Design & Analysis of Algorithms

Matrix Multiplication

1. Traditional method & its Time complexity
2. Divide & Conquer method , its Time complexity
3. Strassen's Matrix Multiplication

Traditional Method (school level)

j=1 j=2 j=3

A 3x3

a11	**a12**	**a13**
a21	a22	a23
a31	a32	a33

i=1, i=2, i=3

B 3x3

b11	**b12**	**b13**
b21	b22	b23
b31	b32	b33

MA → Colum = MB rows

⇒ a[i][k] * b[k][j]

i j

$c_{11} = a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31}$

i

$c_{12} = a_{11} * b_{12} + a_{12} * b_{22} + a_{13} * b_{32}$

⋮

$c_{33} = a_{31} * b_{13} + a_{32} * b_{23} + a_{33} * b_{33}$

C_{3x3}

c11	**c12**	**c13**
c21	c22	c23
c31	c32	c33

```
for i:=1 to 3 (n) step 1 do
{
  for j:1 to 3 step 1 do
  { c[i][j]=0
    for k:1 to 3 step 1 do
    {
      c[i][j] = c[i,j] + A[i][k] * B[k][j];
} } }
```

Traditional Method Algorithm & Time Complexity

$A_{n \times n}$

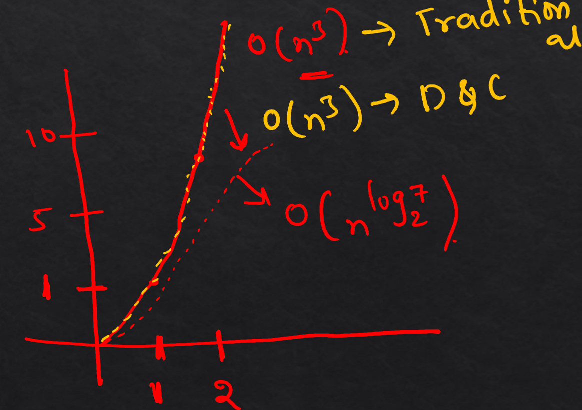
a11	……	a1n
.		
an1		ann

$B_{n \times n}$

b11	….	b1n
….	….	…
bn1	….	bnn

```
Algorithm MatrixMultiplicationGM( A, B, n)
{
    for i:=1 to n step 1 do            → n
    {                                       ×
        for j:=1 to n step 1 do        → n
        {  c[i,j]:=0;                       ×
           for k:=1 to n step 1 do     → n
           {
               c[i,j] = c[i,j] + A[i][k] * B[k][j];
    }} }
```

$T(n) = O(n^3)$ 2×2×2



matrix multipl

$P = \underline{A} \times \underline{B}$

Divide & Conquer Method

$P = A_{2\times 2} * B_{2\times 2}$

$A_{1\times 1}$

a11

$B_{1\times 1}$

b11

small

n=2

small

$A_{2\times 2}$

a11 *	**a12**
a21	a22

$B_{2\times 2}$

b11 *	**b12**
b21	b22

n=1:

$c_{11} = a_{11} * b_{11}$ → 1 unit

$[c_{11}] = [a_{11} * b_{11}]$ ✓

'c' time

$c_{11} = a_{11} * b_{11} + a_{12} * b_{21}$

$c_{12} = a_{11} * b_{12} + a_{12} * b_{22}$

$c_{21} = a_{21} * b_{11} + a_{22} * b_{21}$

$c_{22} = a_{21} * b_{12} + a_{22} * b_{22}$

4×2 = 8 units

'c' time

Divide & Conquer Method

$n=1$ ✗
$n=2$ ✗
$n>2$ ✓

$n \to 2\left(\frac{n}{2}\right)$ ✓

$\frac{n}{2} \times \frac{n}{2}$

$a_{11} \to$ element

$A_{11} \to$ sub matrix

$A_{n \times n}$

A_{11} A_{12}

a11	…….	a1n
.		
an1		ann

A_{21} A_{22}

B_{11} B_{12}

b11	….	b1n
….	….	…
bn1	….	bnn

B_{21} B_{22}

Algorithm m.m.D.C($\underline{A}$, $\underline{B}$, $\underline{n}$)
{
if ($n==1$)
{ return $a_{11} * b_{11}$; } $\frac{1 \text{ unit}}{c}$
}
else if ($n==2$)
{ $C_{11} =$
$C_{12} =$
$C_{21} =$
$C_{22} =$ } $\frac{8 \text{ units}}{c}$

$O(n^2)$

else {
~~mid = $\frac{n}{2}$~~;

m.m.D.C(A_{11}, B_{11}, $n/2$) + mmDC(A_{12}, b_{21}, $n/2$)

mmDC(A_{11}, B_{12}, $\frac{n}{2}$) + mmDC(A_{12}, B_{22}, $n/2$); }

mmDC(A_{21}, B_{11}, $\frac{n}{2}$) + mmDC(A_{22}, B_{21}, $\frac{n}{2}$);

}}. mmDC(A_{21}, B_{12}, $\frac{n}{2}$) + mmDC(A_{22}, B_{22}, $n/2$);

C_{12}

C_{21}

C_{22}

Time Complexity D & C :-

$$T(n) = \begin{cases} \underline{c} & \underline{n=1} \\ \underline{c} & \underline{n=2} \\ 8\,T\left(\frac{n}{2}\right) + n^2 & n > 2 \end{cases}$$

8 multiplication
n/2 matrices

addition

$T(n) = a^k . T\left(\frac{n}{b}\right) + f(n)$

$T(n) = 8\,T\left(\frac{n}{2}\right) + n^2$

$T(n) = O\left(n^{\log_2^8}\right)$

$T(n) = O\left(n^{\log_2 2^3}\right) \Rightarrow O\left(n^{3\log_2^2}\right) \Rightarrow O\left(n^{3\times 1}\right) = \underline{O\left(n^3\right)}$

Strassen's Matrix Multiplication

For two $n \times n$ matrices A and B, divided into four $n/2 \times n/2$ submatrices:

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

Strassen's algorithm computes the following **7 intermediate products:**

1. $P = (A_{11} + A_{22})(B_{11} + B_{22})$
2. $Q = (A_{21} + A_{22})B_{11}$
3. $R = A_{11}(B_{12} - B_{22})$
4. $S = A_{22}(B_{21} - B_{11})$
5. $T = (A_{11} + A_{12})B_{22}$
6. $U = (A_{21} - A_{11})(B_{11} + B_{12})$
7. $V = (A_{12} - A_{22})(B_{21} + B_{22})$

$T(n) = 7 \cdot T(\frac{n}{2}) + f(n)$

Combining the Results

The resulting matrix C is computed using these intermediate products:

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

Where:

$$C_{11} = P + S - T + V$$
$$C_{12} = R + T$$
$$C_{21} = Q + S$$
$$C_{22} = P - Q + R + U$$

$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$

$A_{11} = [1]$

$P = (1+4)(1+2) \Rightarrow 15$

$Q = (7)\ 1 \rightarrow 7$

$R = (1)(1-2) \Rightarrow -1$

$S = (4)(2-1) = 4$

$T = (1+2)(2) = 6$

$U = (3-1)(1+1) = 4$

$V = (2-4)(2+2) = -8$

$C_{11} = 15 + 4 - 6 + (-8) = 5$

$C_{12} = -1 + 6 = 5$

$C_{21} = 7 + 4 = 11$

$C_{22} = 15 - 7 + (-1) + 4 = 11$

$\begin{bmatrix} 5 & 5 \\ 11 & 11 \end{bmatrix}$

Traditional method

$\begin{bmatrix} 1\times1+2\times2 & 1\times1+2\times2 \\ 3\times1+4\times2 & 3\times1+4\times2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1+4 & 1+4 \\ 3+8 & 3+8 \end{bmatrix} \Rightarrow \begin{bmatrix} 5 & 5 \\ 11 & 11 \end{bmatrix}$

Strassen's $\leftarrow$ $T(n) = 7 \cdot T\left(\frac{n}{2}\right) + f(n)$

$= O\left(n^{\log_2 7}\right) < O\left(n^{\log_2 8}\right)$

✓