TD 8

Algorithmique

Exercice 1: Diviser-pour-régner

1. Montrer comment multiplier rapidement deux entiers de n bits en utilisant une méthode divide-and-conquer.
2. Montrer comment calculer efficacement le produit d'une matrice de Toeplitz avec un vecteur. On dit qu'une matrice est une matrice de Toeplitz si pour $i = 2, \ldots, n$ et $j = 2, \ldots, n$, $a_{i,j} = a_{i-1,j-1}$.
3. Transformée de Walsh-Hadamard
   Les matrices de Hadamard $H_0, H_1, H_2, \ldots$ sont définies par:
   - H_0 est la matrice [1],
   - Pour $k > 0$, H_k est la matrice $2^k \times 2^k$:

$$\left[\begin{array}{c|c} H_{k-1} & H_{k-1} \\ \hline H_{k-1} & -H_{k-1} \end{array}\right]$$

   Montrer que si v est un vecteur colonne de taille $n = 2^k$, alors le produit matrice-vecteur $H_k v$ peut se calculer en utilisant $O(n \log n)$ opérations en supposant que les nombres mis en jeu soient petits pour que les opérations arithmétiques de base comme l'addition et la multiplication prenne un temps constant.

Exercice 2: FFT

1. Décrire l'algorithme de multiplication rapide de 2 polynômes
2. Décrire l'algorithme récursif de transformée de Fourier
3. Quel est l'ordre des éléments de la récursion finale sur un tableau de 8 éléments ? Décrire un algorithme qui effectue une permutation miroir d'un tableau: c'est-à-dire que $A[revinv_k(i)] = a(i)$ où $revinv_k(i)$ est la représentation binaire de i sur k bits renversée. Si $a = 0^3$, la valeur 0 sur 3 bits, que vaut l'itération de l'instruction suivante $revinv_3(revinv_3(a) + 1)$ Pouvez-vous décrire un algorithme dont le coût est $O(n)$ au total ?
4. Décrire la version itérative

Exercice 3: FFT sur des entiers

On suppose que n est une puissance de 2.

a) 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 $w = g^k \bmod p$.

b) Expliquer pourquoi la transformation discrète de Fourier et son inverse sont des opérations inverses bin définies modulo p, w étant utilisée comme racine n-ième principale de l'unité.

c) 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.

d) 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: Matrice circulante

Une matrice circulante est une matrice $n \times n$ telle que la i-ième ligne est obtenue par rotation de la première ligne de i positions, pour $0 \leq i \leq n-1$. Une matrice circulante est complètement spécifiée par sa première ligne. On remarque que $c(a)_{i,j} = a_{j-i}$.

1. Montrer que le produit de deux matrices circulantes est une matrice circulante ainsi que l'inverse d'une matrice circulante. (On pourra montrer que $C = P^{-1}CP$ où P est la permutation qui envoie le vecteur $(x_1, \ldots, x_n)^T$ sur $(x_2, \ldots, x_n, x_1)$.)
2. Trouver un algorithme naïf pour multiplier deux matrices circulantes $n \times n$.
3. Montrer que si $a = (a_0, a_1, \ldots, a_{n-1})$ et $b = (b_0, b_1, \ldots, b_{n-1})$, alors

$$F_n^{-1}(F_n a \cdot F_n b) = (\sum_{i=0}^{n-1} a_i b_{-i}, \sum_{i=0}^{n-1} a_i b_{1-i}, \ldots, \sum_{i=0}^{n-1} a_i b_{n-1-i})$$

   où F_n représente la matrice de Fourier et les indices sont pris modulo n.
4. Montrer que

$$c(a) \cdot c(b) = c(F_n^{-1}(F_n a \cdot F_n b))$$

   et en déduire un algorithme pour calculer le produit de deux matrices circulantes.
5. Montrer que si $d(a)$ représente la matrice diagonale avec le vecteur a sur la diagonale, alors

$$F_n^{-1} c(a) F_n = d(F_n a)$$

6. En déduire un algorithme pour inverser une matrice circulante.
7. Montrer comment calculer efficacement le produit matrice-vecteur quand la matrice est une matrice de Toeplitz.