Divide and Conquer

Introduction to Divide-and-Conquer

- Divide-and conquer is a general algorithm design paradigm:
  - Divide: divide the input data S in two or more disjoint subsets S_1, S_2, …
  - Recur: solve the subproblems recursively
  - Conquer: combine the solutions for S_1, S_2, …, into a solution for S
- The base case for the recursion are subproblems of constant size
- Analysis can be done using **recurrence equations**

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 Multiply

$$A \times B = R$$

A_0	A_1
A_2	A_3

$\times$

B_0	B_1
B_2	B_3

$=$

$A_0 \times B_0 + A_1 \times B_2$	$A_0 \times B_1 + A_1 \times B_3$
$A_2 \times B_0 + A_3 \times B_2$	$A_2 \times B_1 + A_3 \times B_3$

- Divide matrices into sub-matrices: A_0 , A_1, A_2 etc
- Use blocked matrix multiply equations
- Recursively multiply sub-matrices

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.

Divide and Conquer Matrix Multiply

$$A \times B = R$$

$$\boxed{a_0} \times \boxed{b_0} = \boxed{a_0 \times b_0}$$

- Terminate recursion with a simple base case

Strassens's Matrix Multiplication

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

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

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

Application

- Top down parser
- Basic Fourier transform
- Sorting
- Multiplying Larger
- Branch & Bound

Scope of Research

Solution to stack overflow

Assignment

Q.1)What is Divide & conquer method?
Q.2)Explain Strassen's matrix multiplication method with an example.
Q.3)How to find analysis of problem i.e. using Divide & conquer method.