Fast Fourier Transform

Dans ce développement, on montre que la multiplication de deux polynômes de degré n peut s'effectuer en temps $O(n \log n)$.

Theorème. *Etant donné $P, Q \in \mathbb{R}_n[X]$, on peut calculer les coefficients de $P.Q$ en temps $O(n \log n)$.*

Idée générale :
Au lieu de multiplier les polynômes, on va les plonger dans un autre domaine dans lequel l'analogue de la multiplication est plus facile à réaliser, puis revenir vers les polynômes. Neper avait déjà eu ce genre d'idée puisque pour calculer un produit de grands nombres, il remarqua qu'il était plus simple de passer au logarithme et d'additionner les logarithmes :

$$\begin{array}{ccccc} x & \times & y & = & z \\ \downarrow & & \downarrow & & \uparrow \\ \log x & + & \log y & = & \log z \end{array}$$

Dans notre cadre, le plongement que l'on va opérer sur les polynômes est tout simplement l'évaluation sur certains points de $\mathbb{C}$, et l'opération inverse correspondra à une interpolation (qui renvoie le bon résultat d'après des résultats mathématiques connus). La multiplication dans le plongement revient juste à une multiplication des évaluations, puisque $(PQ)(x_i) = P(x_i)Q(x_i)$, et est donc facile à réaliser.

Formalisation :
On suppose donné $(x_i)_{1 \leq i \leq m} \in \mathbb{C}^m$, et on associe à tout polynôme P le m-uplet $\tilde{P} = (P(x_1), \ldots, P(x_m))$. Ainsi, pour multiplier deux polynômes, on procedera ainsi :

$$\begin{array}{ccccc} P & \times & Q & = & R \\ \downarrow & & \downarrow & & \uparrow \\ \tilde{P} & \cdot & \tilde{Q} & = & \tilde{R} \end{array}$$

où ici $X \cdot Y$ dénote le produit coordonnées à coordonnées de deux uplets.

Pour la suite, il sera plus pratique d'exprimer $\tilde{P}$ comme le résultat d'un produit de matrice par un vecteur. En effet, si $P(X) = \sum_{i=0}^{n} a_i X^i$, on a :

$$\tilde{P} = \begin{pmatrix} 1 & x_1 & \ldots & x_1^n \\ 1 & x_2 & \ldots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \ldots & x_m^n \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix}$$

La première matrice est une matrice de Vandermonde, on sait qu'elle est inversible si et seulement si les (x_i) sont deux à deux distincts. C'est la seule contrainte qu'on a sur les x_i, et il reste maintenant à bien les choisir pour pouvoir accéler le calcul du produit de matrice.

Le(s) bon(s) choix :

Etant donné w une racine primitive n^{ieme} de l'unité, on prends $x_i = w^i$ pour $i = 1 \dots n$. Pourquoi un tel choix ? Plusieurs raisons :

- La matrice de Vandermonde $V(w)$ associée est plutôt sympathique (quand on simplifie en utilisant $w^n = 1$) :

$$V(w) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & w & w^2 & \dots & w^{n-1} \\ 1 & w^2 & w^4 & \dots & w^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{n-1} & w^{n-2} & \dots & w \end{pmatrix}$$

- L'inverse de $V(w)$ n'est autre que $\frac{1}{n}V(w^{-1}) = \frac{1}{n}V(w^{n-1})$, donc évaluer et interpoler reviennent tous deux à calculer un produit de la forme $V(w) \cdot X$.
- Si n est pair, on a $w^{\frac{n}{2}} = -1$ (car w est primitive), soit $w^{\frac{n}{2}+i} = -w^i$, et donc les lignes numéros i et $\frac{n}{2} + i$ se ressemblent beaucoup et on peut donc espérer économiser des calculs.

Diviser pour régner :

On va maintenant prouver que la multiplication $V(w) \cdot X$ peut être effectué efficacement si n est une puissance de 2. Attention, il ne s'agit pas ici d'une condition pour nous simplifier les calculs, on en a vraiment besoin pour avoir des propriétés sur w.

Lemme. *Si $n = 2^m$, on peut calculer $V(w) \cdot X$ en temps $O(n \log n)$.*

Démonstration. Reprenons plus en détails le fait que $w^{\frac{n}{2}+i} = -w^i$ en regardant les lignes i et $\frac{n}{2} + i$ dans la matrice :

$$V(w) = \begin{pmatrix} 1 & w^i & w^{2i} & w^{3i} & w^{4i} & \dots & w^{i(n-1)} \\ 1 & -w^i & w^{2i} & -w^{3i} & w^{4i} & \dots & -w^{i(n-1)} \end{pmatrix}$$

Ainsi, on a $(V(w) \cdot X)_i = u_i + v_i$ et $(V(w) \cdot X)_{\frac{n}{2}+i} = u_i - v_i$, avec :

$$\begin{cases} u_i &= \sum_{j \equiv 0[2]} w^{ij} X_j \\ v_i &= \sum_{j \equiv 1[2]} w^{ij} X_j \end{cases}$$

En changeant de variable dans les sommes, et en factorisant par w^i dans v_i, on obtient :

$$\begin{cases} u_i = \sum_{k=0}^{n/2} w^{2ik} X_{2k} &= \sum_{k=0}^{n/2} (w^2)^{ik} X_{2k} \\ v_i = w^i \sum_{k=0}^{n/2} w^{2ik} X_{2k+1} &= w^i \sum_{k=0}^{n/2} (w^2)^{ik} X_{2k+1} \end{cases}$$

On voit apparaitre deux nouvelles instances de notre problème, avec comme nouveaux paramètres $\frac{n}{2}$ et w^2. Ainsi, si on note $U = V(w^2) \cdot X_{\text{pair}}, V = V(w^2) \cdot X_{\text{impair}}$, on peut calculer $Z = V(w) \cdot X$ via :

$$\begin{cases} Z_i & = U_i + w^i V_i \\ Z_{\frac{n}{2}+i} & = U_i - w^i V_i \end{cases}$$

Finalement, la complexité $T(n)$ vérifie la relation de récurrence : $T(n) = 2T(n/2) + O(n)$, et par le master theorem, on obtient $T(n) = O(n \log n)$. □

Conclusion :

Pour finir la preuve, il suffit de montrer que la condition que n soit une puissance de 2 n'est pas génante. En effet, lorsque qu'on considère le vecteur associé à P, on peut artificiellement le combler avec des 0 pour tomber sur une puissance de 2 :

$$P(X) = \sum_{i=0}^{n} a_i X^i \longrightarrow \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} \longrightarrow \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$