info803 : informatique

TD 1 : diviser pour rêgner

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22
téléphone : 04 79 75 94 22
email : `Pierre.Hyvernat@univ-savoie.fr`
www : `http://www.lama.univ-savoie.fr/~hyvernat/`

Exercice 0 : polynômes

Question 1 : combien de multiplications sont nécessaires pour évaluer un polynôme de degré n sur un entier ? (On suppose que tous les coefficients sont non-nul.)

Question 2 : quel est le nombre (exact) de multiplications et additions utilisées si on calcule les puissances "normalement", mais on les stocke dans un tableau pour ne pas les recalculer. (On suppose à nouveau que tous les coefficients sont non-nuls.)

Question 3 : même question si on utilise la rêgle de Horner :

$$c_0 + c_1 x + c_2 x^2 + \ldots + c_n x^n \quad = \quad \Big(\ldots \Big(\big((c_n x + c_{n-1}) x + c_{n-2} \big) x + c_{n-3} \Big) \ldots \Big) x + c_0$$

Exercice 1 : analyse du tri fusion

Le tri fusion est un algoritme de tri particulier qui a complexité théorique optimale. (On ne le démontrera probablement pas.) En pratique, d'autres tris sont généralement préferrables (tri par tas ou tri rapide).

Le problème est le suivant : on veut trier un tableau contenant les n premiers nombres entiers $(0, \ldots, n-1)$ et on ne peux que faire des comparaisons entre les différents éléments du tableau.

Question 1 : un tri naif consiste à récupérer l'élément minimal du tableau et le mettre en première position. Ensuite, on s'intéresse à la suite du tableau et on recommence...
Écrivez l'agorithme correspondant (tri par selection) et donnez une analyse asymptotique du nombre de comparaisons utilisées en fonction de la taille du tableau. (Dans le pire cas, le meilleur cas et la moyenne...)

Question 2 : le tri fusion est basé sur le fait que fusionner deux tableaux triés en un seul tableau trié est relativement aisé. Pour trier un tableau, on commence donc par le diviser en deux parties égales, on trie chaque morceau et on fusionne les résultats.

Écrivez la fonction `fusionne(t1,t2)` correspondante et analysez son comportement (nombre de comparaisons utilisées) en fonction de la taille des tableaux. (Meilleur cas, pire cas et moyenne.)

Écrivez l'algorithme `tri-fusion(t)` correspondant. Donnez une relation de récurrence pour le nombre de comparaison effectuées dans le pire cas et trouvez une approximation asymptotique de ce nombre. Qu'en est-il pour le meilleur cas ? Et en moyenne ?

Question 3 : discutez des détails pouvant affecter l'efficacité de ces deux algorithmes.

Exercice 2 : multiplication de Strassen

Nous avons vu en cours une méthode de multiplication des polynômes (ou des nombres) plus rapide (asymptotiquement) que la multiplication naïve : c'était l'algorithme de Karatsuba. Une telle méthode est également disponible pour la multiplication de matrices carrées : la méthode de Strassen.

Question 1 : évaluez le nombre de multiplications et d'additions scalaires utilisées pour la multiplication de deux matrices $n \times n$.

Question 2 : supposons que n soit une puissance de 2. En utilisant récursivement la multiplication par blocs de taille $n/2$, donnez une relation de récurrence pour calculer le nombre de multiplications scalaires nécessaires à la multiplication des matrices.

On rappelle que la multiplication par blocs correspond à :

$$\left[\begin{array}{c|c} A_{1,1} & A_{1,2} \\ \hline A_{2,1} & A_{2,2} \end{array}\right] \times \left[\begin{array}{c|c} B_{1,1} & B_{1,2} \\ \hline B_{2,1} & B_{2,2} \end{array}\right] = \left[\begin{array}{c|c} C_{1,1} & C_{1,2} \\ \hline C_{2,1} & C_{2,2} \end{array}\right]$$

où les $X_{i,j}$ sont des matrices de taille $n/2 \times n/2$ et $C_{1,1} = A_{1,1} \times B_{1,1} + A_{1,2} \times B_{2,1}$, ...

Question 3 : on va maintenant essayer d'améliorer les performances (nombre de multiplications scalaires) : on définit les matrices suivantes :

- $M_1 = (A_{1,1} + A_{2,2}) \times (B_{1,1} + B_{2,2})$
- $M_2 = (A_{2,1} + A_{2,2}) \times B_{1,1}$
- $M_3 = A_{1,1} \times (B_{1,2} - B_{2,2})$
- $M_4 = A_{2,2} \times (B_{2,1} - B_{1,1})$
- $M_5 = (A_{1,1} + A_{1,2}) \times B_{2,2}$
- $M_6 = (A_{2,1} - A_{1,1}) \times (B_{1,1} + B_{1,2})$
- $M_7 = (A_{1,2} - A_{2,2}) \times (B_{2,1} + B_{2,2})$

On pose ensuite

- $C_{1,1} = M_1 + M_4 - M_5 + M_7$
- $C_{1,2} = M_3 + M_5$
- $C_{2,1} = M_2 + M_4$
- $C_{2,2} = M_1 - M_2 + M_3 + M6$

Montrez que la matrice C ainsi obtenue est bien le produit de A et B.

Question 4 : donnez une relation de recurrence pour le nombre de multiplications employées par cet algorithme.
Résolvez cette récurrence et concluez.

Question 5 : discutez les problèmes potentiels liés à cet algorithme.