COURSE HANDOUT

Course Code	ACSC13
Course Name	Design and Analysis of Algorithms
Class / Semester	IV SEM
Section	A-SECTION
Name of the Department	CSE-CYBER SECURITY
Employee ID	IARE11023
Employee Name	Dr K RAJENDRA PRASAD
Topic Covered	Strassen's matrix multiplication
Course Outcome/s	Use the strassen's concept for faster computation of matric multiplication.
Handout Number	19
Date	21 April, 2023

Content about topic covered: Strassen's Matrix Multiplication

Strassen's Matric Multiplication

Let A and B be two n x n matrices. The product matrix C = AB is also an n x n matrix whose $(i,j)^{th}$ element is formed by taking the elements in the i^{th} row of A and j^{th} column of B and multiplying them to get C(i,j).

$$C(i,j) = \sum_{1 \le k \le n} A(i,k) * B(k,j)$$

To compute C(i,j) using this formula we need n multiplications. Since the matrix C has n^2 elements, the time for the resulting matrix multiplication is $O(n^3)$.

The divide and conquer strategy suggests another way to compute the product of two n x n matrices. For simplicity, Assume $n = 2^k$. In case n is not power of 2, then enough rows and columns of 0's can be added to both A and B, so that the resulting dimensions are a power of 2.

Imagine that A and B are each partitioned in to 4 square sub matrices, each matrix having dimensions $\frac{n}{2}$ x $\frac{n}{2}$. Then the product AB is computed as

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} X \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

Then

$$\begin{aligned} C_{11} &= A_{11}\ B_{11} + A_{12}\ B_{21} \\ C_{12} &= A_{11}\ B_{12} + A_{12}\ B_{22} \\ C_{21} &= A_{21}\ B_{11} + A_{22}\ B_{21} \\ C_{22} &= A_{21}\ B_{12} + A_{22}\ B_{22} \end{aligned}$$

To compute AB, we need to perform 8 multiplications of $\frac{n}{2}$ x $\frac{n}{2}$ matrices and 4 additions of $\frac{n}{2}$ x $\frac{n}{2}$ matrices. Since two $\frac{n}{2}$ x $\frac{n}{2}$ matrices can be added in time cn^2 for some constant c, the overall computing time T(n) is given by the recurrence relation

$$T(n) = \begin{cases} b & n \leq 2 \\ 8\,T\left(\frac{n}{2}\right) + cn^2 & n > 2 \end{cases}$$

$$\begin{aligned} T(n) &= 8\,T\left(\frac{n}{2}\right) + cn^2 \\ &= 8\left[8\,T\left(\frac{n}{4}\right) + \frac{cn^2}{4}\right] + cn^2 \\ &= 64\,T\left(\frac{n}{4}\right) + 3\,cn^2 \\ &= \left(2^k\right)^3 + \frac{n}{2^k} + \left(2^k - 1\right)cn^2 \\ &= bn^3 + (n-1)cn^2 \\ &= O(n^3) \end{aligned}$$

Hence no improvement over the conventional method has been made. Matrix multiplications are more expensive than matrix additions.

Strassen has discovered a way to compute C_{ij} using only 7 multiplications and 18 additions or subtractions. His method involves first computing the seven $\frac{n}{2}$ x $\frac{n}{2}$ matrices P, Q, R, S, T, U and V. Then C_{ij}'s are computed using the matrices P, Q, R, S, T, U and V.

$$\begin{aligned} P &= (A_{11} + A_{22})\,(B_{11} + B_{22}) \\ Q &= (A_{21} + A_{22})\,B_{11} \\ R &= A_{11}\,(B_{12} - B_{22}) \\ S &= A_{22}\,(B_{21} - B_{11}) \\ T &= (A_{11} + A_{12})\,B_{22} \\ U &= (A_{21} - A_{11})\,(B_{11} + B_{12}) \\ T &= (A_{12} - A_{22})\,(B_{21} + B_{22}) \end{aligned}$$

Then

$$\begin{aligned} C_{11} &= P + S - T + V \\ C_{12} &= R + T \\ C_{21} &= Q + S \\ C_{22} &= P + R - Q + U \end{aligned}$$

The resulting recurrence relation for T(n) is

$$T(n) = \begin{cases} b & n \leq 2 \\ 7T\left(\frac{n}{2}\right) + an^2 & n > 2 \end{cases}$$

$$
\begin{aligned}
T(n) &= 7\,T\left(\frac{n}{2}\right) + an^2 \\
&= 7\left[7T\left(\frac{n}{4}\right) + a\left(\frac{n}{2}\right)^2\right] + an^2 \\
&= 49\,T\left(\frac{n}{4}\right) + \frac{7}{4}an^2 + an^2 \\
&= an^2\left[1 + \frac{7}{4} + \left(\frac{7}{4}\right)^2 + \cdots + \left(\frac{7}{4}\right)^{2k-1}\right] + 7^k\,T(1) \\
&\leq cn^2\left(\frac{7}{4}\right)^{\log n} + b.7^{log\,n} \\
&\leq cn^{\log 4 + \log 7 - \log 4} + b.n^{log\,7} \\
&= O(n^{log\,7}) \\
&= O(n^{2.81})
\end{aligned}
$$