

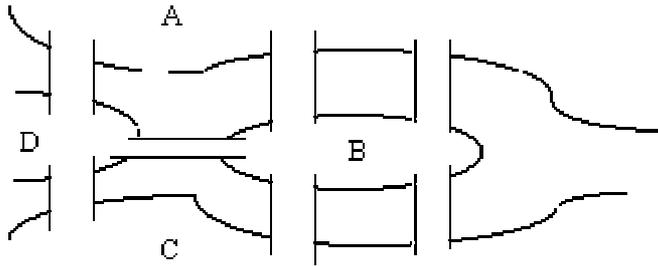


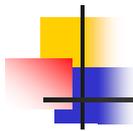
Dans cette partie...

9- Graphes Eulériens

Les ponts de Königsberg

La ville de Königsberg (maintenant Kaliningrad en Russie) est traversée par la rivière Pregolya, et comporte deux îles. Ces îles sont reliées entre elles et aux berges par des ponts, comme sur la figure ci-dessous.



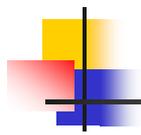


Promenade

Peut-on

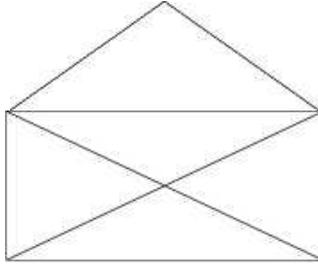
- commencer une promenade sur une île ou une rive,
- terminer la promenade sur n'importe quelle autre (ou la même) île ou rive
- en passant exactement une fois sur chacun des ponts ?

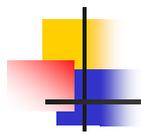
C'est le mathématicien Euler qui en 1735 a trouvé la réponse (d'où le nom "eulérien").



Dessiner une enveloppe

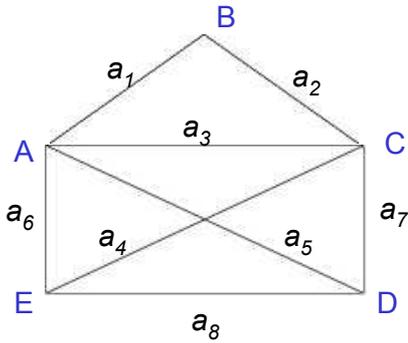
Est-ce possible de dessiner cette enveloppe sans lever le crayon **ET** sans passer deux fois sur le même trait ?



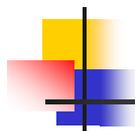


C'est un problème de graphe

Dans un graphe G , est-il possible de trouver une chaîne C qui contient tous les sommets de G et une et une seule fois toutes les arêtes de G ?



Donner une solution pour le graphe ci-contre



Définition de graphe eulérien

Un graphe G est *eulérien* si et seulement si il est possible de trouver une chaîne $C = s_1, a_1, s_2, a_2, \dots, s_m, a_m, s_{m+1}$ telle que :

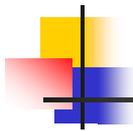
$S(G) = \{s_1, s_2, \dots, s_m, s_{m+1}\}$, $A(G) = \{a_1, a_2, \dots, a_m\}$ et $|A(G)| = m$.

Une telle chaîne est appelée *chaîne eulérienne*.

Si $s_1 = s_{m+1}$ c'est un *cycle eulérien*.

Remarque 1 : Dans la suite de sommets $s_1, s_2, \dots, s_m, s_{m+1}$ certains sommets peuvent apparaître plusieurs fois.

Remarque 2 : Par contre, dans la suite d'arêtes a_1, a_2, \dots, a_m chaque arête apparaît exactement une fois.



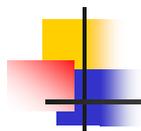
Conditions nécessaires

Théorème :

Un graphe eulérien est nécessairement connexe.

D'autres façons de dire la même chose :

- Tout graphe eulérien est connexe,
- Si un graphe est eulérien, alors il est connexe,
- G eulérien $\Rightarrow G$ connexe,
- Si un graphe n'est pas connexe, alors il n'est pas eulérien,
- G est connexe est une condition nécessaire pour que G soit eulérien.



Preuve du théorème

Soit G un graphe eulérien. Il existe donc une chaîne

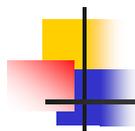
$C = s_1, a_1, s_2, a_2, \dots, s_m, a_m, s_{m+1}$ telle que :

$S(G) = \{s_1, s_2, \dots, s_{m+1}\}$, $A(G) = \{a_1, a_2, \dots, a_m\}$ et $|A(G)| = m$.

Pour démontrer que G est connexe, il suffit de démontrer qu'il existe une chaîne entre s_1 et tout sommet s de G .

Comme $S(G) = \{s_1, s_2, \dots, s_m, s_{m+1}\}$, il existe $i \in \{1, 2, \dots, m+1\}$ tel que $s = s_i$.

Donc, en arrêtant la chaîne C au sommet s_i , on obtient une chaîne entre s_1 et s_i .



Définition de graphe eulérien

Un graphe G est *eulérien* si et seulement si il est possible de trouver une chaîne $C = s_1, a_1, s_2, a_2, \dots, s_m, a_m, s_{m+1}$ telle que :

$S(G) = \{s_1, s_2, \dots, s_{m+1}\}$, $A(G) = \{a_1, a_2, \dots, a_m\}$ et $|A(G)| = m$.

Une telle chaîne est appelée *chaîne eulérienne*.

Si $s_1 = s_{m+1}$ c'est un *cycle eulérien*.

Remarque 1 : Dans la suite de sommets s_1, s_2, \dots, s_{m+1} certains sommets peuvent apparaître plusieurs fois.

Remarque 2 : Par contre, dans la suite d'arêtes a_1, a_2, \dots, a_m chaque arête apparaît exactement une fois.

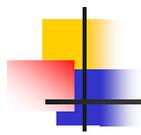


Parité des sommets

Définition

Un sommet est *pair* si son degré est pair.

Un sommet est *impair* si son degré est impair.



Encore une condition nécessaire

Théorème :

Dans un graphe eulérien, le nombre de sommets impairs est nécessairement 0 ou 2.

- Le graphe de la ville de Königsberg n'est donc pas eulérien.
- Pour ce qui concerne l'enveloppe, le théorème ne nous donne pas le droit de prédire que son graphe est eulérien.

Soit G un graphe eulérien. Il existe donc une chaîne

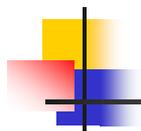
$C = s_1, a_1, s_2, a_2, \dots, s_m, a_m, s_{m+1}$ telle que :

$S(G) = \{s_1, s_2, \dots, s_{m+1}\}$, $A(G) = \{a_1, a_2, \dots, a_m\}$ et $|A(G)| = m$.

■ Examinons un sommet s_i autre que s_1 et s_{m+1} dans cette chaîne.

Un tel sommet est entouré dans C des arêtes a_{i-1} et a_i ce qui donne une contribution de 2 au degré de s_i .

Comme C contient chaque arête exactement une fois, le degré de s_i est pair.



Preuve (suite)

- Examinons le cas des sommets extrémités de la chaîne : s_1 et s_{m+1} .

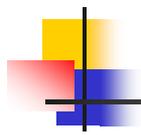
Pour s_1 et s_{m+1} il existe deux cas :

Premier cas : $s_1 \neq s_{m+1}$

s_1 et s_{m+1} sont impairs. Il y a donc 2 sommets impairs.

Deuxième cas : $s_1 = s_{m+1}$

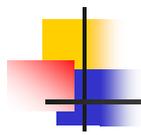
$s_1 = s_{m+1}$ est pair. Il y a donc 0 sommets impairs.



La preuve nous donne plus

Cette preuve nous indique comment trouver une chaîne eulérienne dans un graphe ayant deux sommets impairs :

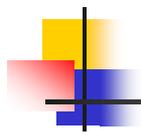
Il faut commencer par un sommet impair, et terminer par l'autre sommet impair.



Théorème.

Un graphe G est eulérien si et seulement si G est connexe et le nombre de sommets impairs de G est 0 ou 2.

- Il nous reste à établir que si G est connexe et possède 0 ou 2 sommets impairs, alors G possède une chaîne passant une seule fois par chacune des arêtes de G .
- Nous allons donner une preuve "constructive" de ce théorème en construisant cette chaîne, dite *eulérienne*.



Une remarque utile sur les cycles

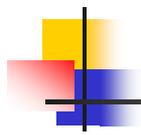
Un **cycle** est une chaîne

$$C = s_1, a_1, s_2, \dots, s_k, a_k, s_{k+1}$$

telle que $|\{a_1, \dots, a_k\}| = k$ et $s_1 = s_{k+1}$.

Remarque :

Si $s_1, a_1, s_2, \dots, s_k, a_k, s_{k+1}$ est un cycle, alors $s_2, a_2, s_3, \dots, a_k, s_1, a_1, s_2$ est aussi un cycle.



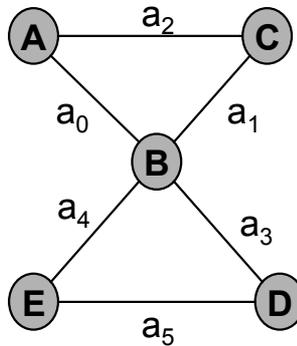
Cas 1 : tous les degrés sont pairs

Algorithme 1 : *construit un cycle C contenant une seule fois chaque arête de G , G étant un graphe connexe où tous les sommets sont pairs.*

1. Choisir un sommet s_1 arbitraire, et former $C = s_1$
2. Tant que le dernier sommet de C a une arête incidente a qui n'appartient pas à C , ajouter à C l'arête a et son sommet extrémité.
(À prouver : C est un cycle après l'étape 2)
3. Si toutes les arêtes de G sont dans C , alors retourner C .
4. Sinon, soit s_i un sommet de C ayant une arête a qui n'appartient pas à C . Former $C' = s_i, a_i, s_{i+1}, \dots, a_m, s_1, a_1, \dots, a_{i-1}, s_i$.
5. Poser $C = C'$, puis continuer à l'étape 2.

Exemple

Faire tourner l'algorithme précédent sur le graphe ci-dessous :



18

$C = A$

$C = A a_0 B$

$C = A a_0 B a_1 C$

$C = A a_0 B a_1 C a_2 A$

On est bloqué. C'est un cycle. Il ne contient pas toutes les arêtes de G . Il y a un sommet (B) qui a (au moins) une arête incidente (a_3) qui n'appartient pas à C . On "casse" le cycle en B , on obtient :

$C' = B a_1 C a_2 A a_0 B$

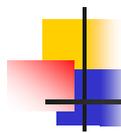
et on continue

$C' = B a_1 C a_2 A a_0 B a_3 D$

$C' = B a_1 C a_2 A a_0 B a_3 D a_5 E$

$C' = B a_1 C a_2 A a_0 B a_3 D a_5 E a_4 B$

On est bloqué. C'est un cycle. Il contient toutes les arêtes de G . C'est fini.



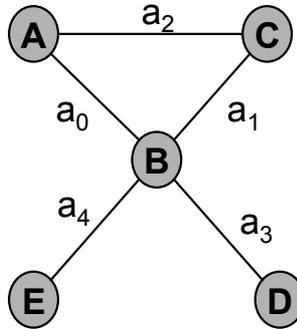
Cas 2 : G a deux sommets impairs

Algorithme 2 : construit une chaîne C contenant une seule fois chaque arête de G , G étant un graphe connexe avec 2 sommets impairs, disons s et t .

1. Former l'arête $a = \{s, t\}$, et un nouveau graphe G' tel que $S(G') = S(G)$, et $A(G') = A(G) \cup \{a\}$. Tous les sommets de G' sont pairs.
2. Déterminer un cycle eulérien C pour G' par l'algorithme 1. On suppose que C contient la sous-chaîne s, a, t dans ce sens (sinon inverser le rôle de s et t).
3. Former $C'' = s, a, t, a_j, s_{i+1}, \dots, s$.
4. Retourner $C = t, a_j, s_{i+1}, \dots, s$.

Exemple

Faire tourner l'algorithme précédent sur le graphe ci-dessous :



20

On rajoute l'arête **a** entre E et D (les deux sommets impairs).

On fait tourner l'algorithme 1 sur le nouveau graphe G' . On obtient le cycle eulérien :

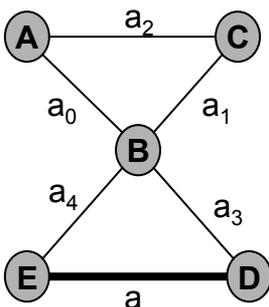
B a1 C a2 A a0 B a3 **D a E a4 B**

On le « casse » en D

D a E a4 B a1 C a2 A a0 B a3 D

On « retire » l'arête **a** et on a la chaîne eulérienne

C = **E a4 B a1 C a2 A a0 B a3 D**





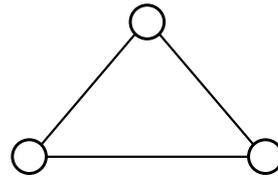
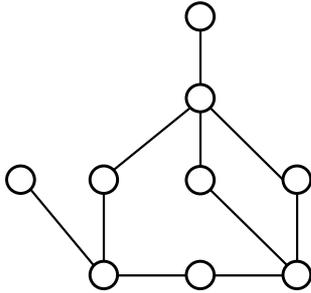
Dans cette partie...

10- Coloration

Coloration

Définition. Une coloration d'un graphe G en k couleurs est une fonction $c : S(G) \rightarrow \{1, \dots, k\}$ telle que, pour tous sommets voisins x et y , $c(x) \neq c(y)$.

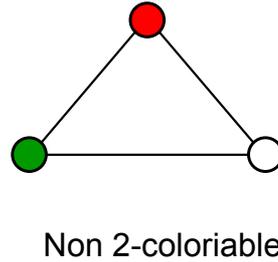
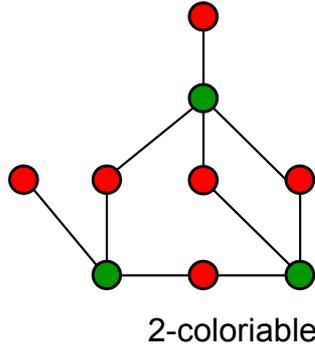
Un graphe qui a une coloration en k couleurs est dit **k-coloriable**.



Coloration

Définition. Une coloration d'un graphe G en k couleurs est une fonction $c : S(G) \rightarrow \{1, \dots, k\}$ telle que, pour tous sommets voisins x et y , $c(x) \neq c(y)$.

Un graphe qui a une coloration en k couleurs est dit **k -coloriable**.



Graphes 2-coloriables et longueurs des cycles

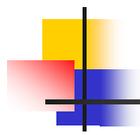
Soit $C = s_1, a_1, \dots, a_m, s_{m+1}$ une chaîne. La longueur de C est son nombre m d'arêtes.

Rappel : C est un cycle si les arêtes a_1, \dots, a_m sont toutes différentes, et si $s_1 = s_{m+1}$. (La chaîne $C = s_1$ est donc un cycle de longueur 0).

Théorème. G est 2-coloriable si et seulement si G n'a pas de cycle de longueur impaire.

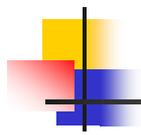
Étape 1 : si G contient un cycle de longueur impaire, alors on ne peut pas le colorier avec 2 couleurs.

Étape 2 : si G ne contient pas de tel cycle, alors il est 2-coloriable. Pour le montrer, on va construire un algorithme de 2-coloration de G .



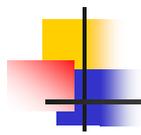
Algorithme de 2-coloration

- On essaie de colorier en deux couleurs, ou bien de trouver un cycle de longueur impaire.
- Quitte à travailler indépendamment sur chaque partie connexe du graphe, on peut supposer G connexe.



Algorithme de 2-coloration

1. Colorier en couleur **1** un sommet arbitraire s_0
2. Tant qu'il existe un sommet s non colorié ayant au moins un voisin colorié :
 - 2.1 si la couleur de tous les voisins coloriés de s est **1**, colorier s en **2**.
 - 2.2 si la couleur de tous les voisins coloriés de s est **2**, colorier s en **1**.
 - 2.3 sinon, deux voisins de s ont des couleurs différentes, et on a un cycle de longueur impaire.



Pourquoi l'algorithme est-il correct ?

À chaque fois qu'on colorie un sommet, on s'assure qu'il sera de couleur différente de celle de ses voisins déjà coloriés.

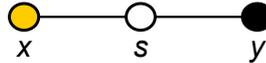
Donc, si l'algorithme colorie tous les sommets, il produit une 2-coloration.

Pourquoi l'algorithme est-il correct ?

Pourquoi y a-t-il un cycle de longueur impaire si l'algorithme détecte que 2 voisins x et y du sommet à colorier s sont déjà coloriés avec 2 couleurs différentes ?

On a détecté :

- (a) une chaîne $C1$ de s_0 à x de sommets coloriés,
- (b) une chaîne $C2$ de s_0 à y de sommets coloriés,
- (c) des arêtes

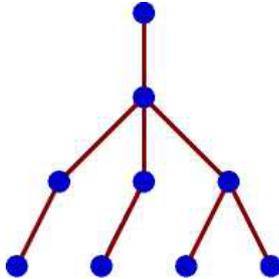


Par (a) et (b), on a une chaîne simple entre x et y , de longueur impaire car x et y ont des couleurs différentes. En fermant cette chaîne par les arêtes (c), on obtient un cycle de longueur impaire.

Dans le cas d'un arbre

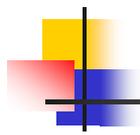
Définition. *Un arbre est un graphe connexe sans cycle.*

Exemple :



Corollaire. *Tout arbre a une coloration en deux couleurs.*

Preuve. Il n'y a aucun cycle, donc pas de cycle de longueur impaire !



Colorier un graphe en 3 couleurs?

Question : Existe-t-il un algorithme "efficace" permettant de savoir si un graphe peut être colorié en trois couleurs? Existe-t-il un critère simple, facilement vérifiable ?

Réponse : on ne sait pas faire beaucoup mieux que de tester toutes les façons possibles de colorier en trois couleurs chacun des sommets et de vérifier que c'est une coloration. Malheureusement, ce n'est pas efficace, il y a 3^n colorations possibles pour un graphe à n sommets!

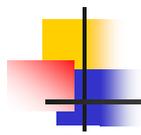
Challenge : Il est offert une prime de 1 million de \$ à qui trouvera un algorithme de complexité n^c (c constante), ou bien démontrera qu'il n'existe pas d'algorithme de complexité polynomiale.



Comment faire ?

Comment faire alors pour colorier **rapidement** un graphe avec **peu** de couleurs?

On utilise une **heuristique**, c'est-à-dire un algorithme qui ne donne pas à coup sûr le nombre minimum de couleurs, mais qui essaie de s'en rapprocher. Cet algorithme doit être efficace.



Heuristique DEGMIN

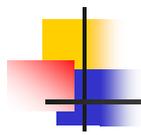
Étant donné un graphe G , donne une coloration pour G

Si G a au moins un sommet, faire :

- Choisir un sommet s de degré minimum d .
- Soit G' le graphe obtenu en supprimant de G le sommet s et toutes ses arêtes incidentes.
- Colorier le graphe G' en réutilisant DEGMIN.
- Rajouter s à G' pour obtenir G .
- Colorier s avec la plus petite couleur différente de ses d voisins.

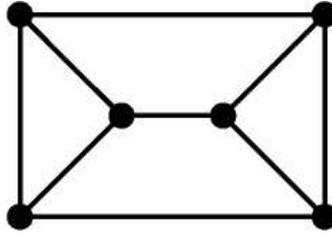
Remarque 1 Pour tout graphe, cet algorithme donne une coloration.

Remarque 2 La complexité est $O(|S(G)| + |A(G)|)$.

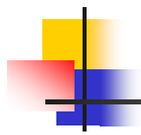


Heuristique DEGMIN : un exemple

Couleurs : **bleu = 1**, **rouge = 2**, **vert = 3**



Il y a à la fin du poly (fichier MSI102-08CI-DEGMIN) les diapos correspondant à chaque étape permettant de faire tourner l'algorithme DEGMIN.



Combien de couleurs utilise DEGMIN ?

Est-ce que DEGMIN marche bien en pratique? Est-ce qu'il colorie avec peu de couleurs?

On va montrer que :

- Si G est un cycle, alors trois couleurs au plus sont utilisées.
- Si G est un arbre, alors deux couleurs au plus sont utilisées.
- Si G est 2-coloriable, alors deux couleurs au plus sont utilisées.
- Si G peut être dessiné sur le plan sans que deux arêtes se croisent, alors six couleurs au plus sont utilisées.



DEGMIN sur arbres et cycles

Cycles

Dans un cycle, chaque sommet a un degré 2. Donc l'heuristique utilise au plus 3 couleurs.

C'est ce qu'on peut faire de mieux si la longueur est impaire.

Arbres (nouvelle preuve qu'ils sont 2-coloriables)

Dans un arbre, il y a un sommet de degré 1 (pourquoi?).

Si on enlève ce sommet, on obtient à nouveau un arbre (?).

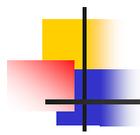
Donc chaque sommet enlevé par l'heuristique a un degré 1.

Donc l'heuristique utilise 2 couleurs au plus.

Graphes 2-coloriables.

Par induction, on utilise 2 couleurs sur G' .

G est 2-coloriable, on réutilise une de ces couleurs pour s



DEGMIN : une propriété importante

Proposition. *Si à chaque étape le sommet s choisi a un degré d , alors DEGMIN utilisera au plus $d + 1$ couleurs pour colorier le graphe.*

Preuve. Par induction sur le nombre de sommets :



DEGMIN : une propriété importante

Proposition. *Si à chaque étape le sommet s choisi a un degré d , alors DEGMIN utilisera au plus $d + 1$ couleurs pour colorier le graphe.*

Preuve. Par induction sur le nombre de sommets :

S'il y a 1 sommet, alors c'est vrai, $d = 0$. Supposons que c'est vrai pour tout graphe à n sommets, et montrons le résultat pour un graphe G à $n+1$ sommets.

Dans G il existe un sommet s de degré d .

Le graphe G' obtenu en supprimant le sommet s de G a n sommets.

Par induction, DEGMIN l'a colorié avec au plus $d + 1$ couleurs.

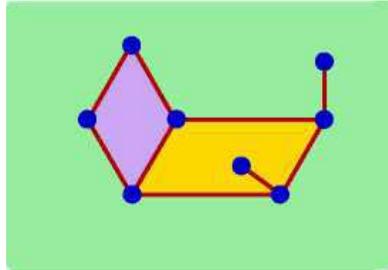
Une couleur de $\{1, \dots, d + 1\}$ est inutilisée sur les d voisins de s .

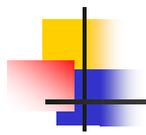
C'est une telle couleur que choisit DEGMIN pour s , donc tous les sommets sont coloriés avec $d + 1$ couleurs.

CQFD

DEGMIN : Graphes planaires

- Un graphe est **planaire** s'il peut être dessiné sur le plan sans que deux arêtes se croisent.
- Une **face** de graphe planaire est une zone du plan délimitée par les arêtes.
- Un graphe fini a une face d'aire infinie. Les autres faces sont d'aire finie.





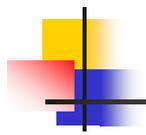
DEGMIN : Graphes planaires

Euler frappe encore !

Théorème. Soit G un graphe planaire connexe, n son nombre de sommets, e son nombre d'arêtes et f son nombre de faces. Alors

$$n - e + f = 2.$$

Preuve : Par induction sur le nombre de sommets.



DEGMIN : Graphes planaires

Preuve. Induction sur le nombre de sommets. Si $n = 1$, toutes les arêtes sont des boucles, chacune détermine une face en plus de la face infinie. Donc $f = e + 1$, et $n - e + f = 2$.

Si $n > 2$, comme G est connexe, il y a une arête entre deux sommets différents. On la « contracte » et on obtient un graphe G' avec $n' = n - 1$ sommets, $e' = e - 1$ arêtes et $f' = f$ faces.

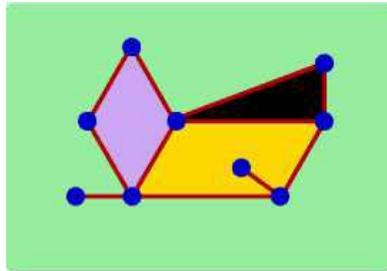
Par induction, $n' - e' + f' = 2$, donc $(n - 1) - (e - 1) + f = 2$, soit $n - e + f = 2$.



Graphes planaires : longueur des faces

La **longueur** d'une face est la longueur de la chaîne qui la délimite.

Exemple.



La longueur de la face noire est 3 celle de la face violette est 4, celle de la face jaune est 6, celle de la face externe est 9. (Noter que l'arête interne de la face jaune, prise "aller-retour" dans la chaîne, contribue pour 2 unités).

Soit $\{F_1, \dots, F_f\}$ l'ensemble des f faces du graphe planaire connexe G et $L(F_i)$ la longueur de la face F_i .

Proposition. Si G est planaire et connexe, alors

$$L(F_1) + \dots + L(F_f) = 2 |A(G)| \quad (1)$$

Preuve. Analogue au 1er théorème sur les degrés.



DEGMIN : Graphes planaires

Théorème. Soit G un graphe planaire simple à au moins 3 sommets. Alors $e < 3n - 6$, et il existe un sommet de degré 5.

Preuve. On peut supposer G connexe, quitte à ajouter des arêtes. Comme $n > 3$, et que G est simple (pas de boucle ni arête multiple) chaque face a une longueur au moins 3. Donc, par (1) :

$$2e \geq 3f.$$

Donc $3n - e = 3n - 3e + 2e \geq 3n - 3e + 3f = 3(n - e + f) = 6$
(car $n - e + f = 2$ par la formule d'Euler), donc $e \leq 3n - 6$.

Si $d(s) \geq 6$ pour tout s , on aurait $2e = \sum d(s) \geq 6n$, impossible.



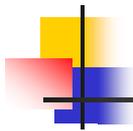
Théorème de 6 couleurs :

Tout graphe planaire est coloriable avec 6 couleurs, sans que deux voisins n'aient la même couleur.

Preuve.

On sait que tout graphe planaire a un sommet de degré ≤ 5 .

On sait que si l'algorithme DEGMIN choisit toujours un sommet de degré d , il colorie le graphe avec au plus $d + 1$ couleurs.



La vie après MSI102

Cette introduction à l'informatique a eu pour objectif d'en donner une idée fidèle, en insistant sur

les aspects algorithmiques, théoriques ou reliés à la recherche,
les applications pratiques dans le domaine de la programmation.

Les notions de ce cours, [graphes](#), [algorithmes](#), [programmes](#), et [complexité](#) interviennent partout en informatique et au-delà, mais...

Ce n'est qu'un avant-goût de ce que vous rencontrerez si vous choisissez des études en informatique.

Bonne continuation !