20MCA 203

Design & Analysis of Algorithms

ASSIGNMENT - 01

Submitted By:

Name :- Riya Saji

RollNo : 24

RMCA 2022-2024

Matrix Multiplication

Brute force Algorithm (Naive method)

Algorithm

```
Alg (a, b, c, n)
{
 for (i=0; i<n; i++)
  {
   for (j=0; j<n; j++)
    {
     c[i][j] = 0
     for (k=0; k<n; k++)
      {
       c[i][j] = c[i][j] + a[i][k] * b[k][j]
      }
    }
  }
  return c
}
```

Time Complexity

$C_{11} = a_{11} b_{11} + a_{12} b_{21}$

$C_{12} = a_{11} b_{12} + a_{12} b_{22}$

$C_{21} = a_{21} b_{11} + a_{22} b_{21}$

$C_{22} = a_{21} b_{12} + a_{22} b_{22}$

$$C_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

$$T(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} c$$

$$= n^3 c$$

$$= \Theta(n^3)$$

$$T(n) = (n+1) + n(n+1) + n \times n(n+1) + n \times n \times n + 2$$

$$= (n+1) + n^2 + n + n^3 + n^2 + n^3 + 2$$

$$= 2n^3 + 2n^2 + 2n + 3$$

$$T(n) = O(n^3)$$

Divide & Conquer Method

The matrix is divided into 4 submatrices in divide and conquer method with dimension $(n/2 \times n/2)$

⇒ Matrix a and b are partitioned into A square submatrix having dimension $(n/2 \times n/2)$

i.e,

$$a = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times b = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = c = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

$$C_{11} = a_{11} b_{11} + a_{12} b_{21}$$

$$C_{12} = a_{11} b_{12} + a_{12} b_{22}$$

$$C_{21} = a_{21} b_{11} + a_{22} b_{21}$$

$$C_{22} = a_{21} b_{12} + a_{22} b_{22}$$

Algorithm

multM ~~Alg~~ (a, b, n)

{ if ($n \leq 2$)

{

$C_{11} = a_{11} b_{11} + a_{12} b_{21}$

$C_{12} = a_{11} b_{12} + a_{12} b_{22}$

$C_{21} = a_{21} b_{11} + a_{22} b_{21}$

$C_{22} = a_{21} b_{12} + a_{22} b_{22}$

}

else { // n > 2, mid- n/2

multiM ($a_{11}, b_{11}, n/2$) + multiM ($a_{12}, b_{21}, n/2$)

multiM ($a_{11}, b_{12}, n/2$) + multiM ($a_{12}, b_{22}, n/2$)

multiM ($a_{21}, b_{11}, n/2$) + multiM ($a_{22}, b_{21}, n/2$)

multiM ($a_{21}, b_{12}, n/2$) + multiM ($a_{22}, b_{22}, n/2$)

}

}

Time Complexity

Here occur 8 matrix multiplication i.e $8T(n/2)$ & 4 addition the time complexity of matrix addition is n^2

Thus,

$$T(n) = \begin{bmatrix} 1 & n \leq 2 \\ 8T(n/2) + n^2 & n > 2 \end{bmatrix}$$

$$T(n) = 8T(n/2) + n^2 \quad -(1)$$

taking n as $n/2$

$$T(n/2) = 8T(n/4) + n^2/4 \quad -(2)$$

putting eqn (2) in (1)

$$T(n) = 8[8T(n/4) + n^2/4]n^2$$

$$= 64T(n/4) + 2n^2 + n^2$$

$$= 64T(n/4) + 3n^2 \quad -(3)$$

taking n as $n/4$ (4)

$$T(n/4) = 64\left[8T(n/8) + \frac{n^2}{16}\right] + 3n^2$$

$$= 512T(n/8) + 4n^2 + 3n^2$$

$$= 512T(n/8) + 7n^2$$

Thus, the equation can be written as

For k times

$$T(n) = k^3 T(n/k) + (k-1)n^2$$

Taking $n/k = 1$

$n = k$

$$T(n) = n^3 T(1) + (n-1)n^2$$
$$= n^3 \times 1 + n^3 - n^2$$
$$= 2n^3 - n^2$$

Time Complexity $= O(n^3)$

Strassen Matrix Multiplication

Strassen's matrix multiplication is the divide of conquer approach to solve the matrix multiplication problems. He discovered a way to compute ab=c of matrix using multiplication and the equation are :

$$P = (a_{11} + a_{22})(b_{11} + b_{22})$$
$$Q = b_{11}(a_{21} + a_{22})$$
$$R = a_{11}(b_{12} - b_{22})$$
$$S = a_{22}(b_{21} - b_{11})$$
$$T = b_{22}(a_{11} + a_{12})$$
$$U = (a_{21} - a_{11})(b_{11} + b_{12})$$
$$V = (a_{12} - a_{22})(b_{22} + b_{21})$$

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

Time Complexity

$$T(n) = \begin{cases} 1 & n \le 2 \\ 7T(n/2) + n^2 & n > 2 \end{cases}$$

$$T(n) = 7T(n/2) + n^2$$

$$= O(n^{\log 7})$$

Example

Matrix A : $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

Matrix B : $\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

Step1: Divide the matrices into submatrices

Matrix A can be divided into four submatrices:

$A11 = [1]$ $A12 = [2]$

$A21 = [3]$ $A22 = [4]$

Matrix B can be divided into four submatrices

$B11 = [5]$ $B12 = [6]$

$B21 = [7]$ $B22 = [8]$

Step 2: compute seven products recursively.

$P1 = A11 * (B12 - B22)$

$P2 = (A11 * A12) * B22$

$P3 = (A21 * A22) * B11$

$P4 = A22 * (B21 - B11)$

$P5 = (A11 + A22) * (B11 + B22)$

$P6 = (A12 - A22) * (B21 + B22)$

$P7 = (A11 - A21) * (B11 + B12)$

Step3 : Compute the resulting submatrices of final product matrix C

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

$C_{11} = P_5 + P_4 - P_2 + P_6 = 19$

$C_{12} = P1 + P2 = 22$

$C_{21} = P3 + P4 = 43$

$C_{22} = P5 + P1 - P3 - P7 = 50$

$$C = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

Strassen's Algorithm is more effecient for large matrices, as the number of recursive steps increases with the size of the matrices.