Algorithmique

I - Passage de la représentation "tableau" à "couples"

1 - Si n est pair $\Leftrightarrow n = 2k$; $k \in \mathbb{N}$

$$\alpha^{\frac{n}{2}} = \alpha^{k} = e^{i\pi} = \cos(\pi) + i \sin(\pi) = -1$$

$$\alpha^{n} = e^{i2\pi} = \cos(2\pi) + i \sin(2\pi) = 1$$

α est donc bien une racine de l'unité

2 - Soit $\alpha^{r} = e^{i\frac{2\pi r}{n}} = \cos\left(\frac{2\pi r}{n}\right) + i \sin\left(\frac{2\pi r}{n}\right)$

Soit $d \neq r$ t.q $\alpha^{r} = \alpha^{d} \Leftrightarrow \frac{d}{n} \in \mathbb{N}$ et $\frac{r}{n} \in \mathbb{N}$

or c'est impossible car r et $d < n$
de plus r et d ne peuvent pas être nuls tous les deux.

Montrons maintenant que ce sont des racines n-ième de l'unité.

Posons $(\alpha^{r})^{n} = 1$, l'hypothèse de récurrence avec $r \in [\![0, n-1]\!]$

$(\alpha^{0})^{n} = 1$ donc vrai pour $r = 0$.

On suppose donc notre hypothèse vraie pour tout r.

On a $(\alpha^r)^n = 1$ par ①

$$(\alpha^r)^n \alpha^n = 1$$
$$\alpha^{rn+n} = 1$$
$$(\alpha^{r+1})^n = 1$$

On a donc que notre hypothèse est vraie au rang $r+1$

Ce sont donc bien des racines n-ièmes

3- Si n est pair $\Leftrightarrow n = 2k$; $k \in \mathbb{N}$

Soit $r \in \{0, \dots, n-1\}$

$$(\alpha^r)^2 = \left(\left(e^{i\frac{2\pi}{n}}\right)^r\right)^2 = e^{\frac{i2\pi r}{k}}$$

et $\left(e^{\frac{i2\pi r}{k}}\right)^{\frac{n}{2}} = e^{i2\pi r}$

$\Leftrightarrow \cos(2\pi r) + i\sin(2\pi r)$
$= 1$

Pour n pair, on a donc bien que les carrés des n nombres $1, \alpha, \alpha^2, \alpha^{n-1}$ sont les racines $\frac{n}{2}$-ièmes de l'unité.

4- Soit $P(x) = a_0 + a_1 x + \dots + a_{n-1}x^{n-1}$, avec n pair

De plus, $P(x) = a_0 + a_2 x^2 + \dots + a_{n-2}x^{n-2}$
$+ x(a_1 + a_3 x^2 + \dots + a_{n-1}x^{n-2})$

Ainsi $P_0(x) = a_0 + a_2 x + \dots + a_{n-2} x^{\frac{n-2}{2}}$

et $P_1(x) = a_1 + a_3 x + \dots + a_{n-1} x^{\frac{n-2}{2}}$

De plus, on a : $P(x) = P_0(x^2) + x P_1(x^2)$

Si on pose : $x = \alpha^i$, on a : $x^2 = (\alpha^i)^2 = \sqrt[\frac{n}{2}]{1}$

Donc : $P(\alpha^i) = P_0(\alpha^{i \times 2}) + \alpha^i P_1(\alpha^{2i})$

5 - Dans le cas n impair, il faut se ramener au cas pair, pour cela on va ajouter un coefficient ($=0$) pour se ramener au polynôme

6) On suppose que n est le nombre de coefficients du polynôme et que R_i correspond au i-ème coefficient du polynôme R.

~~Algo~~ Algo coeff-To-couple (polynôme P)
Si n est impair alors $R_n = 0$ et $\deg(P) = \deg(P) + 1$
Sinon si $(n = 1)$ alors renvoyer R
Sinon $W_n = e^{i\frac{2\pi}{n}}$ et $w = 1$
P_pair = coefficients_pairs
P_impair = coefficients_impairs (P)
A = coeff-To-couple (P_pair)
B = coeff-To-couple (P_impair)
Pour i allant de 0 à $\frac{n}{2} - 1$
$R_i = A_i + w B_i$
$R_{i+\frac{n}{2}} = A_i - w B_i$
$w = w \times W_n$
fin Pour
fin si
renvoyer R

Analyse de complexité :

- Pour cet algorithme, le pire cas, le meilleur cas et le cas moyen sont les mêmes puisqu'il ne dépend pas des valeurs du polynome.

- Maintenant, posons $C(n)$ la fonction de complexité associée à cette fonction.

On a alors : $C(n) = C\left(\frac{n}{2}\right) + C\left(\frac{n}{2}\right) + x$

avec x la complexité d'une itération.
Ici $x = O(n)$ car les fonctions coefficients-pairs coefficients-impairs et la boucle sont en $O(n)$.
Il suffit donc de parcourir une fois les n coefficients du polynôme et d'effectuer un test de parité en $O(1)$

Donc $x = 3\,O(n) = O(n)$

Ainsi $C(n) = 2C\left(\frac{n}{2}\right) + O(n)$

Comme on divise n par 2 à chaque itération, on va donc avoir $\log_2(n)$ itérations.
Sachant que l'on appelle deux fois notre algorithme à chaque itération on va avoir :

$$2^{\log_2(n)} = n$$

On va maintenant prendre en compte le $O(n)$ de chaque itération en montrant que $C(n) \in O(n \log_2(n))$.

$\exists \alpha \in \mathbb{R} \mid C(n) \leq 2\alpha \frac{n}{2} \log_2\left(\frac{n}{2}\right) + \beta n$

$= \alpha n (\log_2(n) - 1) + \beta n$

$= \alpha n \log_2(n) + (\beta - \alpha) n$

Or on peut toujours choisir $\alpha > \beta$ car ce dernier est fixé (3 dans notre cas, la boucle + les deux fonctions sur les coefficients).

$\Rightarrow \beta - \alpha < 0 \Rightarrow C(n) \in O(n \log_2(n))$

II - Algorithme de Karatsuba

1- On choisit la représentation "coefficients" car elle permet une manipulation plus facile des polynômes.

Algo Karatsuba (polynôme A, polynôme B)
* A_0 = première_partie (A)
A_1 = dernière_partie (A)
B_0 = première_partie (B)
B_1 = dernière_partie (B)
Y = Karatsuba ($A_0 + A_1$, $B_0 + B_1$)
U = Karatsuba (A_0, B_0)
Z = Karatsuba (A_1, B_1)

$Y = Y - U - Z$
renvoyer $U + Y \times x^{\frac{n}{2}} + Z \times x^{n}$

* Si $n < 3$ alors renvoyer $(A \times B)$

n est le nombre de coefficients du polynôme de degré le plus élevé.

Analyse de complexité : On négligera $n < 3$

- Pour cet algorithme, comme pour le précédent, les différents cas ont la même complexité.

- On pose $K(n)$ la fonction de complexité associée à notre algorithme.

$$K(n) = 3K\left(\frac{n}{2}\right) + x$$

Ici $x = O(n)$ car les fonctions ~~[illegible]~~ utilisées sont linéaires en la taille des données, tout comme l'addition, la soustraction ou bien le décalage de polynômes.

Ainsi $K(n) = 3K\left(\frac{n}{2}\right) + O(n)$

2- On remarque que notre fonction de complexité est identique à celle vue en cours, on a donc la même complexité que pour les entiers soit :

$$O(n^{\log_2(3)}) \approx O(n^{1,59})$$

III- Expérimentation numérique.

Voir ci-dessous

IV - Pour les courageux

1) Si l'on veut effectuer une multiplication à partir de la représentation tableau et en obtenant une représentation tableau tout en utilisant la multiplication par couples (en $O(n)$) alors il faut effectuer les étapes suivantes :

$O(n \log n)$ Passage représentation tableau $\rightarrow$ couples

$\downarrow$

$O(n)$ Multiplication couples

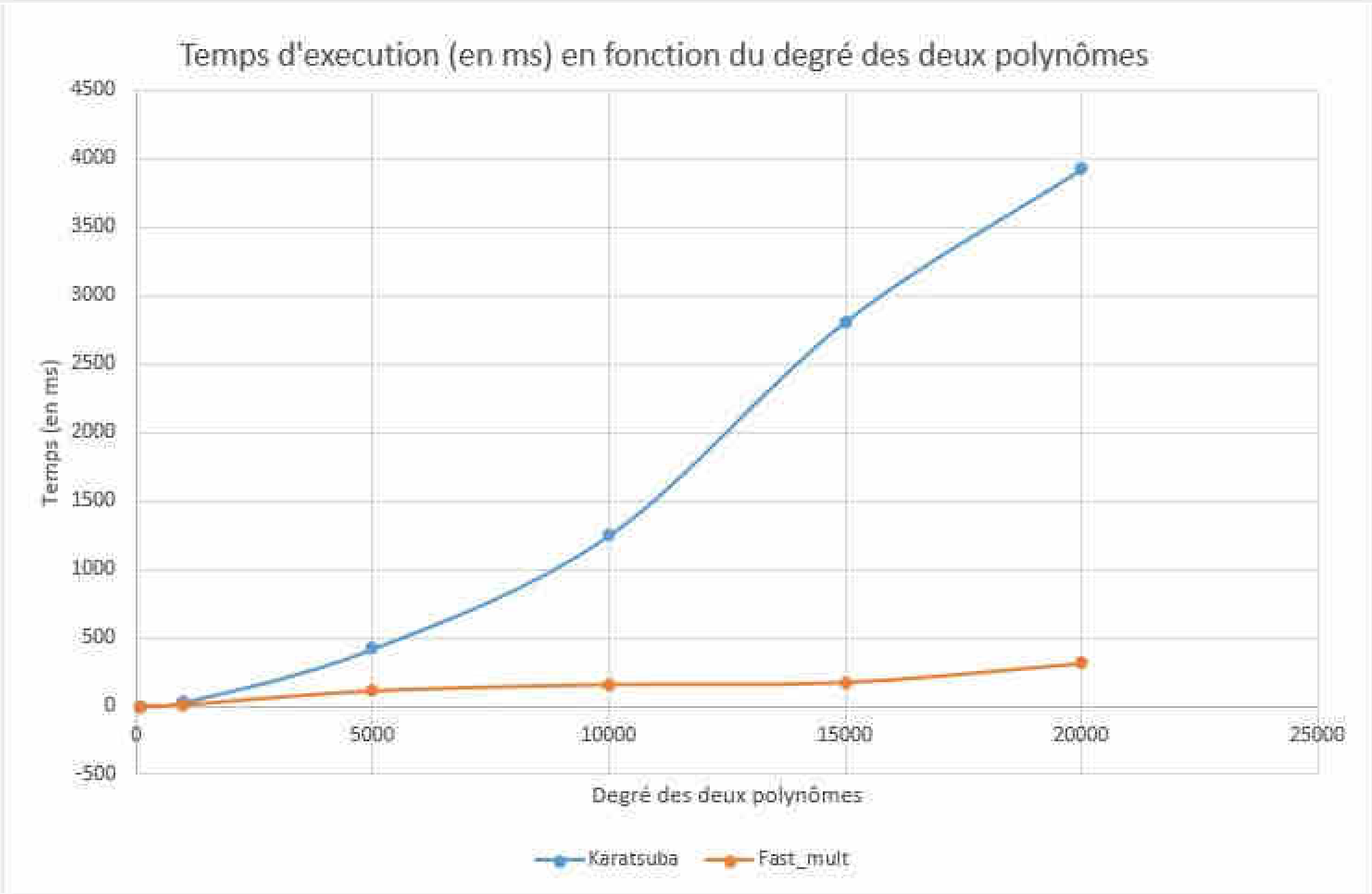
$\downarrow$

x Passage représentation couples $\rightarrow$ tableau

On remarque donc que si cette dernière étape peut se faire en $O(n \log(n))$ alors le coût total sera :

$$O(n \log(n)) + O(n) + O(n \log(n)) = O(n \log n)$$

Il est donc possible de calculer le produit de deux polynômes sous forme "tableau" en $O(n \log(n))$ tout en obtenant un résultat sous la forme rouleau.



Sur ce graphe on peut voir la différence de temps très importante entre Karatsuba et celui consistant à transformer la représentation en couples puis à appliquer l'algorithme en O(n).