FINAL

Aucun document n'est autorisé. Les résultats intermédiaires non démontrés pourront être utilisés tout au long de l'examen. Le barème prendra en compte le niveau de difficulté de chaque exercice.

Exercice 1 Coloration de graphe (sommets)

Le graphe $G(V,E)$ présenté dans la figure 1 sera utilisé tout au long de cet exercice :

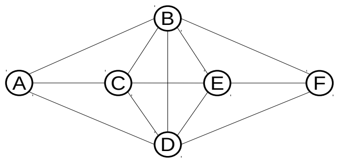

FIGURE 1 – Graphe 1

1. Rappelez la définition générale du nombre chromatique d'un graphe.
2. Calculez la densité du graphe de la figure 1.
3. Le graphe de la figure 1 représente-t-il une cliques ?. Justifiez votre réponse en fonction de la densité calculée précédemment.
4. Donnez trois cliques $G'(V',E')$, $G''(V'',E'')$, et $G'''(V''',E''')$ dont $|V'|$, $|V''|$, $|V'''| \geqslant 4$.
5. Quel est la borne inférieure du nombre de couleurs nécessaire pour colorer le graphe de la figure 1 ?. Justifiez votre réponse.
6. Quel est la borne supérieure du nombre de couleurs nécessaire pour colorer le graphe de la figure 1 ?. Justifiez votre réponse.
7. En déduire un encadrement du nombre chromatique de ce graphe.

A) Algorithme de coloration *Welsh* et *Powell* : l'algorithme de *Welsh & Powell* est un algorithme populaire proposé pour une coloration séquentielle des sommets d'un graphe. Voici la procédure :

1. Repérer le degré de chaque sommet.
2. Ranger les sommets par ordre de degrés décroissants (plusieurs possibilités en cas d'égalité de degrés).
3. Ranger les couleurs par ordre croissant (les couleurs sont des naturels non-nuls).
4. Attribuer au premier sommet i de la liste la première couleur.
5. Suivre la liste en attribuant la même couleur au premier sommet j non adjacent de i.
6. Suivre (si possible) la liste jusqu'au prochain sommet k qui ne soit adjacent ni à i ni à j.
7. Continuer jusqu'à ce que la liste soit finie.
8. Prendre une deuxième couleur pour le premier sommet l non encore coloré de la liste.
9. Répéter les opérations 4 à 7.
10. Continuer jusqu'à avoir coloré tous les sommets.

Utilisez l'algorithme de *Welsh & Powell* pour colorer le graphe de la figure 1 en remplissant le tableau 1 :

Sommets						
Degrés						
Couleur utilisée						
Nombre de couleurs disponibles						

TABLE 1 – *Welsh & Powell*

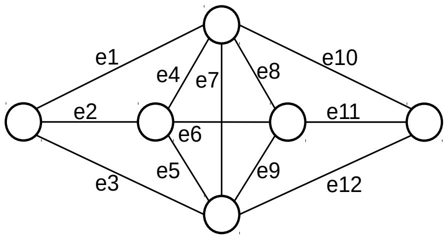

FIGURE 2 – Graphe 2

B) *Welsh* et *Powell* pour les arcs : dans cet exercice, nous traitons le problème de coloration des arcs du graphe de la figure 2. Proposez la version de l'algorithme de *Welsh* et *Powell* pour colorer les arcs du graphe de la figure 2 puis donnez la solution obtenue.

Exercice 2 Recherche Tabou

Dans cet exercice, nous allons étudier le problème du *minimum spanning tree* : *"A minimum spanning tree or minimum weight spanning tree is a subset of the edges of* ***a connected****, edge-weighted undirected graph that* ***connects all the vertices together****,* ***without any cycles*** *and with the minimum possible total edge weight."*

Une compagnie souhaite fournir un service de TV à 5 domiciles $\{A,B,C,D,E\}$ en les liant par un câble coaxial. Lier deux domiciles induit un coût de liaison. Voici les coûts de liaison des domiciles deux-à-deux : $\{AB:20, AC:10, AD:15, BE:30, CE:5, CD:25, DE:40\}$.

1. Donnez le graphe pondéré résultant noté $G(V,E)$.

2. Donnez la solution optimale (le sous graphe G' du graphe G) du problème du *minimum spanning tree* (connecter tous les sommets sans créer de cycle en minimisant le coût des liaisons) en donnant la valeur du coût total obtenu.

Après l'étude du projet, deux contraintes viennent de s'ajouter au problème :

— **contrainte 1** : on ne peut lier A à D que si et seulement si D et E sont liés. Sinon, une pénalité de 100 s'ajoute au coût total.

— **contrainte 2** : au maximum, un des trois liens suivants peut être inclus : AD, CD, et AB. Sinon, une pénalité de 100 s'ajoute au coût total si deux des trois liens sont inclus, et une pénalité de 200 si tous les trois sont inclus.

Donnez la nouvelle valeur du coût total de votre solution optimale en considérant les nouvelles contraintes.

Algorithme de recherche Tabou : cet algorithme a pour objectif de trouver le *minimum spanning tree*. Pour chaque itération, partant d'une solution initiale (G' résultat de la question 2), et en reprenant le graphe initial (G résultat de la question 1) on parcours un par un les arcs existants dans le graphe G et non inclus dans le graphe G'. On intègre l'arc en question puis on supprime un par un les arcs formant un cycle et on calcule le coût total après chaque suppression. Après avoir testé toutes les possibilités d'intégration, on garde la meilleur solution obtenue. Le dernier arc intégré donnant la meilleur solution devient Tabou (ne peut être supprimé). Le résultat de chaque itération est donné dans un tableau comme suit :

Ajouter	Supprimer	Coût total	Mouvement Tabou
BE	CE	275	
⋮	⋮	⋮	⋮

1. Donnez la solution obtenu après trois itérations ainsi que les résultats intermédiaires de chaque itération (tableau et graphe) en continuant l'application de cet l'algorithme avec :
   a. durée Tabou = 2

   b. durée Tabou = 1
2. Qu'en déduisez-vous ?