École Nationale d'Ingénieurs de Carthage
Département d'Informatique
—***—

Examen de la session principale en

Algorithmique Avancée & Complexité

9 janvier 2020

Classe : 2IngInfo$_{A,B,C,D}$ Durée : $1^h30'$ Nombre de pages : 3 pages

Il est conseillé de lire attentivement les énoncées avant de se plonger sur la feuille des réponses. ***Les copies propres et bien soignées sont très appréciées. Il sera tenu compte de la lisibilité des réponses.*** *Le barème donné est indicatif. Aucun document n'est autorisé.*

Bonne chance !

Problème I. Multiplication de matrices carrées d'Ordre n.

Nous considérons dans ce sujet le problème de multiplication matricielle pour des matrices carrées d'ordre n (n étant un entier positif) (ie. n lignes $\times n$ colonnes).

Dans la suite de ce sujet, nous notons par `A` (de terme général $a_{ij}, i, j = 1 \ldots n$), `B` (de terme général $b_{ij}, i, j = 1 \ldots n$),et `C` (de terme général $c_{ij}, i, j = 1 \ldots n$), trois matrices carrées d'ordre n.
Il est toujours question de multiplier les deux matrices `A` et `B` pour obtenir la matrice produit `C`.

$$\begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \\ a_{n1} & & a_{nn} \end{pmatrix} \times \begin{pmatrix} b_{11} & \ldots & b_{1n} \\ \vdots & \ddots & \\ b_{n1} & & b_{nn} \end{pmatrix} = \begin{pmatrix} c_{11} & \ldots & c_{1n} \\ \vdots & \ddots & \\ c_{n1} & & c_{nn} \end{pmatrix}$$

Partie A. Algorithme itératif classique (2 pts)

1. Écrire la procédure `MatMul_iter(...)` qui multiplie les deux matrices `A` et `B` pour obtenir la matrice `C`, en suivant l'algorithme de multiplication matricielle itératif dit classique.

```
Procédure MatMul_iter( var C : Mat ; A,B : Mat ; n : entier)
Var i, j, k : entier
Début
    Pour i de 1 à n Faire
        Pour j de 1 à n Faire
        C[i, j] ← 0
            Pour k de 1 à n Faire
            C[i, j] ← C[i, j] + A[i, k] * B[k, j]
            FPour
        FPour
    FPour
Fin
```

2. Calculer les nombres exactes d'opérations d'additions scalaires, noté $T_{+}^{iter}(n)$, et d'opérations de multiplications scalaires, noté $T_{\times}^{iter}(n)$ effectuées par `MatMul_iter(...)`.

On a : $T_{+}^{iter}(n) = n^3$

<u>et</u> $T_{*}^{iter}(n) = n^3$

3. Déduire la complexité totale de cette procédure en terme d'opérations arithmitiques et en donner une estimation asymptotique en notation $\mathcal{O}()$ pour chaque grandeur.

$\Rightarrow T_{iter}(n) = 2n^3 = O(n^3)$

Partie B. Multiplication Matricielle par blocs (10 pts)

Considèrons, maintenant, la multiplication matricielle en adoptons l'approche `Diviser pour Régner` (`DpR`).

4. Rappeler les trois étapes qu'une procédure, qui suit le paradigme `DpR`, doit contenir. Notons les respectivement `E1`, `E2` et `E3`.
   Expliquer en deux lignes le rôle de chaque étape.

E1 : <u>**Diviser**</u> le problème en un certain nombre de sous-problèmes de taille plus faible.

E2 : <u>**Régner**</u> sur les sous-problèmes de taille plus faible (si la taille d'un ss-pb est assez réduite on le résout directement).

E3 : <u>**Combiner**</u> les solutions des ss-pbs en une solution complète du problème initial.

Comme le suggére le paradigme `DpR`, le traitement d'un problème se fait en le décomposant en des sous-problèmes plus petits. Nous nous proposons alors, dans ce sujet, de décomposer les matrices en des sous-matrices de taille $\frac{n}{3}$ comme dans la figure plus bas. Sans perte de généralité, nous pouvons toujours supposer que $n = 3^k, k \geq 1$.

La matrice produit `C` s'écrit alors :

$$\begin{cases} C_{11} = A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31} \\ C_{12} = A_{11}B_{12} + A_{12}B_{22} + A_{13}B_{32} \\ C_{13} = A_{11}B_{13} + A_{12}B_{23} + A_{13}B_{33} \\ C_{21} = A_{21}B_{11} + A_{22}B_{21} + A_{23}B_{31} \\ C_{22} = A_{21}B_{12} + A_{22}B_{22} + A_{23}B_{32} \\ C_{23} = A_{21}B_{13} + A_{22}B_{23} + A_{23}B_{33} \\ C_{31} = A_{31}B_{11} + A_{32}B_{21} + A_{33}B_{31} \\ C_{32} = A_{31}B_{12} + A_{32}B_{22} + A_{33}B_{32} \\ C_{33} = A_{31}B_{13} + A_{32}B_{23} + A_{33}B_{33} \end{cases}$$

5. Écrire la procédure MatMul_DpR(...). Une attention particulière doit être accordée aux arguments de cette procédure ainsi qu'aux modes de passage.

Procédure MatMul_DpR(var C : Mat ; A,B : Mat ; n : entier)

Var i, j, k : entier

Début

Si n = 1 **alors** Retourner (A * B)

Sinon

MatMul_DpR (M_1, A_{11}, B_{11}, n/3)

MatMul_DpR (M_2, A_{12}, B_{21}, n/3)

MatMul_DpR (M_3, A_{13}, B_{31}, n/3)

MatMul_DpR (M_{27}, A_{33}, B_{33}, n/3)

$C_{11} \leftarrow M_1 + M_2 + M_3$

$C_{12} \leftarrow M_4 + M_5 + M_6$

$C_{33} \leftarrow M_{25} + M_{26} + M_{27}$

Fin

6. Identifier les étapes E1 et E3 de cette procédure et dire combien sont les complexités de ces deux étapes.
7. Soient $T_{+}^{DpR}(n)$ et $T_{\times}^{DpR}(n)$ les complexités de la proécdure MatMul_DpR(...) respectivement en terme d'opérations d'addition scalaire et en terme d'opérations de multiplication scalaire.
Donner les deux récurrences qui régissent ces deux complexités.
Préciser les conditions initiales.

On a : $T_{+}^{DPR}(n) = 18\, T_{+}^{DPR}(n/3)$ avec $T_{+}^{DPR}(1)=1$

et $T_{*}^{DPR}(n) = 27T_{*}^{DPR}(n/3)$ avec $T_{*}^{DPR}(1)=1$

8. a. Calculer $T_{+}^{DpR}(n)$ et $T_{\times}^{DpR}(n)$ en développant ces deux récurrences.
   b. Déduire la complexité totale en terme d'opérations arithmitiques et donner une estimation asymptotique en notation $\mathcal{O}()$ pour chaque grandeur.

On pose $n = 3^k$

$T_+^{DPR}(n) = 18\, T_+^{DPR}(n/3)$

$= 18(18\, T_+^{DPR}(n/3^2)) = \ldots\ldots\ldots = 18\,(18(18\ldots\ldots 18 T_+^{DPR}(n/3^k)))\ldots))))$

$= 18^k\, T_+^{DPR}(1) = 18^k = 18^{\log_3 n} = n^{\log_3 18} = \mathbf{O(n^2)}$

$T_*^{DPR}(n) = 27 T_*^{DPR}(n/3) = 27^k = n^{\log_3 27} = \mathbf{O(n^3)}$

9. Retrouver les mêmes résultats en appliquant le théorème de résolution des récurrences.

$T^{DPR}(n) = 27 T_*^{DPR}(n/3) + O(n^2)$

$a = 27$, $b = 3$ et $f(n) = O(n^2)$

$\log_3 27 = \log_3 3^3 = 3 \log_3 3 = 3$

$f(n) = O(n^{(\log_3 27)-1}) \Rightarrow$ 1 er cas donc $T^{DPR}(n) = \Theta(n^{\log_3 27}) = \Theta(n^3)$