COMP 301 Analysis of Algorithms
Instructor: Zafer Aydın
Lab Assignment 6

Introduction

In this lab you will implement Strassen's algorithm for matrix multiplication. Submit your answers to the questions below in a text file (e.g. Word document). Name your file in name_surname.docx format. Submit your solution document and Java codes (only code files not the whole project) as a folder in name_surname format to Canvas.

You can use the code templates in `matrix.java` in this lab.

Problem Statement

Given two matrices A and B each with size $n \times n$ compute $C = A.B$, which is the product of A and B.

Assignment

1. (a) Implement the Strassen's matrix multiplication algorithm given below in Java. The algorithm assumes that the input matrices are $n \times n$, where n is a power of 2. You can use 2D arrays to represent matrices. You can define array C outside of your method and pass it as input to that method. When you modify that array inside your method the changes will be maintained after the method finishes execution (i.e. it can be used as output at the same time this way). You can use the code templates in `matrix.java`.

Strassen's Algorithm

1. Divide the input matrices A and B and output matrix C into $n/2 \times n/2$ submatrices, as in equation (4.9). This step takes $\Theta(1)$ time by index calculation, just as in SQUARE-MATRIX-MULTIPLY-RECURSIVE.
2. Create 10 matrices $S_1, S_2, \ldots, S_{10}$, each of which is $n/2 \times n/2$ and is the sum or difference of two matrices created in step 1. We can create all 10 matrices in $\Theta(n^2)$ time.
3. Using the submatrices created in step 1 and the 10 matrices created in step 2, recursively compute seven matrix products $P_1, P_2, \ldots, P_7$. Each matrix P_i is $n/2 \times n/2$.
4. Compute the desired submatrices $C_{11}, C_{12}, C_{21}, C_{22}$ of the result matrix C by adding and subtracting various combinations of the P_i matrices. We can compute all four submatrices in $\Theta(n^2)$ time.

COMP 301 Analysis of Algorithms
Instructor: Zafer Aydın
Lab Assignment 6

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \quad C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

$S_1 = B_{12} - B_{22}$,	$P_1 = A_{11} \cdot S_1$	$C_{11} = P_5 + P_4 - P_2 + P_6$.
$S_2 = A_{11} + A_{12}$,	$P_2 = S_2 \cdot B_{22}$	$C_{12} = P_1 + P_2$
$S_3 = A_{21} + A_{22}$,	$P_3 = S_3 \cdot B_{11}$	$C_{21} = P_3 + P_4$
$S_4 = B_{21} - B_{11}$,	$P_4 = A_{22} \cdot S_4$	$C_{22} = P_5 + P_1 - P_3 - P_7$
$S_5 = A_{11} + A_{22}$,	$P_5 = S_5 \cdot S_6$	
$S_6 = B_{11} + B_{22}$,	$P_6 = S_7 \cdot S_8$	
$S_7 = A_{12} - A_{22}$,	$P_7 = S_9 \cdot S_{10}$	
$S_8 = B_{21} + B_{22}$,		
$S_9 = A_{11} - A_{21}$,		
$S_{10} = B_{11} + B_{12}$.		

(b) Verify that your method works correctly for $A = \begin{bmatrix} 1 & 3 \\ 7 & 5 \end{bmatrix}$, $B = \begin{bmatrix} 6 & 8 \\ 4 & 2 \end{bmatrix}$, where C should be $\begin{bmatrix} 18 & 14 \\ 62 & 66 \end{bmatrix}$.

(c) Randomly generate A and B such that their sizes are 16 by 16 and elements are random integers from 0 to 99. Compue $C = A.B$ using the method you implemented in part (a). Report the time it takes to compute C.

(d) Repeat part (c) this time for arrays of size 64 by 64. Report the time it takes to compute $C = A.B$. How much did the time increase as compared to part (c)?

(e) Use the same arrays as in part (c) and compute $C = A.B$ using the `square_matrix_multiply` method available in `matrix.java`. Report the time it takes to compute the C matrix. Compare it to the result you found in part (c).

(f) Use the same arrays as in part (d) and compute $C = A.B$ using the `square_matrix_multiply` method available in `matrix.java`. Report the time it takes to compute the C matrix. Compare it to the result you found in part (d).