C.U.F.R. J-F. Champollion — L2 Informatique S5
Albi — 2015

TP 4 (1 séance)
Coloration des sommets d'un graphe

Le but de ce TP est de mettre en œuvre les algorithmes simples de coloration des sommets d'un graphe non orienté .

1. Coloration naïve

Compléter la méthode suivante de la classe *Liste*

- *public int mini(Liste L) :* qui détermine le plus petit entier >=1 qui n'appartient pas à la liste L. On se servira de cette fonction pour déterminer la plus petite couleur n'appartenant pas à la liste des couleurs interdites.

Compléter la méthode suivante de la classe *GrapheNonOrienteList*

- *public int[] colorNaive()* : qui détermine une coloration du graphe G par l'algorithme naïf.

2. Coloration gloutonne

Compléter la méthode suivante de la classe *GrapheNonOrienteList*

- *public Liste noyau (Liste L) :* qui effectue le calcul du noyau d'un ensemble de sommets, c'est à dire une liste maximale de sommets ne contenant pas de sommets adjacents.
- *public int[] colorGlouton()* : qui détermine une coloration du graphe G par l'algorithme glouton.

3. Coloration de Welsh et Powell

On a vu en TD qu'on diminue sensiblement le nombre de couleurs nécessaires en traitant les sommets dans l'ordre décroissant des degrés.

On peut trier une liste d'objets à l'aide de la fonction python *sorted* en donnant en paramètre la partie de l'objet sur laquelle porte le tri avec la syntaxe :

key=lambda x: "partie de x sur laquelle porte le tri". L'option « reverse » permet d'obtenir un tri dans l'ordre décroissant.

Pour trier les sommets dans l'ordre décroissant des degrés, on peut donc créer la liste Deg de tous les couples (sommet,degré du sommet) et la trier à l'aide de la commande :
sorted(Deg, key=lambda x: x[1],reverse=True)
(Cette construction est donnée dans la fichier fourni)

Compléter la méthode suivante de la classe *GrapheNonOrienteList*

- *public int[] colorWP()* : qui détermine une coloration du graphe G par l'algorithme de Welsh et Powell.