TD2 – Preuve d'algorithmes et complexité

M. Bombar, L. Poyeton

Exercice 1 – Division euclidienne des polynômes (TD01)

Soit K un corps commutatif. On représente les polynômes par des listes de coefficients comme au TD01. Soient $P, Q \in K[X]$, avec $Q \neq 0$.

(Q1) Rappeler l'algorithme de division Euclidienne des polynômes.

(Q2) Exécuter l'algorithme à la main pour $P = X^3 + X + 2$ et $Q = 2X - 1$ dans $\mathbb{Q}[X]$.

(Q3) Donner un *invariant de boucles* pour prouver la correction partielle.

(Q4) Donner un *variant de boucles* qui décroît strictement à chaque tour de boucle. En déduire une preuve de terminaison de cet algorithme.

(Q5) Programmer cet algorithme en Sage, et tester cette implémentation.

(Q6) Quelle condition doit vérifier Q afin que cet algorithme fonctionne dans un anneau commutatif quelconque ?

Exercice 2 – Évaluation et schéma de Horner (TD01)

Soit $P = p_0 + p_1 X + \cdots + p_d X^d$.

(Q1) Écrire une première fonction d'évaluation qui calcule séparément les puissances x^i, puis comparer son coût à celui d'une évaluation mieux organisée.

(Q2) Montrer l'identité

$$P(x) = p_0 + x\big(p_1 + x(p_2 + \cdots + x(p_{d-1} + x p_d)\cdots)\big),$$

et en déduire un algorithme, dû à Horner, pour évaluer un polynôme.

(Q3) Programmer `horner(P,x)` pour un polynôme représenté par une liste.

(Q4) Pour $d \geq 1$, donner le nombre exact d'additions et de multiplications effectuées par Horner.

(Q5) Évaluer $2X^5 - 3X^3 + X - 7$ en $x = 3$ en donnant les valeurs successives de l'accumulateur, puis vérifier avec SageMath.

Exercice 3 – Une fonction mystère

On considère la fonction SageMath suivante :

```
def myst(n):
    if n <= 1:
        return n
    return myst(n-1) + myst(n-2)
```

(Q1) Que calcule cette fonction ?

(Q2) Montrer que pour tout n on a

$$\texttt{myst(n)} = \frac{1}{\sqrt{5}}(\varphi^n - \overline{\varphi}^n)$$

où $\varphi = \dfrac{1+\sqrt{5}}{2}$ et $\overline{\varphi} = \dfrac{1-\sqrt{5}}{2}$.

(Q3) Soit c_n le nombre d'additions effectuées par `myst(n)`. Montrer que

$$c_0 = c_1 = 0, \qquad c_n = c_{n-1} + c_{n-2} + 1 \quad (n \geq 2).$$

(Q4) En déduire que la complexité algébrique de `myst` est exponentielle.

(Q5) Proposer une version itérative de cet algorithme, utilisant $O(n)$ additions et une quantité constante de mémoire supplémentaire. Donner un invariant de boucle prouvant sa correction.

(Q6) La complexité *binaire* de cet algorithme est-elle polynomiale ?

Exercice 4 – Exponentiation binaire

On considère l'algorithme suivant.

Algorithme 1 : Exponentiation binaire

Entrées : $x \in M$ et $e \in \mathbb{N}$
Sortie : ?

```
1 y ← 1
2 z ← x
3 n ← e
4 tant que n > 0 faire
5     si n est impair alors
6         y ← y * z
7     z ← z^2
8     n ← ⌊n/2⌋
9 Renvoyer y
```

(Q1) Exécuter l'algorithme pour $e = 13$, puis pour $e = 93$. À chaque tour, indiquer les valeurs de y, z et e. Que calcule cet algorithme ?

(Q2) Programmer l'algorithme en Sage.

(Q3) En notant e_0 l'exposant initial, montrer que

$$yz^e = x^{e_0}$$

est un invariant de boucle. En déduire la correction et la terminaison.

(Q4) On note N le nombre de bits de e_0, et w le nombre de 1 dans son écriture binaire. Donner le nombre de multiplications effectuées.

Exercice 5 – $(\star)$ Vers le cours 03 : Algorithme de Karatsuba

Dans cet exercice, on suppose que $n = 2^k$ est une puissance de deux, et on pose $m = n/2$. Soient $A, B \in \mathbb{K}[X]$ deux polynômes de degrés $< n$, et on écrit

$$A = A_0 + X^m A_1, \qquad B = B_0 + X^m B_1,$$

où A_0 et B_0 sont les parties de degrés strictement inférieurs à m.

(Q1) Exprimer le produit AB en fonction de $A_i B_j$. De combien de multiplications de polynômes a-t-on besoin ? Bien préciser les degrés, et en déduire la complexité de cet algorithme. Comparer avec la complexité de l'algorithme « Naïf ».

(Q2) L'algorithme de Karatsuba propose de changer le découpage. Posons

$$Z_0 = A_0 B_0, \qquad Z_2 = A_1 B_1, \qquad Z_1 = (A_0 + A_1)(B_0 + B_1) - Z_0 - Z_2.$$

Montrer que

$$AB = Z_0 + X^m Z_1 + X^{2m} Z_2.$$

Combien de produits sont maintenant nécessaires ?

(Q3) Dérouler cet algorithme à la main sur les exemples suivants

$$A = 1 + 2X + 3X^2 + 4X^3, \qquad B = 2 - X + X^2 + 3X^3,$$

.

(Q4) Implémenter l'algorithme en Sage, et le tester sur des exemples. On conserve la représentation des polynômes comme liste de leurs coefficients que nous avons utilisée au TD01, et on pourra supposer que les listes en entrée sont toujours de taille $n = 2^k$, quitte à les compléter par des 0.

(Q5) Montrer que le nombre de produits de coefficients satisfait la relation de récurrence

$$M(n) = 3M(n/2),$$

tandis que les additions et recombinaisons coûtent $\mathcal{O}(n)$ à chaque appel.

(Q6) Démontrer que l'algorithme de Karatsuba effectue $\Theta(n^{\log_2 3})$ opérations dans $\mathbb{K}$.