Créer un graphe 3-COL et 3-COL un graphe  
  
Comment créer un graphe 3-COL:  
On procède de la façon suivante:  
-On ajoute un sommet ***Vi*** au graphe **G** initialement vide. On ajoute ce sommet en lui assignant une couleur ***Ci***parmi les 3 couleurs **C:{*C1, C2, C3*}**.

-On lie ce sommet à tous les sommets ***Vj*** de **G** désirés dont ***Ci*! = *Cj***.

On procède ainsi pour tous les sommets que l'on désire ajouter à **G.** A chaque ajout de sommet, tous ***Vi***de **G** n'ont que des connections valides en terme de coloriage puisqu'il était interdit de les connecter à des sommets de la même couleur.  
  
Notez bien qu'il est possible d'ajouter des sommets qui ne sont pas connectés a aucun sommets et qu'il n'est pas mandatoire d'avoir toutes les couleurs dans le graphe.  
  
A la fin de la procédure, on obtient un graphe de **N** sommets pour lequel le 3-COL existe.  
  
Il est aussi vrai que la séquence d'étapes **(*S1, .., Sn*)** (ajout de sommets) pour construire ce graphe n'est pas unique puisqu'à tout point de la construction du graphe, il est possible d'omettre un sommet ***Va*** qui aurait été ajouté à l'étape **Sa** qui aurait envoyé des connexions a un sous ensemble ***SIGMA*** des sommets (***V1,...,Va-1*)** ajoutés avant lui et qui aurait reçu connexion d'un sous-ensemble **THETA** des sommets  **(*Va+1,...,Vn*)** ajoutés subséquemment et l'ajouter à l'étape Sn en le connectant au sommets des ensembles ***SIGMA*** et ***THETA*.**

On suppose que tous les graphes 3-coloriables peuvent être construits de la façon décrite ci-haut. (Preuve à faire)

### Facon équivalente

On peut aussi créer le même graphe 3 coloriable de façon équivalente à la précédente de cette façon :

-On prend un sommet *Vi* de ***G(V, E),*** on lui donne une couleur ***Ci*** parmi **C :{C1, C2, C3}** et on ajoute toutes ses arrêtes **(*Ei1, …, Eij)*** pour les ***j*** voisins **(*V1, …, Vj)*** qu’il possède et on restreint ses j voisins à prendre des couleurs différentes de Ci *(on leur assigner une étiquette* ***Cj = C – Ci*** *représentant les couleurs pouvant leur être assignées).*

(Comme heuristique, on assigne toujours la couleur de première position de l’étiquette, pour imager qu’on conserve toujours la 3e couleur seulement lorsqu’on en aura besoin)

-On prend un voisin ***i*** avec un nombre restreint de couleur ***Ci***, lui assignons sa couleur et le connectons aussi à ses ***j*** voisins par les arrêtes (**Ei1 , … , Eij)** pour les j voisins **(*V1 , … ,Vj)*** qu’il possède. On actualise l’étiquette de ses voisins.

-On répète cette opération pour tous les sommets de ***G(V, E)***

\*Si certains se voient assignés une étiquette avec une seule couleur, on finit l’itération actuelle et traitons en priorité ce sommet et le colorions de sa couleur.

Lorsqu’on ajoute un sommet ***Vi*** est ajouté avec sa couleur ***Ci***, on restreint tous ses voisins à avoir un ensemble de couleur plus petit, soit pour un voisin ***j***: on restreint ses couleurs a ***Cj*** – ***Ci.***  Compte tenu que ce voisin ***j*** peut avoir des voisins ***k,*** où ***k*** est un voisin de ***Vi*** il se peut que ***Ck*** ne soit réduit qu’à une seule couleur et qu’on soit forcé de lui assigner celle-ci.  
Il est toutefois impossible que 4e voisin ***Vh***  soit connecté à ***Vi, Vj*** et ***Vk***, puisque, dans la construction précédente, l’ajout de ***Vh*** correspondrait à cette suite d’opérations :

-Ajout de Vi avec couleur Ci. Aucune connexion

-Ajout de ***Vj*** avec couleur ***Cj. Ci != Ci***. Connection ***Eij***

-Ajout de ***Vk*** avec couleur ***Ck. Ck != Ci, Ck != Cj.*** Connections ***Eki*** et ***EKj***

-Ajout de ***Vh*** avec couleur ***Vh. Ch != Ci, Ch != Cj, Ch != Ck.*** Connections ***Ehi, Eh***j et ***Ehk***.  
toutefois cette opération était interdite puisque Vh ne peut pas avoir une couleur différente de ***Vi, Vj*** et ***Vk*** puisque la leur sont toutes distinctes et ***|C|=3*** et ***|{Vi, Vj, Vk}| = 3***.  
Alors, si ***Vh*** existe, il n’est pas voisin de ***Vi, Vj et Vk***.

Il s’en suit donc que pour tout ensemble de voisins reliés directement dans un graphe 3-coloriable construit comme ci-haut, il n’existe aucune combinaison d’arrête permettant une assignation de couleur pour cette procédure qui donnera a deux sommets voisins la même couleur.

Cette méthode de construction d’un graphe 3-coloriable est donc équivalente à la précédente

Colorier le graphe 3-COL  
Si on prend le même graphe 3-coloriable ***G(V, E)*** que nous venons de créer et le *décolorions*. Nous pouvons le *recolorer* de la façon suivante :

On choisit un sommet ***Va***de façon arbitraire et lui assignons la couleur de première position dans l’étiquette générale des 3 couleurs disponibles pour le coloriage.

Nous savons que lors de la construction du graphe, ces sommets étaient connectés à notre sommet nouvellement connecté seulement s’ils avaient une couleur différente. Si ***Va*** a une couleur **Ca**, tous ces voisins auront une couleur différente de **Ca***,* nous pouvons donc étiqueter tous ses voisins avec l’ensemble des couleurs qui leur étaient disponible moins **Ca**. Tous les voisins de ***Va*** ont donc un ensemble de couleurs restreints.

On choisit un des voisins ***j*** de ***Va*** ***(Vj, …, Vj+k)*** et nous lui assignons une couleur parmi celles disponibles et actualisons tous ses voisins afin qu’ils aient un nombre restreint de couleurs.

Lorsqu’un sommet n’a qu’une couleur possible, elle lui est automatiquement assignée et on actualise ses voisins.

Aucun sommet autre que le premier n’est ajouté s’il a 3 couleurs disponibles et que d’autres sommets possèdent un nombre moindre de couleurs. Ce sommet « n’a pas encore été ajouté au graphe » en termes de la relation liant les autres sommets et puisque le choix de couleur est arbitraire, il est impossible de spéculer sur la couleur précise qui peut lui être assignée avant qu’un de ses sommets voisins y soit connecté.

Les sommets restant à la fin du coloriage qui auront 3 couleurs disponibles feront donc parti d’un ensemble disjoint du premier.

En coloriant les commets de cette façon, nous effectuons la construction du graphe à rebours, en appliquant la même contrainte que lors de la construction de celui-ci :

Lors de la construction, on savait que le sommet ***Vk*** de couleur ***Ck*** ne pouvait être connecté qu’à l’ensemble des sommets de couleurs ***C-Ck****.*

Lors de ce coloriage, nous affirmons que, pour le même graphe, chaque voisin d’un sommet que nous colorions (partant d’un sommet arbitraire), a été ajouté en respectant la couleur de ce sommet ***Vk****.*

Ou plus simplement, nous effectuons exactement les étapes de création du graphe 3 coloriable correspondant à la 2e méthode (celle équivalent à la première) afin de colorier le graphe (On découvre un sommet à la fois et n’assignons jamais de couleurs a des sommets qui ne sont pas connectés au graphe)

Nous arriverons donc au bout de cette procédure lorsque tous les sommets seront coloriés pour notre graphe 3-COL.

## Algorithme de coloriage

Sachant tout cela, si nous prenons un graphe ***G(V, E) |V|=N*** quelconque et appliquons la même procédure:

-On assigne une étiquette correspondant à l'ensemble ***C :{C1, C2, C3}*** à tous les sommets.

-On choisi un sommet **Vk** au hasard et lui assignons une valeur de façon arbitraire, soit la couleur **Ck.**

-Pour tous ses voisins, on retire la couleur étant assignée à ce graphe de l'ensemble des couleurs pouvant leur être assigné

-On choisi un de ses voisins ***Vk+l (0<= l <= nb de voisins de k)*** et lui assignons une couleur parmi celles étant disponibles en fonction de leur étiquette.

-On actualise tous ses voisins et choisissons un sommet au hasard priorisant ceux avec le moins de couleur disponibles et leur assignons une couleur et répétons la procédure à nouveau.

-On répète la procédure **N** fois, ou que l'on ait rencontré un conflit de coloriage.

(Conflit de coloriage peut se manifester lors de la vérification finale du certificat obtenu ou si on rencontre un sommet dont l'étiquette est vide)

Si nous rencontrons une étiquette vide, il en suit qu’il existe un sommet qui ce graphe n'a pas été construit selon la méthode ci-haut, par conséquent, ce graphe n'est pas 3-COL.  
  
Cet algorithme traite n sommets, pour lesquels il y a un maximum de n-1 connexion

à actualiser les étiquettes des sommets. Cet algorithme est de l'ordre de N^2