École normale supérieure **2012-2013**
Département d'informatique

Algorithmique et Programmation
TD n° 7 : Arithmétique et FFT

Exercice 1. MULTI-EXPONENTIATION

Soit $\mathbb{G}$ un groupe commutatif d'ordre n. Donner un algorithme qui étant donnés $g_1, \ldots, g_\ell \in \mathbb{G}$ et des entiers $e_1, \ldots, e_\ell \in \{0, \ldots, n-1\}$ calcule le produit $\prod_{i=1}^{\ell} g_i^{e_i}$ en $O(\log(n) + 2^\ell)$ multiplications dans $\mathbb{G}$.

Exercice 2. LOGARITHME DISCRET

Soit $\mathbb{G}$ un groupe cyclique d'ordre n engendré par g.

1. On souhaite calculer g^x, pour $x \in \mathbb{Z}$. Supposons que l'on a precalculé g^u, pour $u = [\sqrt{n}]$. Montrer que l'exponentiation dans G coûte $\log n$ opérations dans le pire des cas.
2. Soit $h \in \mathbb{G}$. Le problème du logarithme discret dans $\mathbb{G}$ consiste à trouver l'unique entier x dans $\{1, \ldots, n\}$ tel que $h = g^x$. Proposer un algorithme qui résoud le problème du logarithme discret en $O(\sqrt{n})$ opérations dans le groupe $\mathbb{G}$.

Exercice 3. FFT SUR DES ENTIERS

On suppose que n est une puissance de 2.

1. On suppose que l'on recherche le plus petit entier k tel que $p = kn + 1$ est premier. Montrer heuristiquement que l'on peut s'attendre à ce que la taille de k soit environ $\log n$. Comparer la longueur moyenne de p à la longueur de n.

   Soit g un générateur de $\mathbb{Z}_p^*$ et soit $w = g^k \bmod p$.
2. Expliquer pourquoi la transformation discrète de Fourier et son inverse sont des opérations inverses bien définies modulo p, w étant utilisée comme racine n-ième principale de l'unité.
3. Prouver que la FFT et son inverse peuvent fonctionner modulo p en temps $O(n \log n)$, en supposant que les opérations sur les mots de $O(\log n)$ bits prennent un temps constant. On suppose que l'algorithme a comme paramètre p et w.
4. Calculer la transformée discrète de Fourier modulo $p = 17$ du vecteur $(0, 5, 3, 7, 7, 2, 1, 6)$. Noter que $g = 3$ est un générateur de $\mathbb{Z}_{17}^*$.

Exercice 4. FFT BI-DIMENSIONNELLE

Donner un algorithme qui multiplie deux polynômes (representés par leur coefficients) $P, Q \in \mathbb{C}[X, Y]$ de degrés inférieurs à n en $O(n^2 \log n)$ opérations élémentaires dans $\mathbb{C}$.

Exercice 5. MATRICES DE TOEPLITZ

Une *matrice de Toeplitz* $n \times n$ est une matrice $A = (a_{i,j})$ avec pour pour $i = 2, \ldots, n$ et $j = 2, \ldots, n$, $a_{i,j} = a_{i-1,j-1}$. Montrer comment calculer efficacement en $O(n \log n)$ opérations le produit d'une matrice de Toeplitz avec un vecteur.