Strassen's Matrix Multiplication

Basic Matrix Multiplication

```
void matrix_mult (){
  for (i = 1; i <= N; i++) {
      for (j = 1; j <= N; j++) {
      for(k=1;k<=N;k++){
             compute C_{i,j};
                              }
  }}
```

algorithm

Time analysis

$$C_{i,j} = \sum_{k=1}^{N} a_{i,k} b_{k,j}$$

$$\text{Thus } T(N) = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} c = cN^3 = O(N^3)$$

Basic Matrix Multiplication

Suppose we want to multiply two matrices of size N x N: for example A x $B = C$.

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

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

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

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

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

2x2 matrix multiplication can be accomplished in 8 multiplication.($\mathbf{2^{\log_2 8} = 2^3}$)

Divide and Conquer Matrix Multiplication

- In order to compute AB using the above decomposition, we need to perform 8 multiplications of n/2 X n/2 matrices and 4 additions of n/2 matrices.
- Since two n/2Xn/2 may be added in time Cn^2 for some constant C, the overall computing time, T(n) of the resulting divide and conquer algorithm is given by the recurrence as:

$$T(n) = \begin{cases} b & n \le 2 \\ 8T\left(\dfrac{n}{2}\right) + \Theta(n^2) & n > 2 \end{cases}$$

$$\textit{Compare}\ \text{above recurrence with } T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

$$a = 8, b = 2\ f(n) = n^2$$

$$n^{\log_b a} = n^{\log_2 8} = n^3$$

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

Strassens’s Matrix Multiplication

- Strassen showed that 2x2 matrix multiplication can be accomplished in 7 multiplication and 18 additions or subtractions. .$(2^{\log_2 7} = 2^{2.807})$
- This reduce can be done by Divide and Conquer Approach.

Strassen Algorithm

```
void matmul(int *A, int *B, int *R, int n) {
 if (n == 1) {
    (*R) += (*A) * (*B);
 } else {
    matmul(A, B, R, n/4);
    matmul(A, B+(n/4), R+(n/4), n/4);
    matmul(A+2*(n/4), B, R+2*(n/4), n/4);
    matmul(A+2*(n/4), B+(n/4), R+3*(n/4), n/4);
    matmul(A+(n/4), B+2*(n/4), R, n/4);
    matmul(A+(n/4), B+3*(n/4), R+(n/4), n/4);
    matmul(A+3*(n/4), B+2*(n/4), R+2*(n/4), n/4);
    matmul(A+3*(n/4), B+3*(n/4), R+3*(n/4), n/4);
 }
```

Divide matrices in sub-matrices and recursively multiply sub-matrices

Strassens's Matrix Multiplication

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

$P_1 = (A_{11} + A_{22})(B_{11} + B_{22})$
$P_2 = (A_{21} + A_{22}) * B_{11}$
$P_3 = A_{11} * (B_{12} - B_{22})$
$P_4 = A_{22} * (B_{21} - B_{11})$
$P_5 = (A_{11} + A_{12}) * B_{22}$
$P_6 = (A_{21} - A_{11}) * (B_{11} + B_{12})$
$P_7 = (A_{12} - A_{22}) * (B_{21} + B_{22})$

$C_{11} = P_1 + P_4 - P_5 + P_7$
$C_{12} = P_3 + P_5$
$C_{21} = P_2 + P_4$
$C_{22} = P_1 + P_3 - P_2 + P_6$

Comparison

$$
\begin{aligned}
C_{11} &= P_1 + P_4 - P_5 + P_7 \\
&= (A_{11} + A_{22})(B_{11} + B_{22}) + A_{22} * (B_{21} - B_{11}) - (A_{11} + A_{12}) * B_{22} + \\
&\quad (A_{12} - A_{22}) * (B_{21} + B_{22}) \\
&= A_{11} B_{11} + A_{11} B_{22} + A_{22} B_{11} + A_{22} B_{22} + A_{22} B_{21} - A_{22} B_{11} - \\
&\quad A_{11} B_{22} - A_{12} B_{22} + A_{12} B_{21} + A_{12} B_{22} - A_{22} B_{21} - A_{22} B_{22} \\
&= A_{11} B_{11} + A_{12} B_{21}
\end{aligned}
$$

Analysis

$$T(n) = \begin{cases} b & n \le 2 \\ 7T\left(\dfrac{n}{2}\right) + \Theta(n^2) & n > 2 \end{cases}$$

$$\mathit{Compare}\ \text{above recurrence with } T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

$$a = 7, b = 2\ f(n) = n^2$$

$$n^{\log_b a} = n^{\log_2 7} = n^{2.81}$$

$$T(n) = O(n^{2.81})$$

Time Analysis

$$T(1) = 1 \qquad (\text{assume } N = 2^k)$$

$$T(N) = 7T(N/2)$$

$$T(N) = 7^k T(N/2^k) = 7^k$$

$$T(N) = 7^{\log N} = N^{\log 7} = N^{2.81}$$