Option Informatique en Spé MP et MP*

Devoir surveillé du mercredi 28 mars 2001

Résumé

Il existe un lien logique entre les trois parties de ce sujet. Elles peuvent toutefois être traitées indépendamment les unes des autres.

Veuillez rédiger chaque partie sur une copie séparée.

Table des matières

1 Une récurrence intéressante

► On note $\dbinom{n}{p} = \dfrac{n!}{p!(n-p)!}$. On note $a_n = \dfrac{\binom{2n}{n}}{n+1}$ et $S_n = \sum_{0 \leqslant k \leqslant n} a_k a_{n-k}$.

Question 1 • Calculez a_n et S_n pour $0 \leqslant n \leqslant 4$ Que constatez-vous ?

Question 2 • Trouvez une relation *très simple* entre a_{k+1} et a_k.

► Notons $\mathcal{A}(n)$ l'assertion $S_n = a_{n+1}$; l'assertion $\mathcal{A}(0)$ est immédiate. Supposons $\mathcal{A}(n)$ acquise.

Question 3 • On note $T_n = \sum_{0 \leqslant k \leqslant n} k a_k a_{n-k}$. Trouvez une relation *très simple* entre T_n et S_n.

Question 4 • Donnez une expression *très simple* de $T_{n+1} + S_{n+1}$.

Question 5 • Montrez alors que $S_{n+1} = a_{n+2}$ et concluez.

2 Étude d'un langage

► On note Σ l'alphabet $\{a,b\}$. Un mot v est *préfixe propre* d'un mot u s'il existe un mot w non vide tel que $u = vw$. On note L le langage formé des mots $u \in \Sigma^*$ qui vérifient $|u|_a = |u|_b$, et $|v|_a \geqslant |v|_b$ pour tout préfixe propre v de u. On note $L_n = L \cap \Sigma^{2n}$ l'ensemble des mots de L de longueur $2n$, et $C_n = |L_n|$ son cardinal.

Question 6 • Énumérez L_n pour $0 \leqslant n \leqslant 3$.

Question 7 • On suppose défini le type Caml suivant :

```
type lettre = a | b ;;
```

Notez bien que ce type décrit un ensemble à deux éléments (constructeurs constants `a` et `b`). Rédigez en Caml une fonction de signature

```
bon_mot : lettre list -> bool
```

spécifiée comme suit : `bon_mot u` détermine si u est un mot de L. Par exemple, `bon_mot [a;a;b;b]` donnera la valeur `true`, tandis que `bon_mot [a;b;b;b;a;a]` donnera la valeur `false`.

Question 8 • Le langage L_n est-il rationnel ?

Question 9 • Le langage L est-il rationnel ?

Question 10 • Prouvez l'inégalité $C_n \leqslant \binom{2n}{n}$.

Question 11 $\star\star\star$ • Soit $u \in L \setminus \{\varepsilon\}$. Montrez qu'il existe un et un seul couple (v, w) de mots de L vérifiant $u = avbw$.

Question 12 • Justifiez alors la relation $C_{n+1} = \sum_{0 \leqslant k \leqslant n} C_k C_{n-k}$ pour $n \in \mathbb{N}$.

Question 13 • Utilisez le résultat de la partie 1 pour obtenir une expression simple de C_n.

Question 14 • Montrez que C_n est égal au nombre de chemins qui, dans un quadrillage analogue à celui de la figure 1, mènent du point $(0,0)$ au point (n,n) en vérifiant les deux conditions suivantes :

1. chaque pas se fait vers la droite ou vers le haut ;
2. le chemin ne passe jamais au-dessus de la diagonale.

Sur la figure, un exemple d'un tel chemin est indiqué, en traits **gras**.

► On considère la fonction Caml suivante :

```
let rec c = function
 | 0 -> 1
 | n -> let v = intervalle 0 (n-1) and f k = c(k) * c(n-1-k))
   in it_list (prefix +) 0 (map f v) ;;
```

Les fonctions `map` et `it_list` font partie de la bibliothèque Caml ; `intervalle` construit la liste des éléments d'un intervalle discret.

Question 15 • Montrez que l'évaluation de C_n avec cette fonction a un coût au moins exponentiel.

Question 16 • Proposez une rédaction avec laquelle le coût de l'évaluation de C_n est un $\mathcal{O}(n^2)$.

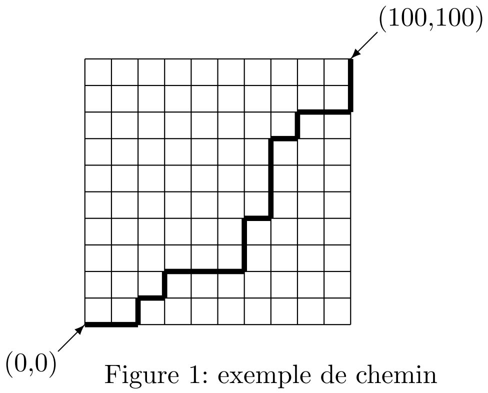

Figure 1: exemple de chemin

3 Calcul optimal d'un produit de n matrices

Question 17 • Rédigez en Caml une fonction de signature

```
produit_matriciel : int vect vect -> int vect vect -> int vect vect
```

spécifiée comme suit : `produit_matriciel a b` effectue le produit des matrices `a` et `b` (si leurs dimensions sont compatibles) ou lève une exception (dans le cas contraire).

▶ Appelons *coût* d'un produit matriciel le nombre de multiplications nécessaires pour calculer ce produit (on néglige les additions).

Question 18 • Montrez que le coût du calcul (par la méthode usuelle, que vous avez utilisée à la question 17) du produit des matrices $A \in \mathcal{M}_{n,p}(\mathbb{K})$ et $B \in \mathcal{M}_{p,q}(\mathbb{K})$ est npq.

Question 19 • Il existe deux façons de calculer le produit $A \times B \times C$. Comparez leurs coûts respectifs lorsque $A \in \mathcal{M}_{3,10}(\mathbb{K})$, $B \in \mathcal{M}_{10,5}(\mathbb{K})$ et $C \in \mathcal{M}_{5,8}(\mathbb{K})$.

Question 20 • Soit $\tau \in \mathbb{N}^*$. Déterminez des naturels n, p, q et r tels que, si $A \in \mathcal{M}_{n,p}(\mathbb{K})$, $B \in \mathcal{M}_{p,q}(\mathbb{K})$ et $C \in \mathcal{M}_{q,r}(\mathbb{K})$ le rapport entre les coûts des calculs de $(A \times B) \times C$ et $A \times (B \times C)$ soit au moins égal à τ.

▶ On se propose de décrire plusieurs algorithmes déterminant le coût minimal du calcul du produit $A_1 \times A_2 \times \cdots \times A_n$ de n matrices. On note ℓ_k le nombre de lignes et c_k le nombre de colonnes de la matrice A_k. On a donc $c_k = \ell_{k+1}$ pour $1 \leqslant k < n$. On décide de noter $\ell_{n+1} = c_n$; ainsi, le problème est entièrement décrit par la liste $\ell = (\ell_1, \ldots, \ell_{n+1})$.

▶ Notons K_n le nombre de façons différentes d'organiser le calcul de ce produit. Ainsi, $K_1 = K_2 = 1$ et $K_3 = 2$.

Question 21 • Déterminez K_4.

Question 22 • Justifiez la relation $K_{n+1} = \sum\limits_{0 \leqslant j \leqslant n} K_j K_{n-j}$ pour $n \in \mathbb{N}$. Quelle relation existe-t-il entre la famille $(K_n)_{n \geqslant 1}$ et la famille $(C_n)_{n \geqslant 0}$ définie dans la partie 2 ?

▶ Pour $1 \leqslant i < j \leqslant n$, on note $\mu(i,j)$ le coût minimal du calcul du produit $A_i \times \cdots \times A_j$. En particulier, $\mu(i,i) = 0$ et $\mu(i,i+1) = \ell_i \ell_{i+1} \ell_{i+2}$.

Question 23 • Pour $1 \leqslant i < j \leqslant n$, justifiez la formule suivante :

$$\mu(i,j) = \min_{i \leqslant k < j} \big(\mu(i,k) + \mu(k+1,j) + \ell_i \ell_{k+1} \ell_{j+1}\big)$$

▶ La formule précédente nous donne un algorithme récursif : si $n = 1$, il n'y a aucun travail à effectuer. Sinon, pour chaque indice $k \in [\![1, n-1]\!]$, on remplace les matrices A_k et A_{k+1} par leur produit ; on détermine le coût de chacun de ces $n-1$ sous-problèmes, puis le minimum des $n-1$ coûts. On note γ_n le nombre de produits matriciels requis pour le calcul de $\mu(1,n)$.

Question 24 • Écrivez une relation simple entre γ_n et γ_{n-1}. Déduisez-en l'expression de γ_n. Cette méthode de programmation est-elle acceptable ?

Question 25 • Pour déterminer le coût minimal, est-il vraiment nécessaire de calculer tous ces produits matriciels ?

▶ En fait, la méthode précédente recalcule de nombreuses fois chaque $\mu(i,j)$. Pour éviter ce gaspillage, on décide de placer les valeurs de $\mu(i,j)$, pour $1 \leqslant i \leqslant j \leqslant n$, dans un tableau M carré de côté n.

Question 26 • Expliquez comment organiser les calculs pour que le coût de construction de ce tableau soit un $\mathcal{O}(n^3)$.

Question 27 • Appliquez cette méthode au calcul du coût optimal, lorsque $\ell = (3, 10, 5, 8, 4)$.

Question 28 • Rédigez en Caml une fonction de signature

```
cout_minimal : int list -> int
```

spécifiée comme suit : `cout_minimal l` calcule le coût minimal associé à la liste de dimensions ℓ.

▶ En fait, nous souhaitons obtenir, non seulement le coût minimal, mais aussi un ordonnancement des calculs qui permettra d'atteindre ce coût.

Question 29 • Montrez que lors de la construction du tableau M, on peut mémoriser des informations desquelles on déduira un tel ordonnancement, pour un coût supplémentaire $\mathcal{O}(n)$.

Question 30 • Rédigez en Caml une fonction de signature

```
ordonnancement : int list -> int list
```

spécifiée comme suit : `ordonnancement l` construit une liste des produits matriciels à effectuer pour réaliser le coût minimal.

Question 31 • Montrez qu'à chaque façon d'évaluer le produit matriciel $A_1 \times \cdots \times A_n$, on peut associer un arbre binaire de recherche, dont les nœuds et les feuilles sont étiquetées par des intervalles discrets de la forme $[\![i, j]\!]$, avec $1 \leqslant i \leqslant j \leqslant n$. Précisez la relation d'ordre utilisée.

▶ On suppose que les matrices $A_1, \ldots, A_n$ sont logées dans une mémoire externe (par exemple un disque) et sont amenées en mémoire vive à la demande. Les résultats partiels sont conservés en mémoire vive tant qu'ils sont utiles. Par exemple, lorsque l'on évalue le produit de deux matrices, les deux opérandes doivent être présents en mémoire ; une fois le calcul terminé, ils disparaissent, le résultat seul subsistant. Il est donc nécessaire, pour ce calcul, de pouvoir loger simultanément en mémoire au moins trois matrices.

Question 32 • Décrivez les arbres binaires pour lesquels, avec une bonne organisation des calculs, on n'aura jamais besoin de loger plus de trois matrices simultanément en mémoire.

Question 33 $\star\star\star$ • Plus généralement, étant donné un arbre du type évoqué à la question 31, comment proposez-vous de minimiser le nombre de matrices simultanément présentes en mémoire, au cours du calcul décrit par cet arbre ?

FIN