Oral d'informatique - MPI (3h30)

Informations préliminaires

Ce document comporte trois par parties.

- La partie 1 constitue le *sujet à traiter*. Les réponses attendues sont généralement courtes et doivent être reportées sur la *fiche réponse à compléter*.
- La partie 2 est la *fiche réponse à compléter*. Elle comporte une valeur u_0 personnelle, inscrite au début de votre fiche, qui sert de donnée d'entrée pour exécuter vos programmes. Sur cette fiche, vous reporterez vos réponses correspondant à cette valeur précise u_0. Cette fiche doit être remise à l'examinateur à la fin de l'épreuve.
- La partie 3 est une *fiche réponse type* qui fournit des réponses attendues pour une donnée d'entrée $\tilde{u}_0$ particulière, notée ainsi pour éviter toute confusion avec u_0. Cette fiche est destinée à vous aider et à vérifier le résultat de vos programmes en les testant avec $\tilde{u}_0$ au lieu de u_0.

Le temps de préparation est de 3 heures. Un échange de 30 minutes avec l'examinateur permettra d'évaluer votre travail. Quand la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. *Vous ne devez en aucun cas recopier le code de vos procédures.* Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, par exemple : $O\left(n^2\right)$, $O(n \log n)$, ...Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de t*ester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main ou bien à l'aide de la fiche réponse type fournie.* Enfin, il est recommandé de lire l'intégralité du sujet avant de commencer afin d'effectuer les bons choix de structures de données dès le début.

Les seuls langages de programmation autorisés sont les langages C et OCaml. Pour traiter le sujet, vous utiliserez, à votre choix, un seul de ces langages.

Partie 1

L'objet de ce sujet est de simuler la propagation de populations de bactéries. L'univers [1] est un tableau bi-dimentionnel de cellules, de taille $n \times n$, sur lequel sont placées k bactéries à des positions différentes appelées *foyers*. Ces bactéries vont contaminer toutes les cellules du tableau selon une règle simple : chaque cellule est contaminée par la bactérie du foyer le plus proche.

Définitions

Une cellule est une paire $(x, y) \in \mathbb{N}^2$. Pour $n \in \mathbb{N}^*$, $C_n = [0, 1, \dots n-1] \times [0, 1, \dots n-1]$ est l'ensemble des cellules de notre univers. Deux cellules (x, y) et (x', y') sont voisines si $|x - x'| + |y - y'| = 1$. À l'instant initial, k bactéries, identifiées par les entiers de 0 à $k-1$, sont placées sur des cellules de C_n. Deux bactéries différentes ne peuvent être placées sur la même cellule. Ces cellules seront les foyers, c'est-à-dire une paire $((x, y), i)$, où i est le numéro de la bactérie et (x, y) une cellule sur laquelle la bactérie i est placée.

Chaque cellule d'un C_n va être contaminée par une bactérie d'un foyer. La règle de contamination est la suivante : chaque cellule (x, y) est contaminée par la bactérie du foyer le plus proche, selon une certaine distance choisie. Si plusieurs foyers sont à même distance d'une cellule, c'est la bactérie de numéro le plus petit qui contamine la cellule parmi les foyers à distance minimum. Au début du sujet, on ne considérera pas de notion de temps. On peut donc supposer que toutes les contaminations se font instantanément. La population de la bactérie i est l'ensemble des cellules contaminées par la bactérie i au cours du processus. L'ensemble des populations de bactéries forme une partition de C_n. La figure 1 illustre une contamination, avec la distance de Manhattan. Les foyers sont représentés par des cercles et les entiers correspondent aux numéros des bactéries.

Question 1. Soit $v_0 = u_0$ et pour tout $i \in \mathbb{N}$:

$$u_{i+1} = 16365 \cdot u_i \bmod 65521 \qquad\qquad v_{i+1} = 16379 \cdot v_i \bmod 65519$$

Calculez :
- ☐ **1.1.** (u_{100}, v_{100})
- ☐ **1.2.** (u_{1000}, v_{1000})
- ☐ **1.3.** (u_{10000}, v_{10000})

Question 2. La position d'une cellule de C_n est définie par le couple $w_{n,i} = (u_i \bmod n, v_i \bmod n)$. Pour initialiser la position des k foyers, on définit l'application $\pi_n : \mathbb{N} \to \mathbb{N}$ comme suit : pour tout i, $\pi_n(i)$ est le plus petit entier naturel tel que, pour tout $0 \leqslant j < i$, $w_{n,\pi_n(i)} \neq w_{n,\pi_n(j)}$. Les foyers sont créés dans l'ordre croissant de leurs numéros, à savoir de 0 à $k-1$. L'ensemble des k foyers sur C_n, noté $\mathcal{F}_{n,k}$, est alors défini par :

$$\mathcal{F}_{n,k} = \left\{ \left(w_{n,\pi_n(0)}, 0\right), \dots, \left(w_{n,\pi_n(k-1)}, k-1\right) \right\}$$

Calculez :
- ☐ **2.1.** $\pi_{20}(100)$
- ☐ **2.2.** $\pi_{100}(1000)$
- ☐ **2.3.** $\pi_{1000}(10000)$

Question 3. Pour $r \in \mathbb{N}^* \cup \{\infty\}$, on note d_r la distance associée à la norme r. Les normes adoptées sont les normes 1, 2 et ∞ en dimension 2. Donnez la distance maximale entre 2 foyers de l'ensemble $\mathcal{F}_{30,10}$ pour :
- ☐ **3.1.** d_1
- ☐ **3.2.** d_2
- ☐ **3.3.** d_∞

Question 4. Soit $\mathcal{P}_{n,k,r} = \{P_1, \dots P_k\}$ l'ensemble des populations, où P_i est la population de la bactérie i, à la fin du processus pour la carte C_n, l'ensemble de foyers $\mathcal{F}_{n,k}$ et la norme r. La figure 1 montre les populations à la fin du processus, dans le cas de la norme 1, pour $n = 30$ et $k = 10$.

Implémentez un algorithme calculant l'ensemble des populations de bactéries à la fin du processus.

Donnez la taille maximum d'une population dans :
- ☐ **4.1.** $\mathcal{P}_{10,10,1}$
- ☐ **4.2.** $\mathcal{P}_{100,100,2}$
- ☐ **4.3.** $\mathcal{P}_{500,500,\infty}$

Question à développer pendant l'oral. Quelle est la complexité de votre algorithme, au pire des cas, en fonction de n et k.

Question 5. On ne considére que la norme 1 et, pour simplifier, $\mathcal{P}_{n,k,1}$ est noté $\mathcal{P}_{n,k}$. Quand n et k sont grands, on veut construire un algorithme qui calcule les populations en un temps raisonnable. On propose le processus suivant qui décrit l'évolution temporelle des populations.

1. En pratique, une *boîte de Pétri*.

- À l'instant 0, les foyers sont infectés par la bactérie correspondante.
- À l'instant $t \in \mathbb{N}^*$, pour chaque bactérie i de 0 à $k-1$, dans cet ordre, chaque cellule infectée à l'instant $t-1$ par la bactérie i infecte toutes ses cellules voisines non encore infectées.

Au bout d'un temps fini, toutes les cellules de C_n sont infectées.

Implémentez un algorithme simulant ce processus.

Donnez la taille maximum d'une population pour les ensembles suivants.

□ **5.1.** $\mathcal{P}_{500,1000}$
□ **5.2.** $\mathcal{P}_{1000,10000}$
□ **5.3.** $\mathcal{P}_{1000,100000}$

Question à développer pendant l'oral. Donnez la complexité en espace et en temps de votre implémentation de l'algorithme, en fonction de n et k.

Question à développer pendant l'oral. Expliquez pourquoi ce processus calcule les populations de bactéries dans le cas de la norme 1.

Question à développer pendant l'oral. Comment modifier l'algorithme pour calculer les populations avec la norme ∞ ? Et pour la norme 2 ?

Question 6. On ne considére toujours que la norme 1. Afin de visualiser les populations de bactéries, on veut colorier chaque population avec une couleur de telle manière que deux populations voisines aient deux couleurs différentes. On cherche également à ce que le nombre de couleurs utilisée ne soit pas trop grand.

On rappelle qu'un *graphe non orienté* est un couple (V, E), où V est l'ensemble de ses sommets et E l'ensemble de ses arêtes. Deux sommets u et v sont adjacents si $\{u, v\} \in E$. Le voisinage d'un sommet u, dénoté $N(u)$ est l'ensemble des sommets adjacents avec v. Le degré d'un sommet u est la taille du voisinage de u.

Deux populations A et B sont *voisines* si $A \neq B$ et s'il existe $(x, y) \in A$ et $(x', y') \in B$ tels que les cellules (x, y) et (x', y') sont voisines. Le *graphe des populations* est le graphe d'ensemble de sommets $\{1, 2, \ldots k\}$ tel que deux sommets i et i' sont adjacents si les populations P_i et $P_{i'}$ sont voisines, où $\mathcal{P}_{n,k} = (P_1, \ldots P_k)$. La figure 2 représente par exemple le graphe des populations de $\mathcal{P}_{30,10}$.

Implémentez un algorithme calculant le graphe des populations de $\mathcal{P}_{n,k}$.

Donnez le nombre d'arêtes et le degré moyen des graphes des populations correspondant à :

□ **6.1.** $\mathcal{P}_{10,30}$
□ **6.2.** $\mathcal{P}_{100,100}$
□ **6.3.** $\mathcal{P}_{500,1000}$
□ **6.4.** $\mathcal{P}_{100,10000}$
□ **6.5.** $\mathcal{P}_{2000,100000}$

Question à développer pendant l'oral. Quelles structures de données avez-vous utilisé pour stocker votre graphe ? Donnez la complexité en espace et en temps de votre implémentation en fonction de n et k.

Question 7. Une *coloration* d'un graphe $G = (V, E)$ est une fonction $c : V \to \mathbb{N}$. Elle est qualifiée de *valide* si pour toute arête $\{u, v\} \in E$, on a $c(u) \neq c(v)$. Soit un graphe $G = (V, E)$ et un sommet $v \in V$. On note $G - v$ le graphe (V', E') où $V' = V \backslash \{v\}$ et $E' = \{\{x, y\} : \{x, y\} \in E \text{ et } v \notin \{x, y\}\}$; $G - v$ est le graphe G privé du sommet v et des arêtes adjacentes v.

Implémentez l'algorithme suivant pour colorier les graphes (on suppose que l'ensemble des sommets du graphe est un ensemble d'entiers, comme c'est le cas pour les graphes de populations).

Algorithme 1 : coloration valide

Entrée : un graphe $G = (V, E)$
Sortie : une coloration de G
Choisir v dans V de plus petit degré. En cas d'égalité, choisir le plus petit v parmi les sommets de plus petit degré.
Calculer récursivement une coloration sur $G' = G - v$.
Pour chaque sommet u dans G', assigner à u dans G la même couleur que dans G'.
Assigner à v dans G la plus petit entier de $\mathbb{N}$ non assigné aux voisins de v dans G'.

Question à développer pendant l'oral. Expliquez pourquoi l'algorithme donne une coloration valide.

Donnez le nombre de sommets de chaque couleur de la coloration calculée par l'algorithme précédent, pour :

□ **7.1.** $\mathcal{P}_{20,50}$
□ **7.2.** $\mathcal{P}_{100,100}$
□ **7.3.** $\mathcal{P}_{1000,1000}$
□ **7.4.** $\mathcal{P}_{2000,20000}$
□ **7.5.** $\mathcal{P}_{5000,500000}$
□ **7.6.** $\mathcal{P}_{20000,2000000}$

Prenez soin de compter le nombre de populations de chaque couleur et non le nombre de cellules de chaque couleur. Il sera peut-être nécessaire d'adapter vos algorithmes pour tenir compte de la mémoire disponible afin de faire passer les dernières instances.

Question à développer pendant l'oral. Donnez la complexité en espace et en temps de votre implantation de l'algorithme, en fonction du nombre de sommets du graphe. Quelles structures de données avez vous utilisées pour obtenir ce temps d'exécution ?

Question à développer pendant l'oral. Compte tenu des résultats obtenus aux questions 6 et 7 , que pouvez vous conjecturer sur le nombre de couleurs utilisé par l'algorithme pour colorier le graphe ? Est-on loin d'une coloration optimale ? Expliquez pourquoi.

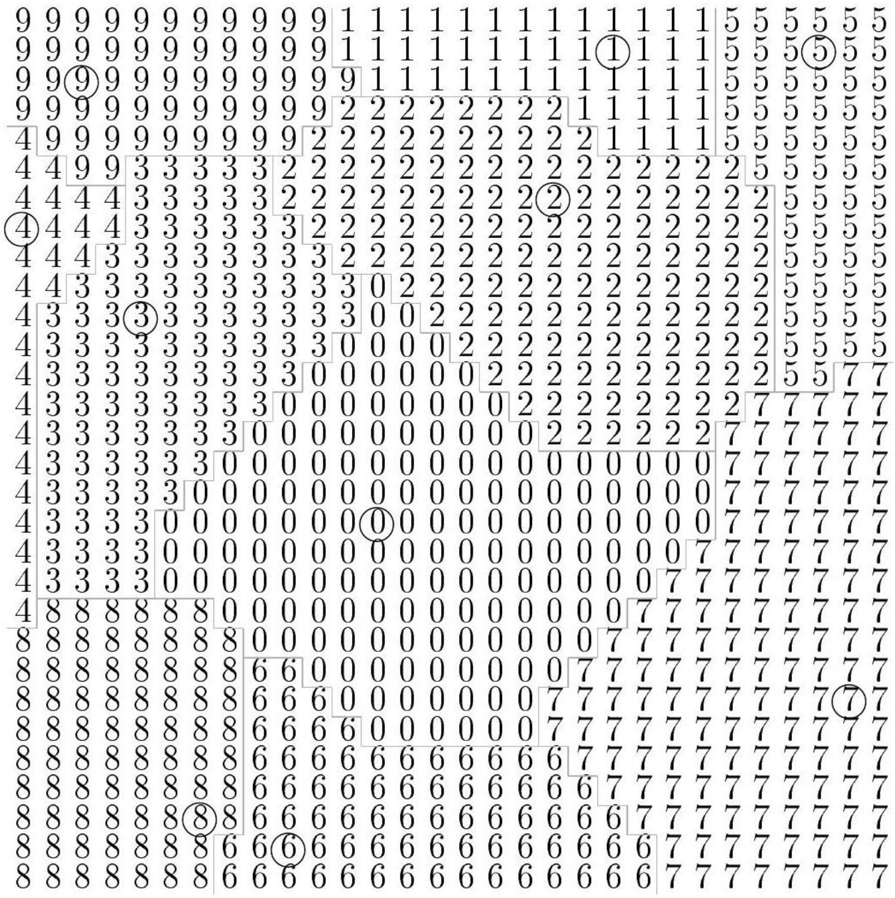

Figure 1 – Les populations de $\mathcal{P}_{30,10,1}$, c'est à dire $n = 30$, $k = 10$ et la distance d_1. Les cercles représentent les foyers et les lignes grises les frontières entre les populations. L'origine est en bas à gauche.

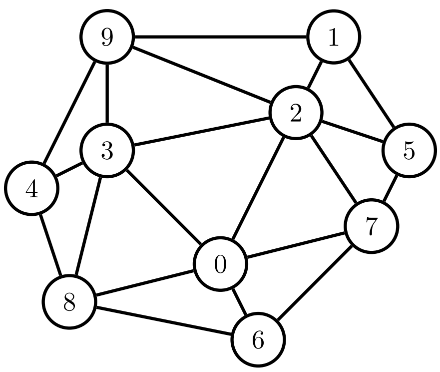

Figure 2 – Le graphe correspondant à $\mathcal{P}_{30,10}$.

Partie 2 - Fiche réponse à compléter

Nom : . . . . **Prénom :** . . . .

Valeur de u_0 **:** . . . .

Question 1.

☐ **1.1.** . . . .

☐ **1.2.** . . . .

☐ **1.3.** . . . .

Question 2.

☐ **2.1.** . . . .

☐ **2.2.** . . . .

☐ **2.3.** . . . .

Question 3.

☐ **3.1.** . . . .

☐ **3.2.** . . . .

☐ **3.3.** . . . .

Question 4.

☐ **4.1.** . . . .

☐ **4.2.** . . . .

☐ **4.3.** . . . .

Question 5.

☐ **5.1.** . . . .

☐ **5.2.** . . . .

☐ **5.3.** . . . .

Question 6.

☐ **6.1.** . . . .

☐ **6.2.** . . . .

☐ **6.3.** . . . .

☐ **6.4.** . . . .

☐ **6.5.** . . . .

Question 7.

☐ **7.1.** . . . .

☐ **7.2.** . . . .

☐ **7.3.** . . . .

☐ **7.4.** . . . .

☐ **7.5.** . . . .

☐ **7.6.** . . . .

Partie 3 - Fiche réponse type

. . . .

Valeur de $\tilde{u}_0$: 42

Question 1.

□ **1.1.** $(49632, 41175)$

□ **1.2.** $(33168, 30138)$

□ **1.3.** $(41199, 46570)$

Question 2.

□ **2.1.** 117

□ **2.2.** 1041

□ **2.3.** 10048

Question 3.

□ **3.1.** 47

□ **3.2.** $\sqrt{1117} = 33.42$

□ **3.3.** 28

Question 4.

□ **4.1.** 20

□ **4.2.** 246

□ **4.3.** 1752

Question 5.

□ **5.1.** 773

□ **5.2.** 419

□ **5.3.** 48

Question 6.

□ **6.1.** $66, 4.40$

□ **6.2.** $265, 5.30$

□ **6.3.** $2852, 5.70$

□ **6.4.** $19800, 3.96$

□ **6.5.** $288827, 5.78$

Question 7.

□ **7.1.** $(15, 14, 13, 8)$

□ **7.2.** $(29, 28, 27, 15, 1)$

□ **7.3.** $(288, 265, 237, 193, 17)$

□ **7.4.** $(5527, 5285, 4910, 3823, 455)$

□ **7.5.** $(138920, 133408, 123341, 94344, 9987)$

□ **7.6.** $(547122, 527386, 489736, 383917, 51839)$