Théorie de Graphes

Cássia Trojahn dos Santos

ctrojahn@univ-tlse2.fr

Comment pouvez-vous résoudre ce problème ?

- Un lycée doit organiser les horaires des examens
- 7 épreuves doivent être planifiées, correspondant aux cours numérotés de 1 à 7
- Certains cours ont des étudiants en communs : (1,2), (1,3), (1,4), (1,7), (2,3), (2,4), (2,5), (2,7), (3,4), (3,6), (3,7), (4,5), (4,6), (5,6), (5,7) et (6,7)
- Comment organiser ces épreuves de façon qu'aucun étudiant n'ait à passer deux épreuves en même temps et cela sur une durée minimale ?

Comment pouvez-vous résoudre ce problème ?

- Un lycée doit organiser les horaires des examens
- 7 épreuves doivent être planifiées, correspondant aux cours numérotés de 1 à 7
- Certains cours ont des étudiants en communs : (1,2), (1,3), (1,4), (1,7), (2,3), (2,4), (2,5), (2,7), (3,4), (3,6), (3,7), (4,5), (4,6), (5,6), (5,7) et (6,7)
- Comment organiser ces épreuves de façon qu'aucun étudiant n'ait à passer deux épreuves en même temps et cela sur une durée minimale ?

- Coloration de graphes !

Comment pouvez-vous résoudre ce problème ?

- Supposez que vous avez à planifier le transport de 7 sortes de produits chimiques, dans différents wagons : A,B,C,D,E,F,G
- Certains produits ne peuvent pas être entreposés dans le même wagon, car il y aurait un risque d'explosion :
  - A avec $\{B,C,E,F,G\}$;
  - B avec $\{A,C\}$;
  - C avec $\{A,B,D,G\}$;
  - D avec $\{C,E,G\}$;
  - E avec $\{A,D,F,G\}$;
  - F avec $\{A,E\}$;
  - G avec $\{A,C,D,E\}$;

Comment pouvez-vous résoudre ce problème ?

- Supposez que vous avez à planifier le transport de 7 sortes de produits chimiques, dans différents wagons : A, B, C, D, E, F, G
- Certains produits ne peuvent pas être entreposés dans le même wagon, car il y aurait un risque d'explosion :
  - A avec $\{B, C, E, F, G\}$;
  - B avec $\{A, C\}$;
  - C avec $\{A, B, D, G\}$;
  - D avec $\{C, E, G\}$;
  - E avec $\{A, D, F, G\}$;
  - F avec $\{A, E\}$;
  - G avec $\{A, C, D, E\}$;
- Coloration de graphes !

Comment pouvez-vous résoudre ce problème ?

- Parmi plusieurs trajets possibles, déterminez la longueur du plus court chemin de Nantes à Grenoble, Bordeaux à Grenoble, et Paris-Montparnasse à Marseille

Comment pouvez-vous résoudre ce problème ?

- Parmi plusieurs trajets possibles, déterminez la longueur du plus court chemin de Nantes à Grenoble, Bordeaux à Grenoble, et Paris-Montparnasse à Marseille
- Parcours minimal dans un graphe !

Comment pouvez-vous résoudre ce problème ?

- Parmi plusieurs trajets possibles, déterminez la longueur du plus court chemin de Nantes à Grenoble, Bordeaux à Grenoble, et Paris-Montparnasse à Marseille
- Parcours minimal dans un graphe !

- Cherchez le trajet minimal permettant à un voyageur de visiter toutes les n villes données

Comment pouvez-vous résoudre ce problème ?

- Parmi plusieurs trajets possibles, déterminez la longueur du plus court chemin de Nantes à Grenoble, Bordeaux à Grenoble, et Paris-Montparnasse à Marseille
- Parcours minimal dans un graphe !

- Cherchez le trajet minimal permettant à un voyageur de visiter toutes les n villes données
- Parcours hamiltonien dans un graphe !

Comment pouvez-vous résoudre ce problème ?

- En considérant différentes connexions fibre-optique possibles entre les villes A, B, C, D, E, F et G et leur coût, trouvez le chemin de coût minimal qui relie chaque une des villes.

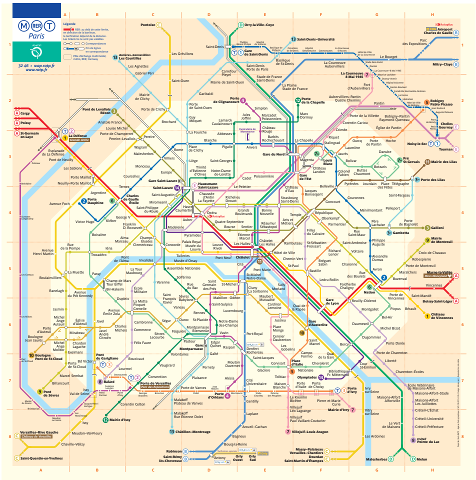
Comment pouvez-vous résoudre ce problème ?

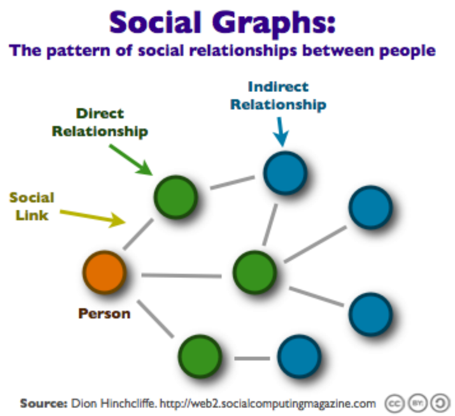
- En considérant différentes connexions fibre-optique possibles entre les villes A, B, C, D, E, F et G et leur coût, trouvez le chemin de coût minimal qui relie chaque une des villes.
- Arbre couvrant minimal !

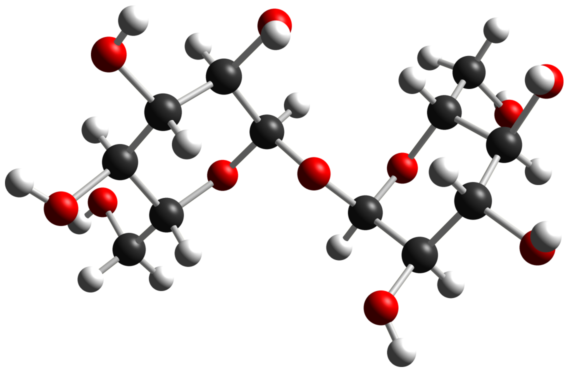
Théorie de graphes

- La théorie de graphes est une théorie **informatique** et **mathématique**
- La configuration de noeuds et leurs connections apparaît dans plusieurs applications :
  - réseaux (de communication, sociaux), circuits électroniques, molécules organiques, plan de villes, etc.
- Ces configurations peuvent être modélisées par des structures combinatoires appelées **graphes**, constituées par deux ensembles :
  1. `sommets` (*vertices*)
  2. `arêtes` (*edges*) (et une relation entre eux)

Théorie de graphes

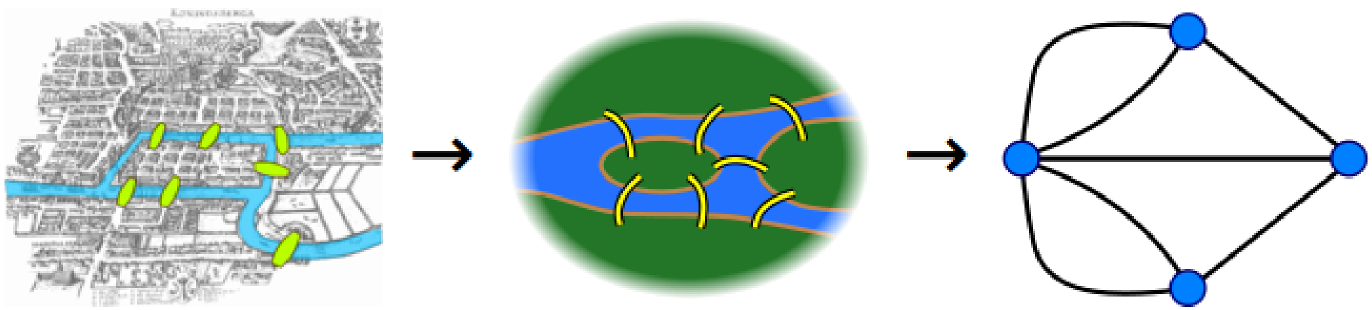






Historique

- La théorie de graphes est née en 1736 quand le mathématicien suisse Leonhard Euler démontra qu'il était impossible de traverser chacun des sept ponts de la ville russe de Konigsberg une fois exactement et revenir au point de départ.



Les catégories de problèmes

- Lorsque un problème donné peut être exprimé en termes d'un graphe, il devient un problème de la "théorie de graphes" ;
- Différentes catégories de problèmes sont connues ;
- Ces problèmes se repartissent en deux catégories :
  - ceux qui peuvent être résolus optimalement par des algorithmes efficaces (rapides) : **polynomiaux**
  - ceux dont la résolution peut pendre un temps exponentiel en fonction de la taille de n : **exponentiels**

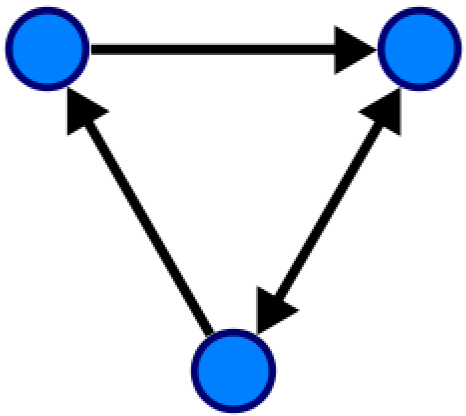
Graphe

Definition (Graphe)

Un **graphe** $G = (S, A)$ est une structure composée de deux ensembles finis S et A. Les éléments de S sont appelés `sommets` (*vertices*) et les éléments de A sont appelés `arêtes` (*edges*). Chaque `arête` a un ou plusieurs `sommets` associés, d'où $A \to S \times S$. Les extrémités d'une arête sont dites `adjacentes`. `L'ordre` d'un graphe est le nombre de sommets $|S(G)|$ qu'il contient.

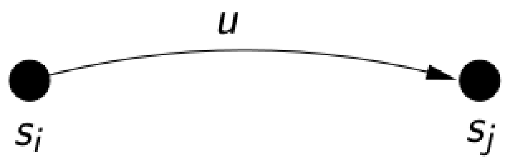
Graphe orienté (*directed graph ou digraph*)

- Dans nombreuses applications, les relations entre les éléments d'un ensemble sont **orientées** ;
- Un graphe orienté est représenté par des sommets et des "flèches" orientées appelées **arcs**, qui relient certains sommets entre eux.



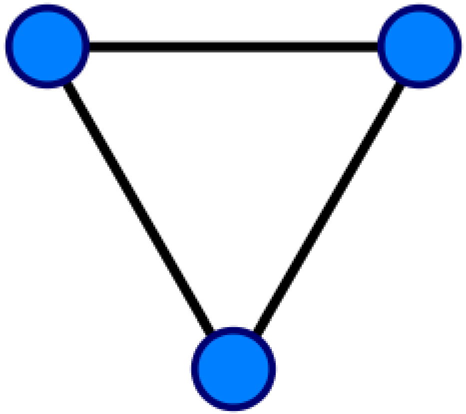
Graphe orienté (*directed graph ou digraph*)

- D'un point de vue mathématique, si S est l'ensemble de sommets, un graphe représente une relation binaire entre des éléments (couples) de S : $u = (s_i, s_j)$



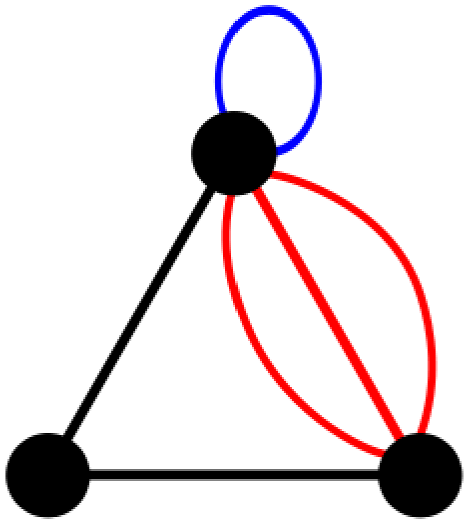
Graphe non-orienté

- Dans certains cas, l'orientation des arcs ne joue aucun rôle ;
- Un graphe non-orienté n'est pas qu'un graphe orienté symétrique :
  - si un `arc` relie le sommet s_1 au sommet s_2, un autre arc relie le sommet s_2 au sommet s_1 ;
  - le trait entre s_1 et s_2 on l'appelle une arête.



Multigraphe et graphe

- Un graphe est **multigraphe** si plusieurs arêtes entre deux sommets existent ;

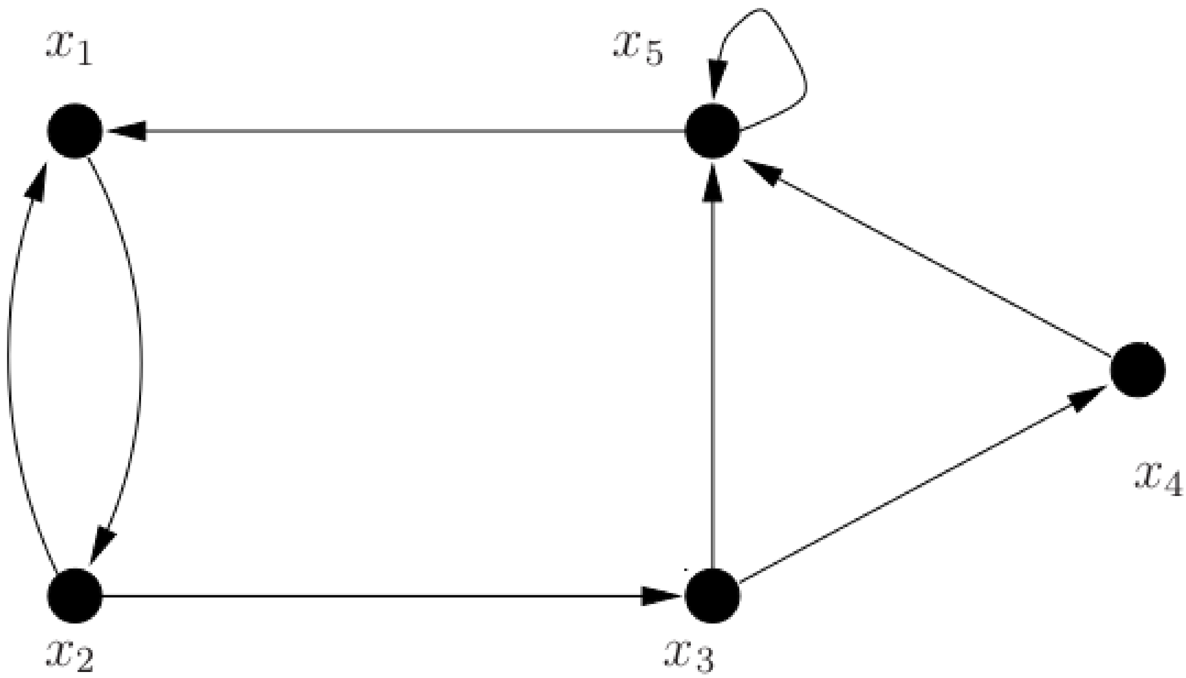


- Un graphe est **simple**
  - s'il n'est pas multigraphe ;
  - s'il n'existe pas de boucle.

Degré d'un sommet

- Dans le cas d'un **graphe orienté** G :
  - le degré sortant d'un sommet s_1 noté $(d^+(s_1))$ est le nombre d'arcs qui partent de s_1 ;
  - le degré entrant, $(d^-(s_1))$ est le nombre d'arcs arrivant au sommet s_1 ;
  - le degré de s_1 est $\Sigma d^+(s_1) + \Sigma d^-(s_1)$
- Dans le cas d'un **graphe non-orienté** G' :
  - le degré d'un sommet s_1 est le nombre d'arêtes rattachées au sommet s_1 (le boucle comptent pour 2).
- Rappel : l'ordre d'un graphe est le nombre de sommets $|S(G)|$ qu'il contient.

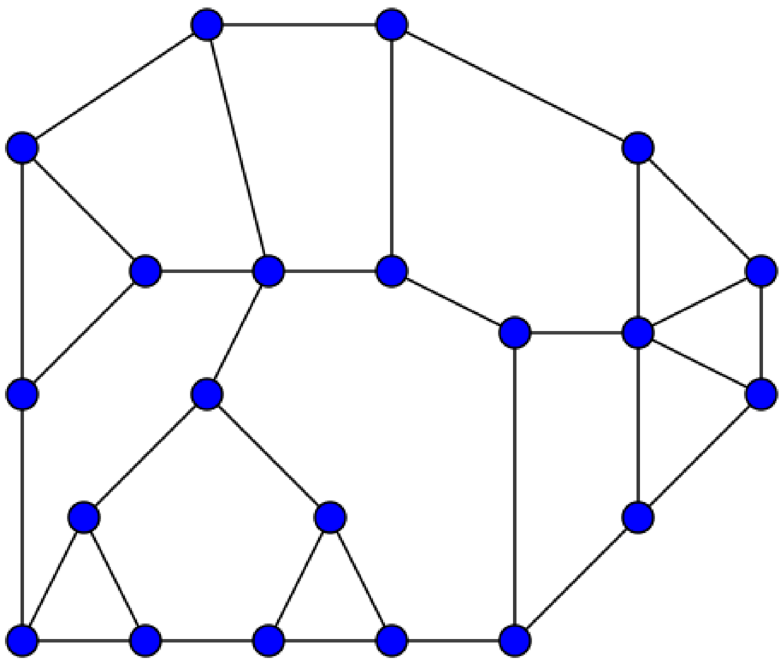
Degré d'un sommet



- $d^+(x_2) = 2$; $d^-(x_2) = 1$; $d(x_2) = 3$
- $d^+(x_5) = 2$; $d^-(x_5) = 3$; $d(x_5) = 5$

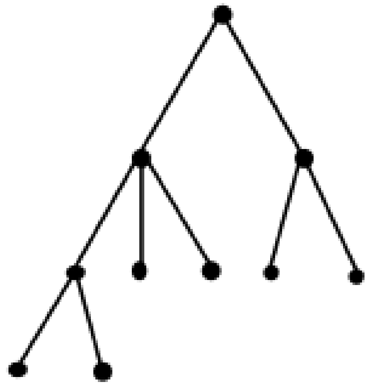
Types de graphes

- **Graphe connexe** :
  - un graphe G est **connexe** si deux sommets quelconques de G sont reliés au moins pour une chaîne ;
  - quels qui soient les sommets s_i et s_j, il existe une suite d'arêtes permettant d'atteindre s_i à partir de s_j.



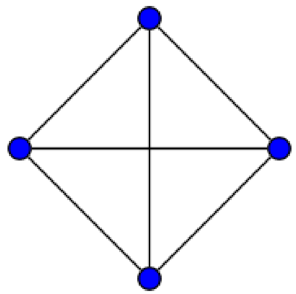
Types de graphes

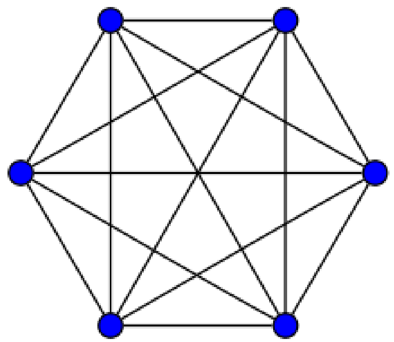
- **Arbre** :
  - il s'agit d'un graphe non-orienté connexe sans cycles ou d'un graphe non-connexe dont le nombre de sommets est égal au nombre d'arêtes plus 1.

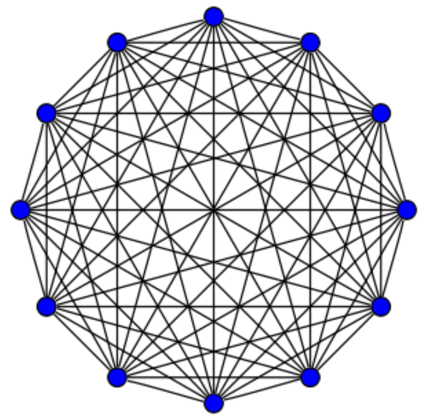


Types de graphes

- **Graphes complets** (clique) :
  - c'est un graphe non-orienté où tous les sommets sont reliés deux à deux par une arête ;
  - dans le cas d'un graphe complet K_n ayant n sommets, le nombre d'arêtes est (n*(n-1)/2).

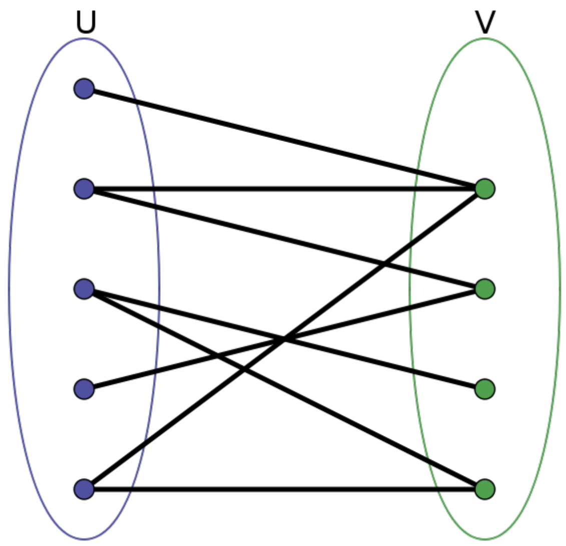






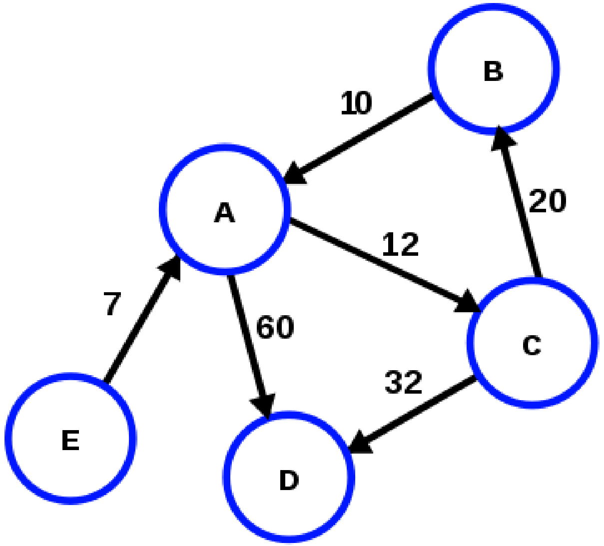
Types de graphes

- **Graphes bipartis** :
  - un ensemble U de sommets d'un graphe est "indépendant" si aucun élément de V n'est connecté à un autre élément de U ;
  - un graphe biparti est un graphe qui représente des relations entre deux ensembles indépendants U et V.



Types de graphes

- **Graphes valués** (ou pondérés) :
  - les arêtes représentent un coût (en temps, en argent, une distance, etc.).

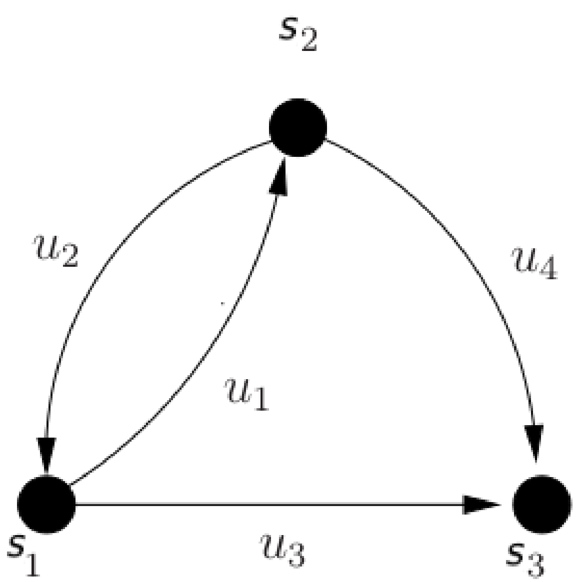


Représentation d'un graphe

- Plusieurs représentations existent pour décrire un graphe :
  - Matrice d'adjacence ;
  - Liste d'adjacence ;
  - Matrice d'incidence sommets-arcs.

Matrice d'adjacence

- La matrice d'adjacence fait correspondre chaque sommet origine (ligne) à un sommet destination (colonne) ;
- Dans une *matrice booléenne* :
  - l'existence d'un arc (ou arête) (s_i, s_j) se traduit par la présence d'un 1 à l'intersection de la ligne s_i et de la colonne s_j ;
  - l'absence d'un arc est représentée par 0.
- Dans le cas d'un graphe non-orienté, la matrice est symétrique par rapport à sa diagonale descendante (on peut mémoriser que la composante triangulaire supérieure)



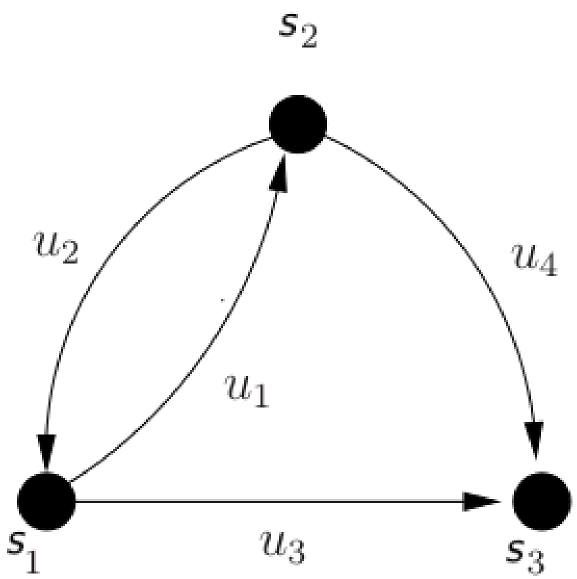
	s_1	s_2	s_3	$\leftarrow$ destination
s_1	0	1	1	
s_2	1	0	1	
s_3	0	0	0	
$\uparrow$ origine				

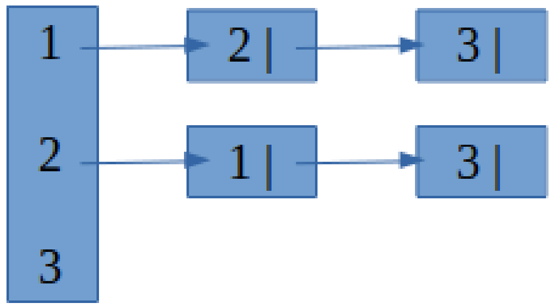
Taille mémoire utilisée : $O(n^2)$ pour un graphe d'ordre n.

Liste d'adjacence

- Il s'agit de créer une liste simplement chaînée pour chaque sommet
  - la liste d'adjacence $L(s_i)$ est une liste chaînée de tous les sommets s_j tels qu'il existe un arc ou une arête (s_i, s_j) ;
  - autrement dit, $L(s_i)$ contient la liste de tous les sommets successeurs de s_i (ordre arbitraire) ;
  - pour un graphe valué, on stocke en plus du numéro du sommet, la valeur de l'arête.

Liste d'adjacence

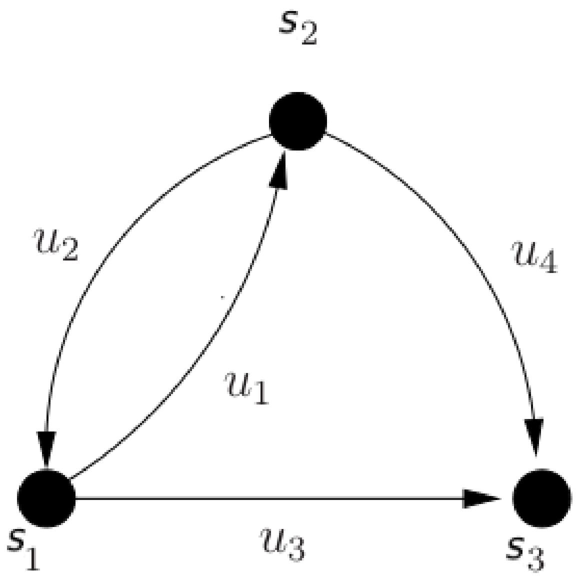




Taille mémoire utilisée : $O(n+m)$ (n sommets et m arêtes).

Matrice d'incidence sommets-arcs

- Chaque ligne de la matrice représente un sommet (s) et chaque colonne un arc (u) – ou arête ;
- Si $u = (s_i, s_j)$, on trouve dans la colonne u : $a_{iu} = 1$ (sortant) ; $a_{ju} = -1$ (entrant) ; tous les autres termes sont nuls.



Taille mémoire utilisée : $O(n \times m)$

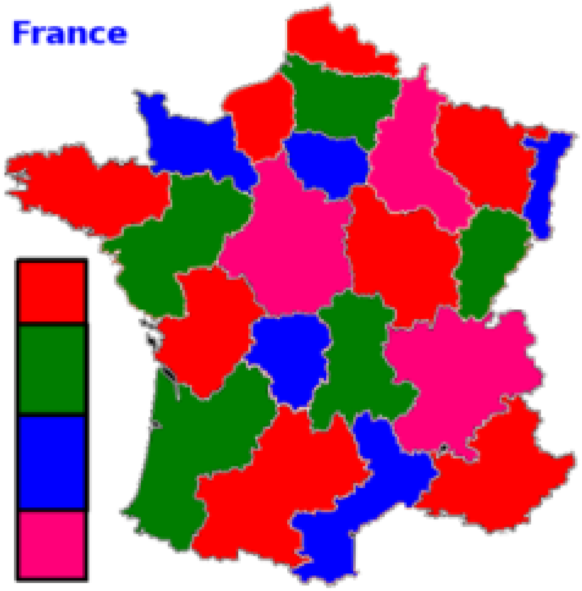
	u_1	u_2	u_3	u_4
s_1	1	-1	1	0
s_2	-1	1	0	1
s_3	0	0	-1	-1

Algorithmes de graphes

- **Coloration**
  - 4-couleurs
  - Heuristique de coloration
- **Chemin le plus court**
  - Algorithme de Dijkstra
  - algorithme A*
- **Arbre couvrant de poids minimum**
  - Algorithme de Kruskal
  - Algorithme de Prim

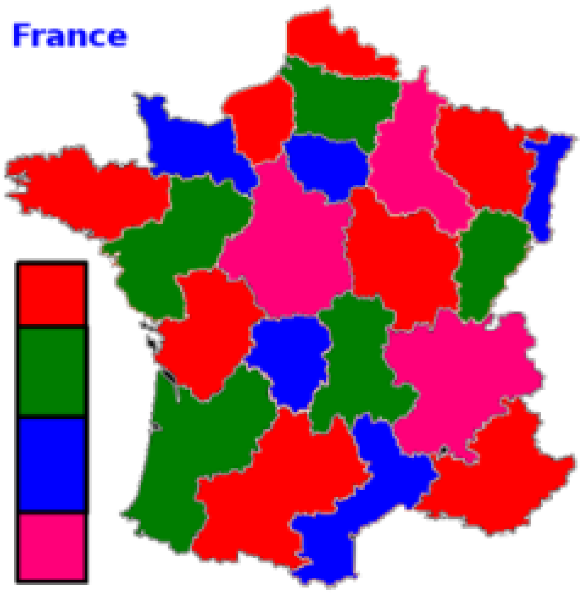
Problème de coloration

- Famille de problèmes qui se présentent par des graphes non-orientés dont les arêtes sont des **contraintes d'incompatibilité** ;
- On cherche à décomposer l'ensemble des sommets du graphe en sous-ensembles indépendants (compatibles).



Problème de coloration : scénario

- Dans la coloration d'une carte de géographie, par exemple, deux régions ayant une frontière commune ne doivent pas être coloriées avec la même couleur :
  - chaque région est représentée par un sommet ;
  - on trace donc une arête entre deux régions.

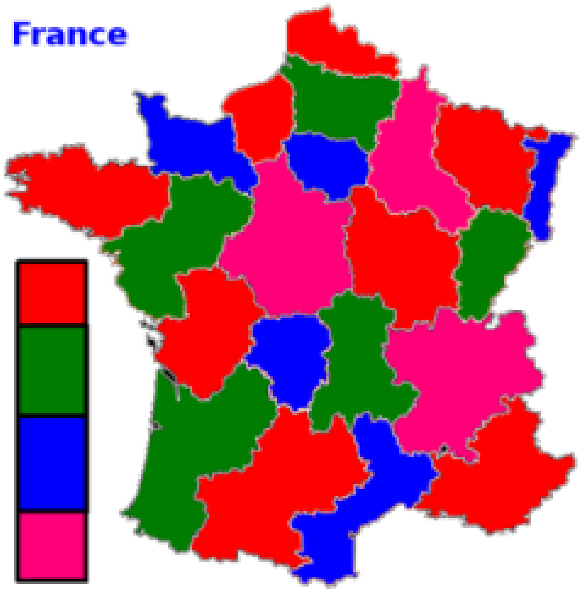


Problème de coloration : scénario

- Certains réseaux de télécommunications sont composés d'émetteurs émettant chacun sur une fréquence particulière. Lorsque deux émetteurs sont trop proches, on ne peut leur allouer la même fréquence à cause des interférences ;
- On associe un graphe au réseau :
  - chaque sommet est un émetteur et chaque arête spécifie que l'on ne veut pas allouer la même fréquence aux deux émetteurs correspondant à ses deux extrémités ;
  - on peut ainsi déterminer une allocation réalisable avec un minimum de fréquences (dont la licence d'exploitation peut entraîner un coût important).
- Ce problème est un cas particulier du problème de la **coloration de graphe**.

Problème de coloration

- Une **k-coloration** est une application qui permet de colorer des sommets d'un graphe avec K couleurs différentes (deux sommets adjacents doivent avoir des couleurs différentes) ;
- Pour chaque couleur k_i, l'ensemble des sommets de couleur k_i est alors un ensemble indépendant.



Problème de coloration

- Le nombre minimal de couleurs k possible s'appelle **nombre chromatique** du graphe G, noté $\gamma(G)$:

Theorem (Graphe complet)

$G = K_n \Leftrightarrow \gamma(G) = n = |S|$ *(nombre de sommets)*

Theorem (Sous-graphe complet)

G contient $K_n \Rightarrow \gamma(G) \geq n$

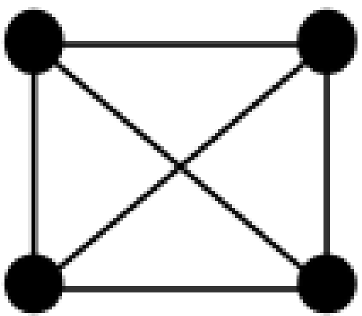
Théorème des 4 couleurs

Theorem (Appel et Haken, 1977)

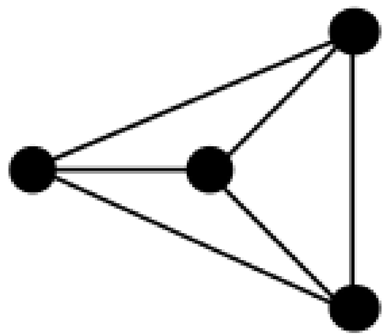
Tout graphe planaire (qui peut être dessiné sur un plan sans croisement d'arêtes) est 4-coloriable. Dans le cas d'un graphe planaire sans triangle, trois couleurs suffisent.

Ce théorème n'a été démontre que grâce à "l'utilisation d'ordinateur", tant que nombre de cas à étudier est grand.

Ce graphe est planaire ...



... parce que on peut le représenter ainsi :



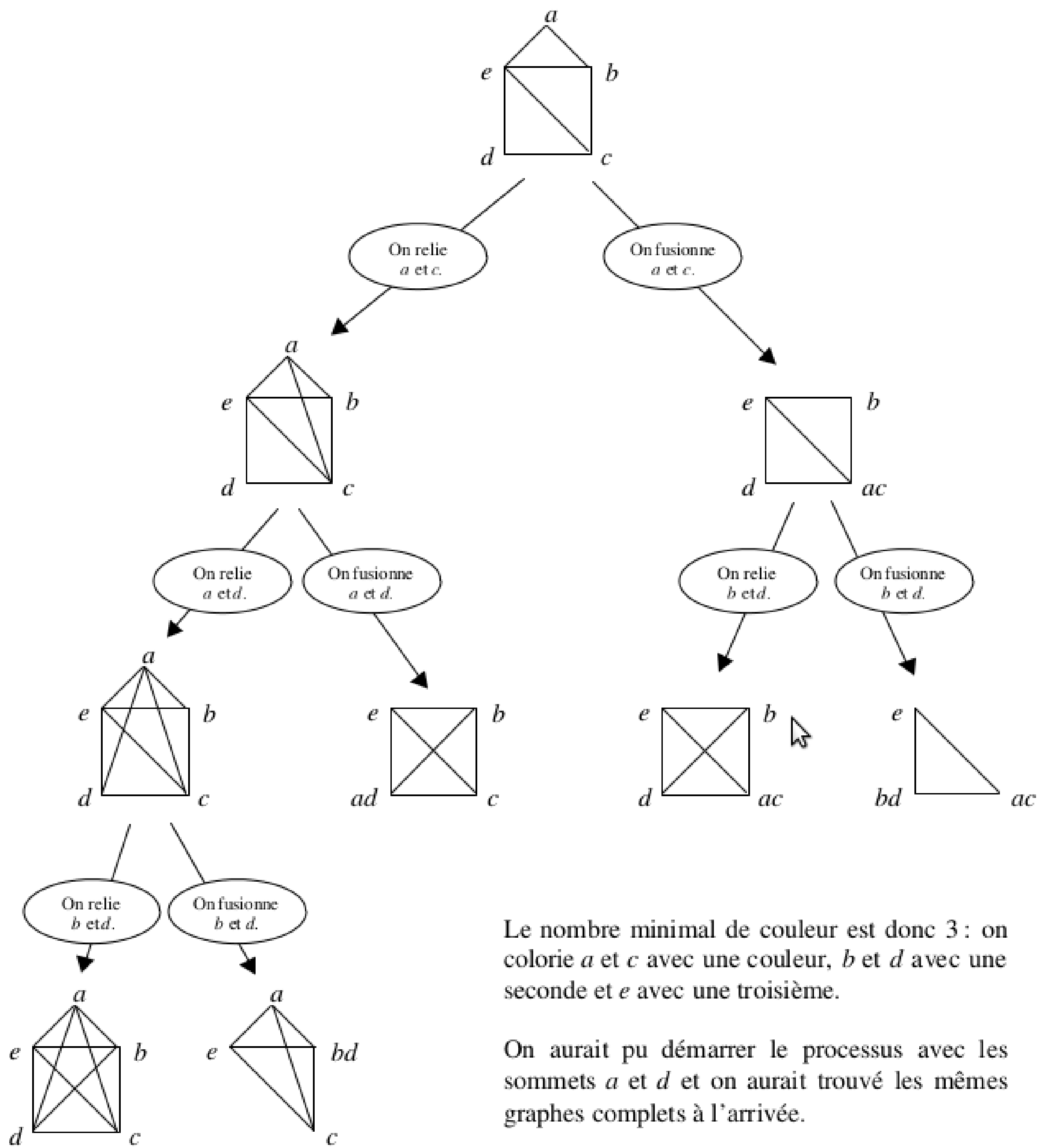
Résolution du problème de coloration

- Déterminer le **nombre chromatique** d'un graphe est un problème **NP-complet** (*no fast solution to it is known !*) dans le cas général
  - rappel : les algorithmes connus pour résoudre des problèmes NP-complets ont un temps d'exécution exponentiel en fonction de la taille des données d'entrée dans le pire cas
- Il s'agit d'un problème de décision : soit G un graphe et k un entier, peut-on colorer G avec moins de k couleurs ?
  - if $k = 2$ il s'agit de vérifier si G est biparti ou pas ;
  - cependant, pour tout $k > 3$ le problème devient NP-complet.
- Il n'existe pas d'algorithme polynomial déterminant le nombre chromatique d'un graphe arbitraire ;
- Pour certaines classes de graphes par contre de tels algorithmes existent ! C'est le cas pour les **graphes triangulés** pour lesquels le problème se résout en temps linéaire.

Colorations possibles pour un graphe triangulé

- Si $G = K_n$ alors if faut n couleurs ;
- Sinon, s'il existe au moins deux sommets s_i et s_j non reliés :
  1. Les colorations de G pour lesquelles les sommets s_i et s_j sont de même couleur sont exactement les colorations du graphe G_1 obtenu à partir de G en identifiant s_i et s_j ;
  2. Les colorations de G pour lesquelles les sommets s_i et s_j ont de couleur distincte sont exactement les colorations du graphe G_2 en identifiant s_i et s_j obtenu à partir de G en ajoutent 1 à l'arête (s_i,s_j). On détermine ensuite les colorations de G_1 et G_2 selon le même principe.

Colorations possibles pour un graphe triangulé



Le nombre minimal de couleur est donc 3 : on colorie a et c avec une couleur, b et d avec une seconde et e avec une troisième.

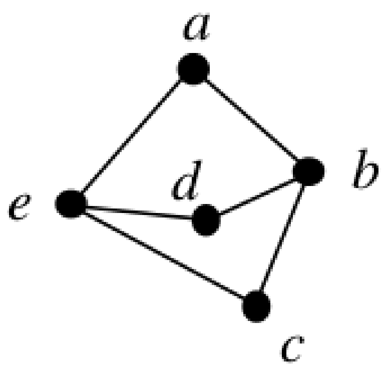
On aurait pu démarrer le processus avec les sommets a et d et on aurait trouvé les mêmes graphes complets à l'arrivée.

Heuristique de coloration

- On peut également définir un processus qui donnera une solution ;
- Par contre, cette solution n'est pas forcement la meilleur !

Heuristique de coloration (Welsh et Powell)

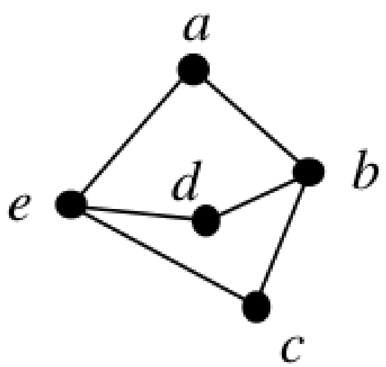
1. On range les sommets dans l'ordre décroissant de leurs degrés : s_1, s_2, ..., s_n.



Sommets	b	e	a	c	d
Degrés	3	3	2	2	2

Heuristique de coloration (Welsh et Powell)

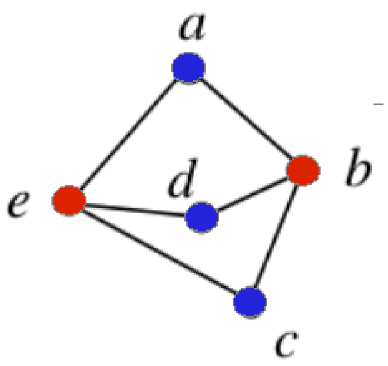
2. On colorie les sommets dans l'ordre précédemment défini ayant comme règle de donner à chaque sommet une couleur, en fonction des **sommets voisins** déjà colorés.



Sommets	b	e	a	c	d
Degrés	3	3	2	2	2
Couleurs	1	1	2	2	2

Heuristique de coloration (Welsh et Powell)

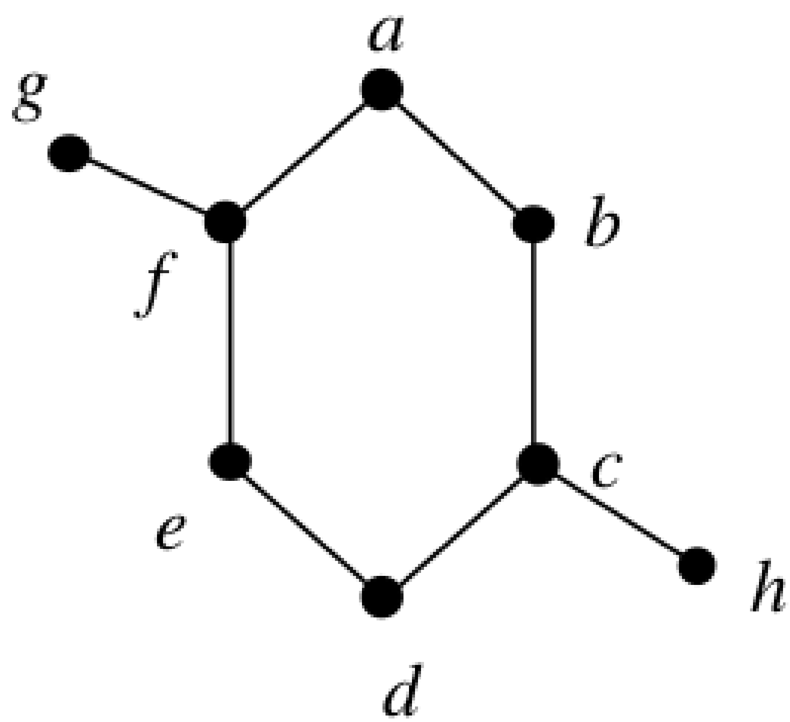
2. On colorie les sommets dans l'ordre précédemment défini ayant comme règle de donner à chaque sommet une couleur, en fonction des **sommets voisins** déjà colorés.



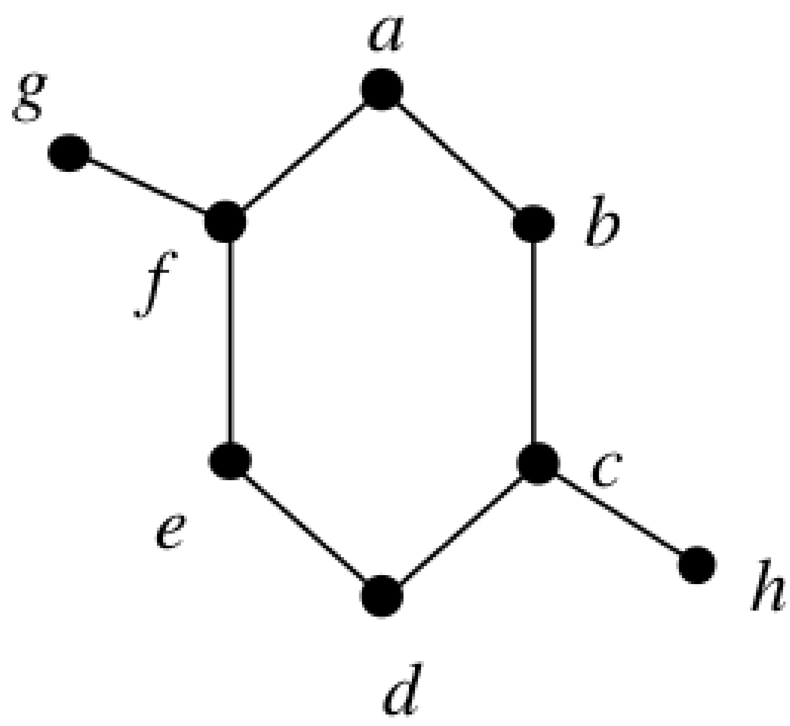
Degrés	3	3	2	2	2
Sommets	b	e	a	c	d
Couleurs	1	1	2	2	2

- **Deux couleurs sont ici nécessaires (et c'est optimal). Mais ce n'est pas toujours le cas !**

Heuristique de coloration

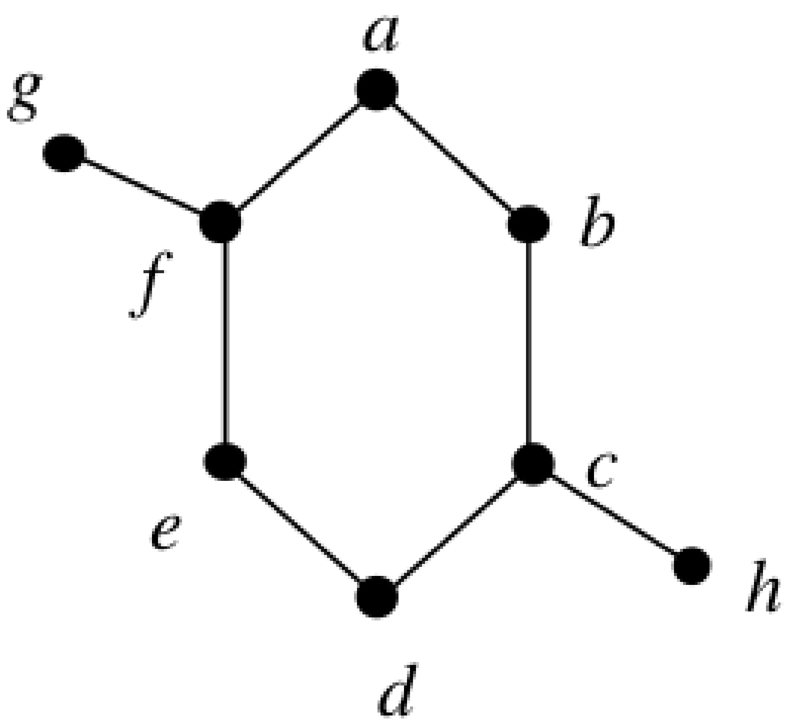


Heuristique de coloration



Le nombre chromatique de ce graphe est 2 : on colorie h, d, f, et b avec une couleur et c, e, a et g avec un autre couleur – ce graphe est biparti !

Heuristique de coloration



Le nombre chromatique de ce graphe est 2 : on colorie h, d, f, et b avec une couleur et c, e, a et g avec un autre couleur – ce graphe est biparti !

Avec Welsh et Powell on trouve 3 couleurs !