Epreuve finale : Algorithmique avancée et Complexité

<u>Exercice 1 :</u> (8 pts)

Considérons deux matrices carrées d'entiers d'ordre *n, A* et *B.* Le produit de *A* par *B* est une matrice carrée *C* définie par :

$$C_{i,j} = \sum_{k=1}^{n} A_{i,k} * B_{k,j}$$

1. Donner un algorithme naïf calculant le produit des deux matrices A et B. Calculer la complexité de cet algorithme **(4 pts).**

< Déjà vu en TD > (3 pts algorithme, 1 pt complexité)

2. Supposons que les matrices A et B peuvent être décomposées en sous matrices carrées de dimension $\frac{n}{2} \times \frac{n}{2}$ comme suit :

$$A = \begin{pmatrix} \begin{pmatrix} (1) & (3) \\ (5) & (10) \end{pmatrix} & \begin{pmatrix} 2 & 7 \\ 4 & 1 \end{pmatrix} \\ \begin{pmatrix} 15 & 6 \\ 9 & 11 \end{pmatrix} & \begin{pmatrix} 12 & 8 \\ 0 & 5 \end{pmatrix} \end{pmatrix}$$

Ainsi, le produit de deux matrices carrées de dimension N peut être calculé en se basant sur ce principe de décomposition jusqu'à atteindre le cas trivial avec des matrices de dimension 1.

- Proposer un algorithme récursif (ou itératif) pour le produit matriciel en vous basant sur la décomposition précitée. Calculer la complexité de cet algorithme **(4 pts)**

```
Mult_R(i, j, k : entier ; A, B : matrice d'entiers) ;
Début
Si k = 1 alors C[i, j] = C[i, j]+ A[i,j]*B[i, j] ;
Retourner ( C );
Sinon k = k/2 ;
//décomposition des deux matrices
A1 = sous_matrice (i, j, k, A) ; A2 = sous_matrice (i, j + k, k, A) ;
A3 = sous_matrice (i + k ,j, k, A) ; A4 = sous_matrice (i+ k, j + k, k, A) ;
B1 = sous_matrice (i, j, k/2, B) ; B2 = sous_matrice (i, j + k, k, B) ;
B3 = sous_matrice (i + k, j, k, B) ; B4 = sous_matrice (i + k,j + k, k, B) ;
 Retourner (concaténation (somme (Mult_R(A1, B1), Mult_R(A2, B3)),
                    Somme (Mult_R(A1, B2), Mult_R(A2, B4)),
                    somme(Mult_R(A3, B1), Mult_R(A4, B3)),
                    somme(Mult_R(A3, B2), Mult(A4, B4))));
Fin;
```

Complexité = O (N^3). La décomposition et la récursivité ne permet pas de réduire l'ordre de complexité dans ce cas

3 pts pour l'algorithme et 1 pt pour la complexité.

<u>Exercice 2 :</u> (12 pts)

Considérons une boite de dominos contenant 28 pièces. Chaque pièce est représentée par deux parties (haut et bas), contenant un certain nombre de points allant de 0 à 6. Chaque pièce peut être utilisée selon une des deux configurations possible (par exemple : $\binom{2}{3}$ $ou\ bien$ $\binom{3}{2}$).

Le problème de correspondance de Post noté PCP, consiste à déterminer s'il existe une séquence P de K dominos distincts, telle que la somme des parties hautes de chaque pièce dans la séquence P est identique à la somme des parties basses. Formellement le problème du PCP se définit comme suit :

- Soit D, l'ensemble de dominos. D = $\{\binom{Hi}{Bi}, avec\ Hi, Bi\ \in [0,\ 6]\}$ Chaque couple $\binom{Hi}{Bi}$ correspond à une pièce du domino à travers le nombre de points dans les parties haute (Hi) et basse (Bi).
- Existe-t-il une séquence P, P ⊂ D, |P| = K, avec la somme $\sum_{j=1}^{K} Hj = \sum_{j=1}^{K} Bj$?

Questions :

1. Illustrer les étapes de construction d'une solution en spécifiant la modélisation la plus adéquate pour K = 10 **(4 pts)**

Modélisation : 2 pts

La solution P correspond à un sous ensemble de k dominos qui peut être représenté par un vecteur d'enregistrement de dimension K. Chaque P [i] est un correspond à une pièce du domino et sa configuration. Initialement le vecteur est vide et il sera construit progressivement à chaque fois qu'une pièce est rajoutée au sous-ensemble.

Une autre modélisation possible serait de considérer la solution comme un vecteur binaire en ne considérant que la présence et l'absence des pièces de dominos.

Illustration : 2 pts

Il s'agit ici de tracer l'arbre de résolution qui correspond à la modélisation en spécifiant la profondeur maximale, le nombre de successeurs lorsque k = 10.

2. Estimer approximativement la taille de l'arbre de résolution et en déduire l'ordre de complexité de l'algorithme **(2 pts)**

En fonction de la modélisation et de l'arbre de résolution.

1 pt pour la taille de l'arbre. 1 pt pour l'ordre de complexité

3. Quels sont les critères que doit satisfaire une solution donnée S' pour être valide **(1 pt)**

Ils sont au nombre 4 (0,25 pour chaque) :

- La taille de la solution est égale à K (vecteur de taille K pour la première modélisation. Ou bien nombre de 1 est égale à K dans la modélisation binaire)
- L'unicité de la pièce du domino dans la solution
- L'unicité de la configuration

– Somme des parties hautes est égale à la somme des parties basses des dominos pris

4. Proposer un algorithme de validation d’une solution donnée S’et calculer sa complexité. **(4 pts)**

3 pts pour l’algorithme. 1 pt pour la complexité

En fonction de la modélisation, certains critères n’ont pas besoin d’être vérifié par exemple l’unicité de la pièce dans le cas de la modélisation binaire, ou bien le nombre de dominos dans le cas d’un vecteur d’enregistrement.

Pour la modélisation binaire les contraintes qui doivent être prise en compte sont : l’unicité de la configuration, le nombre de dominos égale à K (nombre de 1 dans la solution est égale à 1) et la somme des parties hautes et basses des dominos à 1 est égale.

Pour la modélisation en utilisant un tableau d’enregistrement de langueur K, les contraintes à prendre en compte sont : l’unicité de chaque pièce de dominos, l’unicité de la configuration et l’égalité de la somme.

5. En déduire la classification associée au problème étudié. Justifier votre réponse **(1 pt)**

L’algorithme de validation est polynomial (N^2 à cause de la vérification de l’unicité), cela signifie que le problème appartient à la classe NP.

Le problème ne peut pas appartenir à la classe P puisque la taille de l’arbre de résolution est exponentielle