Design and Analysis of Algorithms

Lab - 3

Divide and Conquer

A divide and conquer algorithm is a strategy of solving a large problem by breaking the problem into smaller sub-problems, solving the sub-problems, and combining them to get the desired output.

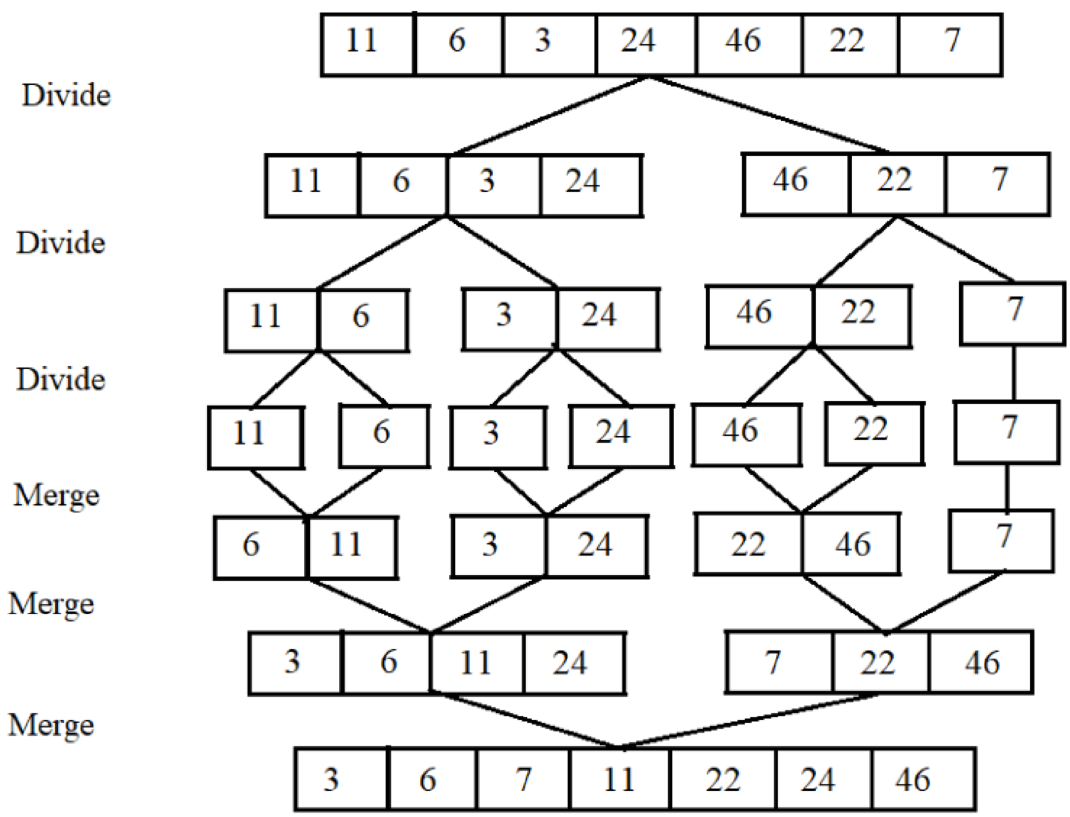
A. Write a C/C++ program for the implementation of Merge Sort.
B. Write a C/C++ program for the Matrix Multiplication using Strassen's algorithm for square matrices of order n,

Do the run time analysis and time complexity analysis with the different values of input size. Maintain the tabular data (n, execution time) and plot it graphically using data plotting tools.

Suggestion:

Merge sort is defined as a sorting algorithm that works by dividing an array into smaller subarrays, sorting each subarray, and then merging the sorted subarrays back together to form the final sorted array.

For example:



Strassen's algorithm is recursively used to divide the matrix A and B of size n x n into 4 sub-matrices of size n/2 x n/2 and compute the corresponding Matrix C.

$$A = \begin{array}{cc} A_{11} & A_{12} \\ \multicolumn{2}{c}{\left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{24} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{array}\right]} \\ A_{21} & A_{22} \end{array} \qquad B = \begin{array}{cc} B_{11} & B_{12} \\ \multicolumn{2}{c}{\left[\begin{array}{cc|cc} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{24} & b_{23} & b_{24} \\ \hline b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{array}\right]} \\ B_{21} & B_{22} \end{array}$$

It requires 8 submatrices multiplication,

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

but it can be reduced into 7 multiplication as

$$\begin{aligned} M_1 &= (A_{11} + A_{22})\,(B_{11} + B_{22}) \\ M_2 &= (A_{21} + A_{22})B_{11} \\ M_3 &= A_{11}(B_{12} - B_{22}) \\ M_4 &= A_{22}(B_{21} - B_{11}) \\ M_5 &= (A_{11} + A_{12})B_{22} \\ M_6 &= (A_{21} - A_{11})(B_{11} + B_{12}) \\ M_7 &= (A_{12} - A_{22})(B_{21} + B_{22}) \end{aligned}$$

$$\begin{aligned} C_{11} &= M_1 + M_4 - M_5 + M_7 \\ C_{12} &= M_3 + M_5 \\ C_{21} &= M_2 + M_4 \\ C_{22} &= M_1 - M_2 + M3 + M_6 \end{aligned}$$

And that results in less time complexity.