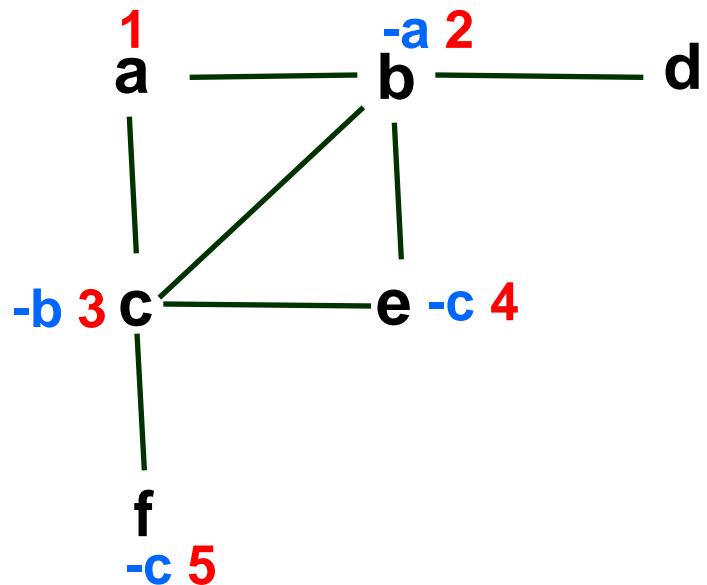
Appliquons l'algorithme



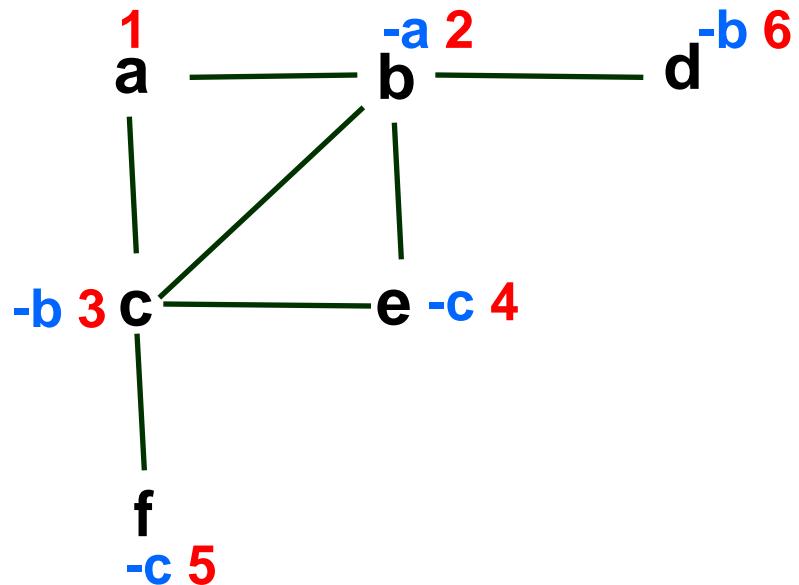


C'est un parcours en « profondeur d'abord »
 En rouge = ordre de parcours
 En bleu = sommet d'où l'on vient

c a pour voisin **b** qui est déjà classé donc on passe au voisin **e**



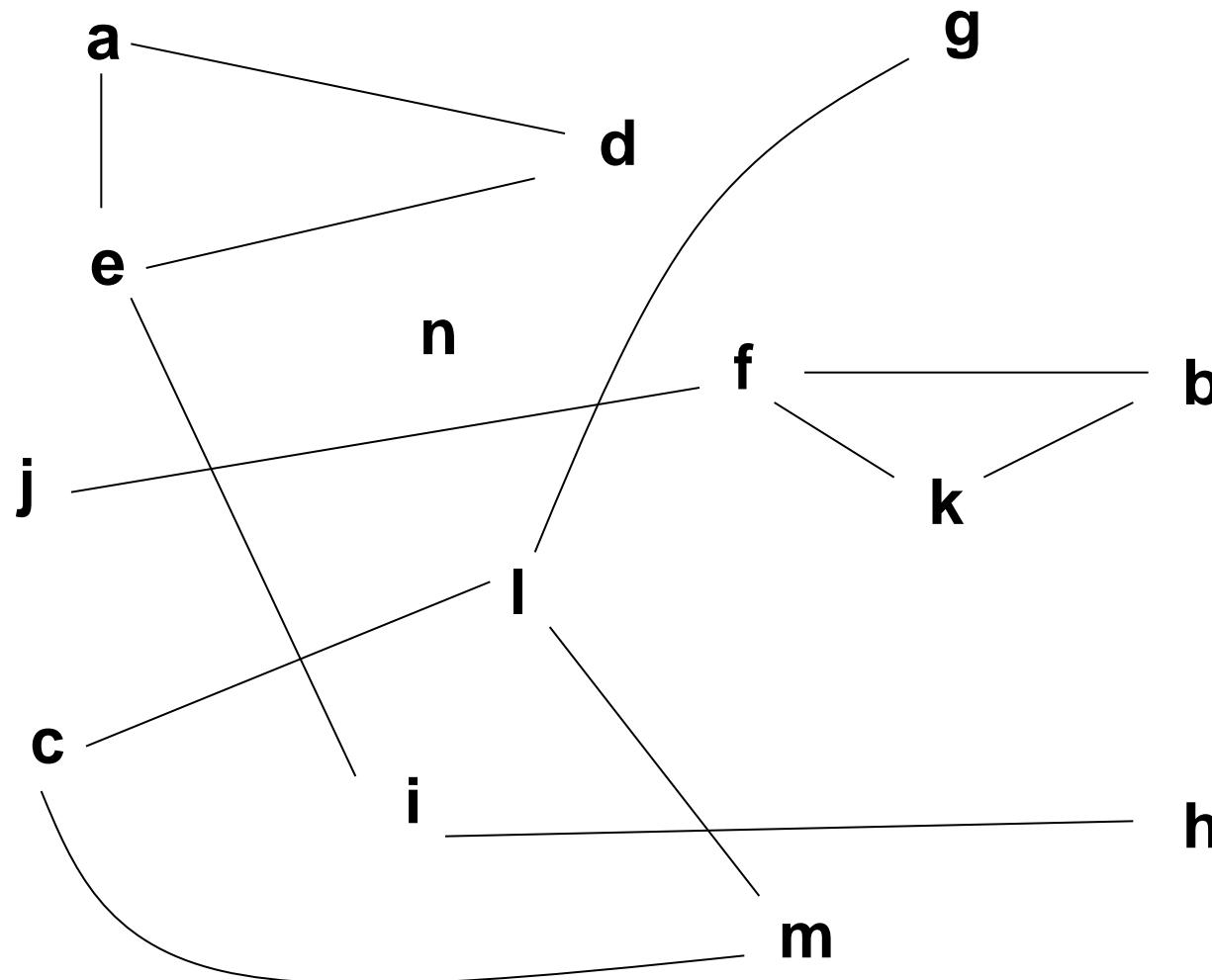
e a pour voisin **b** qui est déjà classé donc on revient sur **c** (backtrack)
 Et on explore le voisin **f** de **c**



f n'a pas de voisin non classé donc on revient sur **c** (backtrack)
 Qui lui-même n'a pas de voisin non classé
 Et on revient sur **b** (backtrack)
 Qui a un voisin **d** non classé

Ensuite, par backtrack de **d** on remonte à **b** puis **a** et c'est terminé

Appliquer l'algorithme à ce graphe et donner les composantes connexes

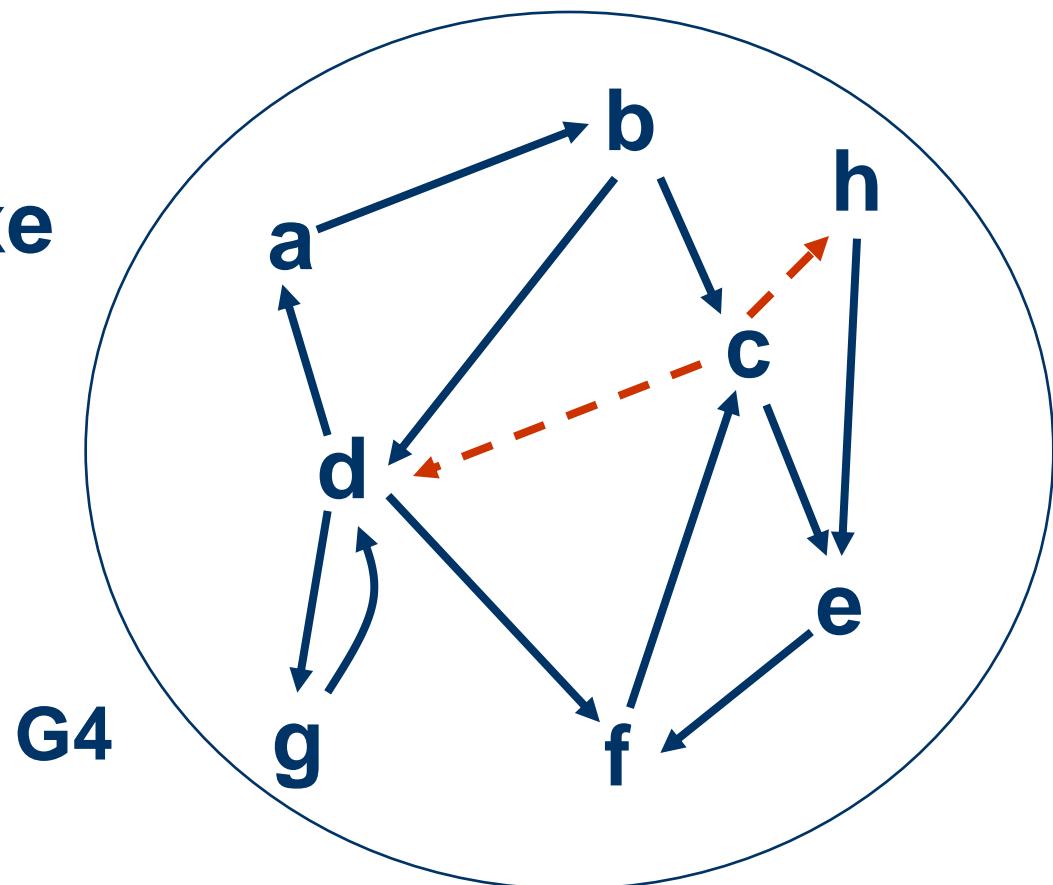


Forte connexité : un graphe orienté est f-connexe si tout couple de sommet est relié par un chemin

EXAMPLE

G4 non f-connexe

Mais si on ajoute (c,d) et (c,h) , G4 devient f-connexe

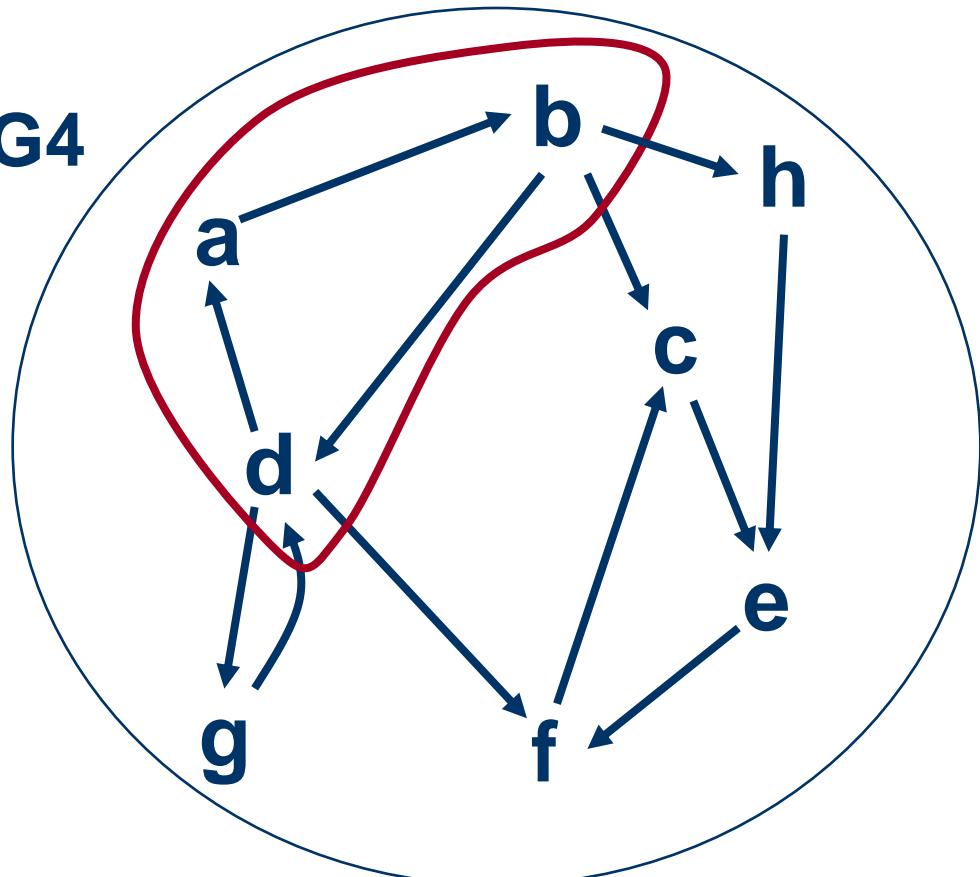


Composante fortement-connexe de G: sous-graphe de G f-connexe maximal

EXAMPLE

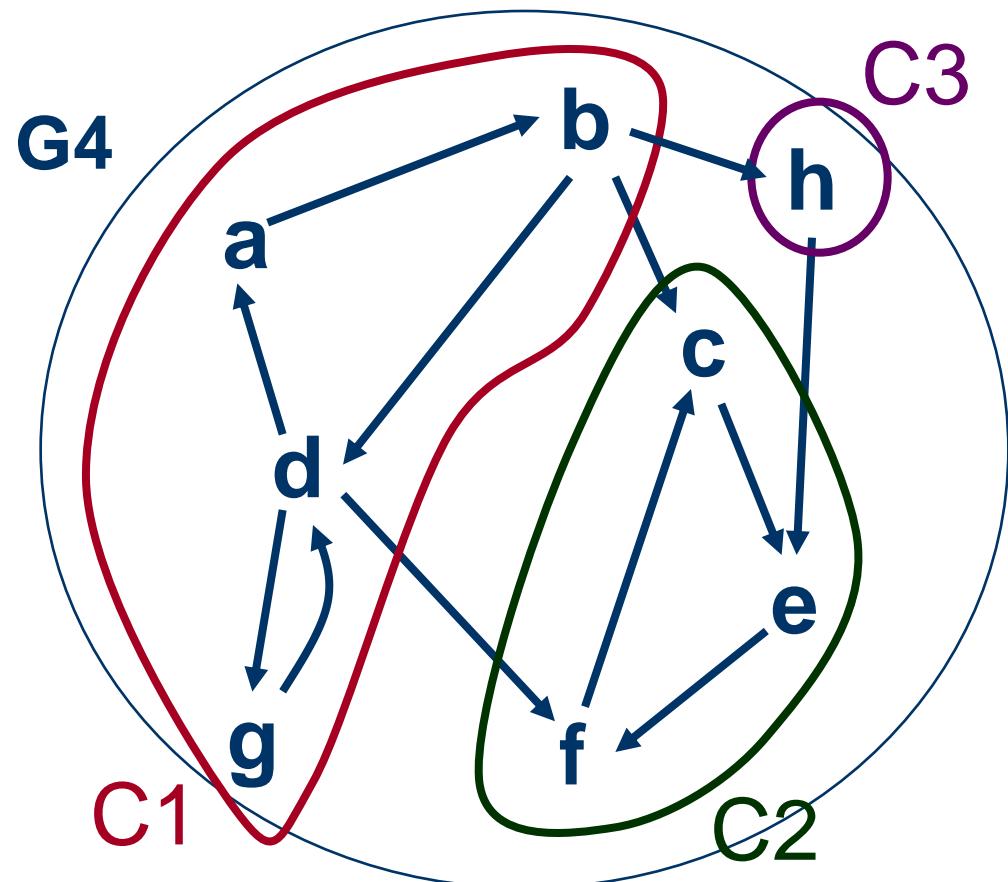
{a,b,d} défini un sous-graphe f-connexe mais non maximal donc ne définit pas une composante

G4



Composante f-connexe de G : sous-graphe de G f-connexe maximal

EXEMPLE
 G_4 a 3 composantes f-connexes



Algorithme de recherche des composantes fortement connexes d'un graphe

entrée: graphe $G=(X,A)$

sortie: liste des composantes f-connexes

principe:

- déterminer une composante f-connexe C en partant d'un sommet quelconque
- retirer C du graphe et recommencer

Procédure composantes_f-connexes de G=(X,A):

***E $\leftarrow X;$**

Tant que¹ **E non vide** faire

***marquer + et - un sommet x de E;**

tant que² **c'est possible** faire

***marquer par + tout successeur (non encore marqué par +) d'un sommet déjà marqué + ;**

***marquer par - tout prédecesseur (non encore marqué par -) d'un sommet déjà marqué - ;**

fait²;

***Écrire C l'ensemble des sommets marqués + et -;**

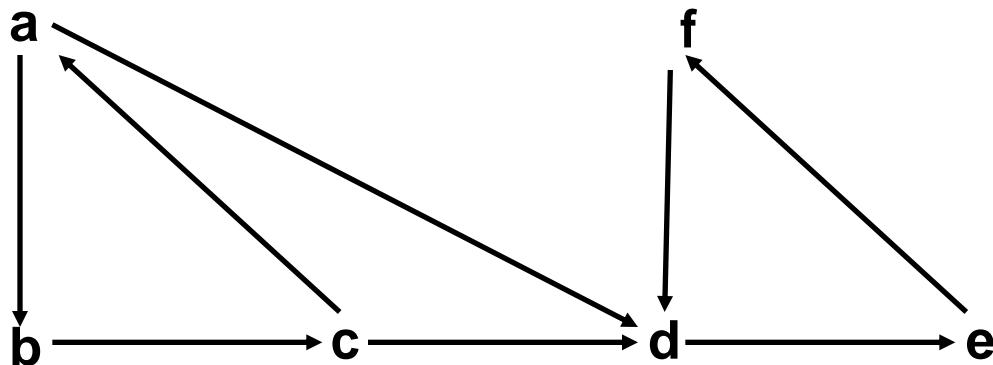
***Le sous-graphe de G dont les sommets sont ceux de C est une composante f-connexe de G;**

***E $\leftarrow E - C;$ C $\leftarrow \emptyset;$**

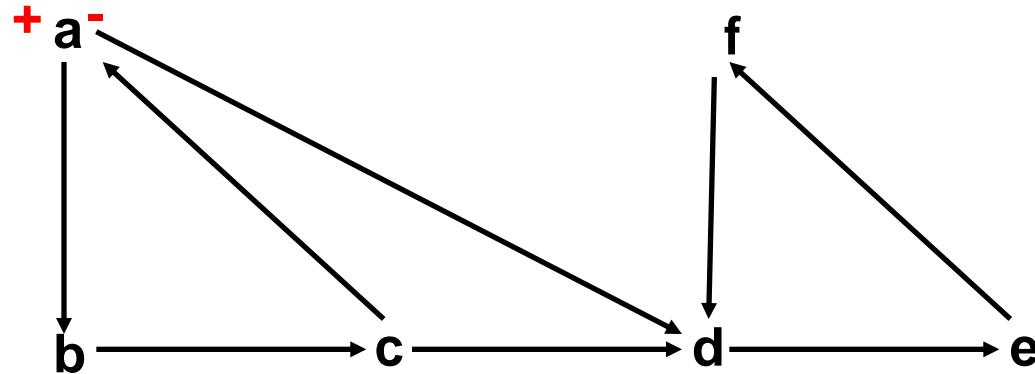
fait¹;

fin;

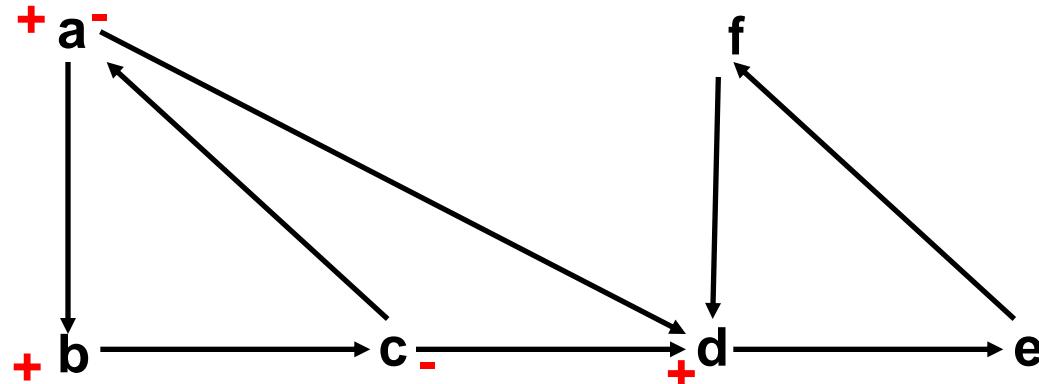
Appliquons l'algorithme

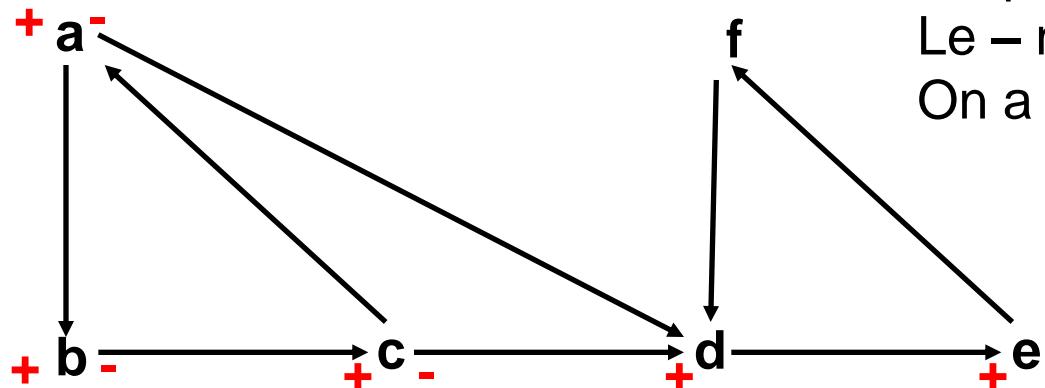


On part de **a** par exemple



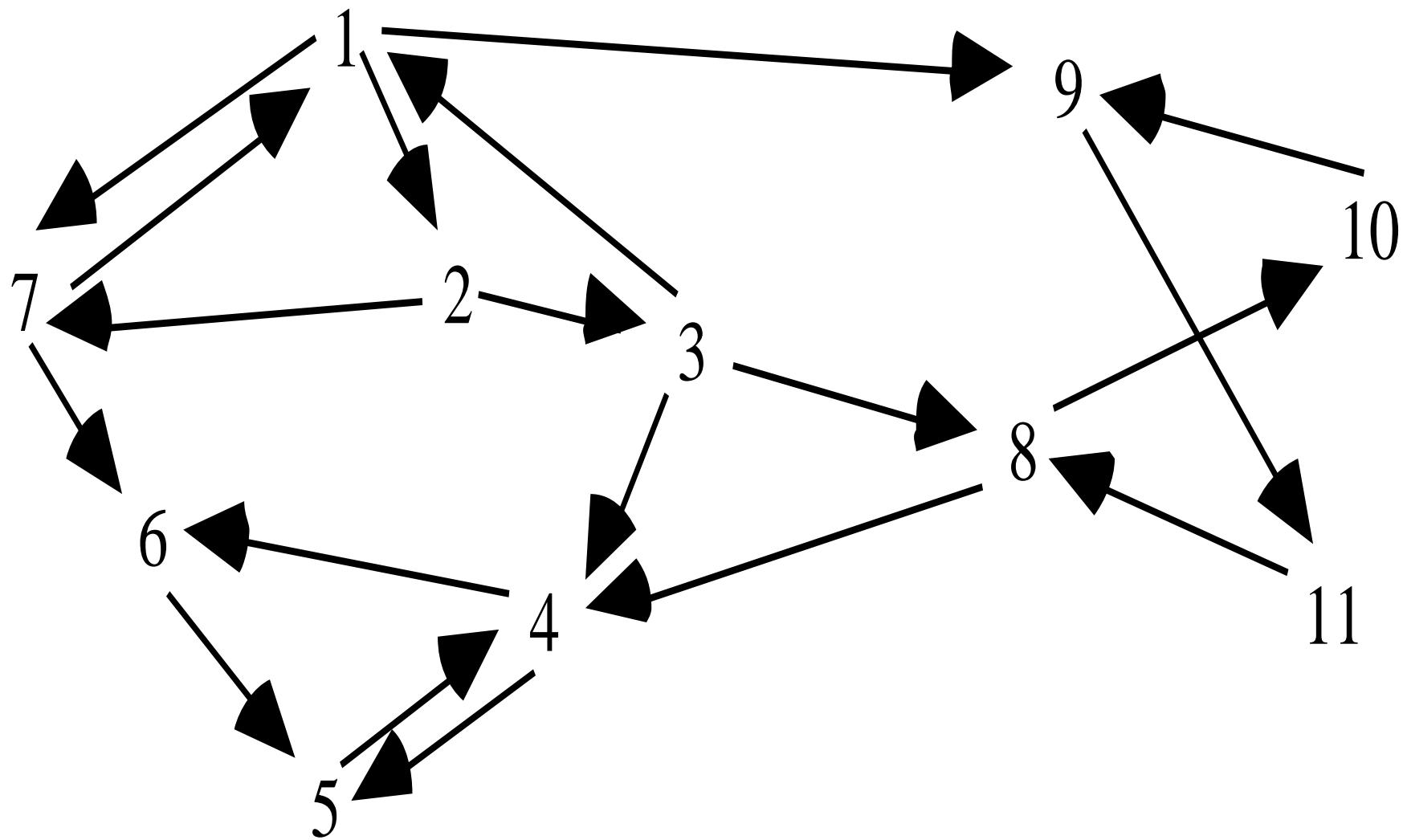
On propage le + et le -

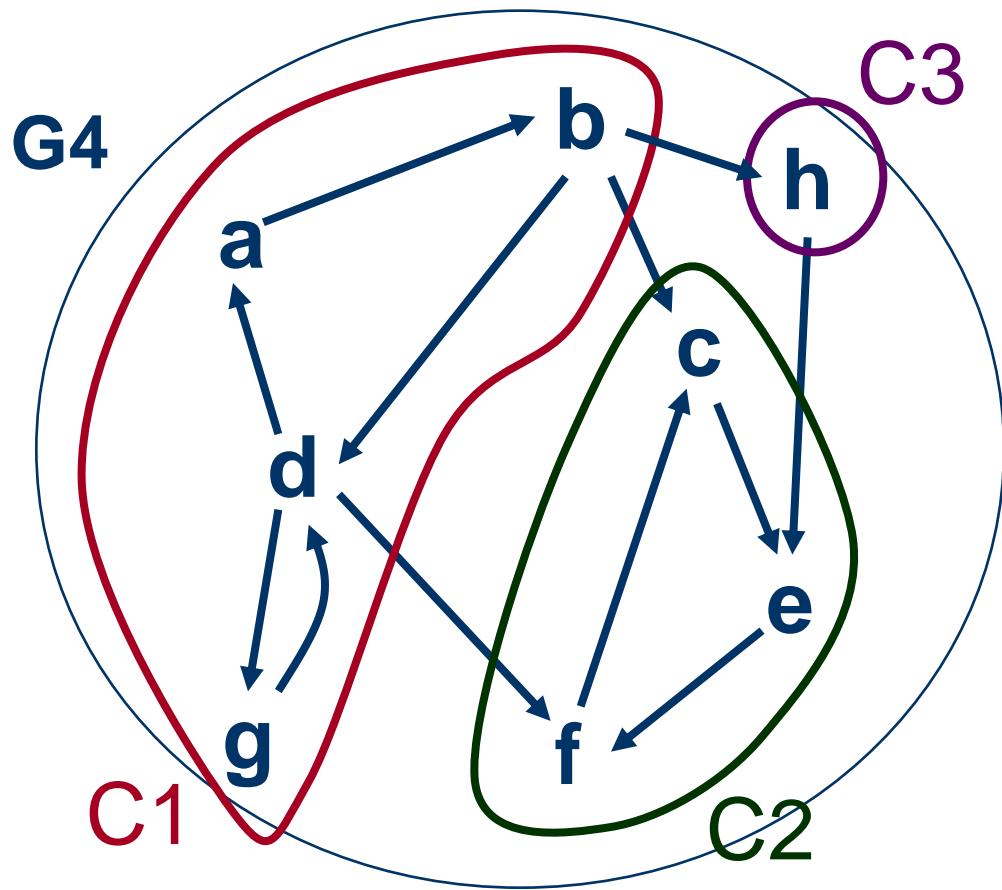




On propage le + et le -
Le - ne se propage plus
On a une cfc {a , b , c}

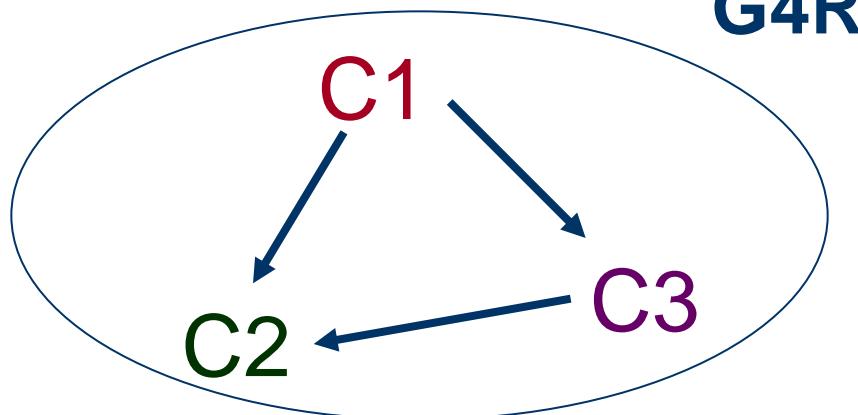
Trouver les cfc de ce graphe





GR est sans circuit !

Graphe réduit



THEOREME

Soit G un graphe connexe avec au moins un arc, alors:

**G est f -connexe
si et seulement si
par tout arc de G il passe un circuit (au moins)**

Tri topologique ou mise en ordre d'un graphe

Procédure **Tri_topologique**

début

Y \leftarrow **X**; **N** \leftarrow 1;

tant que **Y** non vide faire

**mettre sur le niveau N tous les sommets de Y n'ayant
aucun prédécesseur dans Y;**

soit X_N l'ensemble des sommets de niveau N;

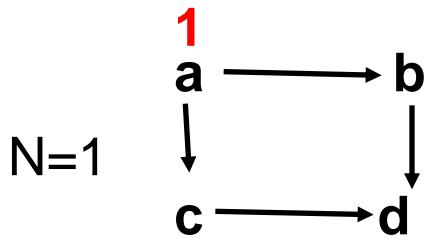
Y \leftarrow **Y -** X_N ;

N \leftarrow **N+1**;

fait;

fin;

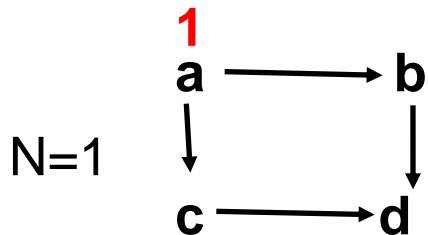
Appliquons l'algorithme



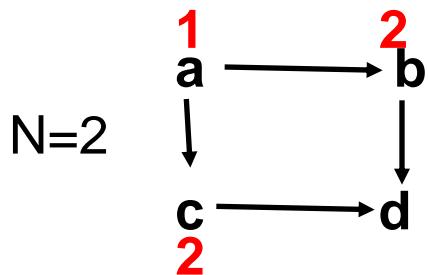
$Y = a, b, c, d$

a n'a aucun prédecesseur

Appliquons l'algorithme

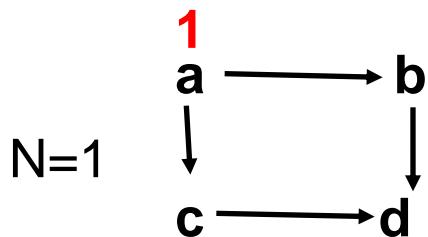


$Y = a, b, c, d$
a n'a aucun prédecesseur

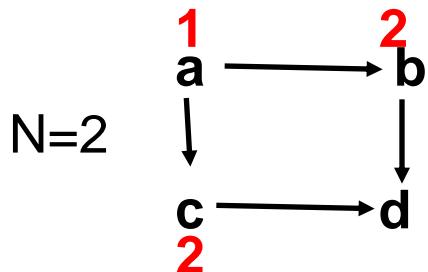


$Y = b, c, d$
b et c n'ont aucun prédecesseur dans Y

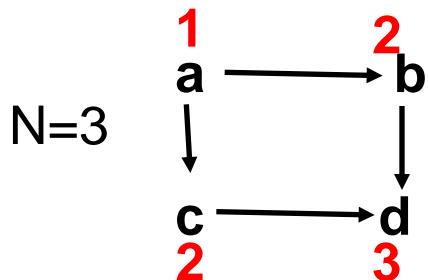
Appliquons l'algorithme



$Y = a, b, c, d$
a n'a aucun prédécesseur



$Y = b, c, d$
b et c n'ont aucun prédécesseur dans Y

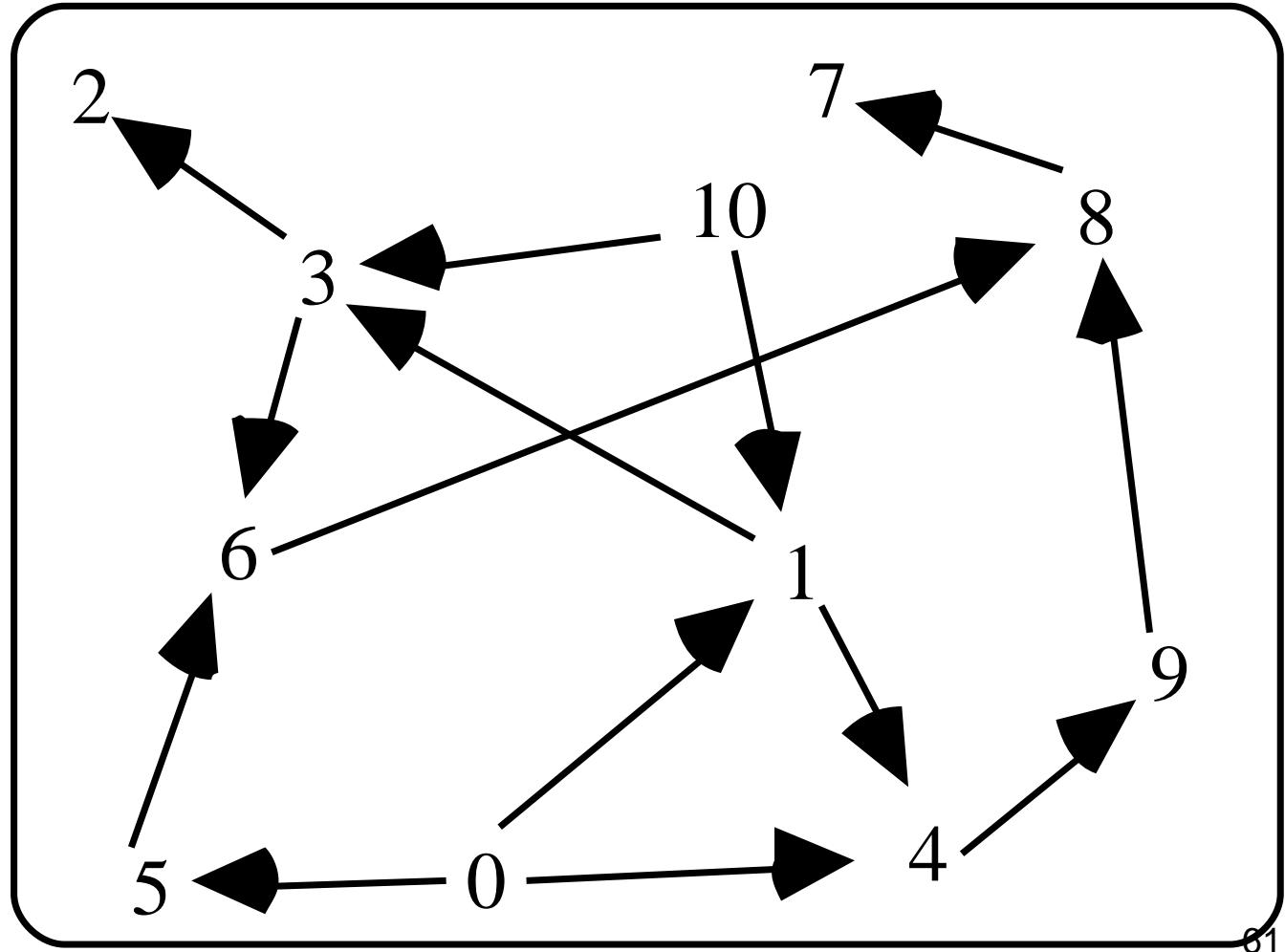


$Y = d$
d n'a aucun prédécesseur dans Y

Les prédécesseurs d'un sommet de niveau N sont de niveaux < N

Tri topologique ou mise en ordre d'un graphe

Mettre en niveaux ce graphe



Exploration en profondeur d'un graphe

Procédure Num_en_profondeur

Tant que tous les sommets n'ont pas été numérotés
faire

 Choisir un sommet de départ S non numéroté;

 Numéroter S;

 Tant que S non fini faire

 Tant que c'est possible faire

 S \leftarrow un successeur de S non numéroté;

 Numéroter S;

 Fin tant que;

 Marquer S par fini;

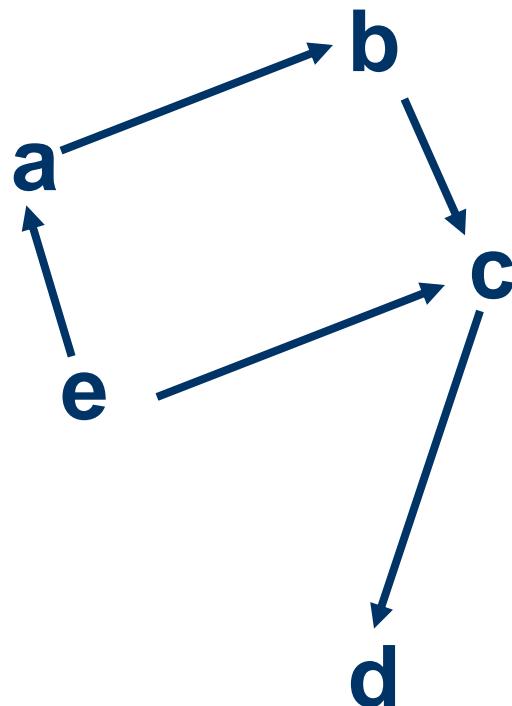
 S \leftarrow sommet qui a permis de numéroter S si S
n'est pas le sommet de départ;

 Fin tant que;

Fin tant que;

Fin;

Matrice booléenne associée à un graphe



	a	b	c	d	e
a	0	1	0	0	0
b	0	0	1	0	0
c	0	0	0	1	0
d	0	0	0	0	0
e	1	0	1	0	0

Matrice sommets x sommets appelée **matrice d'adjacence**

Fermeture transitive d 'un graphe

Soit $G=(X,U)$

$\tau(G)=(X,\tau(U))$ est la fermeture transitive de G

si, pour tout $(x,y) \in X^2$,

$$(x,y) \in \tau(U)$$

\Leftrightarrow

il existe un chemin de x à y dans G

Graphe partiel

$G=(X,U)$ orienté ou pas

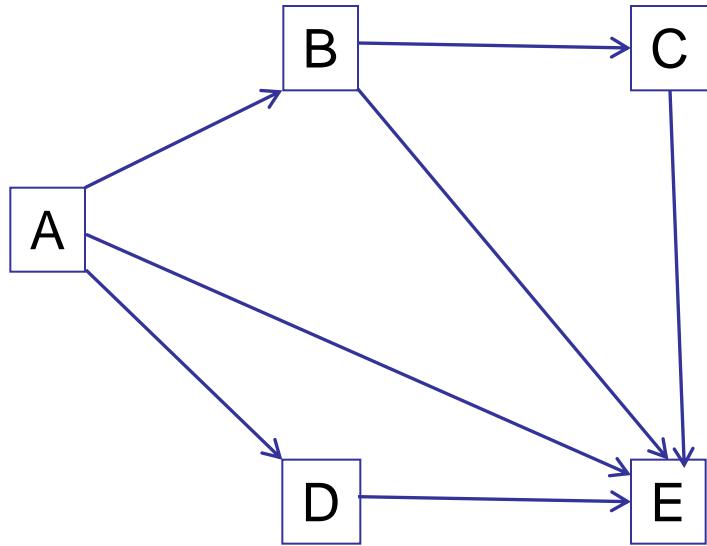
Graphe partiel de G : on retire des arcs (cas orienté)
ou des arêtes (cas non orienté)

$U' \subset U$, $G'=(X,U')$

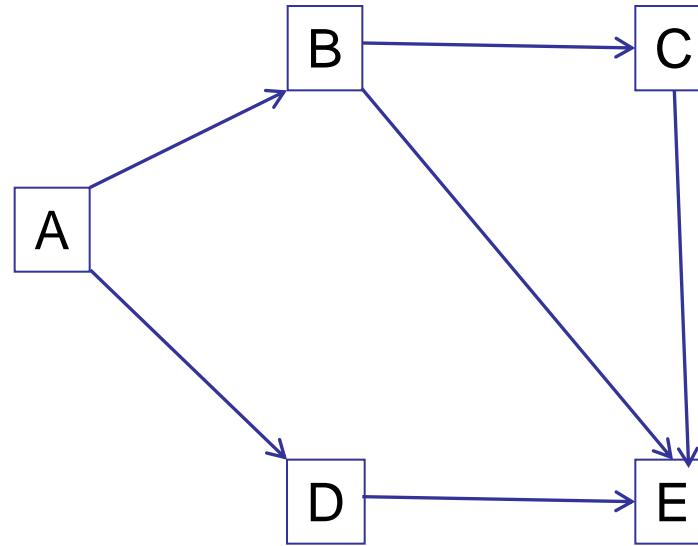
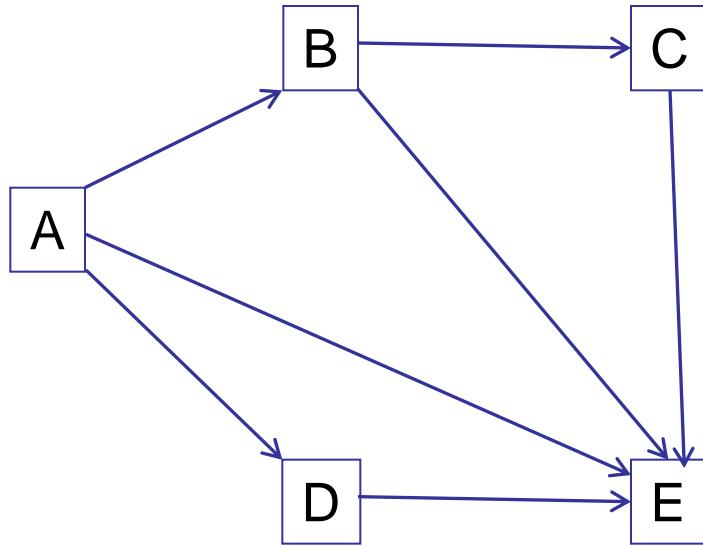
Question:

Peut-on retirer des arcs à G sans altérer sa fermeture transitive ?
Et combien ?

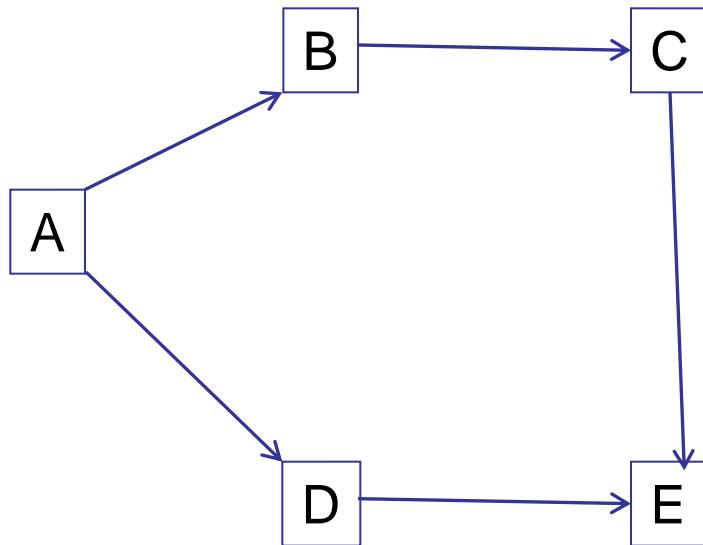
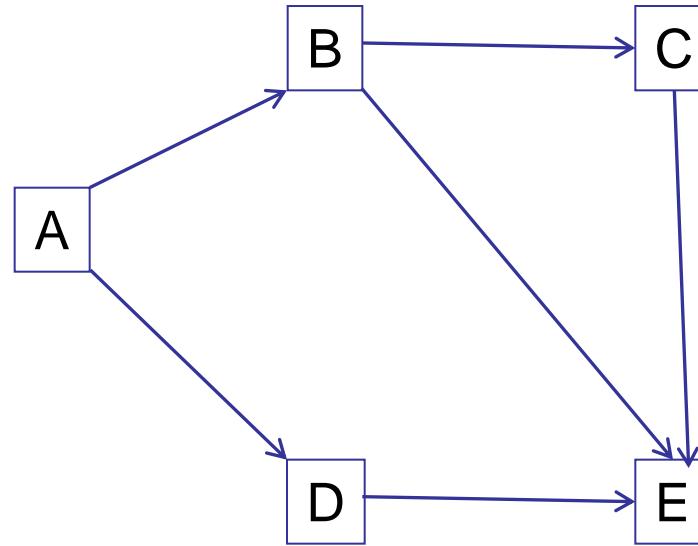
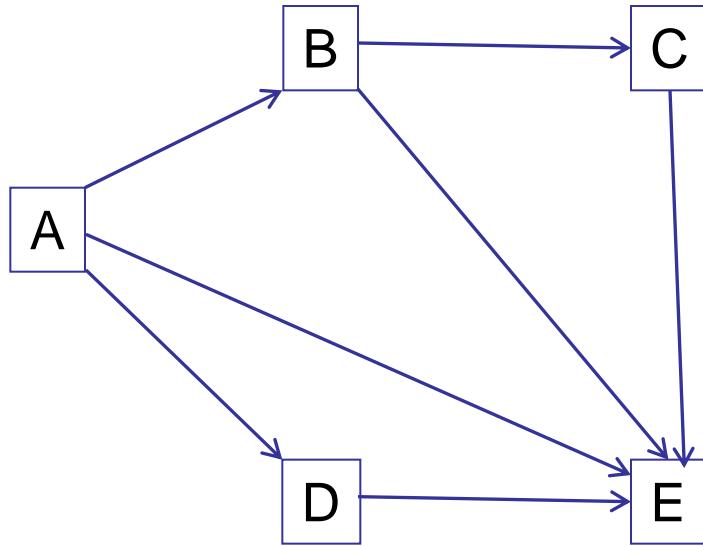
Peut-on retirer des arcs à G sans altérer sa fermeture transitive ?



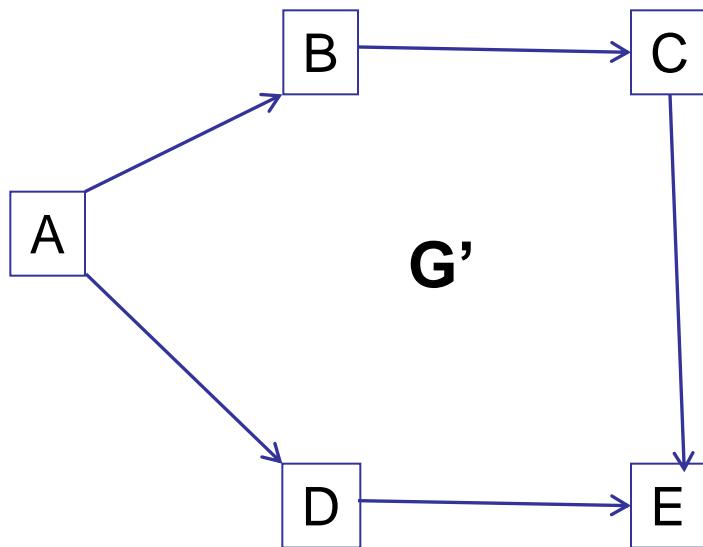
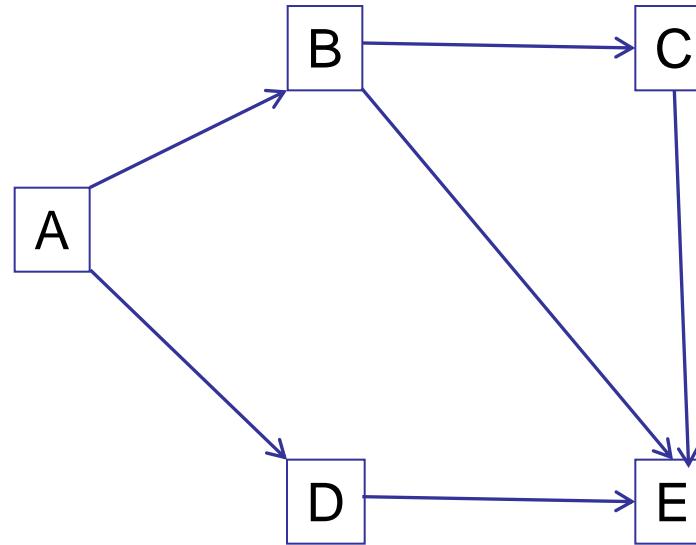
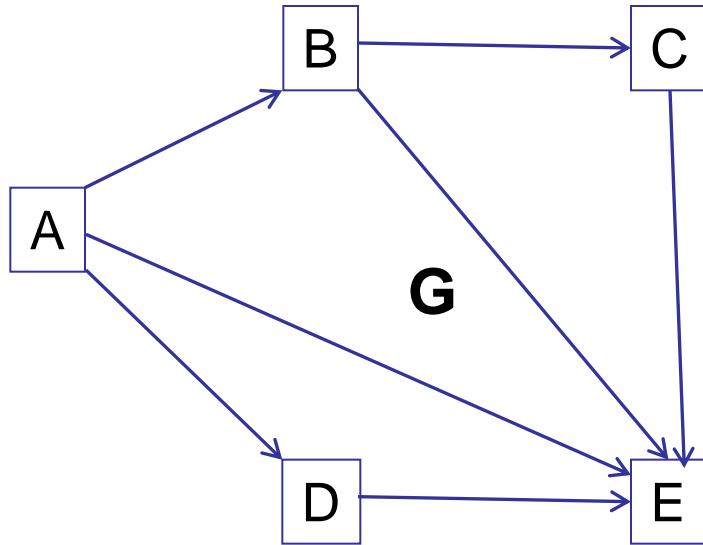
Peut-on retirer des arcs à G sans altérer sa fermeture transitive ?



Peut-on retirer des arcs à G sans altérer sa fermeture transitive ?



Peut-on retirer des arcs à G sans altérer sa fermeture transitive ?



On ne peut plus retirer d'arcs
sinon la fermeture transitive change

G' est τ -minimal τ -équivalent à G

Fermeture transitive d 'un graphe τ -minimalité

Deux graphes G et G' sont τ -équivalents

$$\Leftrightarrow$$

$$\tau(G) = \tau(G')$$

Théorème

Si G est sans circuit, il n'existe qu'un seul graphe τ -minimal τ -équivalent à G

Fermeture transitive d 'un graphe

Algorithme de Roy-Warshall

Procédure **fermeture_transitive**

Principe

$\tau(G)$ = graphe donné

pour tout sommet faire

 relier chacun de ses prédécesseurs dans $\tau(G)$
 à chacun de ses successeurs dans $\tau(G)$;

$\tau(G) \leftarrow$ graphe ainsi obtenu

fait;

Fermeture transitive d 'un graphe

Implémenter l'algorithme de Roy-Warshall

avec la matrice d'adjacence

Comment identifier les successeurs d'un sommet ?

Comment identifier les prédecesseurs d'un sommet ?

Relier les prédecesseurs d'un sommet aux successeurs ?

2

PROBLEMES

ET

COMPLEXITE

*Des problèmes qui se ressemblent
et pourtant...*

un problème "facile"
existe t'il une chaîne eulérienne dans G ?

et

un problème "difficile"
existe t'il un chemin hamiltonien dans G ?

Mais que veut dire "complexité" ?

Qu'est la "théorie de la complexité" ?

Attention!
Nous donnons ici une
réponse INTUITIVE

Problème de Décision:
Réponse par OUI ou NON

Classe NP

De manière intuitive, un problème de décision est dans la classe *NP* si, quand on sait que la réponse est OUI, on peut facilement convaincre un tiers que c'est vrai.

(Mais on ne peut pas forcément trouver que la réponse est oui.)

Exemple si Carlos sait qu'il existe un cycle hamiltonien dans un graphe donné, il peut facilement vous en convaincre



**Mais si le graphe est grand,
Carlos ne pourra pas savoir si ce cycle existe**

Classe *P*

Un problème de *NP* est "facile" (**polynomial**) si on peut le résoudre par un algorithme "efficace" (temps polynomial en fonction de la taille de l'instance)

Exemple

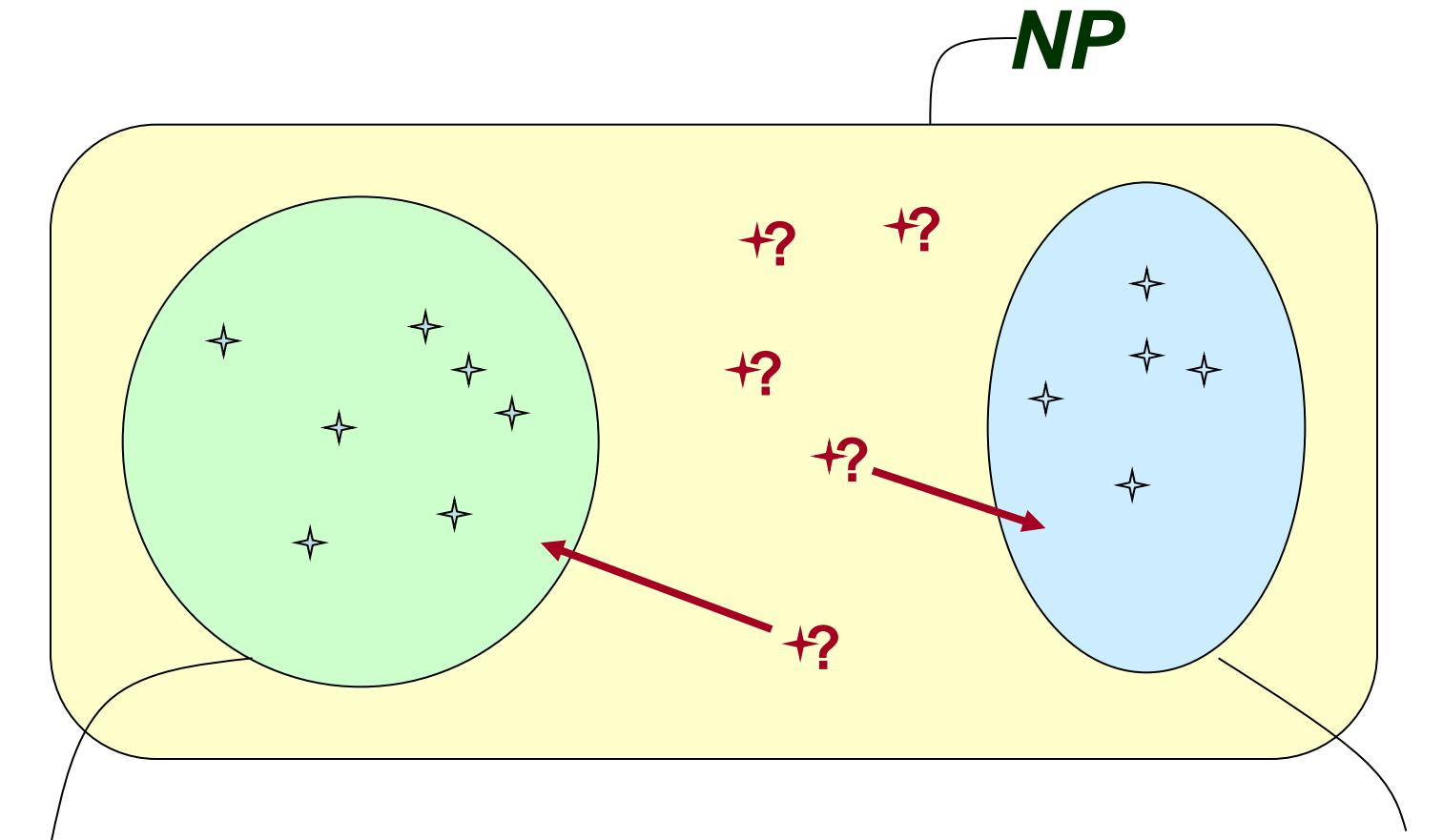
L'existence d'un cycle eulérien dans un graphe

Un problème est "difficile" si les seules méthodes connues pour le résoudre exigent un temps de calcul exponentiel en fonction de la taille de l'instance

Exemple
L'existence d'un cycle hamiltonien dans un graphe

Classe *NP-C*

Un problème de *NP* est *NP-complet* si
"savoir le résoudre efficacement"
implique
"savoir résoudre efficacement TOUS les
problèmes de *NP*" .



★ Problèmes

★? Problèmes non classés

Pour montrer qu'un problème Π est polynomial

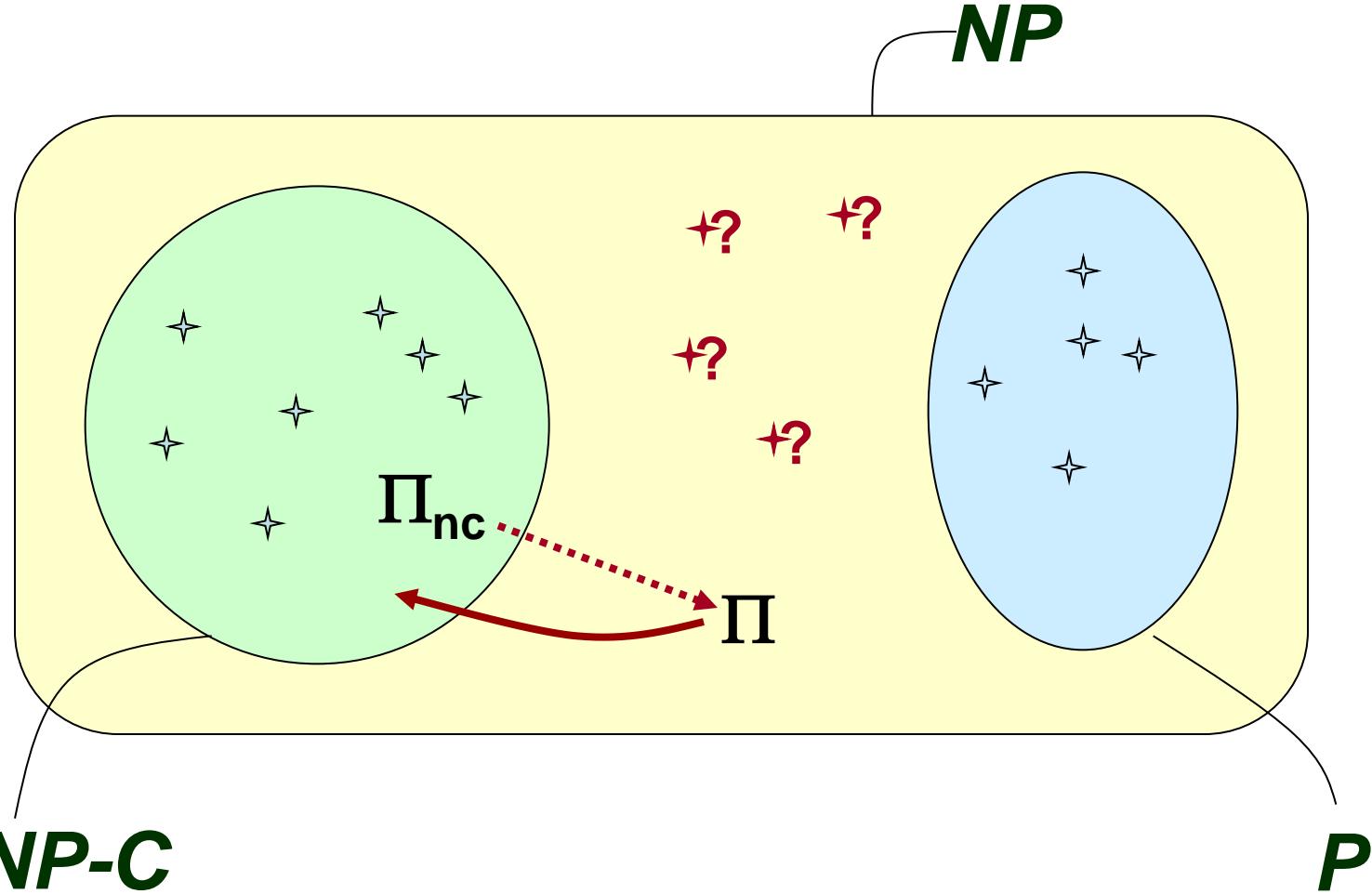
il faut trouver un algorithme pour le résoudre et prouver que cet algorithme s'exécute en un temps qui augmente de façon polynomiale en fonction de la taille de l'instance traitée

Pour montrer qu'un problème Π est NP -complet, on choisit un problème déjà connu pour être NP -complet, soit Π_{nc} , et on montre que Π_{nc} peut se "transformer" en Π .

Donc, si on savait résoudre Π , on saurait résoudre Π_{nc} .

Or, on ne sait pas résoudre Π_{nc} : donc il va sûrement être difficile de résoudre Π .

Π va, à son tour, être classé NP -complet.



Si on savait résoudre facilement Π on saurait résoudre
 aussi Π_{nc} ; or on ne sait pas résoudre Π_{nc}
 Π est donc sûrement difficile à résoudre

Les problèmes sont classés de façon incrémentale: la classe d'un nouveau problème est déduite de la classe d'un ancien problème.

L'établissement d'un "premier" problème NP -complet pour classer tous les autres s'est donc avéré nécessaire.

Le problème SAT

"satisfiabilité" d'une expression logique

Exemple

$$(x \vee \bar{y} \vee z) \wedge (\bar{x} \vee \bar{y} \vee t) \wedge (y \vee \bar{z} \vee t) \wedge (x \vee z \vee \bar{t})$$

x est vrai ou faux

x vrai \iff \bar{x} faux

Peut-on affecter des valeurs vrai ou faux aux variables de telle façon que l'expression soit vraie ?

Exemple

une solution: x=vrai y=faux t=vrai z=vrai

Le théorème de Cook

**Stephen Cook
a classé le
problème SAT
comme *NP-complet***

**SAT est le premier problème
NP-complet connu**

Voulez-vous gagner 1 000 000 \$?

Prix Clay

Il "suffit" de démontrer la **conjecture suivante**

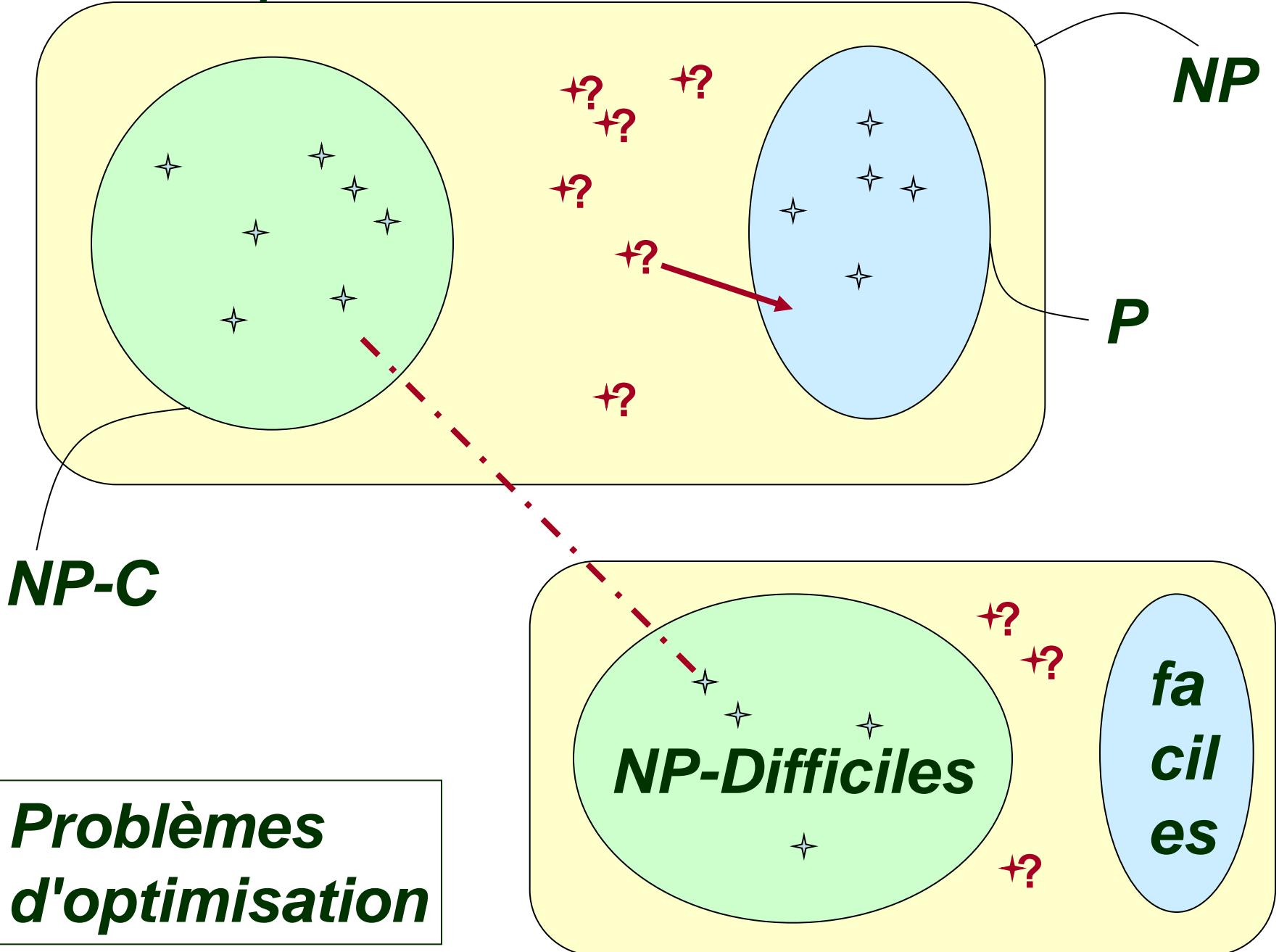
$$P \neq NP$$

(ou bien de prouver que

$$P = NP$$

Pour prouver que $P = NP$ il faudrait résoudre l'un des problèmes NP -complets avec un algorithme polynomial. Faire "tomber" un seul de ces problèmes dans la classe P ferait tomber l'ensemble de la classe NP

problèmes de décision



Les problèmes d'optimisation sont plus difficiles que les problèmes de décision.

Problème d'optimisation : trouver un stable max

Problème de décision :
existe-t-il un stable de taille au moins k

Si on sait résoudre (polynomialement)
le problème d'optimisation ,
on sait a fortiori résoudre le problème de décision

*Tous ces problèmes
(de décision ou d'optimisation)
présentent des enjeux industriels et
économiques très importants:
production, cryptographie,
écologie...*

*Beaucoup sont des problèmes de
graphes:
chemins, flots, ordonnancements,
planification, coloration, ...*

3

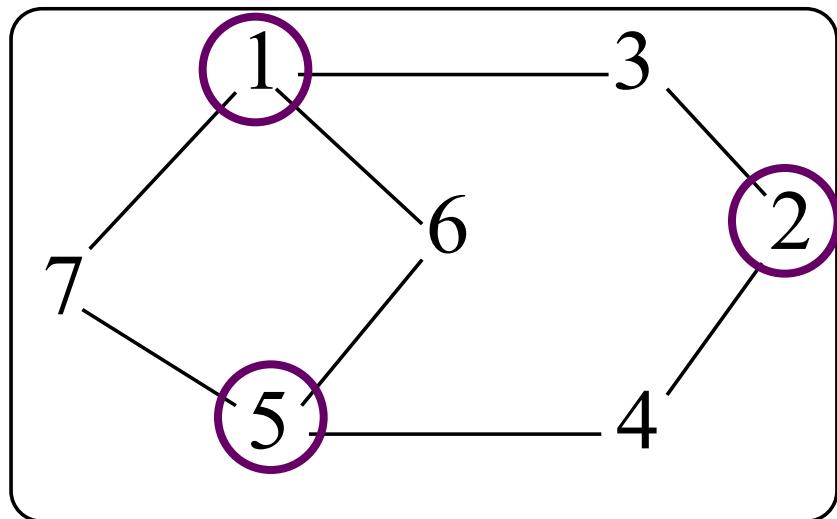
GRAPHES ET SOUS-GRAPHES PARTICULIERS

Graphes sans boucle

Stabilité

définition :

soit G un graphe orienté ou non orienté,
un ensemble stable de G est un sous-ensemble S
de sommets tel que deux sommets de S ne sont
jamais voisins.



$$S' = \{1, 2\} \text{ stable}$$

$$S'' = \{1, 2, 5\} \text{ stable maximal}$$

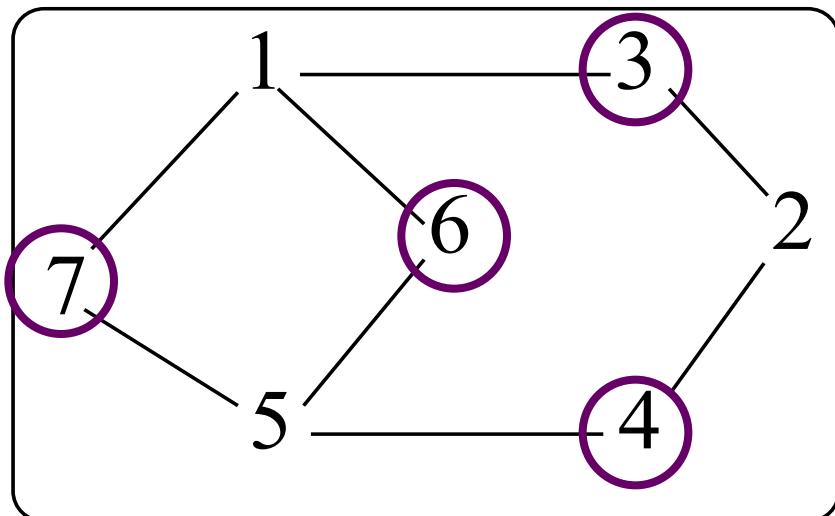
Stabilité

ensemble stable :

$$S \cap \Gamma(S) = \emptyset$$

nombre de stabilité :

$\alpha(G)$ = cardinal du plus grand ensemble stable



$$\alpha(G)=4$$

$S''= \{3,4,6,7\}$
stable de cardinal max

Stabilité

Théorème :

Dans un graphe de n sommets et de degré maximal h , la cardinalité de tout ensemble stable maximal est supérieure ou égale à $[n/h+1]^+$

$[n/h+1]^+$ est la partie entière supérieure de $n/h+1$

Graphe complet

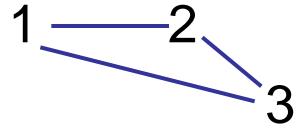
graphe simple avec toutes les arêtes

K_n : n sommets et $n(n-1)/2$ arêtes

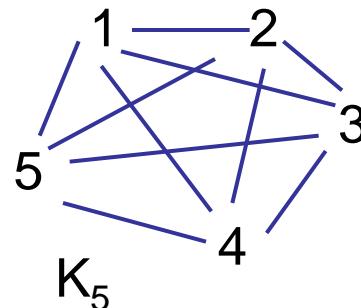
1 K_1 un sommet et aucune arête



K_2



K_3



K_5

Degré des sommets de K_n ?

Taille d'un stable max. de K_n ?

Clique

Définition :

Soit un graph G non orienté,

Une clique de G est un sous-graphe complet de G

Partition des sommets en cliques

$$P = \{C_1, C_2, \dots, C_k\}$$

C_1, C_2, \dots, C_k cliques de G tel que $X(C_i) \cap X(C_j) = \emptyset$

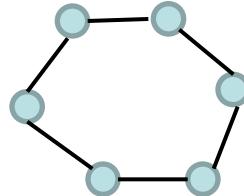
$X(C_1) \cup X(C_2) \cup \dots \cup X(C_k) = X(G)$ = sommets de G

Théorème:

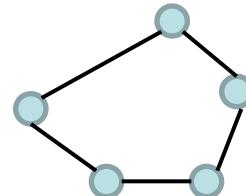
soit un stable S de G et P une partition en cliques des sommets de G , alors $|S| \leq |P|$.

Si $|S|=|P|$ alors

S est un stable maximum et P une partition minimum



$$|S|=|P|=3$$



$$|S|=2 \quad |P|=3$$

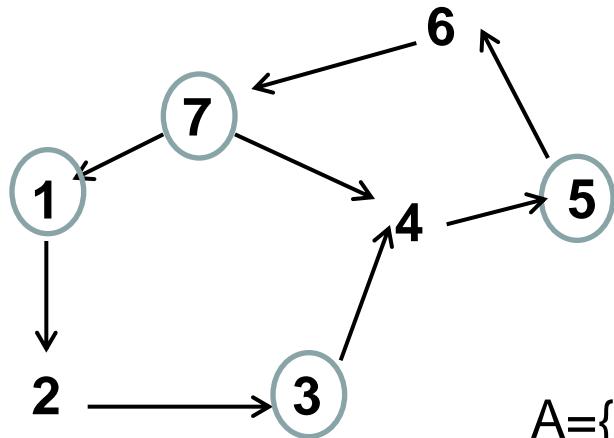
Ensemble absorbant

Définition : soit un graphe G orienté

Un ensemble absorbant de G est un sous-ensemble A de sommets de G tel que tout sommet n'appartenant pas à A a un successeur dans A

nombre d'absorption :

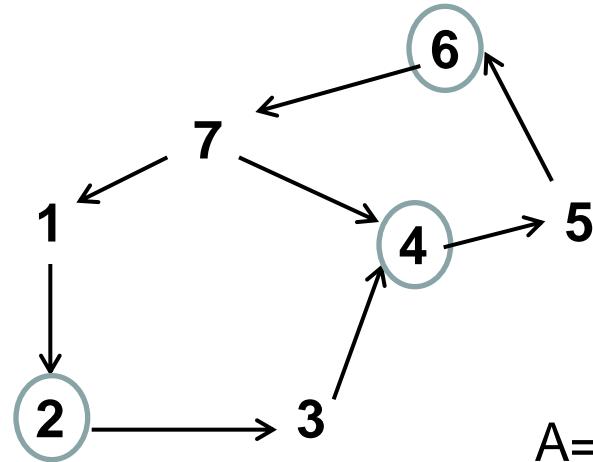
$\beta(G)$ = cardinal du plus petit ensemble absorbant



$A = \{1, 3, 5, 7\}$ est absorbant
2 a un successeur dans A
4 a un successeur dans A
6 a un successeur dans A

A est minimal : si on retire un sommet de A il n'est plus absorbant

Est-il minimum ?



$A = \{2, 4, 6\}$ est absorbant
 1 a un successeur dans A
 3 a un successeur dans A
 5 a un successeur dans A
 7 a un successeur dans A

A est minimum : il n'existe pas d'absorbant plus petit (de taille 2)

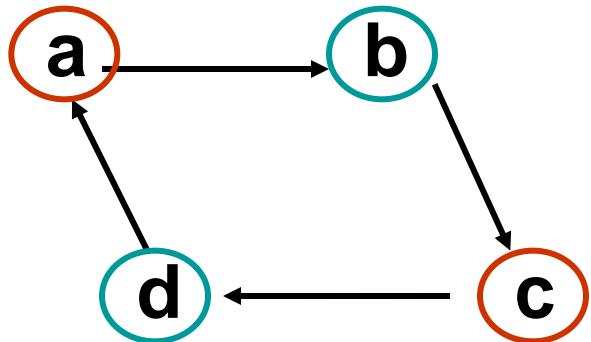
$$\beta(G) = 3$$

Noyau

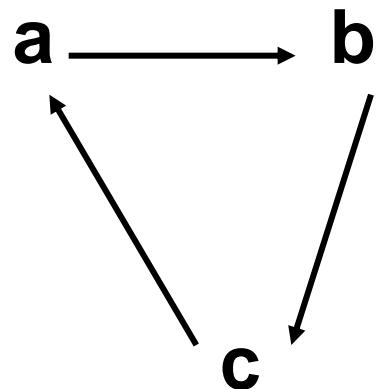
Définition : soit un graphe G orienté

Un noyau de G est un sous-ensemble de sommets de G qui est à la fois stable et absorbant

Noyau



Graphe à 2 noyaux



Graphe sans noyau

Fonction de Grundy

Définition : soit un graphe G orienté

g est une fonction de Grundy de G si

g est une application de X dans \mathbb{N} telle que

**$g(x)$ est le plus petit entier non attribué aux
successeurs de x**

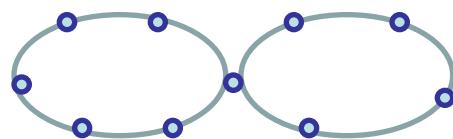
Fonction de Grundy

Propriété : soit un graphe G orienté

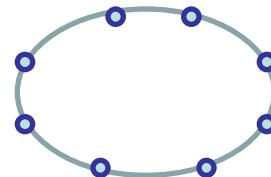
Si G admet une fonction de Grundy,
l'ensemble des sommets de G tels que $g(x)=0$
forme un noyau de G

Cycles

Cycle : chaîne qui ne contient pas deux fois le même arc et dont le sommet initial et le sommet terminal coïncident.



Cycle élémentaire: cycle qui passe une fois au plus par chaque sommet.



Cocycles

A ensemble de sommets

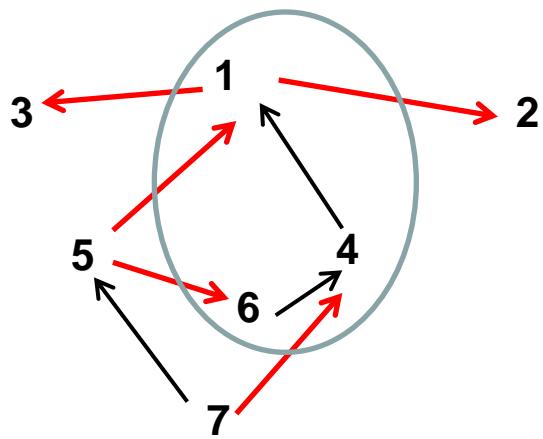
$\omega^+(A)$ =arcs sortant de A

$\omega^-(A)$ =arcs entrant dans A

$\omega(A) = \omega^+(A) \cup \omega^-(A)$

Cocycle : ensemble d'arcs de la forme $\omega(A)$

Cocycle élémentaire: cocycle minimal au sens de l'inclusion



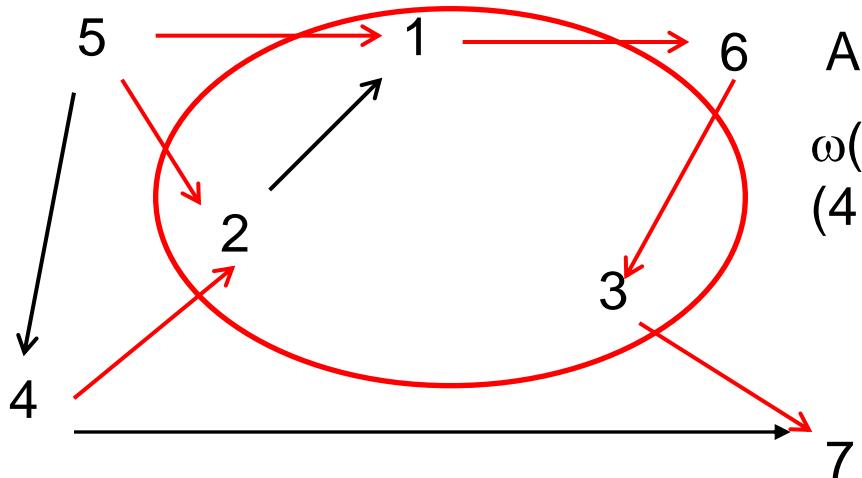
Cocycle

$A = \{1, 6, 4\}$, $\omega^+(A) = \{(1, 2), (1, 3)\}$, $\omega^-(A) = \{(5, 1), (5, 6), (7, 4)\}$

$$\omega(A) = \omega^+(A) \cup \omega^-(A)$$

Cycle = { (7,5) , (5,6) , (6,4) , (4,7) }

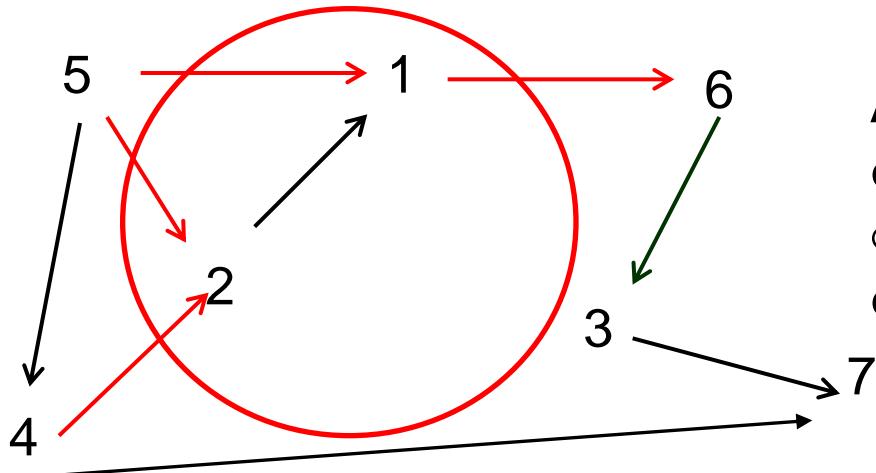
L'arc (7,4) parcouru à l'envers \Rightarrow on lui affecte le signe -



$$A = \{1, 2, 3\}$$

$$\omega(A) = (1, 6), (3, 7), (5, 1), (5, 2), (4, 2), (6, 3)$$

$\omega(A)$ pas minimal
on peut retirer (3,7)
et (6,3)



$$A' = \{1, 2\}$$

$$\begin{aligned} \omega(A') &= (5, 1), (5, 2), (4, 2), (1, 6) \\ &\subset \omega(A) \\ \omega(A') &\text{ minimal} \end{aligned}$$

Vecteur caractéristique de cycle

vecteur de cycle C : vecteur μ indexé par les arcs t.q.

- $\mu_a = 1$ si l'arc a est orienté dans le sens de parcours de C
- $\mu_a = -1$ si l'arc a est orienté dans le sens contraire du parcours de C
- $\mu_a = 0$ si l'arc a n'est pas dans C

Vecteur caractéristique de cocycle

vecteur de cocycle $\omega(A)$: vecteur ω indexé par les arcs t.q.

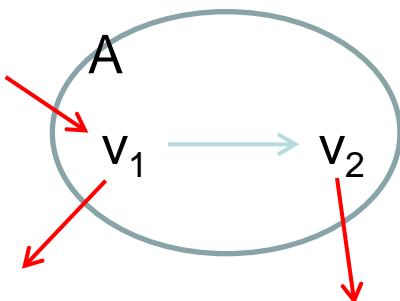
$\omega_a = 1$ si l'arc a est orienté vers l'extérieur de A

$\omega_a = -1$ si l'arc a est orienté vers l'intérieur de A

$\omega_a = 0$ si l'arc a n'est pas dans $\omega(A)$

Propriété: vecteurs de cocycles et de cycles sont \perp

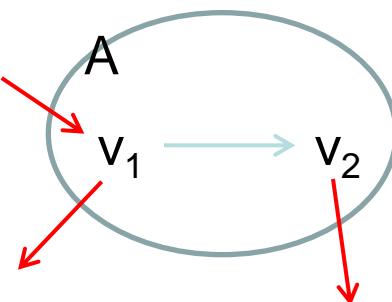
1. Un vecteur de cocycle $w(A)$ peut toujours se décomposer en une somme $\sum_{v \in A} w(v)$



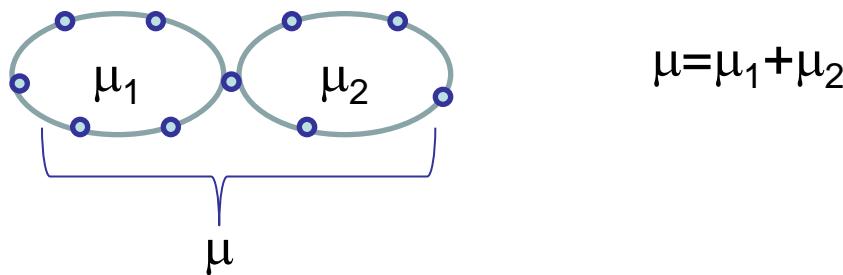
L'arc sort de v_1 est compté +1
Quand il rentre dans v_2 compté -1
Seule reste la contribution des arcs sortant ou rentrant dans A

Propriété: vecteurs de cocycles et de cycles sont \perp

1. Un vecteur de cocycle $w(A)$ peut toujours se décomposer en une somme $\sum_{v \in A} w(v)$



2. Un vecteur de cycle μ peut toujours se décomposer en une somme de vecteurs de cycles élémentaires



Propriété: vecteurs de cocycles et de cycles sont \perp

3. Faisons le produit scalaire d'un vecteur de cycle et vecteur de cocycle

On peut déjà considérer un cycle élémentaire μ

Le cocycle se décompose en $\sum_{v \in A} w(v)$. Dans le produit scalaire, seuls les sommets v du cycle élémentaire μ interviendront :

$$\langle \mu | \sum_{v \in A} w(v) \rangle = \sum_{v \in \mu \cap A} \langle \mu | w(v) \rangle = 0$$



Dans le cycle μ les arcs sont comptés +1
Dans le cocycle $w(v)$ l'un est compté -1 l'autre +1
Dans le produit scalaire $-1 \times 1 + 1 \times 1 = 0$



Dans le cocycle $w(v)$ les arcs sont comptés +1
Dans le cycle μ l'un est compté -1 l'autre +1
Dans le produit scalaire $-1 \times 1 + 1 \times 1 = 0$

Cycles et cocycles

Théorème : Soit G un graphe de n sommets, m arcs et p composantes connexes, la dimension d'une base de cycles est le nombre cyclomatique de G :

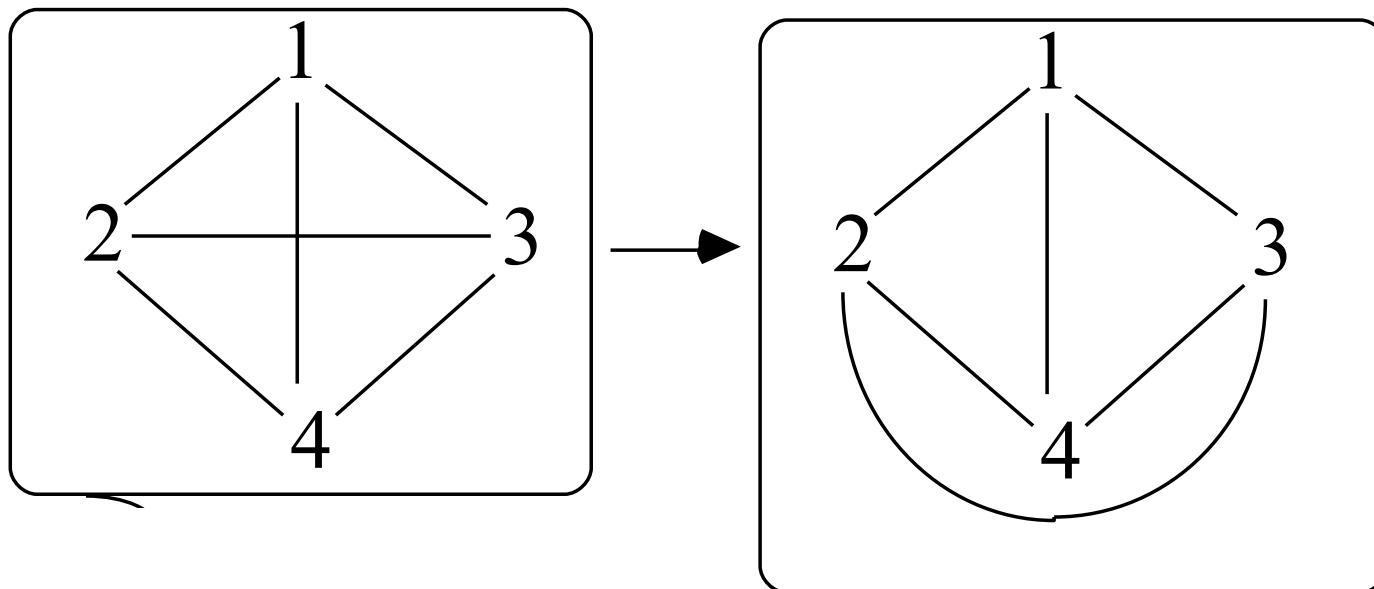
$$v(G) = m - n + p.$$

la dimension d'une base de cocycles est le nombre cocyclomatique de G :

$$\lambda(G) = n - p.$$

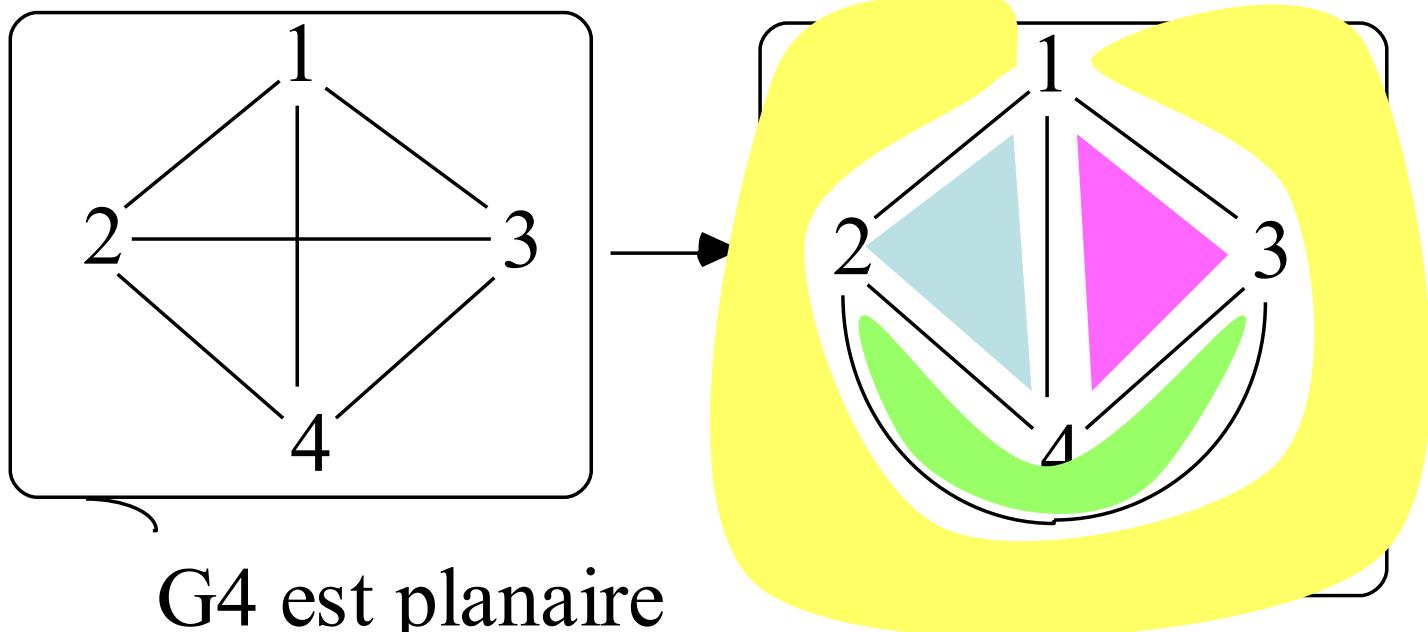
Planarité

graphe planaire: graphe qui peut être représenté sur un plan de sorte que 2 arêtes ne se rencontrent pas en dehors de leurs extrémités

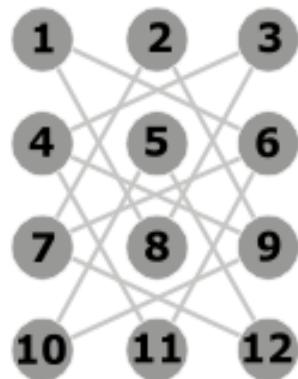


Planarité

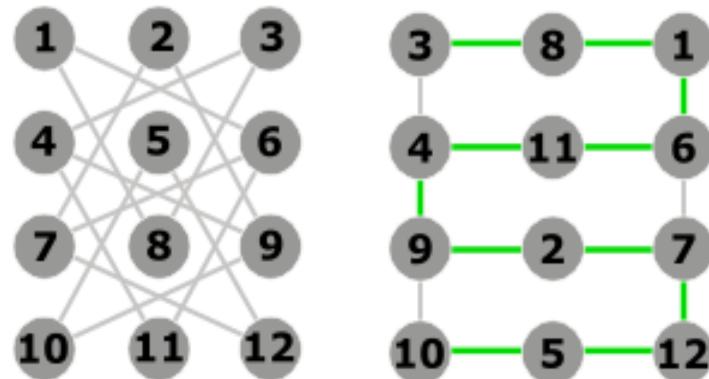
Face: région du plan limitée par des arêtes dont l'ensemble constitue la frontière + face infinie



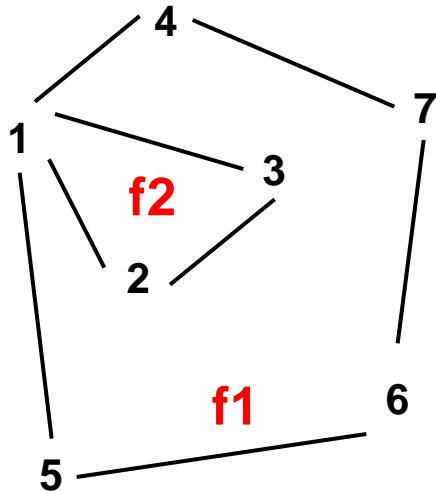
Ce graphe est-il planaire ?



Ce graphe est planaire



Frontière et contour



Graphe planaire topologique est une représentation planaire d'un graphe planaire

Contour de $f_1 = 1-4-7-6-5-1$

Frontière de $f_1=1-3-2-1-5-6-7-4-1$

La frontière dépend de la représentation pas le contour

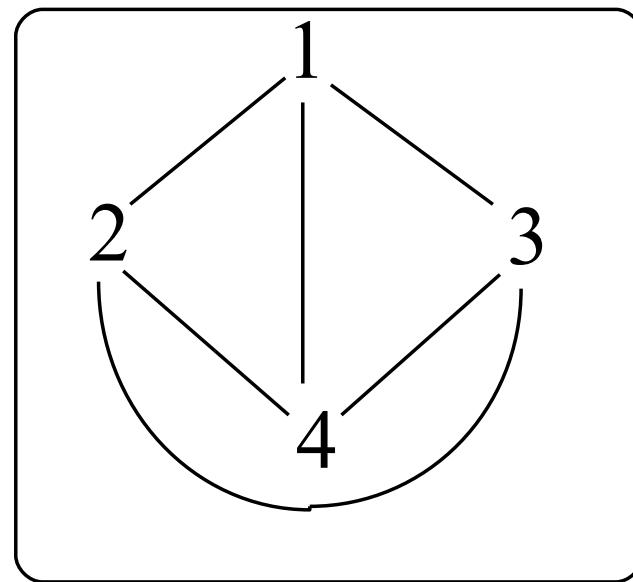
Les contours sont des **cycles élémentaires**

Le contour est le cycle élémentaire qui contient en son intérieur toutes les arêtes de la frontière

Théorème:
***dans un graphe planaire topologique,
les contours des faces finies
constituent une base de cycles***

Démonstration par récurrence sur le nombre de faces

Si le graphe n'est pas orienté
on oriente arbitrairement les arêtes



face: région du plan limitée par des arêtes (+ face infinie)

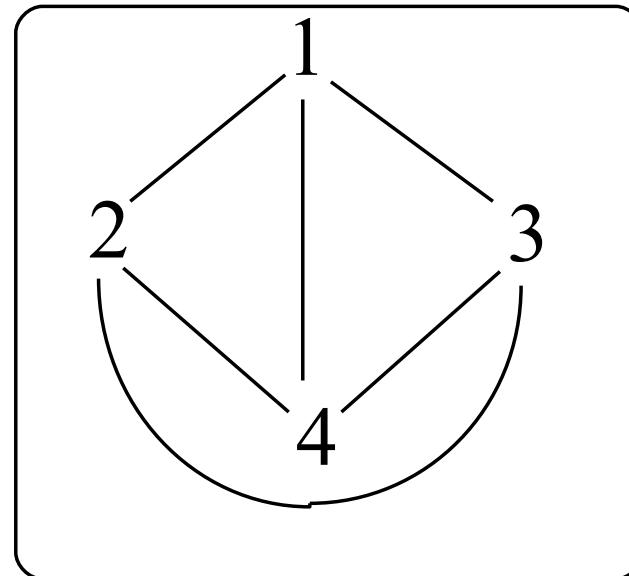
formule d'Euler: Dans un graphe planaire de n sommets, m arêtes et f faces, on a

$$n-m+f = 2$$

Exemple

nombre de faces

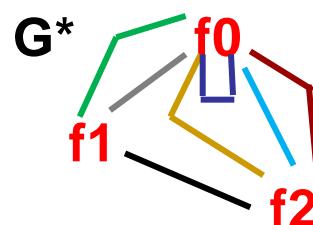
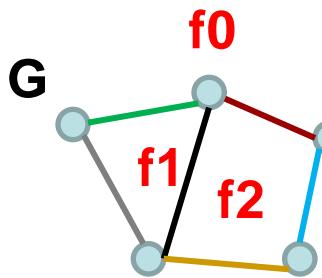
$$f = m-n+2 = 6-4+2 = 4$$



Graphe dual d'un graphe planaire topologique

G^* graphe dual d'un graphe G planaire topologique

- Sommets de G^* = faces de G
 - Arêtes de G^* = arêtes de G
- $a^* = (f^*, g^*)$ est une arête de G^* ssi les faces f et g de G partagent l'arête a



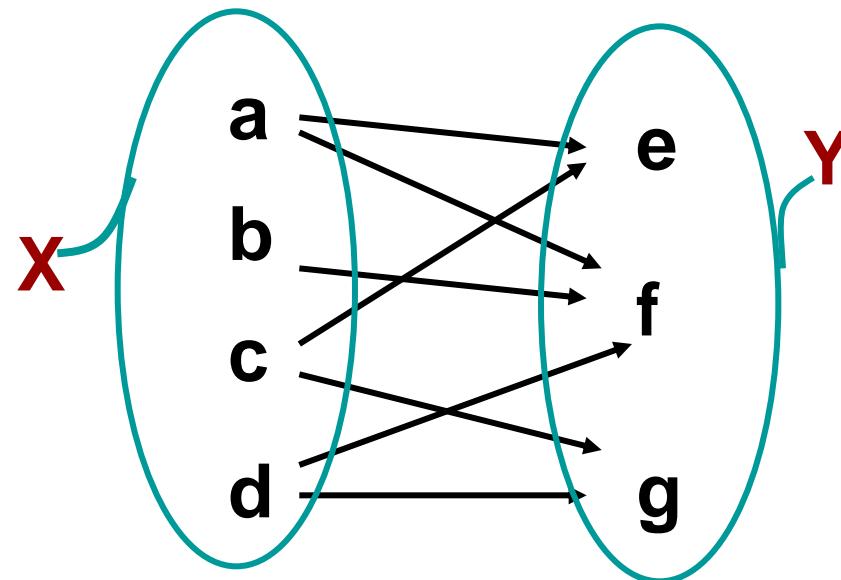
G^* est planaire

Graphe biparti

$G = (X, Y, U)$ tel que

$\forall (x, y) \in U$

$x \in X$ et $y \in Y$



Graphe face-arêtes associé à un graphe planaire G :

Graphe biparti $H=(F,A,E)$ tel que

$F=\{\text{faces de } G\}$

$A=\{\text{arêtes de } G\}$

$E=\{(x,y)\}, x \text{ dans } F \text{ et } y \text{ dans } A$

telle que y est sur le contour de $x\}$

graphe face-arêtes associé à un graphe planaire G

- Tracer $H=(F,A,E)$ le graphe face-arête de K_4
où F = faces et A = les arêtes de K_4

Dans le cas général, soit $H=(F,A,E)$ le graphe face-arête d'un graphe G planaire qui contient au moins une face finie (un cycle).
On note $f = |F|$ le nombre de faces de G et $m = |A|$ le nombre d'arêtes de G

- Quel est le degré min des sommets de F ?
- Quel est le degré max des sommets de A ?
- En déduire que $3f \leq |E| \leq 2m$

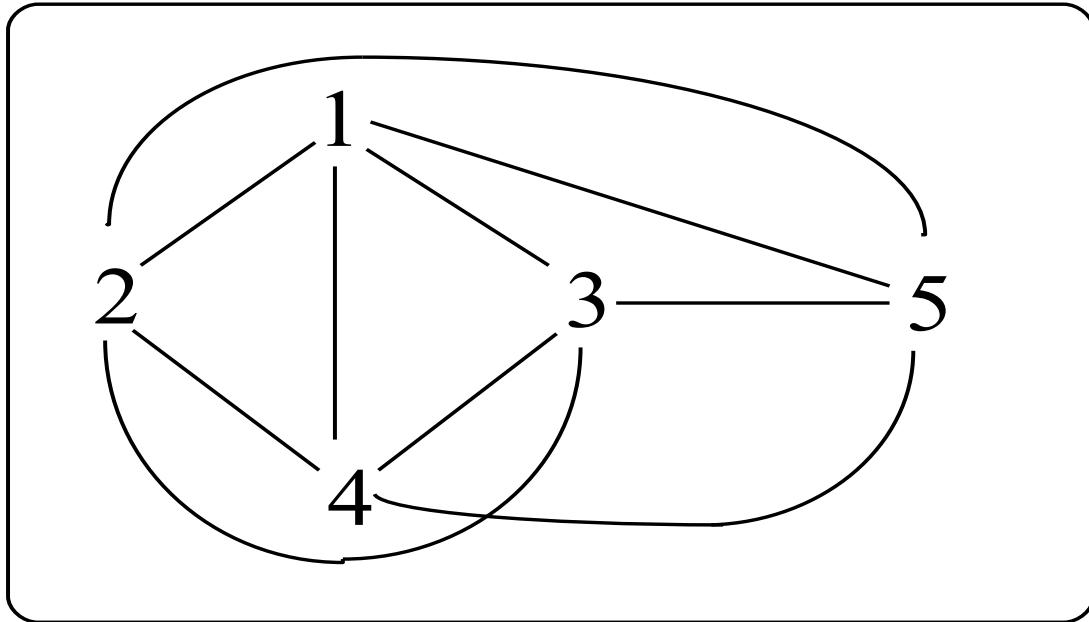
Théorème:

Si G est un graphe simple planaire connexe contenant au moins 1 face finie alors $m < 3n - 5$

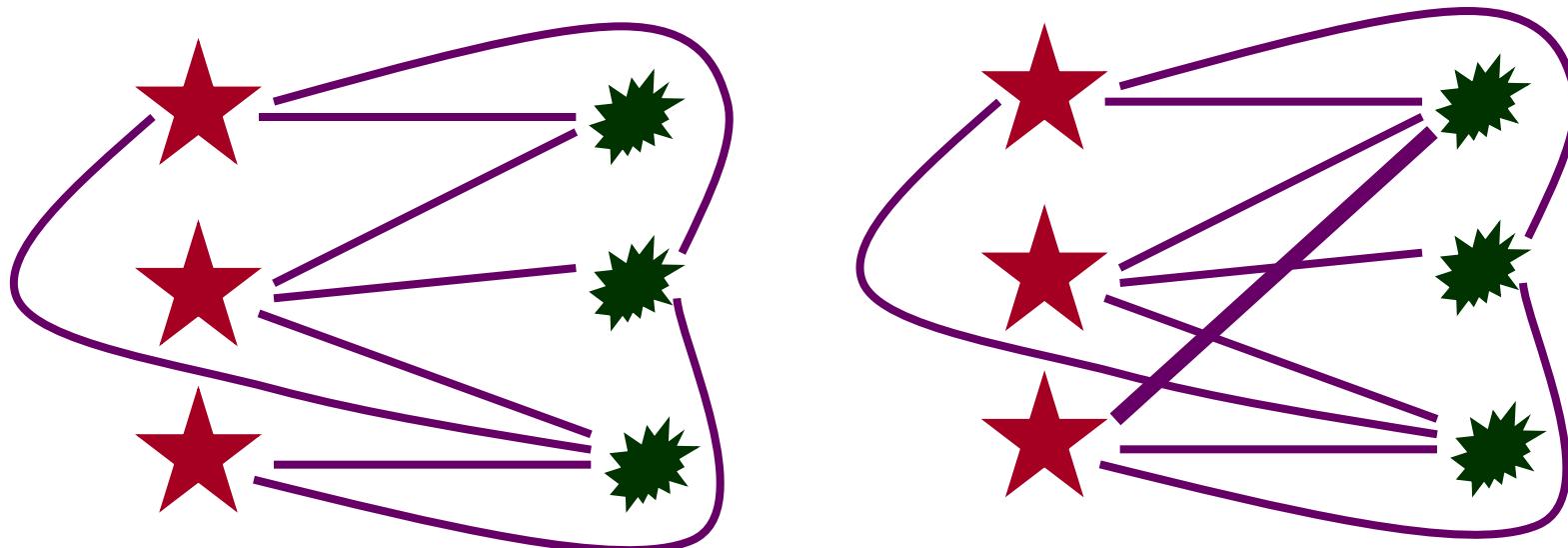
Théorème:

Dans un graphe simple planaire connexe contenant au moins 1 face finie il existe au moins un sommet x de degré $d(x) < 6$

Le graphe complet de 5 sommets n'est pas planaire



Le graphe $K_{3,3}$ n'est pas planaire.



Une subdivision d'un graphe est obtenue en rajoutant (éventuellement) des sommets sur les arêtes (ou les arcs) du graphe

Théorème de Kuratowski (1930):

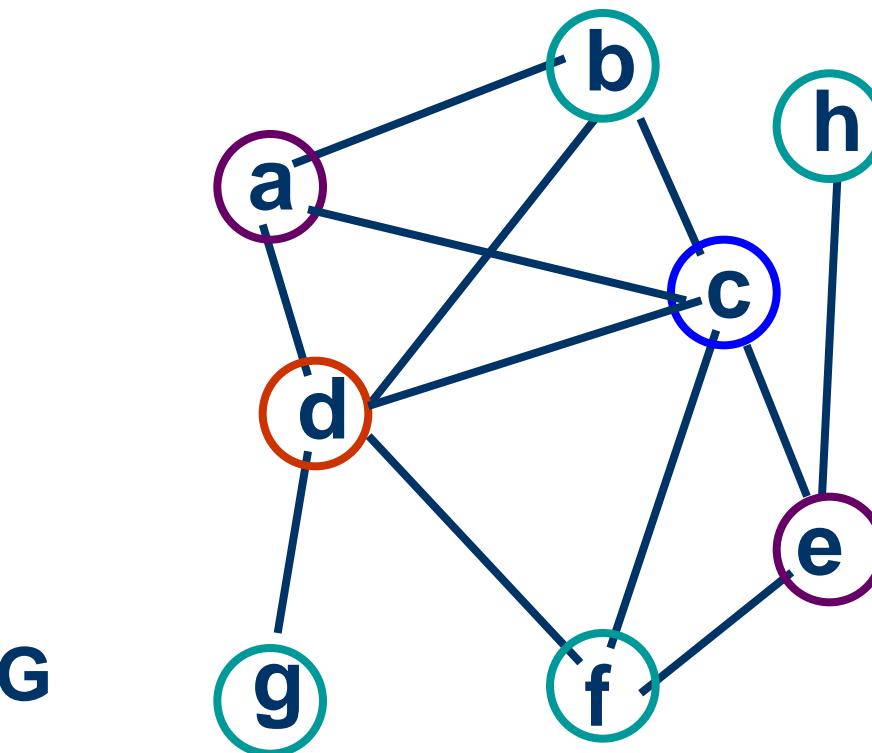
Un multigraphe est planaire si et seulement si il ne contient pas comme sous-graphe partiel une subdivision K_5 ou de $K_{3,3}$

Coloration

Colorier les sommets de G sans que 2 voisins aient la même couleur

Une couleur
=
un stable

G

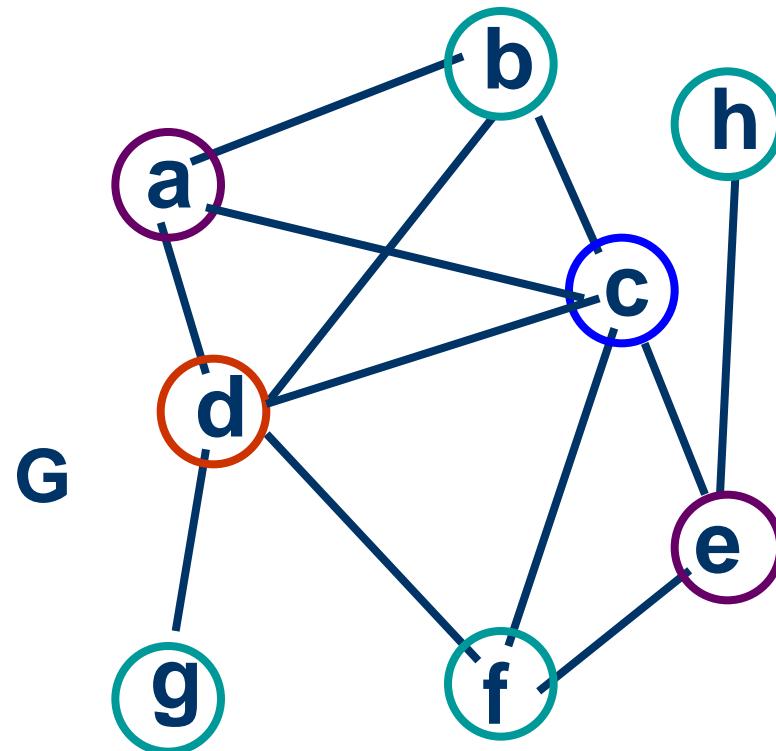


Coloration

nombre chromatique: $\gamma(G)$
plus petit nombre de couleurs
nécessaires pour colorier les sommets
de G sans que 2 voisins aient la même
couleur

Trouver $\gamma(G)$ est
un problème
"difficile"

Exemple
 $\gamma(G) = 4$



théorème des 4 couleurs:

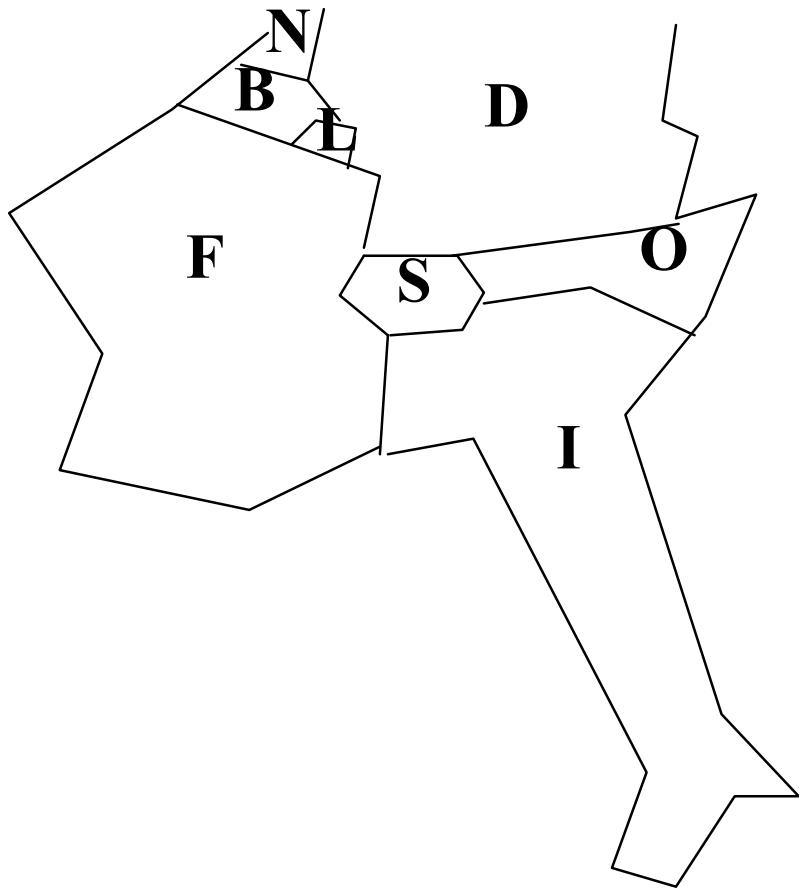
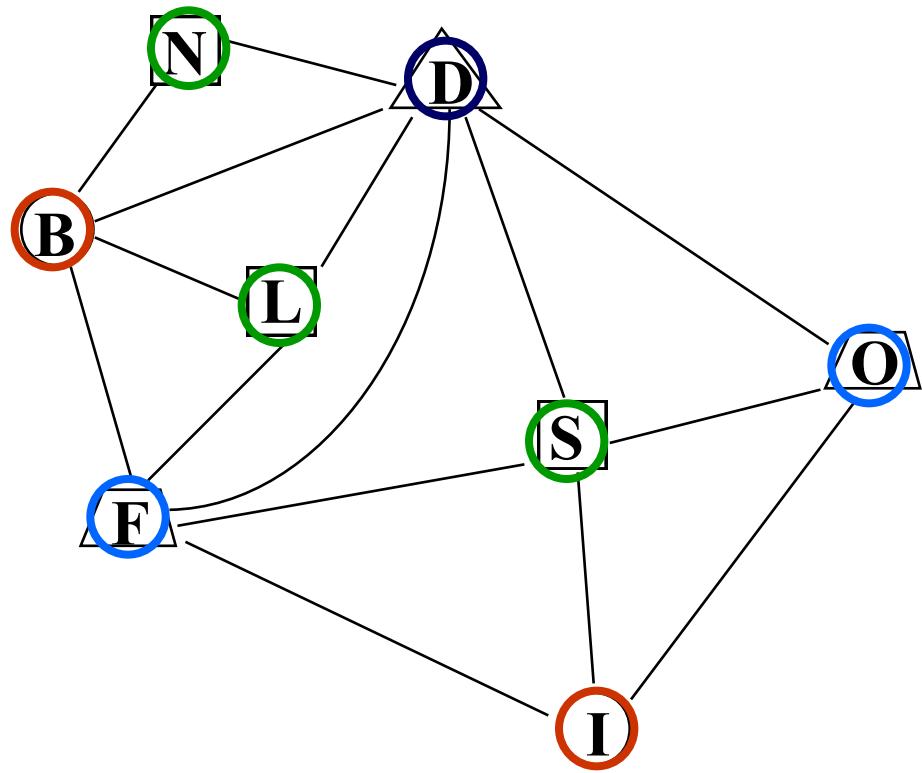
Si G est un graphe planaire alors

$$\gamma(G) \leq 4$$



Cette carte
peut être
coloriée avec
seulement
4 couleurs ou
moins.

Graphe associé
un pays = un sommet
une frontière=une arête



théorème des 4 couleurs:

Si G est un graphe planaire alors

$$\gamma(G) \leq 4$$

Conjecture formulée pour la première fois par l'Écossais *Francis Guthrie* en 1852. (Travaux de Paul Valéry).

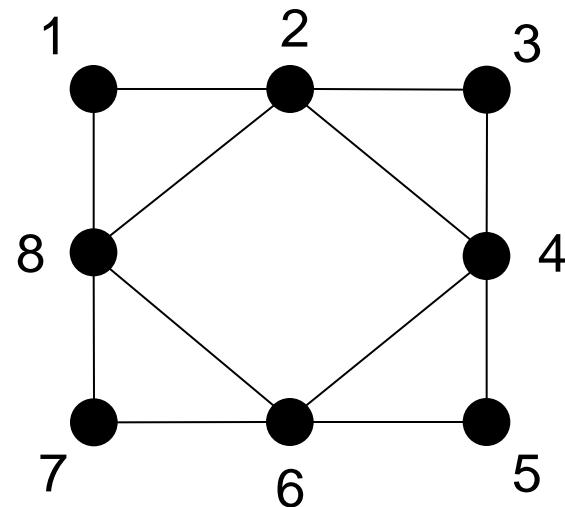
(pour la coloration de cartes de géographie
Preuve en... 1976: *K. Appel et W. Haken.*

Premier théorème de l'histoire des mathématiques qui a nécessité l'usage systématique de l'ordinateur

Un problème d'ordonnancement

ordonner n tâches sur k machines en le minimum de temps, certaines tâches ne pouvant être exécutées en parallèle (partage des ressources)

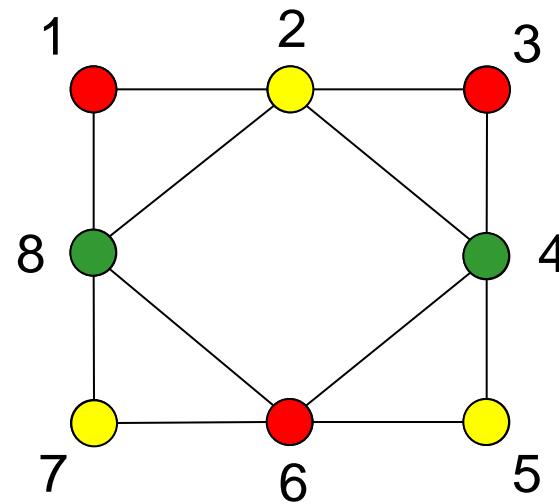
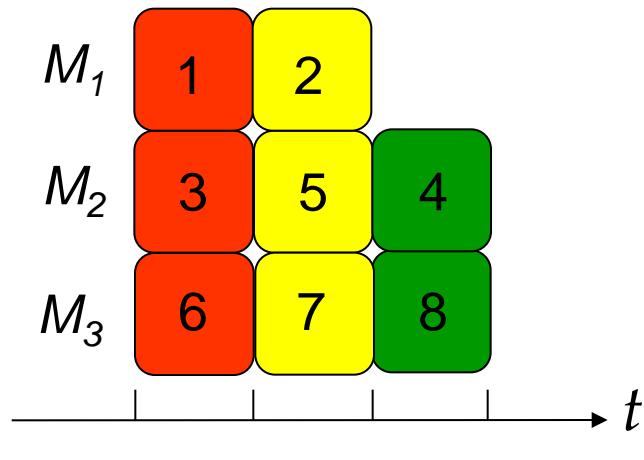
conflits entre les tâches → graphe d'exclusion mutuelle



*c'est un problème de coloration
lorsque toutes les tâches ont le même temps d'exécution*



ordonnancement = coloration du graphe tel que chaque couleur n'apparaisse pas plus de k fois



un algorithme « glouton » de coloration

Welsh-Powell

Début

- Soit $G' = G$
- numéroter les sommets par ordre de degrés décroissants: x_1, x_2, \dots, x_n
- *tant que* G' n'est pas vide *faire*
 - Sélectionner dans l'ordre de la liste tout sommet non voisin (dans G') d'un sommet déjà sélectionné;
 - Colorier de la même couleur tout les sommets sélectionnés;
 - Retirer tous les sommets sélectionnés de G' ;

fait;

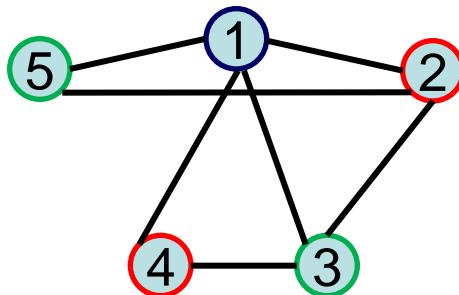
Fin;

Heuristique rapide mais non nécessairement optimale

Borne supérieure

Supposons les sommets numérotés par degré décroissant
 $d_1 \geq d_2 \geq \dots d_n$

Théorème: l'algorithme de Welsh-Powell colorie le graphe avec au plus $\max\{ \min\{k, d_k+1\} : k=1, \dots, n \}$ couleurs



$$\max \{ \min\{1,5\} \min\{2,4\} \ min\{3,4\} \ min\{4,3\} \ min\{5,3\} \ } = 3$$

algorithme relier-contraction

Principe

Soit un graphe G et 2 sommets a et b non adjacents

Relier $\rightarrow G^R = G + [a,b]$

Contraction $\rightarrow G^C$: un seul sommet ab relié à
 $G(a) \cup G(B)$

Coloration de G

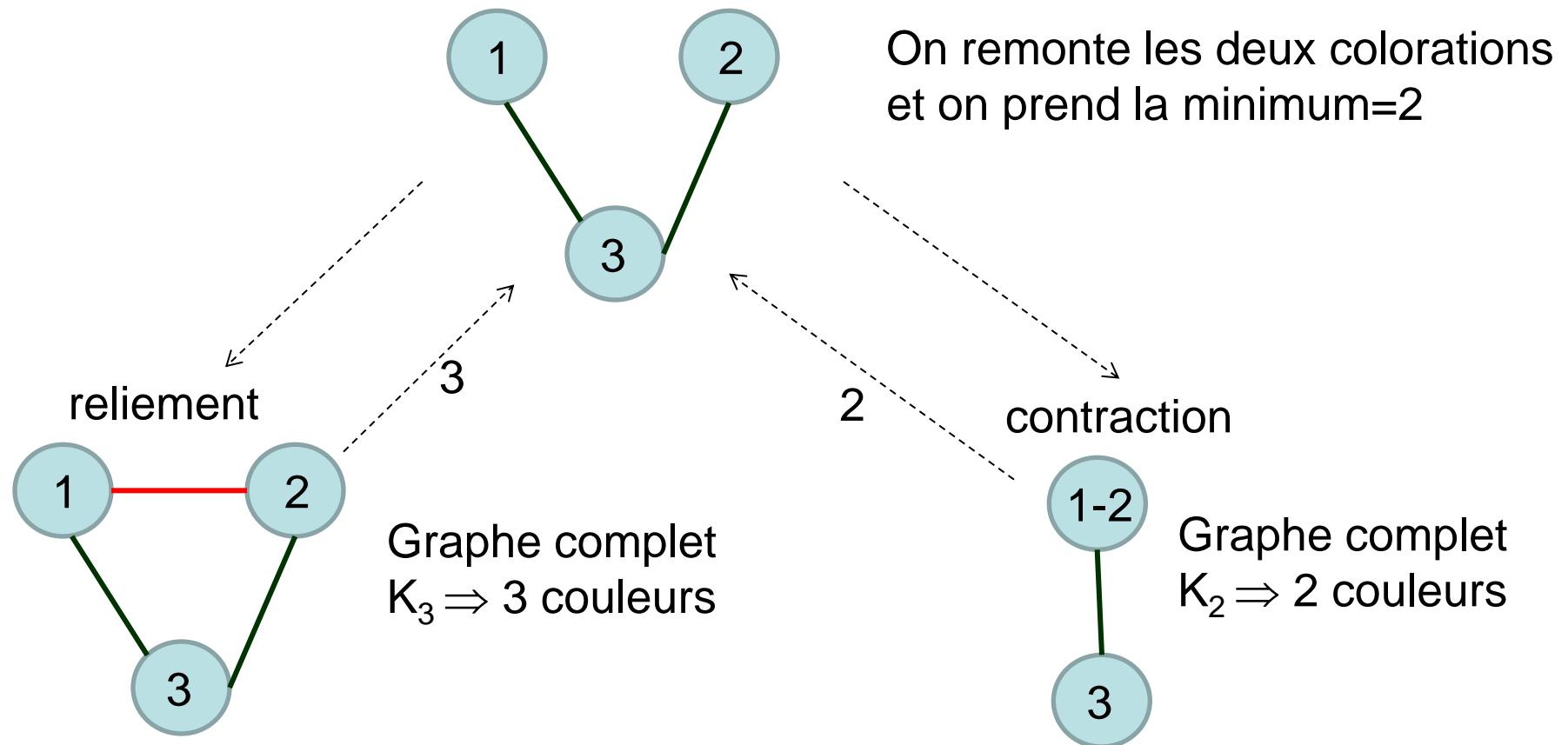
a et b même couleur:
coloration de G^C

a et b de couleurs différentes:
coloration de G^R

Contracter et/ou relier tant que graphe pas complet

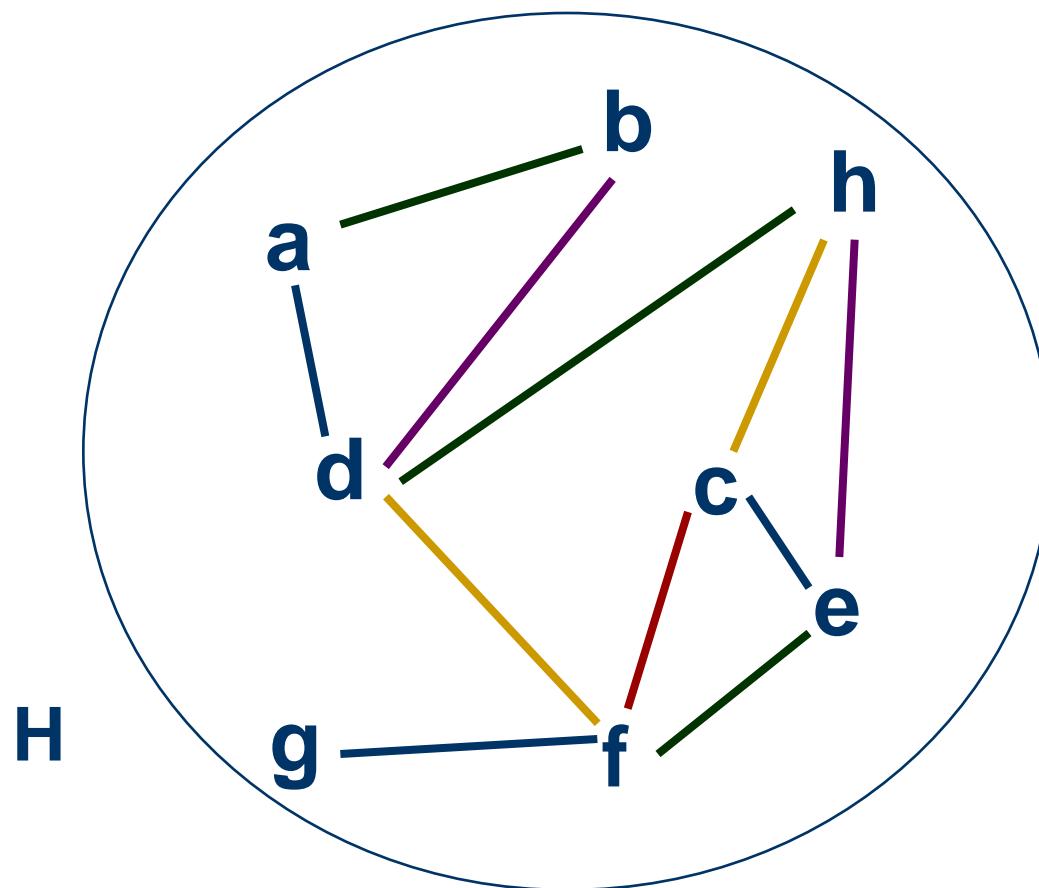
Algorithme optimal mais exponentiel

algorithme reliemment-contraction



Coloration des arêtes

2 arêtes adjacentes n'ont pas la même couleur



Coloration des arêtes

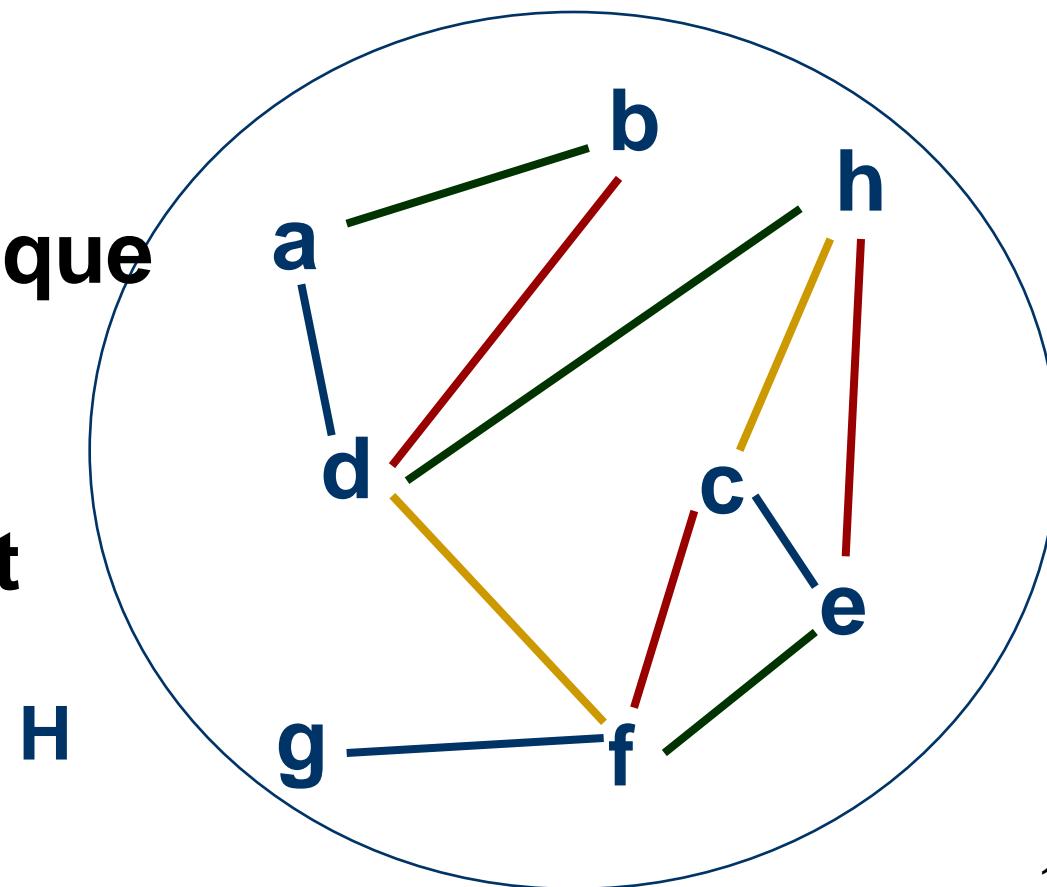
Indice chromatique:

Nombre minimal de couleurs pour les arêtes

**Exemple:
indice chromatique**

$$\gamma'(H) = 4$$

**il y a un sommet
De degré 4**



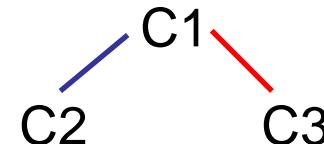
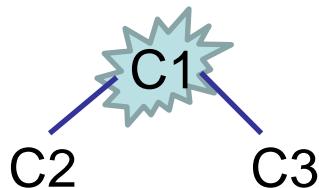
Allocation de fréquences

Réseau de capteurs

- Les capteurs pas trop éloignés communiquent 2 à 2
- A chaque paire de capteurs pas trop éloignés est alloué une fréquence sur laquelle ils communiquent
- Il faut des fréquences différentes pour éviter les interférences

On veut utiliser un **minimum de fréquences**

Même fréquence \Rightarrow interférences quand
C2 et C3 communiquent en même temps avec C1



On doit affecter des fréquences différentes

Coloration des arêtes

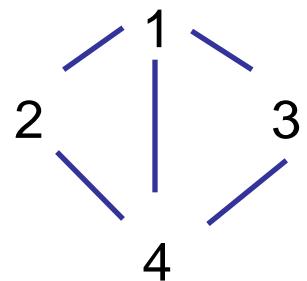
« Line graph » ou « graphe des arêtes »

Graphe $L(G)=(Y,B)$ associé à $G=(X,A)$

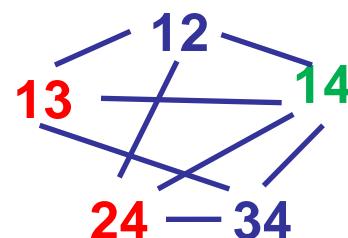
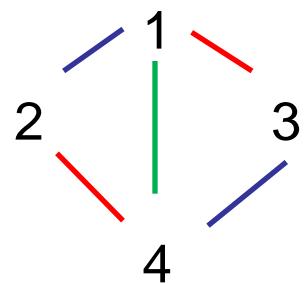
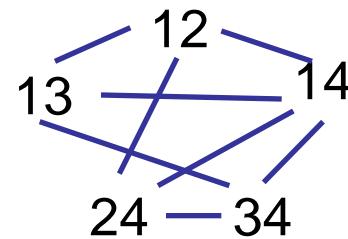
- $Y = A$ (un sommet par arête de G)
- $[a,b]$ arête de L si et seulement si a et b ont une extrémité commune dans G

L'indice chromatique de G est égal au nombre chromatique de $L(G)$

Graphe G



Graphe des arêtes de G



Coloration arêtes \Leftrightarrow Coloration sommets

Coloration des arêtes

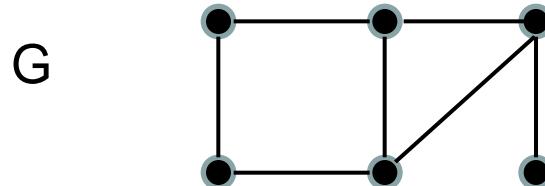
**La recherche de l'indice chromatique
d'un graphe est un problème difficile**

**Mais on sait qu'il est égal soit à h soit à $h+1$
où h est le degré maximal du graphe (Th.de Vizing 1964)**

Couplage d'un graphe G

sous ensemble d'arêtes de G tq. 2 arêtes quelconques sont non adjacentes

Couplage maximum = couplage de cardinal maximum

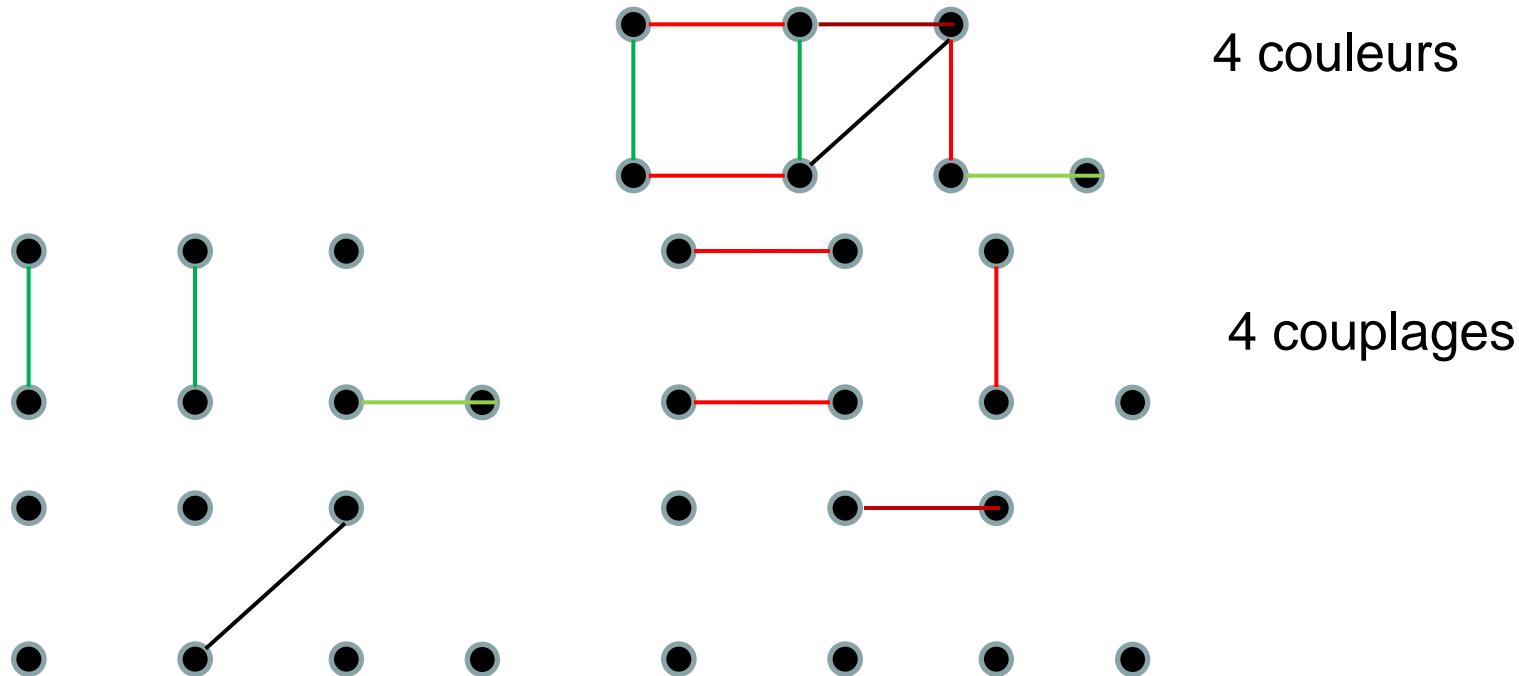


2 couplages maximum



Coloration des arêtes et couplages

Une coloration des arêtes revient à faire une partition des arêtes en couplages
Un couplage = une couleur



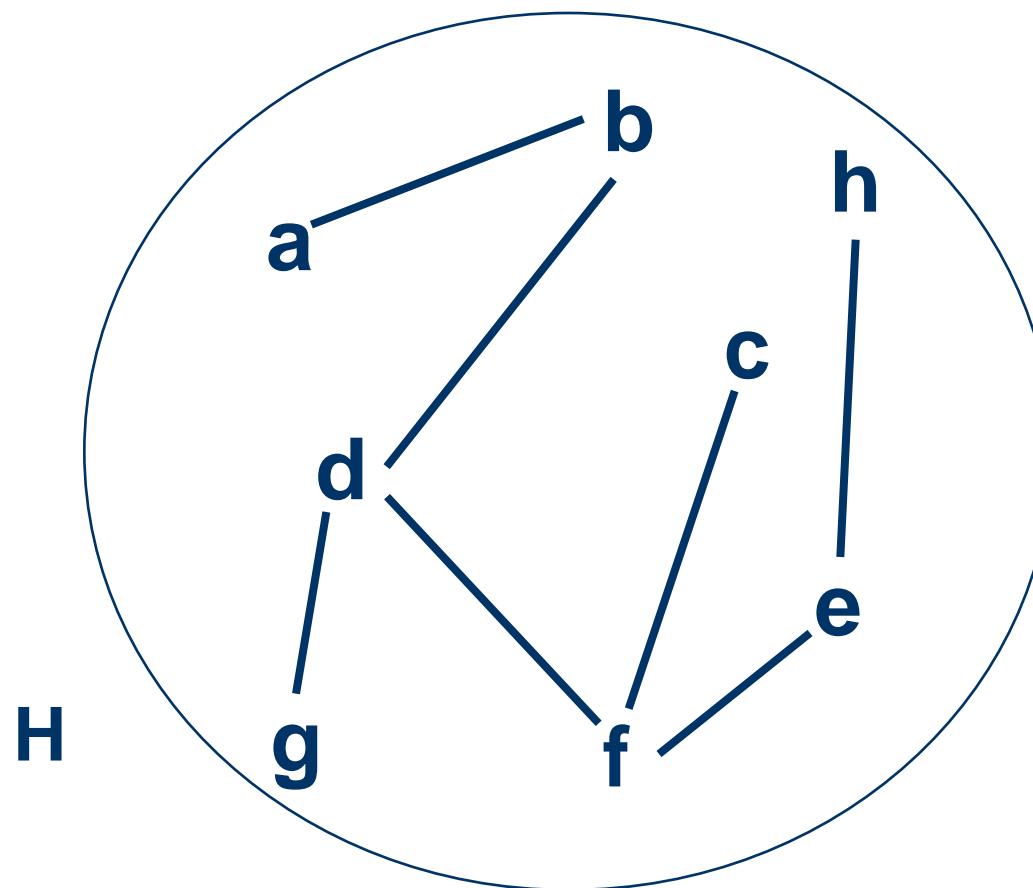
4

ARBRES

ARBRE COUVRANT D'UN GRAPHE (Algorithmes gloutons)

Arbres

arbre: Graphe connexe et sans cycle



arbre: définitions équivalentes

$H = (X, U)$, n sommets ($n \geq 2$)

- (1) H connexe et sans cycle
- (2) H sans cycle et $n-1$ arêtes
- (3) H connexe et $n-1$ arêtes
- (4) H sans cycle et en ajoutant une arête on crée un et un seul cycle
- (5) H connexe et si on supprime une arête , il n'est plus connexe
- (6) une chaîne et une seule entre toute paire de sommets

Démonstration

Rappel

nombre cyclomatique : $v(G) = m - n + p$

1⇒2 H connexe et sans cycle

\Rightarrow H sans cycle et $n-1$ arêtes

2⇒3 H sans cycle et $n-1$ arêtes

\Rightarrow H connexe et $n-1$ arêtes

3⇒4 H connexe et n-1 arêtes

⇒ H sans cycle et en ajoutant
une arête on crée un et un
seul cycle

**4⇒5 H sans cycle et en ajoutant une arête
on crée un et un seul cycle**

⇒ H connexe et si on
supprime une arête , il n'est
plus connexe

Démonstration (fin)

5⇒6 H connexe et si on supprime une

arête , il n'est plus connexe

⇒ une chaîne et une seule

**entre toute paire de
sommets**

6⇒1 une chaîne et une seule entre toute

paire de sommets

⇒ H connexe et sans cycle

Propriétés d'un arbre H

de m arêtes et n sommets

- **H sans cycle**
- **H est connexe**
- **H a $m=n-1$ arêtes**
- **En ajoutant une arête à H on crée un et un seul cycle**
- **Si on supprime une arête de H, H n'est plus connexe**
- **Il y a dans H une chaîne et une seule entre toute paire de sommets**

Théorème

Un graphe G admet un graphe partiel qui est un arbre si et seulement si il est connexe

Construction d'une base de cycle d'un graphe

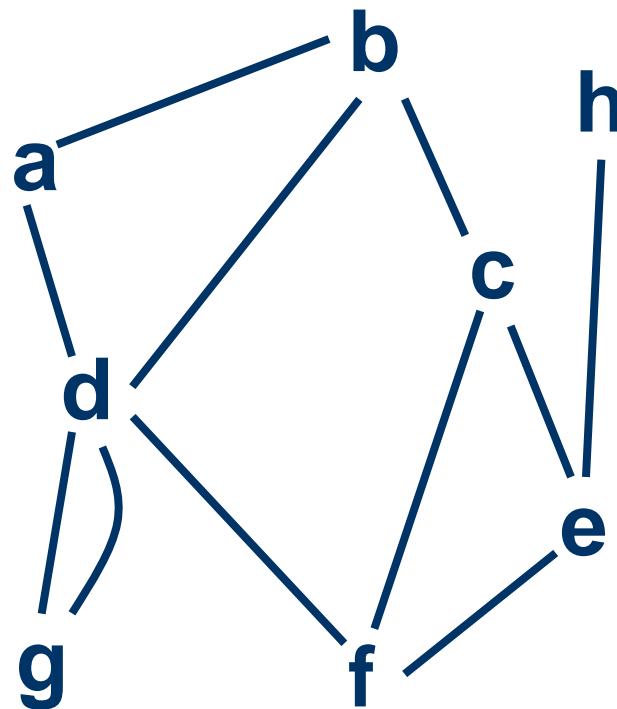
Soit un graphe G connexe et H un arbre couvrant de G .

Les cycles construits pas ajout des arêtes de G n'appartenant pas à H (une par une) forment une base de cycles.

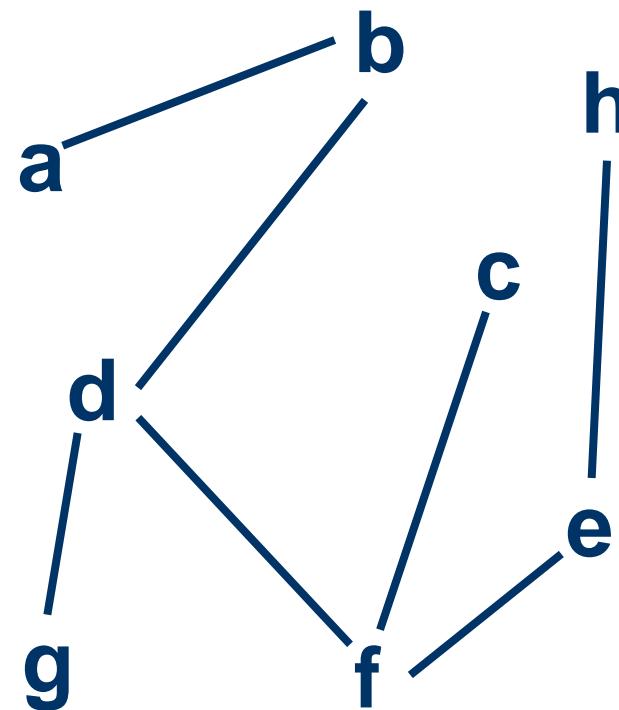
LE PROBLEME DE L'ARBRE

COUVRANT MINIMAL

Comment relier des objets avec un nombre minimal de liens (choisis dans un ensemble de liens donnés)?



G:graphe des liens possibles



H: arbre couvrant de G

PROBLEMES D'OPTIMISATION

un problème → **plusieurs solutions**

une solution → **une valeur**

problème d'optimisation:

**recherche d'une solution de valeur optimale
(min ou max)**

algorithme de résolution:

- **reconnaissance d'une solution**
- **évaluation d'une solution**
- **sélection d'une des meilleures solutions**

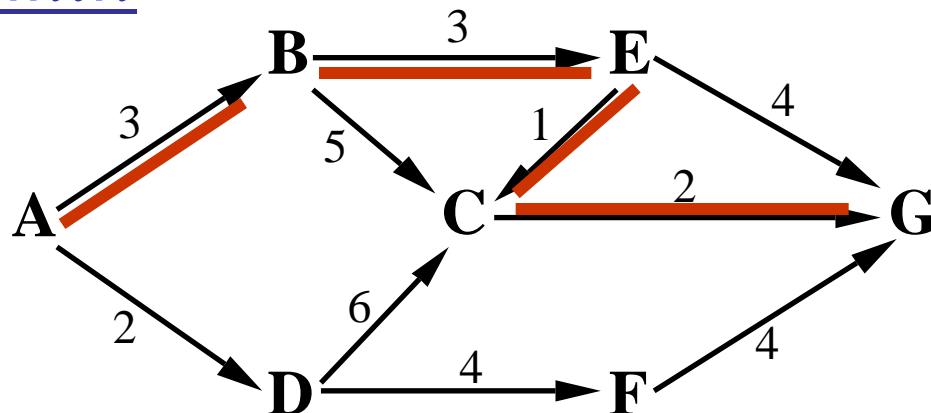
problèmes d'optimisation

→ faciles

→ difficiles

EXEMPLES

E1) chemin min



problème facile

(v=9;ABECG)

E2) sac-à-dos (en nombres entiers)

problème "assez" difficile

aliments	A	B	C
poids unitaire hg	6	4	3
valeur nutritive unitaire	14	10	6
poids max des aliments: 17			

2 solutions optimales:

$$1.A + 2.B + 1.C \rightarrow v = 40 \quad (p=17)$$

$$4.B \rightarrow v = 40 \quad (p=16)$$

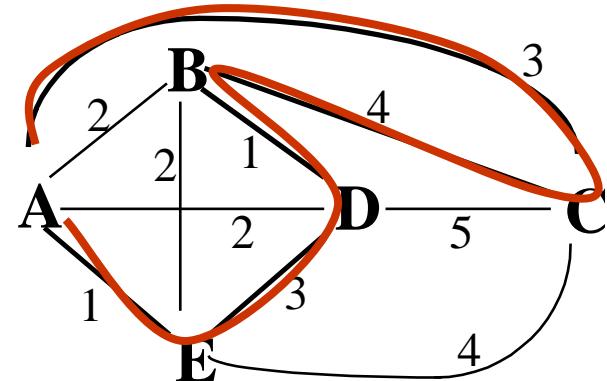
E3) voyageur de commerce

problème difficile

ici, 2 solutions optimales

ACBDEA et ACEBDA

$v = 12$



nombre de solutions: 24 ($n-1!$)

problème de petite taille

→ énumération possible

problème de grande taille

→ énumération impossible

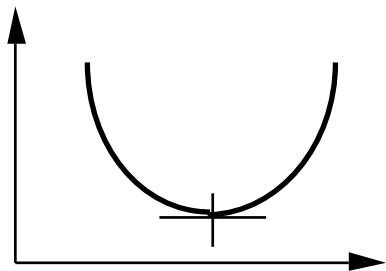
**INTERDIT D'ÉNUMÉRER DANS LE
COMBINATOIRE**

optimisation discrète \neq optimisation continue

ALGORITHMES GLOUTONS

choix glouton = choix localement optimal

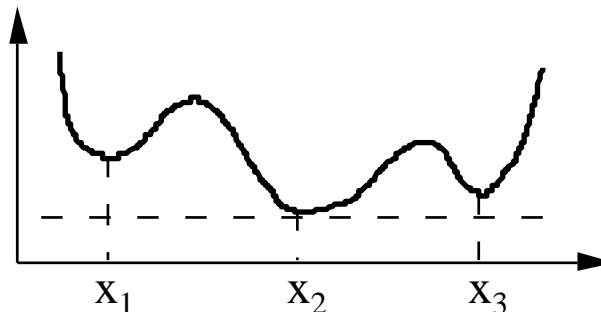
optimum local \neq optimum global



**fonction concave
(ou convexe)**

**optimum local =
optimum global**

continu \neq entier



x_1 et x_3 : optima locaux

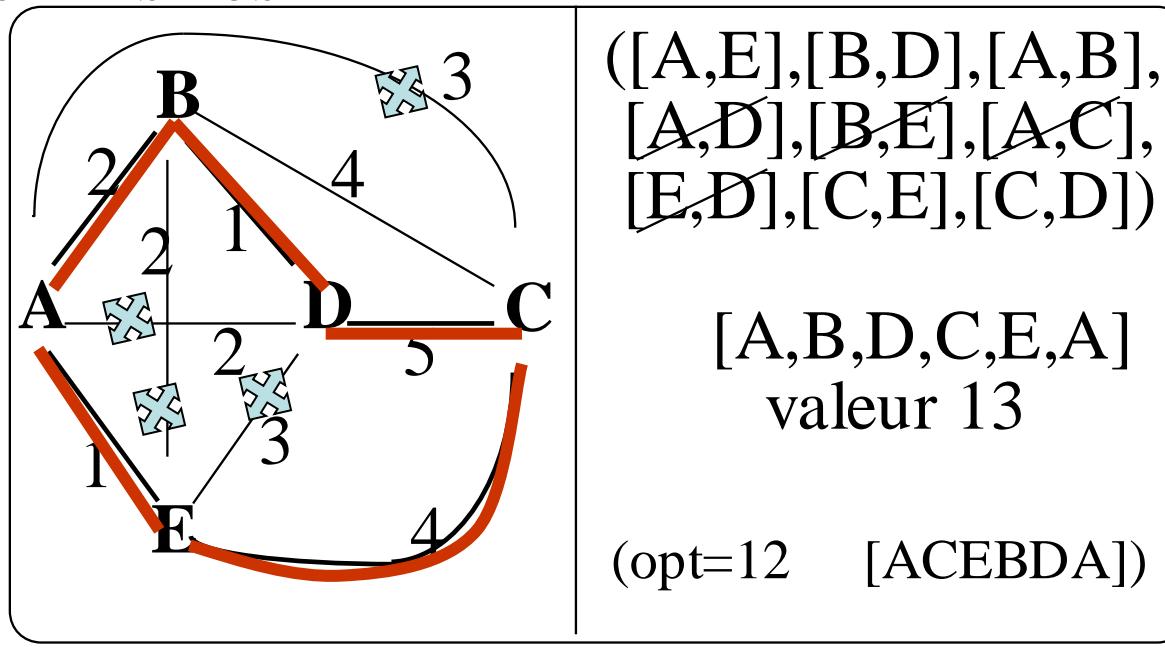
**x_2 optimum global
(et local)**

algorithme glouton: pas toujours optimal

EXAMPLE

un algo glouton pour le PVC:

- trier les arêtes (coûts croissants)
- sélectionner dans l'ordre les arêtes non "parasites"



algorithme glouton: à chaque étape
choix le plus intéressant à cet instant

→ facile à concevoir

→ difficile de vérifier l'optimalité

→ efficace (faible complexité)

ARBRE COUVRANT MINIMAL

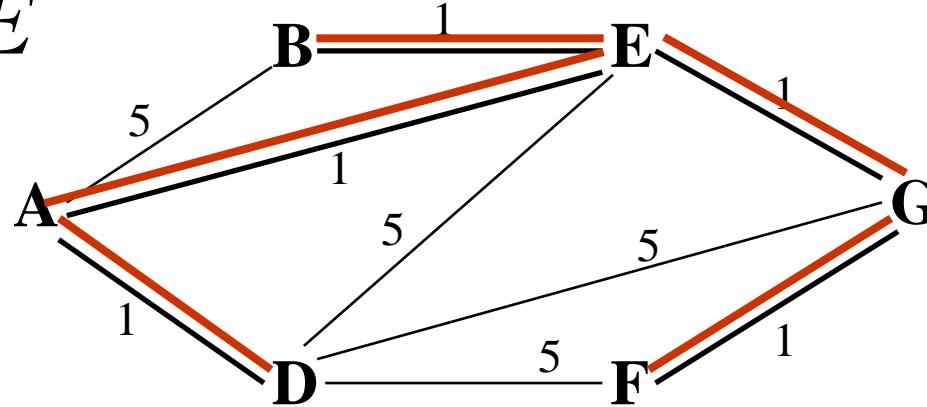
LE PROBLÈME

**relier des objets avec une
longueur totale minimale des liens**

graphé valué associé:

- sommets = objets**
- arête = lien possible**
- poids d'une arête = longueur du lien**

EXAMPLE



solution optimale:

sans cycle et connexe → **ARBRE**

par tous les sommets → **COUVRANT**

de longueur totale min → **MINIMAL**

ALGORITHME DE KRUSKAL

rappel: arbre $\rightarrow m = n-1$ arêtes

entrée: $G = (X, U, P)$

n sommets, m arêtes

sortie: A arbre couvrant min de G

procedure **kruskal** (in G: graphe; out A: arbre):

var k: entier :=0; V: {arêtes} := \emptyset ;

début

trier les arêtes de G par ordre de poids croissant ;

tant que k < n-1 **faire**

parcourir la liste triée ;

sélectionner la première arête qui ne forme pas de cycle avec les précédentes, w ;

k := k+1 ;

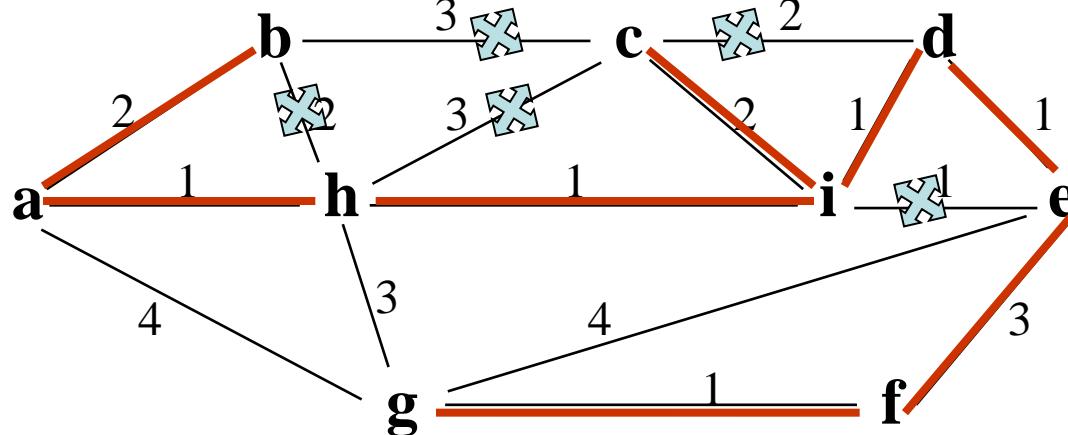
V := V \cup {w} ;

fait ;

A := **graphe** (X,V) ;

fin ;

EXAMPLE



liste triée:

ah	de	di	ei	hi	fg	ab	bh	ci	cd	bc	ch	ef	gh	ag	ge
1	1	1	1	1	1	2	2	2	2	3	3	3	3	4	4

$n = 9 \rightarrow$ stop après 8 sélections

$$P(A) = (1+1+1+1+1+2+2+3) = 12$$

procedure **kruskal** (in G: graphe; out A: arbre):

var k: entier :=0; V: {arêtes} := \emptyset ;

début

trier les arêtes de G par ordre de poids croissant ;

tant que $k < n-1$ faire

parcourir la liste triée ;

sélectionner la première arête qui ne forme pas de cycle avec les précédentes, w ;

$k := k+1$;

$V := V \cup \{w\}$;

fait ;

$A := \text{graphe}(X, V)$;

fin ;

PREUVE DE L'OPTIMALITE

$G = (X, U)$ le graphe

$A = (X, V)$ l'arbre couvrant obtenu

$p(u)$: poids de l'arête u

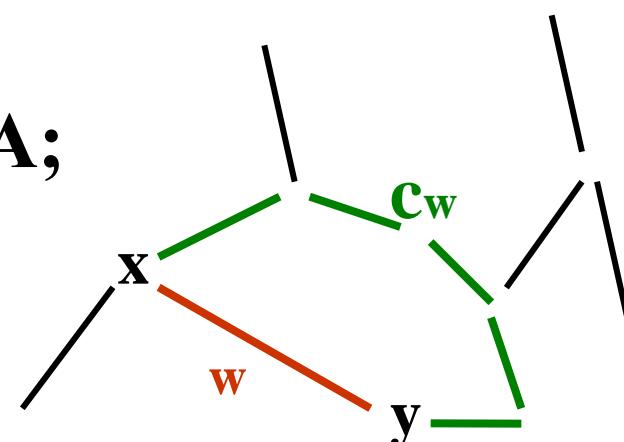
propriété 1

soit $w \in U - V$, $w = [x, y]$ et

c_w : chaîne de x à y dans A ;

alors,

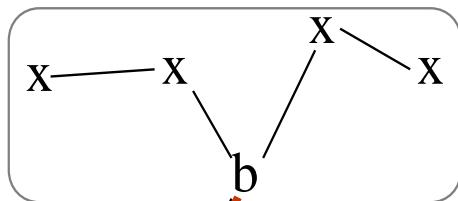
$p(w) \geq \max_{u \in c_w} p(u)$



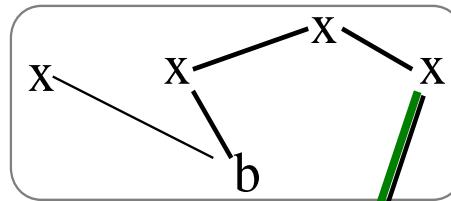
A': arbre optimal de poids $p(A')$

montrons que $p(A) = p(A')$

soit $u \in A' - A$



C_1



C_2

$$A' = (X, V')$$

$$A = (X, V)$$

$u \in V' - V$ relie C_1 et C_2 dans A'

$v \in c_u$ relie C_1 et C_2 dans A

propriété 1 \Rightarrow $p(u) \geq p(v)$ et

A' minimal \Rightarrow $p(v) \geq p(u)$

\Rightarrow $p(u) = p(v)$

soit $A'' = A' + \{v\} - \{u\}$: $p(A'') = p(A')$

\rightarrow **A'' optimal** et $v \in A''$

soit $u' \in A'' - A, \dots$ (idem... $\rightarrow A'''$)

...

fin: $A'''' = A$ et $p(A''''') = p(A') = p(A)$

difficultés:

- **implémentation**
- **complexité $O(m \log m)$ (TRI)**

PREUVE DE L 'OPTIMALITE...

graphes orientés

racine : sommet r tel qu'il existe un chemin de r à tout autre sommet du graphe

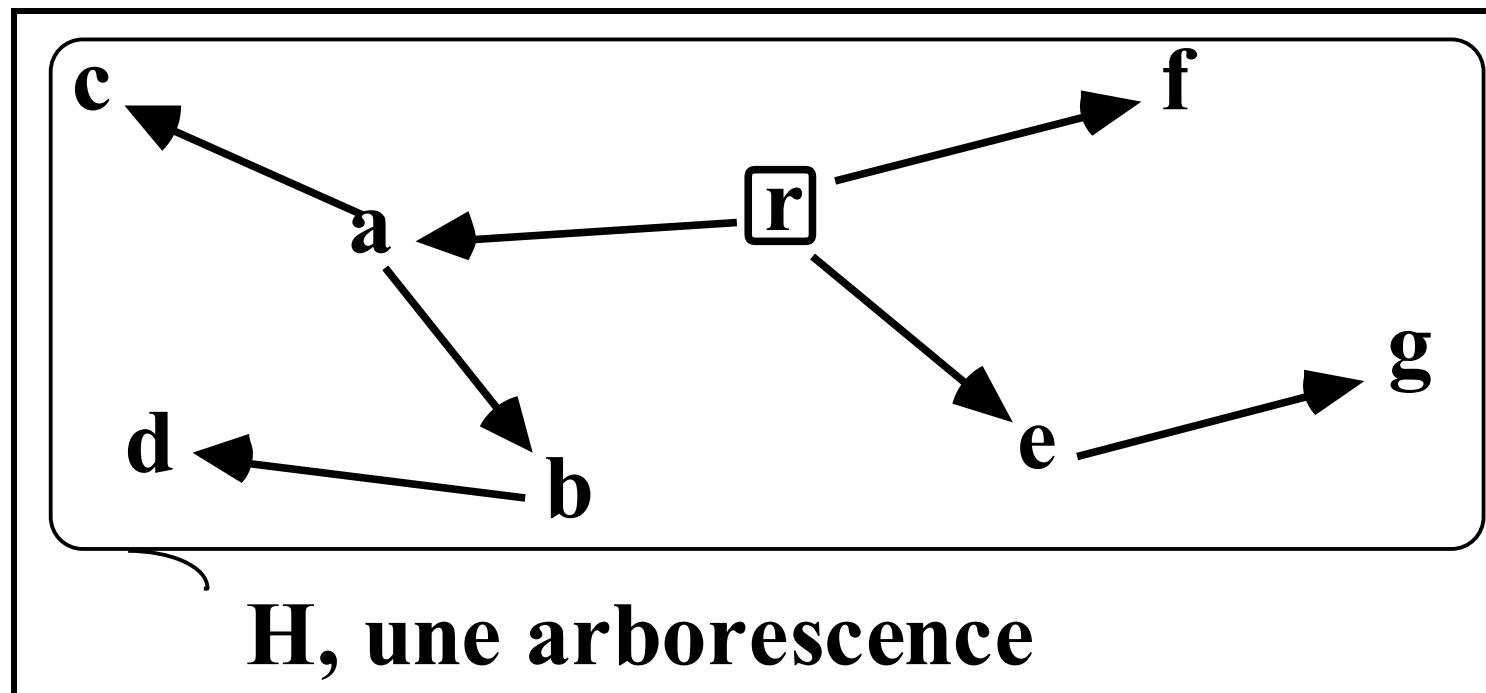
degré intérieur(resp.extérieur): d'un sommet x : nombre d'arcs d'extrémité terminale (resp. initiale) x notés $d^-(x)$ et $d^+(x)$

arborescence: définitions équivalentes

$H=(X,U)$ n sommets ($n \geq 2$)

- (1) H arbre avec une racine r
- (2) H arbre et $\exists r \in X$, relié à tout $x \in X$ par un chemin unique
- (3) H connexe et
 $\exists r \in X$ t.q. $d^-(r)=0$ et
 $d^-(x)=1$ pour tout $x \neq r$

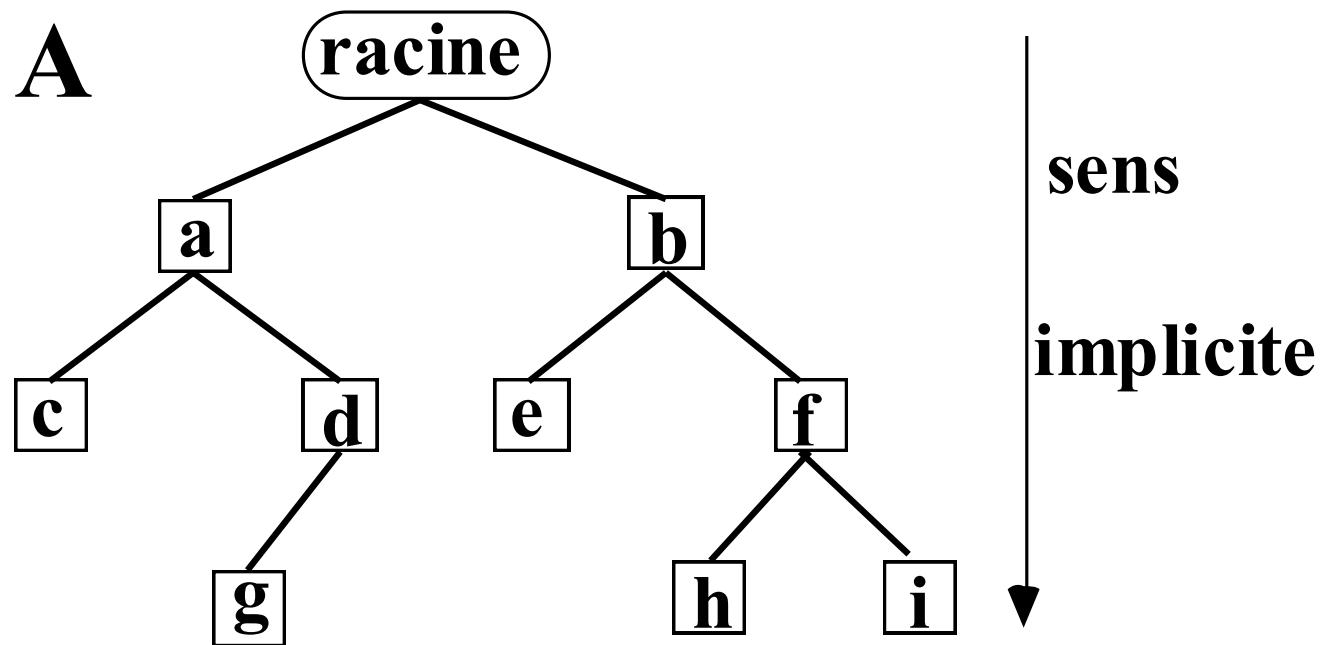
(4) H sans cycle et \exists un sommet $r \in X$ t. q. $d^-(r)=0$ et $d^-(x)=1$ pour tout $x \neq r$



arborescence =
"arbre enraciné" (rooted tree)
= "arbre" en informatique

Ex: arbre généalogique, tournois, arbre des espèces animales,...

arborescence binaire:



5

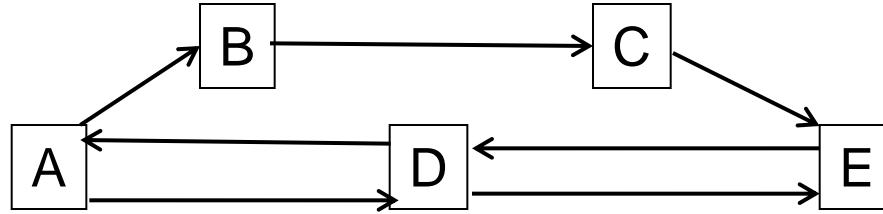
Centre

Diamètre

Centre , diamètre

- La distance $d(x,y)$ entre 2 sommets x,y d'un graphe est le nombre minimum d'arcs pour aller de x à y .
- L'écartement $e(x)$ du sommet x est la distance max entre x et les autres sommets du graphe
- Le (ou les) centre du graphe est le (ou les) sommet d'écartement min
- Le rayon $\rho(G)$ d'un graphe G est l'écartement d'un centre
- Le diamètre $\delta(G)$ est la distance max entre deux sommets du graphe

graphe G



$$\begin{aligned}e(A) &= 2 \\e(B) &= 4 \\e(C) &= 4 \\e(D) &= 3 \\e(E) &= 4\end{aligned}$$

\Rightarrow

A est le centre de G
 $\rho(G)=2$
 $\delta(G)=4$

- Rayon fini si G admet une racine (un sommet x_0 t.q. il existe un chemin de x_0 vers chacun des autres sommets)
- Diamètre fini si G est fortement connexe

- Intérêt d'un centre
 - placer un équipement dans un réseau qui soit le plus « proche » des autres sommets
 - le plus « proche » = la distance la plus longue est minimale
 - on place l'équipement sur un centre

$d^+(x)$ = demi-degré extérieur de x = nombre d'arcs sortant de x

Théorème 1: Soit $G(X, U)$, $|X|=n$

Soit $p=\max \{d^+(x): x \in X\}$ et $p \geq 2$

$$\rho(G) \geq \lceil \frac{\log(1-(1-p)n)}{\log(p)} - 1 \rceil$$

Exemple:

$$p=2, n=5 \quad \rho(G) \geq \lceil \frac{\log(6)}{\log(2)} - 1 \rceil = \lceil 1,58 \rceil = 2$$

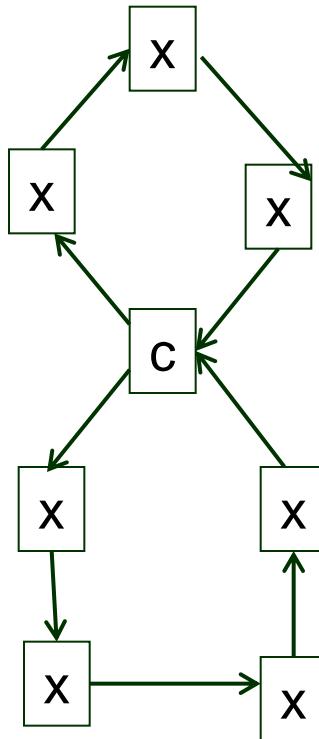
Théorème 2: Soit $G(X, U)$, $|X|=n$, $|U|=m$
 G fortement connexe

$$\rho(G) \geq \lceil (n - 1) / (m-n+1) \rceil$$

Exemple:

$$n=5, m=7 \quad \rho(G) \geq \lceil 4 / 3 \rceil = \lceil 1,33 \rceil = 2$$

Rosace



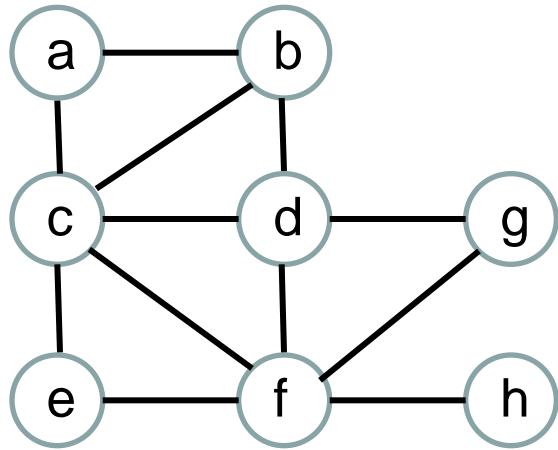
c centre
 $\rho(G) = 4$

Borne théorème 1: $p=2, n=8 \rho(G) \geq \lceil \log(9) / \log(2) - 1 \rceil = \lceil 2,16 \rceil = 3$

Borne théorème 2: $n=8, m=9 \rho(G) \geq \lceil 7 / 2 \rceil = \lceil 3,5 \rceil = 4$

Graphe non orienté

- Les notions précédentes s'étendent aux graphes non orientés
- $d(x,y) = \text{nombre min d'arêtes pour aller de } x \text{ à } y$
- Écartement, rayon, diamètre se définissent ensuite de la même façon



$d(x,y)$	a	b	c	d	e	f	g	h	$e(x)$
a									
b									
c									
d									
e									
f									
g									
h									

Centre =
 Rayon =
 Diamètre =

6

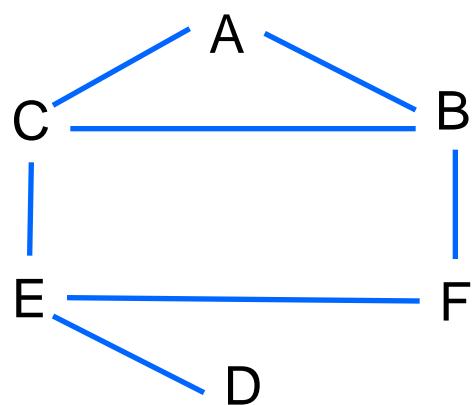
COUPLAGES

- On doit former des équipes de 2 personnes
- L'ensemble des personnes A,B,C,D,E,F
- Compatibilités entre les personnes

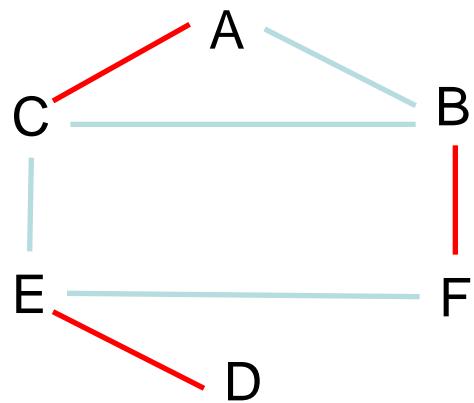
Former un maximum d'équipes

compatibilité	A	B	C	D	E	F
A	-	+	+			
B	-	-	+			+
C	-	-	-		+	
D	-	-	-	-	+	
E	-	-	-	-	-	+
F	-	-	-	-	-	-

Graphe des compatibilités

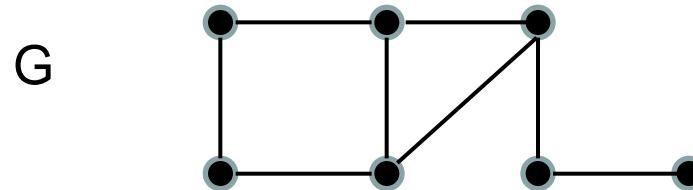


3 équipes



Couplage d'un graphe G

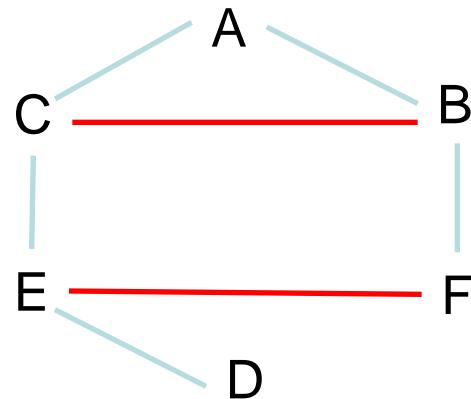
sous ensemble d'arêtes de G tq. 2 arêtes quelconques sont non adjacentes



Exemples de couplages



Couplage maximal = maximal pour l'inclusion d'arêtes

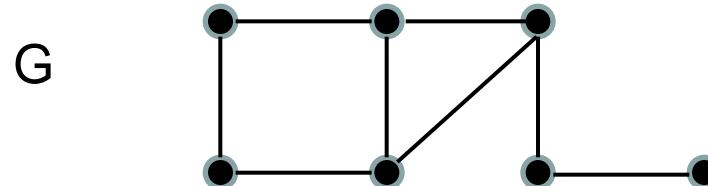


Ce couplage en rouge est maximal . L'ajout d'une autre arête fait qu'il n'est plus un couplage

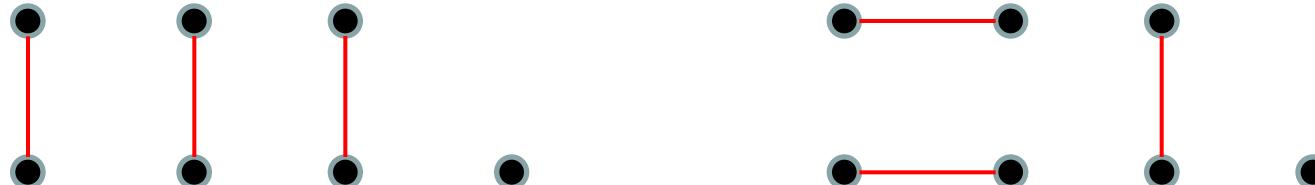
Couplage d'un graphe G

sous ensemble d'arêtes de G tq. 2 arêtes quelconques sont non adjacentes

Couplage maximum = couplage de cardinal maximum

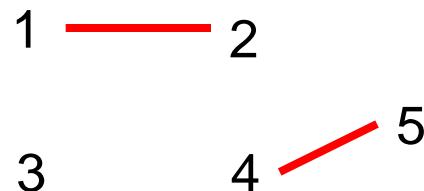
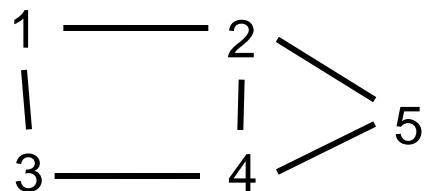


couplages maximum



Sommet saturé par un couplage

Un sommet est dit saturé par un couplage K s'il est l'extrémité d'une arête de K



1 , 2 , 4 , 5 saturés par le couplage rouge

Couplage parfait = couplage qui sature tous les sommets

Un couplage parfait est maximum

X1 — X2 — X3 — X4



Le couplage (rouge) est parfait
Tous les sommets saturés

X5 — X6 — X7 — X8

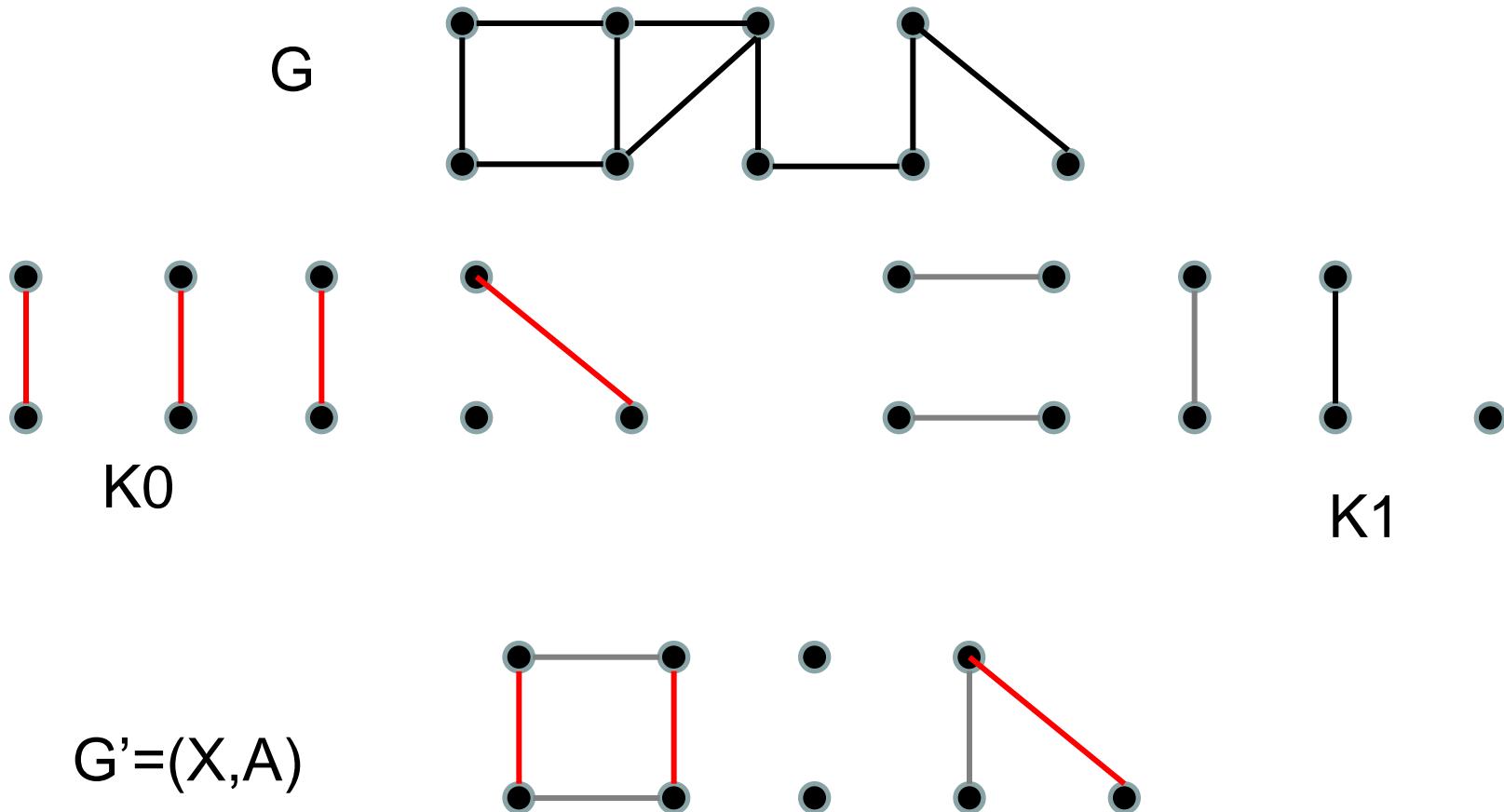
Théorème.

Le cardinal d'un couplage maximal vaut au moins la moitié du cardinal d'un couplage maximum

Dans l'exemple du graphe des compatibilités

- K couplage maximal $|K|=2$
- K^* couplage maximum $|K^*|=3$
- $2=|K| \geq \frac{1}{2}|K^*|=1,5$

Soit 2 couplages K_0 et K_1 de G et soit $A=(K_0-K_1)\cup(K_1-K_0)$
Graphe partiel $G'=(X,A)$



Rappel

Une chaîne élémentaire est une chaîne qui ne repasse pas deux fois par le même sommet

Cycle élémentaire idem

Soit 2 couplages K_0 et K_1 de G et soit $A = (K_0 - K_1) \cup (K_1 - K_0)$
Graphe partiel $G' = (X, A)$

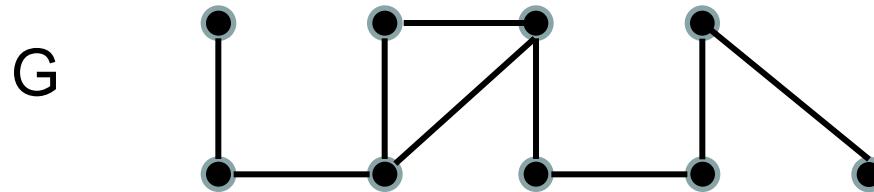
Théorème: $G' = (X, A)$ est fait des composantes connexes de 3 types:

- des points isolés
- des cycles élémentaires pairs faits d'arêtes alternativement dans K_0 et K_1
- des chaînes élémentaires faites d'arêtes alternativement dans K_0 et K_1

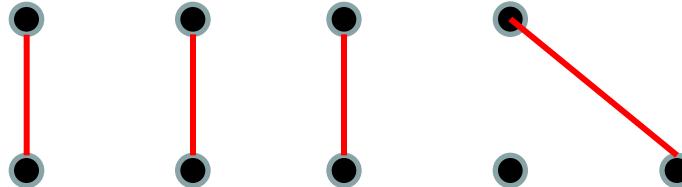
Soit K un couplage de $G=(X,E)$

Une chaine alternée est une chaîne composée alternativement d'arêtes de K et de $E-K$

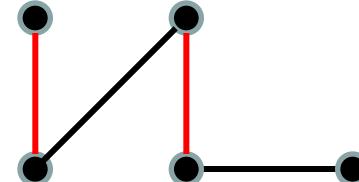
Note: une chaîne alternée est nécessairement élémentaire sinon K n'est pas un couplage



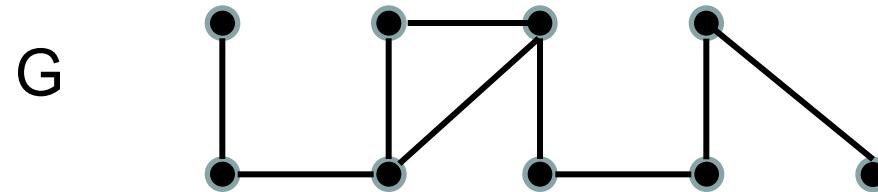
K un couplage de G



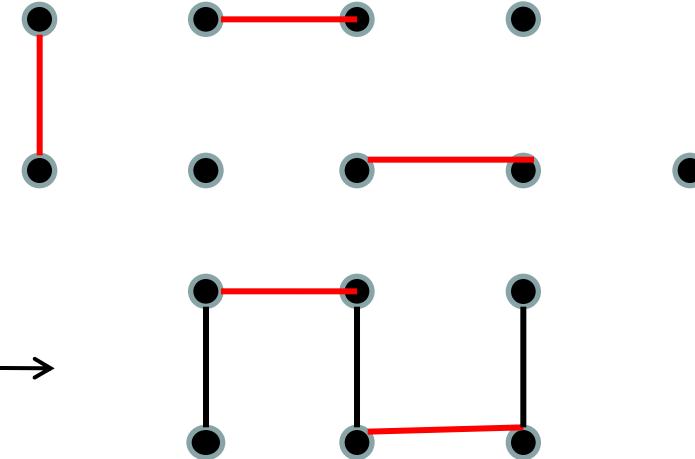
une chaîne alternée



Théorème: Un couplage K est maximum si et seulement si il n'existe pas de chaîne alternée ayant ses 2 extrémités non saturées par K .

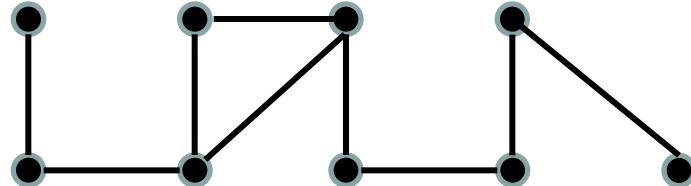


un couplage K de G

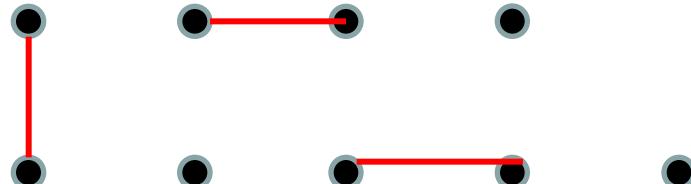


chaîne alternée avec les 2 extrémités non saturées par K
donc le couplage K est non maximum

G

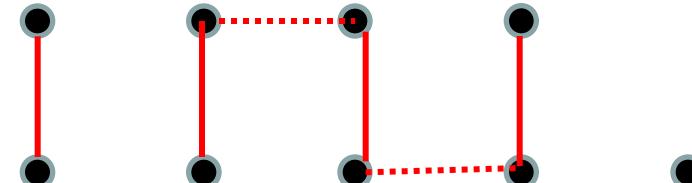


un couplage K de G



chaine alternée avec les 2 extrémités non saturées par K
donc le couplage K est non maximum

un couplage K' de G
plus grand de taille 4.
On remplace les noires
par les rouges dans la chaine alternée



Recherche d'un couplage maximum

On part d'un couplage maximal

Puis augmente sa cardinalité en cherchant des chaînes alternées augmentantes c'est-à-dire avec les extrémités non saturées par le couplage courant



Transfert le long de la chaîne augmente le cardinal du couplage de 1



Dans les graphes bipartis la recherche de chaînes alternées augmentantes est simplifiée

Couplage maximum dans un graphe biparti $G=(X,Y,E)$

Partir d'un couplage maximal K , $\text{STOP}=\text{false}$

Tant que pas STOP

chaine augmentante trouvée=false

marquer + tout sommet de X insaturé

tant que marquage possible et pas chaine augmentante trouvée

marquer + x tout sommet de Y relié à $x \in X$ par arête $\notin K$

marquer - y tout sommet de X relié à $y \in Y$ par arête $\in K$

si on a marqué un sommet de Y non saturé par K

alors chaine augmentante trouvée = true

fin tant que

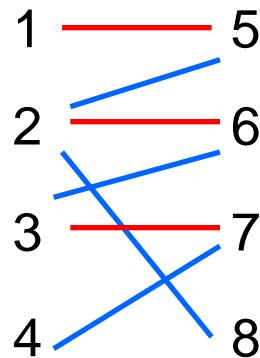
Si pas chaine augmentante trouvée alors K est maximum, $\text{STOP}=\text{true}$
sinon soit μ la chaine trouvée,

effectuer un transfert le long de cette chaine μ :

arête $\notin K$ est mise dans K et arête $\in K$ est sortie de K

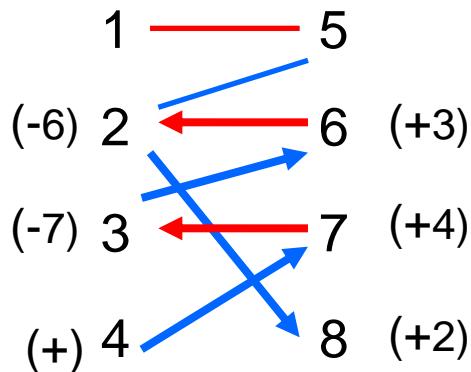
effacer marquage

Fin tant que



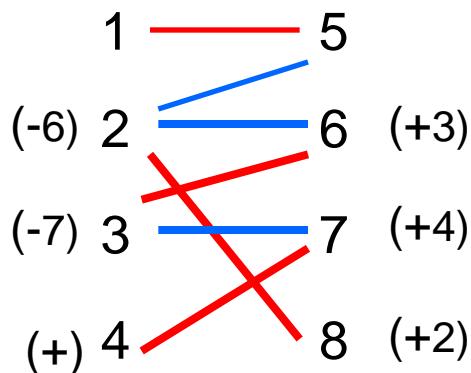
couplage maximal

3 arêtes rouges



marquage

chaine augmentante trouvée



transfert

4 arêtes rouges