Proiectarea algoritmilor

Lucrare de laborator nr. 8

Paradigma *Divide_et_Impera*

Înmulţirea matricelor pătratice

Cuprins

1 Algoritm clasic 1

2 Algoritm *Divide_et_Impera* **clasic - exemplu** 2

3 Metoda lui *Strassen* 2

4 Sarcini de lucru şi barem de notare 3

1 Algoritm clasic

Considerăm algoritmul clasic de înmulţire a matricelor pătratice:

```
procedure inmultireMatrice(A, B, C, n )
   for i ← 0 to n-1
       for j ← 0 to n-1
           C[i,j] ← 0
           for k ← 0 to n-1
               C[i,j] ← C[i,j] + A[i,k] * B[k,j]
end
```

Mult timp s-a crezut că bariera de $O(n^3)$ nu poate fi depăşită.

Strassen a demonstrat însă că se poate obţine un algoritm mai bun.

Metoda lui *Strassen* foloseşte un algoritm de înmulţire bazat pe descompunerea matricelor în sferturi.

2 Algoritm *Divide_et_Impera* clasic - exemplu

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}\begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix} = \begin{bmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}$$

$$C_{1,1} = A_{1,1}B_{1,1} + A_{1,2}B_{2,1}$$
$$C_{1,2} = A_{1,1}B_{1,2} + A_{1,2}B_{2,2}$$
$$C_{2,1} = A_{2,1}B_{1,1} + A_{2,2}B_{2,1}$$
$$C_{2,2} = A_{2,1}B_{1,2} + A_{2,2}B_{2,2}$$

$$\mathbf{AB} = \begin{bmatrix} 3 & 4 & 1 & 6 \\ 1 & 2 & 5 & 7 \\ 5 & 1 & 2 & 9 \\ 4 & 3 & 5 & 6 \end{bmatrix}\begin{bmatrix} 5 & 6 & 9 & 3 \\ 4 & 5 & 3 & 1 \\ 1 & 1 & 8 & 4 \\ 3 & 1 & 4 & 1 \end{bmatrix}$$

$$A_{1,1} = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix} \quad A_{1,2} = \begin{bmatrix} 1 & 6 \\ 5 & 7 \end{bmatrix} \quad B_{1,1} = \begin{bmatrix} 5 & 6 \\ 4 & 5 \end{bmatrix} \quad B_{1,2} = \begin{bmatrix} 9 & 3 \\ 3 & 1 \end{bmatrix}$$

$$A_{2,1} = \begin{bmatrix} 5 & 1 \\ 4 & 3 \end{bmatrix} \quad A_{2,2} = \begin{bmatrix} 2 & 9 \\ 5 & 6 \end{bmatrix} \quad B_{2,1} = \begin{bmatrix} 1 & 1 \\ 3 & 1 \end{bmatrix} \quad B_{2,2} = \begin{bmatrix} 8 & 4 \\ 4 & 1 \end{bmatrix}$$

Figura 1: Exemplu de înmulţire a matricelor prin descompunerea în sferturi

$T(n) = 8T(n/2) + O(n^2)$
Din teorema complexităţii *Divide_et_Impera* rezultă $T(n) = O(n^3)$!! Nici un progres !

3 Metoda lui *Strassen*

Strassen a utilizat o strategie similară inmulţirii numerelor întregi şi a arătat că pot fi utilizate doar 7 înmulţiri în loc de 8.

Cele 7 înmulţiri sunt următoarele:
$M_1 = (A_{1,2} - A_{2,2})(B_{2,1} + B_{2,2})$
$M_2 = (A_{1,1} + A_{2,2})(B_{1,1} + B_{2,2})$
$M_3 = (A_{1,1} - A_{2,1})(B_{1,1} + B_{1,2})$
$M_4 = (A_{1,1} + A_{1,2})B_{2,2}$
$M_5 = A_{1,1}(B_{1,2} - B_{2,2})$
$M_6 = A_{2,2}(B_{2,1} - B_{1,1})$
$M_7 = (A_{2,1} + A_{2,2})B_{1,1}$

Apoi:
$C_{1,1} = M_1 + M_2 - M_4 + M_6$
$C_{1,2} = M_4 + M_5$
$C_{1,3} = M_6 + M_7$
$C_{1,4} = M_2 - M_3 + M_5 - M_7$

$T(n) = 7T(n/2) + O(n^2)$. Rezultă $T(n) = O(n^{\log 27}) = O(n^{2.81})$!!!

4 Sarcini de lucru şi barem de notare

Sarcini de lucru:

1. Scrieţi un program C/C++ care implementează metoda *Divide_et_Impera* clasică pentru înmulţirea a două matrice pătratice.
2. Scrieţi un program C/C++ care implementează metoda lui Strassen pentru înmulţirea a două matrice pătratice.

Barem de notare:

1. Scrierea pseudocodului algoritmului *Divide_et_Impera* clasic: 3p
2. Implementarea algoritmului *Divide_et_Impera* clasic: 3p
3. Aplicarea metodei lui Strassen: 3p
4. Baza: 1p

Bibliografie

[1] M. A. Weiss, *Data Structures and Algorithm Analysis in C*, The Benjamin/Cummings Publishing Company, Inc., 1992.