L3-INFO – INF6B
Structures discrètes
2020-2021

8. Coloration des sommets d'un graphe (introduction)

1. Exemple introductif

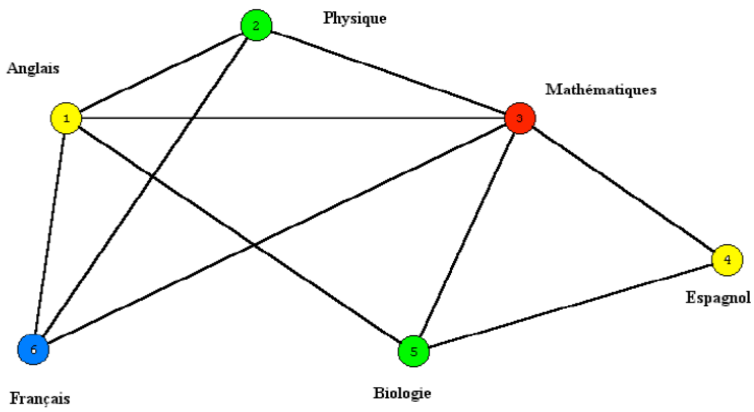
Enoncé. Cinq étudiants doivent passer des écrits d'examen.

- Maxime en Anglais, Physique, Math
- Aude en Espagnol, Biologie, Math
- Marion en Math, Français, Anglais
- Amélie en Anglais, Biologie
- Laurent en Physique, Français

Chaque examen nécessitant une 1/2 journée, quelle est la durée minimale de la période des examens ?

Modélisation sous forme d'un problème de coloration.
- sommets = {Anglais, Physique, Math, Espagnol, Biologie, Français}
- arête entre 2 sommets i et j ssi il existe au moins un étudiant devant passer à la fois les examens i et j

Ci-dessous une solution avec 4 couleurs :



NB : Il n'existe pas de coloration avec 3 couleurs. En effet, les sommets 1, 2, 3 et 6 forment une 4-clique.

2. Définitions

Soit G = (V, E) un graphe non orienté.

Un sous-ensemble S de V est **stable** s'il ne comprend que des sommets non adjacents deux à deux.

Le cardinal du plus grand stable est le **nombre de stabilité** de G ; on le note a(G).

La coloration des sommets d'un graphe G consiste à affecter à chaque sommet de G une couleur telle que deux sommets adjacents soient de couleur différente.

Une coloration avec k couleurs est donc une partition de l'ensemble des sommets en k stables.

Le nombre chromatique du graphe G, noté g(G), est le plus petit entier k pour lequel il existe une partition de V en k sous-ensembles stables.

3. Estimation du nombre chromatique

Soit G = (S, A)) un graphe non orienté.

3.1 Majoration du nombre chromatique

$g(G) \leq r + 1$, où r est le plus grand degré de ses sommets.

Soit un graphe et r le degré maximum de ses sommets. Donnons-nous une palette de (r + 1) couleurs. Pour chaque sommet du graphe on peut tenir le raisonnement suivant : ce sommet est adjacent à r sommets au plus, et le nombre de couleurs déjà utilisées pour colorer ces sommets est donc inférieur ou égal à r. Il reste donc au moins une couleur non utilisée dans la palette, avec laquelle nous pouvons colorer notre sommet.

$g(G) \leq 1 + n - a(G)$

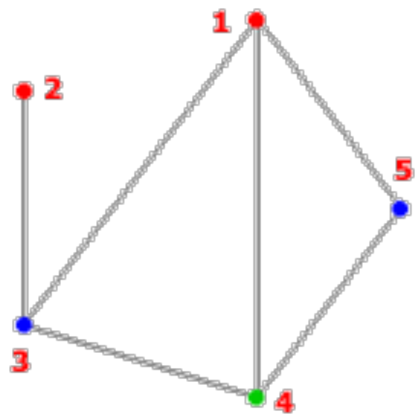
Considérons S un sous-ensemble stable de V de cardinal a(G). Une coloration possible des sommets consiste à colorer les sommets de S d'une même couleur et les n - a(G) autres sommets de couleurs toutes différentes. On en déduit que $g(G) \leq 1 + n - a(G)$.

3.2 Minoration du nombre chromatique

$g(G) \geq w(G)$

Par définition, pour une clique d'ordre m, tous les sommets sont adjacents entre eux ; il faudra donc m couleurs différentes pour en colorer les sommets. Le nombre chromatique du graphe est forcément supérieur ou égal à l'ordre de sa plus grande clique noté w(G).

Sur le graphe ci-dessous, on a eu besoin de trois couleurs de sorte que deux sommets adjacents aient des couleurs différentes. On a donc trois stables : {1, 2}, {3, 5} et {4}. On ne peut pas utiliser moins de couleurs, à cause des K3 cliques 1-4-5 et 1-3-4.



Enfin, le sommet 2 aurait pu aussi être vert. La coloration minimale n'est donc pas forcément unique.

4. Algorithme de coloration de WELSH et POWELL

Cet algorithme couramment utilisé permet souvent d'obtenir une assez bonne coloration d'un graphe, c'est-à-dire une coloration n'utilisant pas un trop grand nombre de couleurs. Cependant il n'assure pas que le nombre de couleurs utilisé soit minimal (et donc égal au nombre chromatique du graphe).

4.1 Sous forme usuelle

Étape #1. Classer les sommets du graphe dans l'ordre décroissant de leur degré.

Étape #2. En parcourant la liste dans l'ordre, attribuer une couleur non encore utilisée au premier sommet non encore coloré, et attribuer cette même couleur à chaque sommet non encore coloré et non adjacent à un sommet de cette couleur.

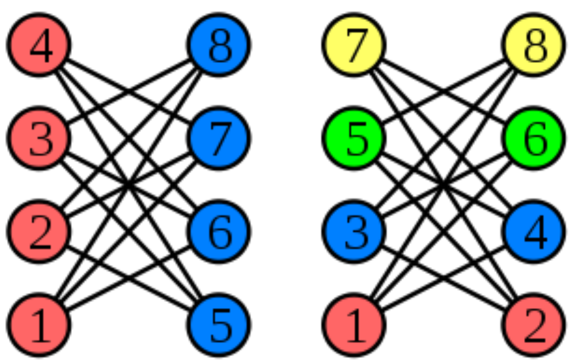
Étape #3. S'il reste des sommets non colorés dans le graphe, revenir à l'étape 2. Sinon, la coloration est terminée.

4.2 Pseudo-code

```
début
Trier les sommets du graphe G par ordre décroissant de leur degré ;
tantque tous les sommets ne sont pas colorés faire
        Choisir une nouvelle couleur K ;
        Colorer le 1er sommet s non coloré en lui affectant la couleur K ;
        pour tous les sommets t non encore colorés faire
                Si t n'est pas adjacent à s alors colorer t avec la couleur K ;
fintantque
fin
```

Cet algorithme donne « souvent » de bons résultats, mais il peut aboutir à de (très) mauvaises colorations, si l'on ne choisit pas bien l'ordre dans lequel on colore les sommets.

4.3 Sur l'exemple ci-dessous *(graphe biparti complet)*, tous les sommets sont de même degré égal à trois.



Si on classe d'abord les sommets « de gauche », puis les sommets « de droite », alors on obtient une 2-coloration (qui est optimale). Mais, si on utilise un classement alternatif « gauche/droite » alors l'algorithme utilisera *4* couleurs.

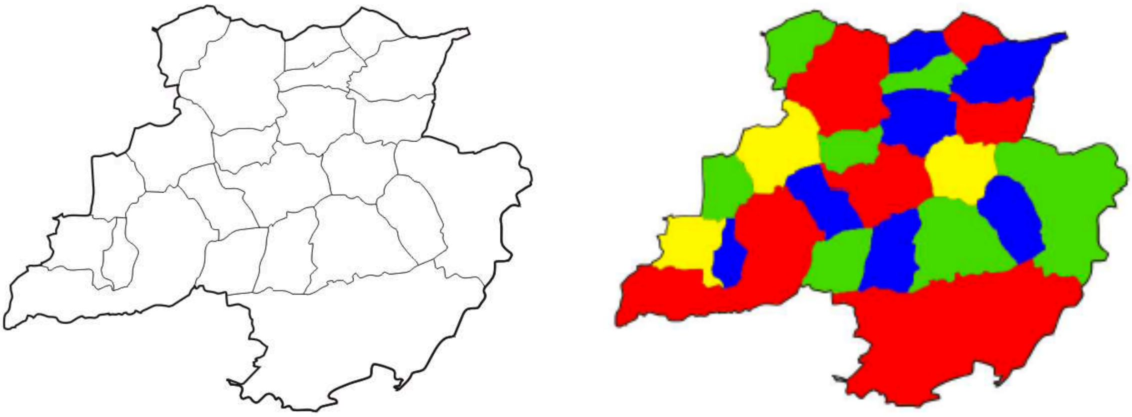
En savoir plus
https://fr.wikipedia.org/wiki/Graphe_couronne

5. Coloration des sommets d'un graphe planaire

5.1 Théorème des quatre couleurs

Il est possible, en utilisant au plus quatre couleurs différentes, de colorier n'importe quelle carte découpée en régions connexes, de sorte que deux régions limitrophes soient de couleurs distinctes.

Exemple :



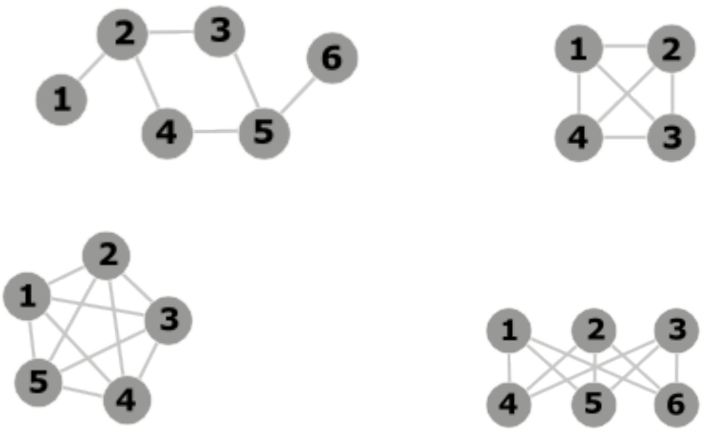
Il y a de nombreuses colorations possibles... avec 4 couleurs

Cette conjecture a été formulée pour la première fois par l'Écossais F. Guthrie en 1852. La preuve de ce théorème n'arriva qu'en... 1976, grâce à K. Appel et W. Haken. Leur démonstration nécessitait l'usage d'un ordinateur pour étudier les 1478 cas critiques (plus de 1200 heures de calcul). Le problème de la preuve du théorème se trouve alors déplacé vers le problème de validation de l'algorithme d'exploration, et de sa mise en œuvre sous forme de programme.

Depuis 2005 une version entièrement formalisée a été formulée en `Coq`[1] par G. Gonthier et B. Werner, qui permet à un ordinateur de complètement vérifier la preuve du théorème des quatre couleurs.

5.2 Graphe planaire

Un graphe planaire a la particularité de pouvoir se représenter sur un plan sans qu'aucune arête (ou arc pour un graphe orienté) n'en croise une autre.
Exemples : https://fr.wikipedia.org/wiki/Graphe_planaire



5.3 Lemme. On peut colorer tous les sommets d'un graphe planaire en utilisant (au plus) quatre couleurs.

[1] https://www.inria.fr/equipes/coq

VERTEX COLORING: WELSH-POWELL ALGORITHM

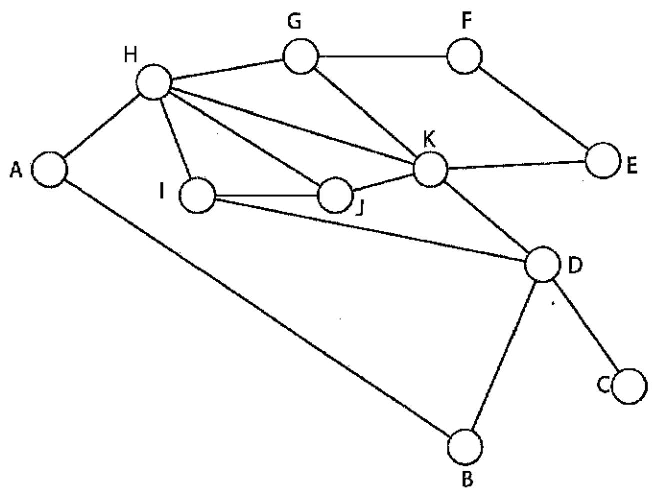
One algorithm that gives a good solution to a vertex-coloring problem is the *Welsh-Powell algorithm*. It may not always give the best solution, but it will usually perform better than just coloring the vertices without a plan will.

The Welsh-Powell algorithm consists of the following steps:

1. Find the valence for each vertex.
2. List the vertices in order of descending valence (you can break ties any way you wish).
3. Color the first vertex in the list (the vertex with the highest valence) with color 1.
4. Go down the list and color every vertex not connected to the colored vertices above the same color. Then cross out all colored vertices in the list.
5. Repeat the process on the uncolored vertices with a new color – always working in descending order of valence until all the vertices have been colored.

The idea is that, by starting with the vertices that have the highest valences, you will be able to take care of the vertices with the largest number of conflicts as early as possible.

EXAMPLE:



Vertex	Valence
A	2
B	2
C	1
D	4
E	2
F	2
G	3
H	5
I	3
J	3
K	5

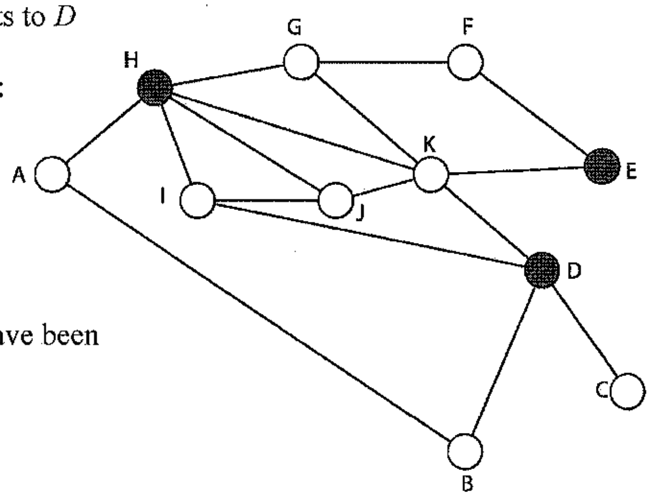
Arrange the list of vertices in descending order of valence. If there's a tie, you can randomly choose how to break it. (To get the ordering below, we used alphabetical order to break ties.) The new order will be:

H, K, D, G, I, J, A, B, E, F, C

Now we color the vertices, in the order listed above. Here's the thought process for the first color:

H color red
K don't color red since it connects to *H*
D color red
G don't color red since it connects to *H*
I don't color red since it connects to *H*
J don't color red since it connects to *H*
A don't color red since it connects to *H*
B don't color red since it connects to *D*
E color red
F don't color red since it connects to *E*
C don't color red since it connects to *D*

Now the graph should look like this:



If we now ignore the vertices that have been already colored, we're left with:

K, G, I, J, A, B, F, C

We can repeat the process now with a second color (blue):

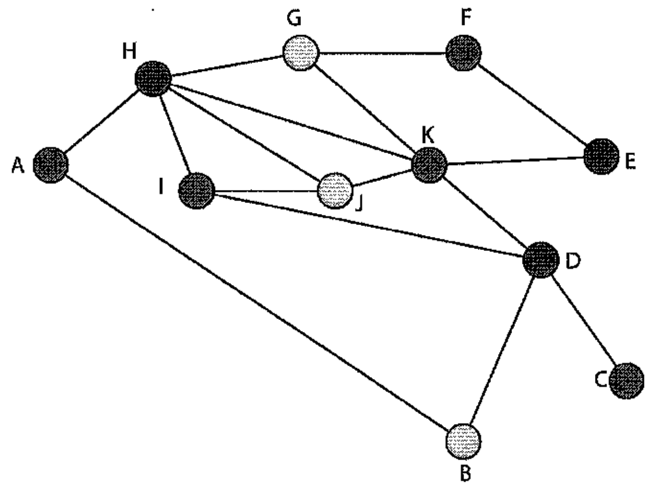
K color blue
G don't color blue since it connects with *K*
I color blue
J don't color blue since it connects with *I*
A color blue
B don't color blue since it connects with *A*
F color blue
C color blue

Again, cross out the colored vertices, leaving you with *G, J, B*, and start at the top of the new list with a new color (yellow):

G color yellow
J color yellow
B color yellow

The final colored graph is shown at right.

We see that by using Welsh-Powell we can get away with using just three colors. In this case, three colors turns out to be the optimal solution since the graph contains at least one triangle (for example, there's the triangle *HIJH*), and it will always take at least 3 colors to color a graph with a triangle in it.

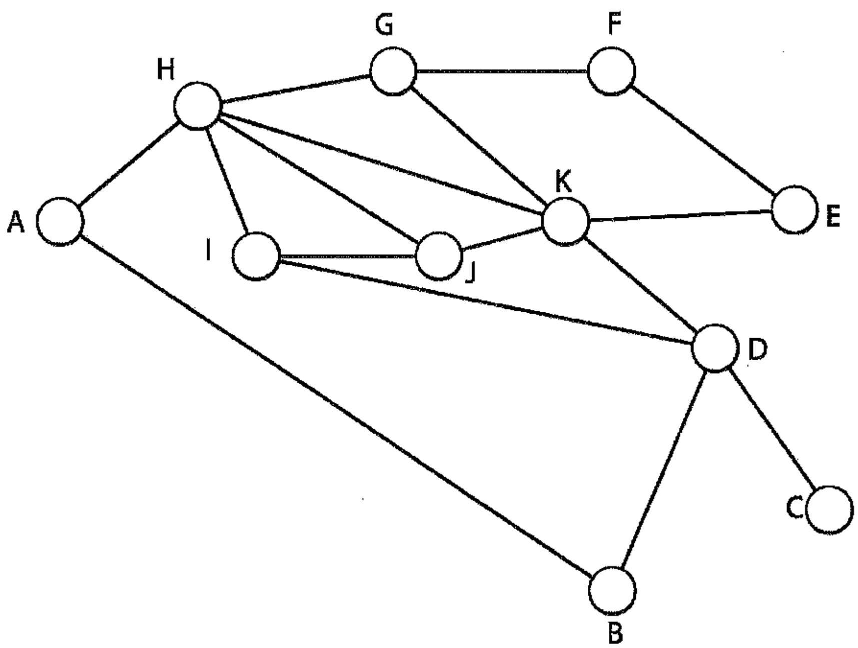


QUESTION

Why would a graph with a triangle in it require at least three colors?

EXERCISE

What would happen if you don't reorder the list of vertices according to their valence? That is to say, what if you just start going through the vertices in alphabetical order? How many colors would be necessary? Color the graph below to find the answer.



Welsh Powell Algorithm: Example