Révisions

Exercice 1 – Programmation linéaire

Chaque jour, il est conseillé de consommer au moins 2500 kilo-calories, et au moins $80g$ de gras. Un étudiant (sans le sou) cherche à optimiser ses prochaines courses au supermarché en calculant le coût minimum de ce qu'il doit acheter pour couvrir ses besoins d'alimentation basiques quotidiens. Une portion de nouilles instantanées coûte 2 euros, une demi-pizza 5 euros et une portion de riz 1 euro.

	Calories	Gras
Nouilles instantanées	525	20
Demi-pizza	700	20
Riz (portion)	400	0

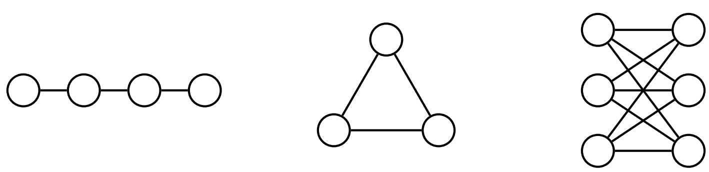
1. Modéliser le problème comme un programme linéaire.
2. Écrire son dual.
3. L'étudiant mange finalement quatre parts de nouilles et une part de riz. Est-ce satisfaisant pour sa santé ? Est-ce optimal pour son porte-monnaie ?

   Indication : Pour déterminer l'optimalité, on pourra utiliser le théorème des écarts complémentaires.

Exercice 2 – Coloration d'arêtes

Dans cet exercice, on s'intéresse au problème de *coloration d'arêtes*. Étant donné un graphe G, le but est de déterminer le plus petit nombre de couleurs nécessaire pour pouvoir assigner une couleur à chaque arête, de sorte que deux arêtes qui partagent une extrémité reçoivent des couleurs différentes.

1. Donner une coloration d'arêtes optimale de chacun des graphes suivants. *On ne demande pas de justifier l'optimalité des colorations obtenues.*



On rappelle que $\Delta(G)$ est le *degré maximum* du graphe G, c'est-à-dire le nombre maximum de voisins que peut avoir un sommet de G

2. Montrer qu'une coloration d'arêtes d'un graphe de degré maximum Δ utilise nécessairement au moins Δ couleurs.

On considère l'algorithme glouton suivant : on colore les arêtes une à une avec la plus petite couleur disponible (c'est-à-dire qui n'est pas présente sur une arête incidente à l'arête considérée).

3. Montrer que l'algorithme glouton utilise toujours au plus $2\Delta - 1$ couleurs quand on l'applique à un graphe de degré maximum Δ.
4. Construire un graphe où l'algorithme glouton ne renvoie pas une coloration d'arêtes optimale.
5. Quel est le pire exemple que vous pouvez construire ? Autrement dit, quel est le nombre maximum de couleurs que vous pouvez forcer l'algorithme glouton à utiliser sur un graphe de degré maximum Δ ?

Exercice 3 – Facteur commun

Un *facteur* d'un mot X est un ensemble de lettres de X *pas forcément* consécutives. Par exemple, les mots $A = ababab$ et $B = bbbaab$ contiennent tous les deux le facteur $bbab$. Par contre, $aabb$ est un facteur de A mais pas de B. Le but est de trouver le plus long facteur qui apparaît à la fois dans le mot $A = a_1 \cdots a_n$ et dans le mot $B = b_1 \cdots b_m$.

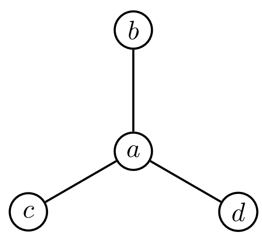
On note $F[i, j]$ la longueur du plus long facteur commun des mots $a_1 \cdots a_i$ et $b_1 \cdots b_j$. Par convention, on note $F[i, 0] = 0$ et $F[0, i] = 0$.

1. Que valent $F[2, 4]$ et $F[5, 3]$ si $A = abbbaa$ et $B = bbaaba$?
2. Donner une équation de récurrence satisfaite par $F[i, j]$ si $a_i = b_j$.
3. Donner une équation de récurrence satisfaite par $F[i, j]$ si $a_i \neq b_j$.
4. En déduire un algorithme de programmation dynamique pour résoudre le problème. Quelles sont ses complexités en temps et en espace ?
5. Pouvez-vous améliorer la complexité spatiale de votre algorithme ?
6. Modifier votre algorithme pour pouvoir déterminer le plus long facteur commun à A et B (et pas seulement sa longueur).

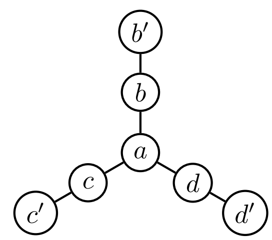
Exercice 4 – Gestion de salles

Dans cet exercice, on considère le problème d'*affectation de salles*. Une *instance* de ce problème est une liste de couples $[(d_1, f_1), \ldots, (d_n, f_n)]$ représentant n cours ; le i-ème cours commençant à l'heure d_i et finissant à l'heure f_i. On souhaite affecter une salle à chaque cours en minimisant le nombre total de salles utilisées.

Modélisation Pour résoudre ce problème, on modélise les instances comme suit : on construit un graphe dont les sommets représentent les cours, et où les sommets i et j sont reliés par une arête si les créneaux des cours correspondants se chevauchent, autrement dit si $[d_i, f_i] \cap [d_j, f_j] \neq \varnothing$. On cherche alors une coloration des sommets de ce graphe.



Graphe G



Graphe H

1. Donner une instance du problème d'affectation de salles modélisée par le graphe G.
2. Montrer que dans toute instance modélisée par G, l'un des créneaux correspondant à b, c ou d est inclus dans le créneau de a.
3. En déduire qu'aucune instance ne peut être modélisée par le graphe H.

Résolution Soit G un graphe obtenu lors de la modélisation d'une instance. On considère l'algorithme glouton suivant. On parcourt les sommets de G par ordre croissant de d_i, c'est-à-dire en partant du sommet qui représente le cours commençant plus tôt, jusqu'au sommet représentant le cours commençant le plus tard. À chaque nouveau sommet considéré, on attribue la plus petite couleur disponible (c'est-à-dire qui n'apparaît pas sur un sommet voisin déjà coloré).

4. Soit v un sommet qui reçoit la couleur i pendant l'exécution de l'algorithme glouton. Montrer que v est contenu dans une clique (pas forcément maximum) de taille i dans G.
5. En déduire que cet algorithme glouton renvoie une ω-coloration, où ω est la taille d'une clique maximum dans G.
6. Cet algorithme glouton est-il optimal ?