

Université Catholique de Lille

# Projet Algorithmique

Problème de coloration de graphe



FACULTÉ DE  
GESTION,  
ÉCONOMIE  
& SCIENCES

Johann De Almeida | Raphael Bauvin  
22/01/2020

## Le projet

L'objectif de ce projet est d'implémenter des algorithmes de coloration de graphe. Pour cela il a d'abord fallu implémenter les librairie utiles comme une librairie de modélisation de graphe.

Tout d'abord notre solution fonctionne uniquement sur les graphes non orientés.

Pour modéliser des graphes dans notre application nous avons gardé la méthode de stocker les données relative aux sommets et aux arcs sous forme de matrice d'adjacences. Cette matrice est stockée dans notre structure graphe qui est aussi composé d'un tableau de degrés qui sont les degrés de chaque sommet, le nombre de sommet et un tableau de coloration correspondant à la couleur de chaque sommet.

La structure a été le premier choix retenu sur le projet, le second choix a été la syntaxe d'écriture du fichier. Nous avons fait le choix d'une syntaxe simple, soit la matrice sans séparation entre les colonnes et un retour à la ligne pour chaque ligne. Cela nous a permis de réaliser un analyseur syntaxique plus simple par la suite mais aussi de comparer les résultats avec les autres étudiants de la classe puisque pour la majorité ils ont choisi la même syntaxe.

Le troisième choix important a été pris plus loin dans le projet lors de l'implémentation des algorithmes de coloration. Pour mieux gérer « les listes » d'élément tel que les sommets adjacents, nous avons créé une librairie pour le type Liste. Elle nous permet de simplifier l'écriture des algorithmes et la manipulations de nos données.

## Les bibliothèques développées

La première bibliothèque implémenter est celle des graphes. Nous avons commencé par les fonctions de base comme **l'initialisation** de graphe, **l'ajout** ou la **suppression** d'un arc dans le graphe et **l'affichage** du graphe. Nous nous sommes ensuite penchés sur la matrice du graphe, pour cela nous avons d'abords conçu une fonction qui **génère une matrice aléatoire**. Ensuite nous avons créé les fonctions de **lecture** et d'**écriture** d'un graphe dans un fichier. Pour ces fonctions nous avons dû ajouter des fonctions des fonctions plus secondaire comme **ouvrir un fichier**, obtenir le **nombre de sommet** du fichier et **vérifier** que le graphe soit bien non orienté.

Avant de passer à la coloration, nous avons besoin des degrés de chaque sommet pour l'algo de Welsh Powell, nous avons donc ajouter les fonctions de **mise à jour des degrés** des du graphe et de **récupération du degrés** d'un sommet.

Les algorithmes étant assez complexe à l'écriture en C dû à la gestion des tableaux et de leurs nombres d'éléments, nous avons créé une librairie de Liste (plus précisément une ArrayList). Nous y avons implémenté plusieurs fonctions qui nous ont permis de simplifier nos algorithmes :

- **initialiser** liste
- **ajouter** un élément
- **retirer** un élément
- **récupérer** un élément à un index donné
- **fusionner** 2 listes

- vérifier si un élément est **inclus** dans la liste
- **trier** par ordre croissant de la liste (tir par sélection)

La troisième bibliothèque que nous avons développée est celle de la coloration de graphe, elle a besoin des 2 précédentes pour fonctionner. La bibliothèque est principalement composée de l'algorithme de **Welsh Powell** et celui de **Glouton**, tous deux servant à proposer une coloration pour un graphe. Pour qu'ils fonctionnent la librairie dispose d'une fonction pour récupérer une **liste de sommets adjacent à un sommet** et pour récupérer **la liste des sommets d'un graphe par degrés décroissant**.

Afin de la compléter, nous sommes venus ajouter les fonction de **lecture, écriture, vérification** et **suppression** d'une coloration.

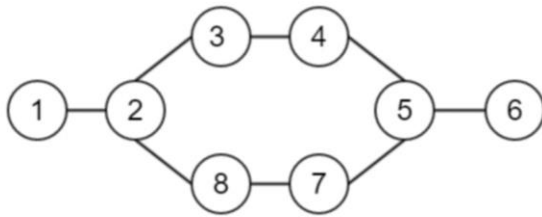
## Les algorithmes

Le premier algorithme implémenté est l'algorithme de Welsh Powell. Le principe est de récupérer la liste des sommets trier par ordre de degrés décroissant. On prend le premier sommet de la liste et on lui ajoute une couleur, ensuite on regarde le prochain sommet de la liste qui ne fait pas partie de sommets adjacents du sommet que l'on vient de colorer. On ajoute la même couleur à ce sommet et on regarde une nouvelle fois le prochain sommet qui n'est pas dans la liste des adjacents de ces 2 sommets. On répète plusieurs fois cette action jusqu'à qu'il n'y est plus de sommet qui ne sont pas adjacents à ceux coloré, et on passe au sommet suivant non coloré en lui ajoutant une nouvelle couleur. On répète ces actions jusque tous les sommets soient colorés. C'est ici l'ordre des sommets qui peut causer des trous dans la coloration et donc l'utilisation de couleur de trop dans certains cas.

Le second algorithme est basé sur le principe de « Greedy Search » aussi appelé algorithme Glouton. Cet algorithme est moins performant au niveau complexité mais permet d'obtenir une coloration plus optimale, c'est à dire avec moins de couleurs que l'algorithme de Welsh Powell. Le principe est de traiter chaque sommet au cas par cas. Pour cela on part simplement du premier sommet du graphe et on lui donne une couleur correspondant à la couleur 0. On passe au sommet suivant et on regarde quel est la couleur la plus petites qui n'est pas présente dans les couleurs de ses sommets adjacents. Pour le second sommet s'il est adjacent au sommet 0 se sera donc 1. On réalise cette recherche de la couleur minimale pour colorer chaque sommet. Cependant comme expliqué plus en dessous dans le résultat, cet algorithme n'est pas sans défaut non plus.

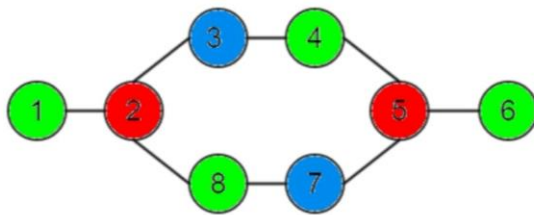
## Le résultat

Pour le premier test, nous nous sommes basés sur ce graphe à 8 sommets. Nous avons rédigé sa matrice d'adjacences dans un fichier texte, nous l'avons ensuite lu à l'aide de nos fonctions développées et nous l'avons affiché voici le résultat :



0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	1
0	1	0	1	0	0	0	0
0	0	1	0	1	0	0	0
0	0	0	1	0	1	1	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	1
0	1	0	0	0	0	1	0

Nous avons ensuite exécuté l'algorithme de Welsh Powell pour effectuer sa coloration, ce qui donne ce résultat :



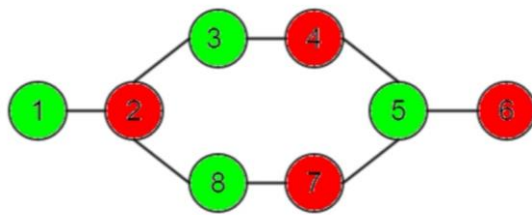
```

Les colorations par Welsh Powell :
Sommet 1 -> couleur 1
Sommet 2 -> couleur 0
Sommet 3 -> couleur 1
Sommet 4 -> couleur 2
Sommet 5 -> couleur 0
Sommet 6 -> couleur 1
Sommet 7 -> couleur 1
Sommet 8 -> couleur 2

Coloration valide !

```

Comme on peut l'apercevoir, la coloration est valide mais non optimale, en effet il est possible de colorer ce graphe avec seulement 2 couleurs. Voici donc le résultat quand on appelle notre algorithme de Glouton (Greedy search) :



```

Les colorations par Glouton :
Sommet 1 -> couleur 0
Sommet 2 -> couleur 1
Sommet 3 -> couleur 0
Sommet 4 -> couleur 1
Sommet 5 -> couleur 0
Sommet 6 -> couleur 1
Sommet 7 -> couleur 1
Sommet 8 -> couleur 0

Coloration aussi valide !

```

On constate donc que sur ce graphe si, le second algorithme est plus efficace pour colorer le graphe avec le moins de couleurs.

Maintenant les comparaisons du nombre de couleur avec 50 et 100 sommets :

50 sommets :

Sommet 48 -> couleur 4  
Sommet 49 -> couleur 1  
Sommet 50 -> couleur 4

Nombre de couleurs : 13

Sommet 48 -> couleur 11  
Sommet 49 -> couleur 10  
Sommet 50 -> couleur 11

Nombre de couleurs : 12

100 sommets :

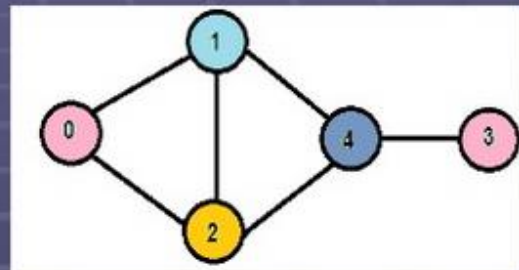
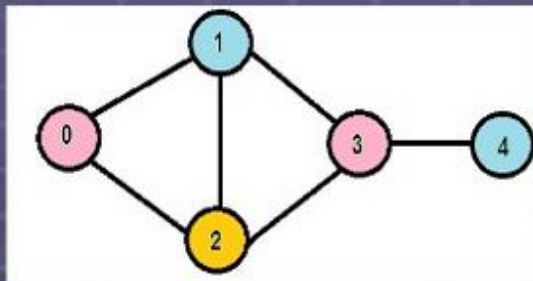
Sommet 97 -> couleur 14  
Sommet 98 -> couleur 14  
Sommet 99 -> couleur 6  
Sommet 100 -> couleur 7

Nombre de couleurs : 20

Sommet 97 -> couleur 17  
Sommet 98 -> couleur 18  
Sommet 99 -> couleur 6  
Sommet 100 -> couleur 19

Nombre de couleurs : 22

## Analysis of Basic Algorithm



Comme on peut voir sur ce test à 100 sommets, l'algorithme de Glouton c'est avéré moins efficace. Ce problème est dû à l'ordre des sommets, comparé à Welsh Powell nous ne trions pas les sommets dans un ordre précis. Ce qui peut changer la coloration, comme dans l'exemple ci-dessous où l'algorithme est appliqué en commençant la coloration par la gauche dans un premier temps, puis en appliquant l'algorithme par la droite :

## Les références

Welsh Powell :

<https://m6colorationgraphes.wordpress.com/2015/11/30/partie-iii-la-coloration-par-welsh-powell/>

[https://www.youtube.com/watch?v=-4\\_F5OTFoyk](https://www.youtube.com/watch?v=-4_F5OTFoyk)

Glouton (Greedy search) :

[https://fr.wikipedia.org/wiki/Algorithme\\_glouton](https://fr.wikipedia.org/wiki/Algorithme_glouton)

[https://www-npa.lip6.fr/~blin/Enseignement\\_files/ALRE/201602\\_ALRE\\_Coloration.pdf](https://www-npa.lip6.fr/~blin/Enseignement_files/ALRE/201602_ALRE_Coloration.pdf)

<https://www.slideshare.net/PriyankJain26/graph-coloring-48222920>