UL245

SESSION 2005

Filière : 2ème concours

INFORMATIQUE

Durée : 3 heures

L'usage des calculatrices électroniques de poche à alimentation autonome, sans imprimante et sans document d'accompagnement est autorisé. Cependant, une seule calculatrice à la fois est admise sur la table ou le poste de travail. Aucun échange n'est permis entre les candidats.

Tournez la page S.V.P.

Le problème comporte 8 questions. Il est recommandé de lire l'ensemble du sujet avant de commencer la rédaction. Il est également conseillé de traiter les questions dans l'ordre de l'énoncé. On pourra cependant aborder une question en admettant les résultats des questions précédentes. Les algorithmes pourront être écrits dans un langage au choix du candidat, en utilisant les structures de contrôle habituelles. Une attention particulière sera portée à l'analyse des algorithmes proposés (en particulier, les coûts annoncés seront clairement justifiés).

Soient $\mathbb{N}$ l'ensemble des entiers naturels, $\mathbb{N}^*$ l'ensemble des entiers strictement positifs, $\mathbb{C}$ l'ensemble des nombres complexes et $\mathbb{C}[x]$ l'anneau des polynômes en la variable x et à coefficients dans $\mathbb{C}$.

On appelle *opération arithmétique dans* $\mathbb{C}$ (et on abrégera par "op") n'importe quelle opération dans $\mathbb{C}$ parmi $\{+, -, \times, /\}$.

Pour $n \in \mathbb{N}^*$, on introduit la fonction $\mathsf{M}(n)$ pour désigner le coût (en nombre d'ops) de la *multiplication de deux polynômes de degré* $< n$: si $a(x) = \sum_{i=0}^{n-1} a_i x^i$ et $b(x) = \sum_{i=0}^{n-1} b_i x^i$ sont donnés par leurs coefficients $a_i, b_i \in \mathbb{C}$, on peut calculer les coefficients $c_i \in \mathbb{C}$ du produit $c(x) = a(x)b(x) = \sum_{i=0}^{2n-2} c_i x^i$ en $\mathsf{M}(n)$ ops. On a $\mathsf{M}(1) = 1$ et l'algorithme de multiplication "classique" donne $\mathsf{M}(n) \leq 2n^2 - 2n + 1$. En utilisant la notation O habituelle ("grand O"), on a donc $\mathsf{M}(n) = O(n^2)$ et on dira que la *constante associée au terme dominant*, ici n^2, vaut 2. On admettra enfin dans toute la suite de ce document que, pour n pair,

$$\mathsf{M}(n/2) \leq \frac{1}{2}\mathsf{M}(n).$$

Pour $n, p \in \mathbb{N}^*$, on introduit une autre fonction, $\mathsf{E}(n, p)$, qui cette fois désigne le coût (toujours en nombre d'ops) de l'*évaluation en p points d'un polynôme de degré* $< n$: si $\alpha_1, \alpha_2, \ldots, \alpha_p \in \mathbb{C}$ sont donnés et si $a(x) = \sum_{i=0}^{n-1} a_i x^i$ est donné par ses coefficients $a_i \in \mathbb{C}$, on peut calculer les p valeurs $a(\alpha_1), a(\alpha_2), \ldots, a(\alpha_p) \in \mathbb{C}$ en $\mathsf{E}(n, p)$ ops.

Pour $m \in \mathbb{N}^*$, on désigne par $\mathbb{C}[x]^{m \times m}$ l'ensemble des matrices $m \times m$ dont les éléments sont dans $\mathbb{C}[x]$. Cet ensemble coïncide avec l'ensemble des polynômes en x et à coefficients dans $\mathbb{C}^{m \times m}$. On peut donc voir tout élément de $\mathbb{C}[x]^{m \times m}$ non seulement comme une *matrice de polynômes* mais aussi comme un *polynôme de matrices*. Un exemple avec $m = 2$ est:

$$\begin{bmatrix} 2x+1 & 4x+3 \\ -x+2 & -2x+4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ -1 & -2 \end{bmatrix} x + \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}.$$

Soit $A(x) \in \mathbb{C}[x]^{m \times m}$ non nulle. On appelle *degré de* $A(x)$ la plus grande puissance de x apparaissant parmi les m^2 polynômes éléments de $A(x)$. Si $A(x)$ est de degré d alors $A(x)$ admet

l'écriture polynomiale suivante:

$$A(x) = \sum_{i=0}^{d} A_i x^i \text{ avec } A_0, A_1, \ldots, A_d \in \mathbb{C}^{m\times m} \text{ et où } A_d \text{ n'est pas la matrice nulle.}$$

Les $d+1$ matrices A_i seront appelées les *coefficients de* $A(x)$. L'exemple ci-dessus montre donc une matrice 2×2 de degré 1, de la forme $A(x) = A_1 x + A_0$ et dont les deux coefficients sont A_0 et A_1. On notera que, lorsque $m = 1$, on retrouve les notions habituelles de degré et de coefficients d'un polynôme en une variable.

Soient $\alpha \in \mathbb{C}$ et $A(x) \in \mathbb{C}[x]^{m\times m}$. On appelle *valeur de* $A(x)$ *en* α et on note $A(\alpha)$ la matrice dans $\mathbb{C}^{m\times m}$ obtenue en remplaçant x dans $A(x)$ par α. Par exemple, pour $\alpha = 0$ on a $A(0) = A_0 \in \mathbb{C}^{m\times m}$.

Soit $m \in \mathbb{N}^*$. **Dans toute la suite du document, on suppose désormais donnés**
- un vecteur $v_0 \in \mathbb{C}^m$ non nul;
- deux matrices $A_0, A_1 \in \mathbb{C}^{m\times m}$ définissant $A(x) \in \mathbb{C}[x]^{m\times m}$ non nulle et de la forme $A(x) = A_1 x + A_0$.

Pour $n \in \mathbb{N}^*$, soit $v_n \in \mathbb{C}^m$ le vecteur défini par le produit

$$v_n = A(n-1)A(n-2)\cdots A(1)A(0)v_0. \tag{1}$$

Le problème qui suit a pour but la mise au point d'un algorithme permettant de calculer le n-ième terme v_n de la suite $\{v_i\}_{i\in\mathbb{N}}$ définie par (1) et dont le coût en nombre d'ops soit sous-linéaire en le paramètre n. (Par sous-linéaire en n, on entend une fonction de la forme $o(n)$ avec la notation o habituelle ("petit o").)

Hypothèses importantes. Pour simplifier, on prendra dans toute la suite de ce document $n = 4^q$ avec $q \in \mathbb{N}$; on admettra de plus que $\mathsf{M}(n) = O(n \log n)$ où log désigne le logarithme en base 2. Enfin, on suppose disposer des deux fonctions suivantes:

- la fonction MulPoly qui, étant donnés les coefficients de $a(x), b(x) \in \mathbb{C}[x]$ avec $a(x)$ et $b(x)$ de degré $< n$, retourne les coefficients du produit $c(x) = a(x)b(x)$ en $\mathsf{M}(n)$ ops.

- la fonction EvalPoly qui, étant donnés $\alpha_1, \alpha_2, \ldots, \alpha_p \in \mathbb{C}$ et les coefficients de $a(x) \in \mathbb{C}[x]$ avec $a(x)$ de degré $< n$, retourne les valeurs $a(\alpha_1), a(\alpha_2), \ldots, a(\alpha_p)$ en $\mathsf{E}(n, p)$ ops.

On pourra faire appel à ces fonctions pour simplifier l'écriture des algorithmes demandés.

1 Préliminaires

Cette première partie permet d'étudier deux cas particuliers pour $A(x)$ et de donner, dans le cas général, un algorithme naïf.

Question 1
On suppose dans cette question que $m = 1$, $v_0 = 1$ et $A(x) = x + 1$.

1. Que vaut v_n ?
2. Donner un algorithme qui calcule le n-uplet $(v_0, v_1, \ldots, v_{n-1})$ en $O(n)$ ops.
3. Soient $\alpha \in \mathbb{C}$ et $a(x) = \sum_{i=0}^{n-1} a_i x^i \in \mathbb{C}[x]$. Pour $j \in \mathbb{N}$, soit $a^{(j)}(x)$ la dérivée d'ordre j de $a(x)$. En utilisant le fait que, pour $0 \leq j < n$, $a^{(j)}(\alpha)$ est égal au coefficient en x^{n-1-j} de $(\sum_{i=0}^{n-1} v_i a_i x^{n-i-1}) \times (\sum_{i=0}^{n-1} \frac{\alpha^i}{v_i} x^i)$, montrer à l'aide de la question précédente qu'à partir de α et des a_i on peut calculer les coefficients $\tilde{a}_i$ du polynôme $a(x+\alpha) = \sum_{i=0}^{n-1} \tilde{a}_i x^i$ en au plus $M(n) + 5n$ ops.

Question 2
On suppose dans cette question que $m \in \mathbb{N}$, $v_0 \in \mathbb{C}^m$ et $A(x) = A_0 \in \mathbb{C}^{m \times m}$.

1. Que vaut v_n ?
2. Soient $P, Q \in \mathbb{C}^{m \times m}$ et $v \in \mathbb{C}^m$. Montrer que l'on peut calculer le produit matrice-vecteur Pv en moins de $2m^2$ ops et le produit matrice-matrice PQ en moins de $2m^3$ ops. (On pourra nommer les fonctions correspondant à ces deux calculs et, dans la suite, faire appel à elles directement afin de simplifier la description des algorithmes.)
3. Donner un algorithme qui calcule v_n en $O(m^3 \log n)$ ops. Préciser la constante associée au terme dominant.

Question 3
On considère désormais le cas général où $A(x)$ est de la forme $A(x) = A_1 x + A_0$ avec $A_0, A_1 \in \mathbb{C}^{m \times m}$ données.

1. Pour $\alpha \in \mathbb{C}$, donner un algorithme qui calcule $A(\alpha) \in \mathbb{C}^{m \times m}$ à partir de A_0 et A_1 et indiquer le nombre exact d'ops qu'il utilise.
2. Décrire un algorithme qui calcule v_n en $O(m^2 n)$ ops et préciser la constante associée au terme dominant.
3. Lorsque $m = O(1)$ et $n = 2^{14}$, de combien l'algorithme de la question 3.2 est-il plus lent que celui de la question 2.3 ? (On se contentera de donner un ordre de grandeur.)

2 Calcul de v_n via les matrices de polynômes

Cette deuxième partie montre comment réduire le coût en n du calcul de v_n en utilisant de façon judicieuse une matrice de polynômes $N_i(x)$ définie ci-dessous à partir de $A(x) = A_1x + A_0$.

On suppose que $i \in \mathbb{N}^*$ est une puissance de 2 et on définit $N_i(x) \in \mathbb{C}[x]^{m\times m}$ par le produit suivant:

$$N_i(x) = A(x+i-1)\cdots A(x+1)A(x). \tag{2}$$

Question 4

1. Soient $P(x)$ et $Q(x)$ dans $\mathbb{C}[x]^{m\times m}$, chacune de degré $< d$. Donner un algorithme qui calcule les coefficients du produit matrice-matrice $P(x)Q(x)$ en moins de $2m^3\mathsf{M}(d)$ ops.
2. Vérifier rapidement que le degré de $N_i(x) \in \mathbb{C}[x]^{m\times m}$ définie en (2) est $\leq i$.
3. Donner un algorithme qui calcule les coefficients de $A(x+1), A(x+2), \ldots, A(x+i-1)$ à partir de ceux de $A(x) = A_1x + A_0$. Quel est son coût ?
4. Montrer que l'on peut calculer les coefficients de $N_i(x)$ à partir de ceux de $A(x) = A_1x+A_0$ en $O(m^3\mathsf{M}(i)\log i)$ ops.

Question 5

1. Soient $n \in \mathbb{N}^*$, $\alpha \in \mathbb{C}$ et $P(x) = \sum_{i=0}^{n-1} P_ix^i \in \mathbb{C}[x]^{m\times m}$ donnée par ses coefficients P_i. En utilisant la question 1.3, donner un algorithme qui calcule les coefficients de $P(x+\alpha)$ en $O(m^2\mathsf{M}(n))$ ops.
2. En déduire un algorithme récursif qui calcule les coefficients de $N_i(x)$ en $O(m^3\mathsf{M}(i))$ ops.

Question 6

On suppose dans cette question que i divise n et on note $k = n/i$.

1. Exprimer v_n en fonction de v_0 et du produit de k valeurs prises par $N_i(x)$.
2. En déduire un nouvel algorithme pour calculer v_n. En quoi cet algorithme généralise-t-il celui de la question 3.2 ?
3. Majorer le coût de l'algorithme obtenu en 6.2 en fonction de $\mathsf{M}(\cdot)$, $\mathsf{E}(\cdot,\cdot)$, m, i et k. (Rappel: la fonction $\mathsf{E}(n,p)$, définie au début de ce document, est le coût de l'évaluation en p nombres complexes d'un polynôme de degré $< n$ donné par ses coefficients.)
4. En supposant que $\mathsf{E}(n,n) = O(\mathsf{M}(n)\log n)$ et en rappelant que $\mathsf{M}(n) = O(n\log n)$, donner un algorithme qui calcule v_n en un nombre d'ops sous-linéaire en n.

3 Évaluation rapide d'un polynôme en plusieurs points

Le but de cette dernière partie est de montrer que $\mathsf{E}(n,n) = O(\mathsf{M}(n)\log n)$.

Soient $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{C}$ et $a(x) = \sum_{i=0}^{n-1} a_i x^i$ donné par ses coefficients $a_i \in \mathbb{C}$.

Question 7

1. Donner un algorithme qui calcule la valeur $a(\alpha_1)$ à partir de la donnée de α_1 et des a_i. Préciser son coût en fonction de n. Quelle est la constante associée au terme dominant ?

2. En déduire une majoration pour $\mathsf{E}(n,p)$. En quoi est-elle très satisfaisante lorsque $p = 1$? En quoi n'est-elle pas satisfaisante lorsque $p = n$?

Question 8
Soit $b(x) = (x-\alpha_1)(x-\alpha_2)\cdots(x-\alpha_{n/2}) \in \mathbb{C}[x]$.

1. Montrer comment calculer les coefficients $b_i \in \mathbb{C}$ de $b(x) = \sum_{i=0}^{n/2} b_i x^i$ défini ci-dessus en $O(\mathsf{M}(n/2)\log n)$ ops.

2. Soit $(q(x), r(x))$ l'unique paire de polynômes telle que $a(x) = b(x)q(x) + r(x)$ avec $r(x)$ de degré $< n/2$. Montrer que si les coefficients de $a(x)$, $b(x)$ et $q(x)$ sont connus alors on peut déduire ceux de $r(x)$ en $\mathsf{M}(n/2) + n/2$ ops.

3. En admettant qu'on sait déduire les coefficients de $q(x)$ de ceux de $a(x)$ et $b(x)$ en $O(\mathsf{M}(n/2))$ ops, donner un algorithme récursif qui calcule $\{a(\alpha_i)\}_{1\leq i\leq n}$. Majorer son coût en nombre d'ops en fonction de n et $\mathsf{M}(\cdot)$.

4. Conclure que $\mathsf{E}(n,n) = O(\mathsf{M}(n)\log n)$.

— Fin de l'énoncé —