TD ALBA algorithme sur les polynômes

Exercice 1 :
Soit $P \in \mathbf{A}[X]$ un polynôme de degré n et $a \in \mathbf{A}$.

1. Donner un algorithme direct pour calculer le polynôme $Q(X) = P(X + a)$. Quelle est sa complexité ?
2. Montrer que l'on peut calculer Q en $O(M(n) \log n)$.

Exercice 2 :
Soient $P_1, \cdots, P_t$ des polynômes de $\mathbf{A}[X]$ de degrés $d_1, \cdots, d_t$ avec $\sum d_i = N$. Montrer que l'on peut effectuer le produit $P_1 \cdots P_t$ peut s'effectuer en $O(M(N) \log t)$ opérations dans $\mathbf{A}$.

Exercice 3 :
Soit $\mathbf{A}$ un anneau. Soient $A, B \in M_d(\mathbf{A}[X])$ des matrices dont les coefficients sont des polynômes de degré au plus n.

1. Donner la complexité de l'algorithme naïf qui calcule le produit de A et B.
2. On suppose l'existence d'une fonction $MM(d)$ telle que pour deux matrices carrées de taille d à coefficients dans $\mathbf{A}$, le produit de ces deux matrices prend $MM(d)$ opérations arithmétiques dans $\mathbf{A}$. Montrer que l'on peut trouver un algorithme qui calcule le produit de A et B de complexité $O(d^2 M(n) + nMM(d))$.

Exercice 4 :
Soient $x_1, \cdots, x_n \in \mathbf{K}$. On note $P_K = x_1^k + \cdots + x_n^k$ la k-ième somme de Newton.

1. Soit $x \in \mathbf{K}$, montrer que l'on peut calculer tous les x^k pour k une puissance de 2 inférieure ou égale à n en $O(\log(n))$ opérations dans $\mathbf{K}$.
2. En déduire que l'on peut calculer tous les P_k pour k une puissance de 2 inférieure ou égale à n en $O(n \log n)$ opérations dans $\mathbf{K}$.
3. Montrer l'égalité suivante dans les séries formelles
$$\frac{X}{1 - X} = X + X^2 + X^3 + \cdots$$
4. En déduire que
$$\sum_{i=1}^{n} \frac{x_i X}{1 - x_i X} = \sum_{k \geq 1} P_k X^k.$$
5. En déduire en algorithme que l'on écrira en pseudo-code permettant de calculer tous les P_k pour $k \leq n$ en $O(M(n) \log n)$ opérations dans $\mathbf{K}$ qui utilise l'algorithme rapide de calcul de somme de fractions.