**ANNÉE UNIVERSITAIRE 2023/2024**

**4TMA701U Calcul Formel**
**Devoir Surveillé**
**Date : 08/11/2023   Heure : 15h30   Durée : 1h30**

Documents autorisés.

**Collège Sciences et Technologies**

Vous rendrez à la fin de l'examen une copie papier ainsi qu'un fichier sage contenant vos programmes (lisible, commenté et nettoyé si possible..) au format DS-Nom-Prenom.ipynb (feuille Jupyter) ou DS-Nom-Prenom.sage (fichier texte). Le fichier est à envoyer par e-mail à votre enseignant.e de TD (christine.bachoc@u-bordeaux.fr ou leo.poyeton@u-bordeaux.fr).

L'objectif de ce sujet est de montrer l'existence d'un algorithme rapide pour l'évaluation simultanée en $n$ nombres d'un polynôme de degré inférieur à $n$. On supposera pour simplifier que $n$ est une puissance de 2, et que $K$ est un corps de caractéristique différente de 2 et contenant les racines $2^k$-ièmes de l'unité pour tout $k \geq 1$. On rappelle que la transformée de Fourier rapide conduit à un algorithme de multiplication de deux polynômes de $K[X]$ de degrés au plus $n$ de complexité algébrique $O(n \log n)$. On admettra qu'il existe aussi un algorithme pour leur division euclidienne de même complexité algébrique.

Soit donc $P \in K[X]$, $n = 2^k$, $\deg(P) < n$, et soit $(u_0, u_1, \ldots, u_{n-1}) \in K^n$. On veut calculer efficacement $(P(u_0), \ldots, P(u_{n-1}))$.

1. Rappelez sans justification l'ordre de grandeur de la complexité algébrique de l'algorithme de Horner calculant l'évaluation $P(a)$ de $P$ en $a \in K$. En déduire un algorithme naif de complexité algébrique quadratique pour calculer $(P(u_0), \ldots, P(u_{n-1}))$.
2. On suppose dans cette question que $w$ est une racine primitive $n$-ième de l'unité dans $K$, et que $u_i = w^i$ pour tout $0 \leq i < n$. Quel algorithme vu en cours permet de calculer $(P(u_0), \ldots, P(u_{n-1}))$ avec une meilleure complexité algébrique que l'algorithme naif ?
3. On définit les polynômes suivants :

$$M_{i,j} = \prod_{\ell=0}^{2^i-1} (X - u_{j2^i+\ell}), \ 0 \leq i \leq k-1, \ 0 \leq j \leq 2^{k-i} - 1$$

   Explicitez ces polynômes dans le cas $k = 3$ (on pourra les représenter dans un arbre binaire, cela peut aider).
4. Montrez les propriétés suivantes :
   (1) $M_{0,j} = X - u_j$ pour $0 \leq j \leq 2^{k-i} - 1$.
   (2) $\deg(M_{i,j}) = 2^i$ pour $0 \leq i \leq k-1$
   (3) $M_{i+1,j} = M_{i,2j}M_{i,2j+1}$ pour $0 \leq i \leq k-2, \ 0 \leq j \leq 2^{k-i-1} - 1$

5. Montrez que l'algorithme suivant calcule la liste des polynômes $M_{i,j}$ avec une complexité algébrique en $O(n(\log n)^2)$ :

   **Algorithme 1** [POLYMIJ]
   *Entrées :* $n = 2^k$, $(u_0, \dots, u_{n-1}) \in K^n$
   *Sortie : Les* $M_{i,j}$
   *1. Pour* $j = 0, \dots, 2^{k-i} - 1$, $M_{0,j} = X - u_j$
   *2. Pour* $i = 0, \dots, k-2$ *:*
       *Pour* $j = 0, \dots, 2^{k-i-1} - 1$ *:*
           $M_{i+1,j} = M_{i,2j} M_{i,2j+1}$
   *3. Sortir* $[[M_{i,j},\ 0 \leq j \leq 2^{k-i} - 1],\ 0 \leq i \leq k-1]$.

6. Implémentez POLYMIJ et le tester pour $k = 2, 3$.
7. Soit $P_0 = \mathrm{rem}(P, M_{k-1,0})$ et $P_1 = \mathrm{rem}(P, M_{k-1,1})$ (respectivement les restes de $P$ dans la division euclidienne par $M_{k-1,0}$ et par $M_{k-1,1}$). Montrez que $P(u_i) = P_0(u_i)$ pour $0 \leq i \leq n/2 - 1$ et que $P(u_i) = P_1(u_i)$ pour $n/2 \leq i \leq n-1$.
8. Utilisez le résultat de la question précédente pour écrire un algorithme **récursif** que vous nommerez MULTIEVAL prenant en entrées $n$, $P$, et la liste des $M_{i,j}$ et sortant $(P(u_0), \dots, P(u_{n-1}))$.
9. Soit $T(n)$ la complexité algébrique de l'algorithme MULTIEVAL. Montrez que $T(n) = 2T(n/2) + O(n \log n)$.
10. Montrez que $T(n) = O(n(\log n)^2)$. *Indication : on pourra s'inspirer de la preuve du Lemme maître vue en cours...*
11. Implémentez et testez MULTIEVAL.
12. Donnez en conclusion un algorithme répondant au problème initial et analysez sa complexité algébrique.