ÉCOLE NORMALE SUPÉRIEURE DE LYON

Concours d'admission session 2021

Filière universitaire : Second concours

COMPOSITION D'INFORMATIQUE

Durée : 3 heures

L'utilisation des calculatrices n'est pas autorisée pour cette épreuve.

⋆ ⋆ ⋆

Le sujet comporte trois parties indépendantes.

Notations

Si $i, j \in \mathbb{N}$, on notera $[i, j]$ l'ensemble des entiers $\{i, i+1, \ldots, j\}$.

Graphes
Un *graphe* (non-orienté) G est donné par un ensemble de sommets V, et un ensemble d'arêtes E reliant certains sommets. Un tel graphe sera noté $G = (V, E)$, avec V l'ensemble des sommets, habituellement numérotés $\{1, 2, \ldots\}$, et E l'ensemble des arêtes, une arête étant de la forme $\{i, j\} \subseteq V$. Deux sommets i, j sont appelés *voisins* si $\{i, j\} \in E$, c'est-à-dire s'ils sont reliés par une arête.

Un *chemin* de i à j dans un graphe $G = (V, E)$ est une séquence de sommets $i_0, i_1, \ldots, i_k$ telle que $i_0 = i$, $i_k = j$, et pour tout $p \in [0, k-1]$, on a $\{i_p, i_{p+1}\} \in E$. La *longueur* d'un tel chemin est k.

Un graphe $G = (V, E)$ est *connexe* si pour tous sommets $i \neq j$ de V, il existe un chemin de i à j dans G.

Matrice d'adjacence:
Soit $G = (V, E)$ un graphe de taille n, avec $V = [1, n]$. On définit la *matrice d'adjacence* de G, notée $A(G)$ de la manière suivante: c'est la matrice de taille $n \times n$, dont le coefficient a_{ij} vaut 1 si $\{i, j\} \in E$ et 0 sinon. On peut remarquer que cette matrice est symétrique (c'est-à-dire $a_{ij} = a_{ji}$ pour tous $i, j \in [1, n]$), et que les coefficients diagonaux sont égaux à 0 (c'est-à-dire $a_{ii} = 0$ pour tout $i \in [1, n]$).

Produit de matrices:
Rappelons que l'on peut définir le produit de 2 matrices de la manière suivante. Si $A = (a_{ij})$ est de taille $n \times m$, et $B = (b_{ij})$ est de taille $m \times t$, leur produit $C = A \times B$ est de de taille $n \times t$, et pour chaque $i \in [1, n]$ et $j \in [1, t]$, son coefficient c_{ij} est défini par

$$c_{ij} = \sum_{k=1}^{m} a_{ik} \times b_{kj} = (a_{i1} \times b_{1j}) + (a_{i2} \times b_{2j}) + \cdots + (a_{im} \times b_{mj}).$$

Il est également possible de définir le produit de matrices en utilisant d'autres opérations que la somme et le produit classiques sur les coefficients. Si on note $(\oplus, \otimes)$ les nouvelles opérations de somme et de produit respectivement, le coefficient c_{ij} de la matrice $C = A \times B$ s'écrit alors:

$$c_{ij} = \bigoplus_{k=1}^{m} a_{ik} \otimes b_{kj} = (a_{i1} \otimes b_{1j}) \oplus (a_{i2} \otimes b_{2j}) \oplus \cdots \oplus (a_{im} \otimes b_{mj}).$$

Si A est une matrice carrée et k un entier, on note $A^k = \overbrace{A \times A \times \cdots \times A}^{k \text{ fois}}$, le produit de k copies de la matrice A.

Partie 1: Itérations de la matrices d'adjacence

Matrices booléennes

On considère des coefficients booléens: 0 ou 1. L'opération de somme sur ces coefficients est le "ou" booléen, noté $\vee$, et l'opération de produit est le "et" booléen, noté $\wedge$. Ainsi dans toute cette section, le produit de deux matrices booléennes reste une matrice booléenne.

Question 1. Dessinez le graphe dont la matrice d'adjacence est $A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

Question 2. Soit B la matrice suivante: $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$

Que vaut le produit $C = A \times B$, où A est la matrice de la question 1 ? On rappelle que les opérations de somme et produit utilisées sont $\vee$ et $\wedge$ respectivement.

Question 3. Écrire un algorithme qui décide si une matrice booléenne donnée en argument est la matrice d'adjacence d'un graphe.

On fixe un graphe quelconque $G = (V, E)$ pour le reste de la section **Matrices booléennes**. Soit $n = |V|$ le nombre de sommets de G, et $A(G)$ sa matrice d'adjacence.

Question 4. Montrer que pour tout entier $k \geq 1$, le coefficient (i, j) de $A(G)^k$ vaut 1 si et seulement s'il existe un chemin de i à j de longueur k dans G.

Question 5. Soit I la matrice identité $n \times n$ comportant des 1 sur la diagonale et 0 partout ailleurs. Soit $B = I + A(G)$, et $k \geq 1$. Montrer que le coefficient (i, j) de B^k vaut 1 si et seulement s'il existe un chemin de i à j de longueur au plus k dans G.

Question 6. Soit $k \geq n$, montrer que les trois propositions suivantes sont équivalentes:

1. G est connexe.
2. La matrice B^k ne contient que des 1.
3. La première ligne de la matrice B^k ne contient que des 1.

Question 7. Supposons que l'on dispose d'un algorithme effectuant le produit de matrices boolénnes $n \times n$ en temps $O(n^\alpha)$, avec $2 < \alpha < 3$. En s'inspirant de la question précédente, écrire un algorithme permettant de déterminer si G est connexe, en temps $O(n^\alpha \log n)$.

Indice: il est utile de choisir une puissance de 2 pour l'entier k, et de remarquer que si $i \in \mathbb{N}$, on a $B^i \times B^i = B^{2i}$

Somme et produit sur les entiers

On considère maintenant que les matrices sont à coefficients entiers, avec les opérations $+$ et $\times$ usuelles sur les entiers. Les coefficients des matrices d'adjacence restent 0 et 1, mais on peut maintenant obtenir d'autres entiers en multipliant ces matrices entre elles.

Question 8. Soit G un graphe et $k \geq 1$ un entier. Montrer que le coefficient (i,j) de la matrice $A(G)^k$ représente le nombre de chemins de i à j de longueur k dans G.

Question 9. Donner une caractérisation des graphes G tels que tous les coefficients de $A(G)^k$ restent bornés quand k tend vers l'infini.

Autres opérations

Question 10. Soit G un graphe et n son nombre de sommets. On va maintenant prendre des coefficients dans l'ensemble $\mathbb{N} \cup \{\infty\}$, et utiliser le minimum (min) de deux entiers comme opération de somme, et la somme $(+)$ de deux entiers comme opération de produit. Pour tout $x \in \mathbb{N} \cup \{\infty\}$, on prend comme convention $\min(x,\infty) = \min(\infty, x) = x$ et $x + \infty = \infty + x = \infty$.

Soit A la matrice obtenue à partir de $A(G)$ en remplaçant les coefficients nuls non diagonaux par ∞. Soit $C = A^n$, calculée en utilisant les opérations $(\min, +)$ pour somme et produit respectivement. Montrer que pour tous $i, j \in [1, n]$, le coefficient c_{ij} de C contient la longueur du plus court chemin de i à j.

Question 11. On veut être capable de décider en calculant une matrice A^k s'il existe un chemin de longueur **paire** et au plus k dans un graphe G.

Montrer qu'il est possible de choisir un ensemble **fini** F de coefficients, muni d'opérations $\oplus : F^2 \to F$ et $\otimes : F^2 \to F$, ainsi qu'une manière de représenter tout graphe G par une matrice A à coefficients dans F, pour satisfaire la propriété ci-dessus. C'est-à-dire que pout tout entier $k \geq 1$, le coefficient (i,j) de la matrice A^k, calculée en utilisant les opérations $\oplus$ et $\otimes$, nous permette de dire s'il existe un chemin de longueur paire et au plus k, de i à j dans G.

Partie 2: Allumer la lumière

Soit $G = (V, E)$ un graphe, avec $V = [1, n]$. On considère que les sommets de G sont des ampoules munies d'un interrupteur. Actionner un interrupteur change l'état (allumé ou éteint) de son ampoule ainsi que de toutes les ampoules voisines dans le graphe.

Initialement, toutes les ampoules de V sont éteintes, le but est d'atteindre un état où elles sont toutes allumées.

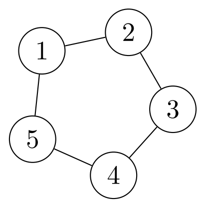
Question 12. Montrer que l'ordre des actions n'importe pas, et qu'il n'est jamais nécessaire d'actionner plusieurs fois le même interrupteur. En déduire qu'une stratégie pour allumer toutes les ampoules peut se décrire par un élément de $\{0,1\}^n$.

On dira qu'une stratégie $x \in \{0,1\}^n$ est *gagnante* si elle allume effectivement toutes les ampoules.

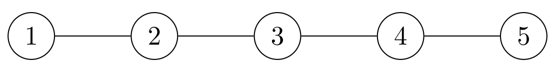
Question 13. On note $A = (a_{ij})$ la matrice d'adjacence du graphe G, à coefficients dans $\{0,1\}$, et $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ une stratégie. Donner une condition sur les a_{ij} et les x_i exprimant le fait que la stratégie x est gagnante.

Question 14. Supposons que l'on dispose d'une fonction $\mathrm{bit}(x,i) : \mathbb{N}^2 \to \{0,1\}$ nous donnant le i^{eme} bit d'un entier x, en une étape de calcul. Écrire un algorithme exhaustif, c'est-à-dire testant toutes les stratégies possibles, qui prend en entrée un graphe, et donne une stratégie gagnante s'il en existe une, ou répond "Impossible" s'il n'en existe pas. Quelle est la complexité de cet algorithme ?

Question 15. Montrer qu'il est possible d'allumer toutes les ampoules si le graphe est un cycle C_n à n sommets, pour tout $n > 1$. Ci-dessous C_5 est donné en exemple.



Question 16. Montrer qu'il est possible d'allumer toutes les ampoules si le graphe est une ligne L_n, pour tout $n > 1$. Ci-dessous L_5 est donné en exemple.



Partie 3: Réseau d'ordinateurs

On considère un nombre impair $N = 2n+1$ d'ordinateurs, numérotés de 1 à N, et branchés ensemble en réseau. Parmi ces ordinateurs, au moins $n+1$ fonctionnent correctement, et sont appelés *bons*. Les autres ordinateurs sont *mauvais*. La seule opération autorisée sur ces ordinateurs est la suivante: pour tous $i, j \in [1, N]$, on peut demander à l'ordinateur i si l'ordinateur j est bon ou mauvais. Si l'ordinateur i est bon, il répondra la vérité. Si au contraire l'ordinateur i est mauvais, il peut répondre oui ou non de manière imprévisible. On pourra noter $Q(i,j)$ cette question posée à l'ordinateur i, sur le statut de l'ordinateur j, avec $Q(i,j) = 1$ si la réponse est "bon" et $Q(i,j) = 0$ si la réponse est "mauvais".

Dans la suite, la complexité d'un algorithme sera le nombre de questions posées.

Question 17. Etant donné un ordinateur j, donner un algorithme pour savoir à coup sûr s'il est bon. Quelle est la complexité dans le pire cas ?

Question 18. Donner un algorithme naïf permettant de trouver à coup sûr un bon ordinateur. Quelle est sa complexité dans le pire cas ?

Dans la suite, on va tenter de trouver un meilleur algorithme.

Question 19. Expliquer pourquoi il n'est jamais utile de poser une question de la forme $Q(i, i)$.

Question 20. Montrez que si $Q(i, j) = 0$, alors retirer i et j de l'ensemble des ordinateurs préserve la propriété que strictement plus de la moité des ordinateurs sont bons.

Question 21. Supposons que l'on parvient à construire une séquence d'ordinateurs $i_1, i_2, \ldots, i_k$, tel que pour chaque $p \in [1, k-1]$, on a $Q(i_p, i_{p+1}) = 1$, et supposons de plus qu'il existe $p \in [1, k]$ tel que i_p est bon. Montrer que dans ce cas i_k est bon.

Question 22. Déduire des deux questions précédentes un algorithme de complexité linéaire $O(N)$ pour trouver à coup sûr un bon ordinateur. Quelle est la complexité exacte de cet algorithme dans le pire cas ?