CONCOURS COMMUN INP

ÉPREUVE MUTUALISÉE AVEC E3A-POLYTECH

ÉPREUVE SPÉCIFIQUE - FILIÈRE MP

INFORMATIQUE

Durée : 4 heures

N.B. : le candidat attachera la plus grande importance à la clarté, à la précision et à la concision de la rédaction. Si un candidat est amené à repérer ce qui peut lui sembler être une erreur d'énoncé, il le signalera sur sa copie et devra poursuivre sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.

RAPPEL DES CONSIGNES

- *Utiliser uniquement un stylo noir ou bleu foncé non effaçable pour la rédaction de votre composition ; d'autres couleurs, excepté le vert, bleu clair ou turquoise, peuvent être utilisées, mais exclusivement pour les schémas et la mise en évidence des résultats.*
- *Ne pas utiliser de correcteur.*
- *Écrire le mot FIN à la fin de votre composition.*

Les calculatrices sont interdites.

Le sujet est composé de trois parties indépendantes.

Partie I - Coloration de graphes

L'objectif de cette partie est de proposer une implémentation en **Python** d'une solution au problème de coloration d'un graphe.

I.1 - Définitions et propriétés

Soit $G = (S, A)$ un graphe fini non orienté avec S son ensemble de sommets et A son ensemble d'arêtes. On suppose que le graphe est simple c'est-à-dire qu'il ne comporte pas de boucles et que chaque paire de sommets est reliée par au plus une arête. On note n, le cardinal de l'ensemble S. Les sommets sont numérotés de 0 à $n-1$. Étant donné un entier naturel k, une k-coloration des sommets de G est une application $c : S \to \{0, 1, ..., k-1\}$ telle que pour chaque arête $\{x, y\}$ d'extrémités x et y, $c(x) \neq c(y)$. Si $c(x) = i$, on considèrera que la couleur i est affectée au sommet x. Si G admet une k-coloration, il est k-coloriable. On définit le nombre chromatique $\chi(G)$ d'un graphe G fini par $\chi(G) = \min\{k \in \mathbb{N}, \text{G est } k\text{-coloriable}\}$.
Une clique est un sous-ensemble de sommets du graphe, adjacents 2 à 2. On dit qu'un graphe est complet si il est une clique. On notera K_p le graphe complet à p sommets.
On pose $\omega(G) = \max\left\{p \in \mathbb{N} | K_p \text{ est une clique de G}\right\}$, avec $\mathbb{N}$ l'ensemble des entiers naturels.

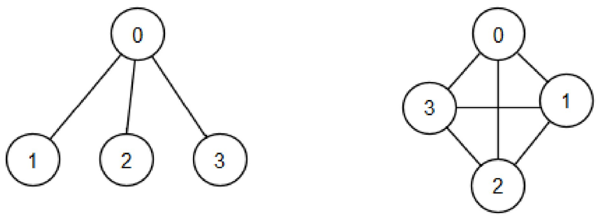

Figure 1 - de gauche à doite, le graphe G_1 et le graphe K_4

Q1. Le graphe G_1 de la **figure 1** ci-dessus est-il 2-coloriable ? Justifier votre réponse.

Q2. Pour un entier naturel $n \geq 1$, déterminer le nombre chromatique du graphe K_n.

Q3. Montrer que pour tout graphe G à n sommets, on a $\omega(G) \leq \chi(G) \leq n$.

I.2 - Algorithmique et programmation en Python (Informatique Commune)

La coloration d'un graphe G avec $\chi(G)$ couleurs est un problème complexe. Dans cette sous-partie, nous présentons une heuristique permettant de construire une coloration d'un graphe donné.
Dans la suite, on implémente un graphe par **un dictionnaire (type `dict` en Python), contenant les listes d'adjacence des sommets**. Les clés du dictionnaire sont les numéros des sommets et la valeur correspondant à la clé `i` du dictionnaire est la liste d'adjacence du sommet numéro `i`.
Notons qu'il serait ici possible d'implémenter le graphe par des listes d'adjacence dans un tableau, dans la mesure où les sommets sont numérotés de 0 *à* $n-1$.

Q4. Définir en Python le dictionnaire `d1` correspondant au graphe G_1.

Q5. Écrire une fonction Python `degres_sommets(d)` qui prend en paramètre un dictionnaire `d` représentant un graphe et renvoie une liste de couples `(di,i)`, `i` étant le numéro d'un sommet parcourant l'ensemble $\{0, ..., n-1\}$ et `di` le degré du sommet `i`.

Q6. On suppose qu'on dispose d'une fonction Python `tri(l)` qui trie une liste de couples `l` dans l'ordre décroissant par rapport à la première composante du couple. En déduire une fonction `tri_degres(d)` qui prend en paramètre un dictionnaire `d` représentant un graphe et renvoie une liste contenant les numéros des sommets, triés par degrés décroissants.

Q7. Écrire une fonction Python `test(d,c)` qui prend en paramètre un dictionnaire `d` représentant un graphe G, un dictionnaire `c` dont les clés représentent les sommets du graphe G et les valeurs, leurs couleurs. La fonction renvoie `True` si `c` est une 2-coloration pour G.

On considère ci-dessous, l'algorithme de coloriage de Welsh-Powel.

Algorithme 1 : Welsh-Powel (coloration de graphe)

Entrée : un graphe `G` à `n` sommets
Sortie : liste d'entiers contenant en position `i` la couleur du sommet numéro `i`

Début
- Ordonner les sommets selon les degrés décroissants dans une liste `li` ;
- `colorie` : dictionnaire vide qui à terme, associera à chaque clé `i`, la couleur du sommet `i` ;
- **Tant qu'** *il reste des sommets à colorier* **faire**
  - Chercher dans `li` le premier sommet non colorié et le colorier avec la plus petite couleur `c` non utilisée ;
  - Colorier avec cette même couleur, en respectant leur ordre dans `li`, tous les sommets non coloriés et non adjacents à des sommets de couleur `c` ;
- **Fin**
- Retourner `colorie`

Fin

Q8. Que contient `colorie` si on déroule l'algorithme de coloriage ci-dessus avec le graphe G_1 en entrée ?

Q9. Écrire une fonction Python `adjacent(d,dc,s,c)` qui prend en paramètre un graphe représenté par un dictionnaire `d`, un dictionnaire `dc` contenant la couleur des sommets coloriés, le numéro d'un sommet `s`, une couleur `c` et renvoie `True` si le sommet `s` est adjacent à l'un des sommets de couleur `c`, `False` sinon.

Q10. Proposer une implémentation en Python de l'algorithme de Welsh-Powel.

Application

Le tableau ci-dessous représente les liens d'amitiés entre huit étudiants : Alice (A), Béatrice (B), Carl (C), David (D), Eloïs (E), Fanny (F), Gary(G) et Hedge (H).

Prénom	A	B	C	D	E	F	G	H
Ami·e avec	B,C,G	A,C,E,F	A, B	E,F	B,D,F	B,D,E,H	A,H	F,G

On souhaite créer des groupes de travail. Dans le contexte de l'application, un groupe contient au moins 2 étudiants tel que chaque étudiant soit dans un groupe différent de celui de ses amis.
Q11. Modéliser la situation par un graphe et en déduire une solution.