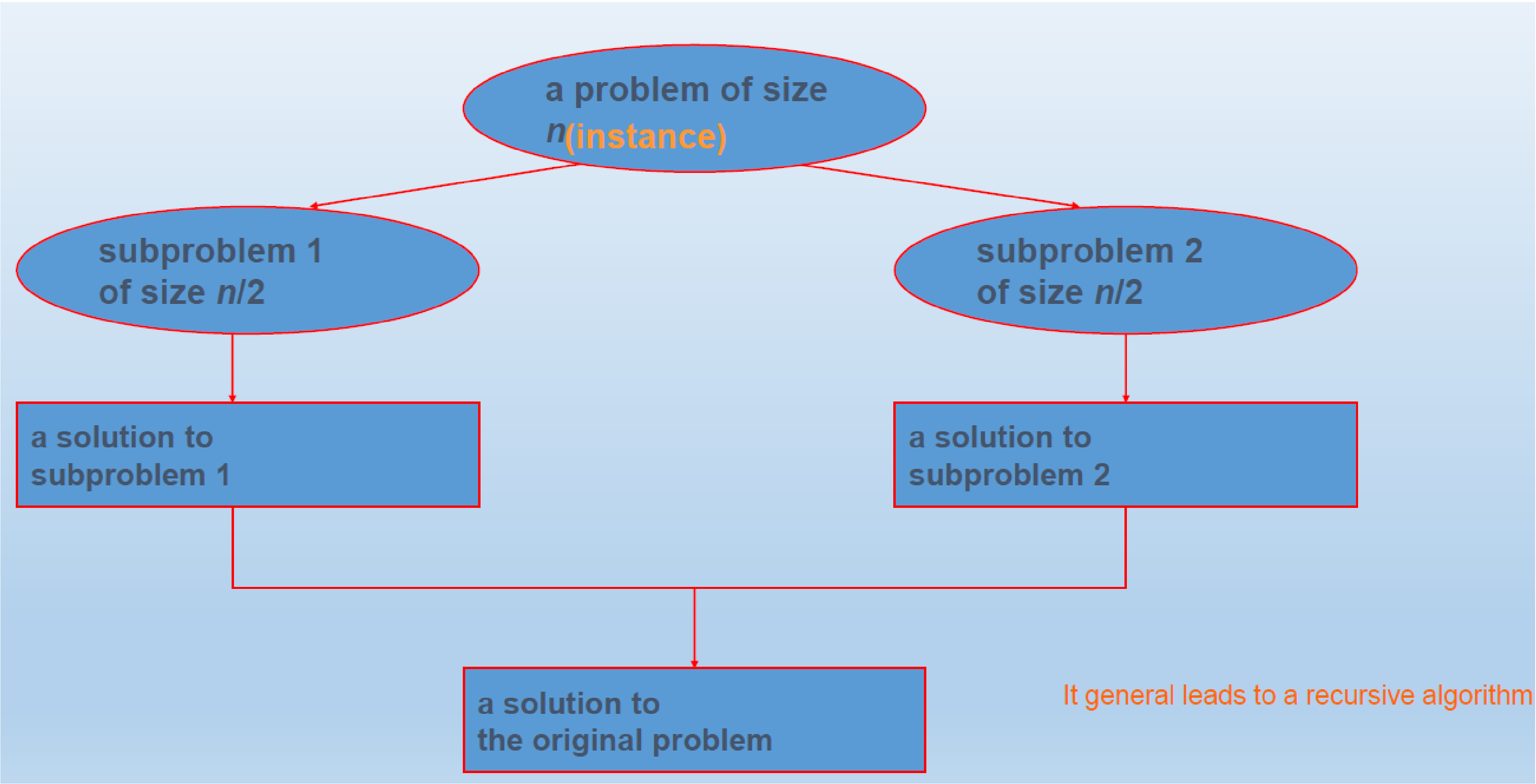
Design Technique: Divide and Conquer

Divide-and-Conquer : Algorithm Design Strategy

The most-well known algorithm design strategy:

- Divide instance of problem into two or more smaller instances
- Solve smaller instances recursively
- Obtain solution to original (larger) instance by combining the solutions

Divide-and-Conquer : Algorithm Design Strategy



Divide-and-Conquer : Approach

```
DAC(P)
{

if (small (P))
// P is small, and solution is obvious
return solution (P);

k= divide(P); // divide the problem into k sub problems
return combine ( DAC(P1), DAC(P2), DAC(P3)..
DAC(PK))

}
```

Divide-and-Conquer : Examples

- Sorting: mergesort and quicksort
- Binary search
- Binary tree traversals
- Matrix multiplication: Strassen's algorithm
- And many more

Matrix Multiplication

The time complexity of Naïve approach of matrix multiplication is $O(N^3)$ because three loops of size n is required to multiply two matrices. The naïve matrix multiplication algorithm is given below-

Algorithm: Matrix-Multiplication (X, Y, Z)

```
for i = 1 to n do
for j = 1 to n do
Z[i,j] := 0
for k = 1 to n
do Z[i,j] := Z[i,j] + X[i,k] × Y[k,j]
```

Can we design a divide and conquer (recursive) approach to multiply two matrices? If yes, what will be the time complexity for the recursive approach?

Matrix Multiplication : Recursive

In divide and conquer based approach, main problem is divided into smaller problems until we get a problem that can be solved easily. For the matrix multiplication, the smaller problem size will be 2.

In some cases, the matrix size may not be in the multiplication of 2. In such scenario, some auxiliary rows and column with value 0 is added in the matrix.

For any matrix of size 2x2 matrix, the matrix multiplication is performed as follows-

C11= a*e + b*g

C12= a*f + b*h

C21= c*e + d*g

C22= c*f + d*h

$$\underbrace{\begin{bmatrix} a & b \\ c & d \end{bmatrix}}_{A} \times \underbrace{\begin{bmatrix} e & f \\ g & h \end{bmatrix}}_{B} = \underbrace{\begin{array}{cc} C11 & C12 \\ \multicolumn{2}{c}{\begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}} \\ C21 & C22 \end{array}}_{C}$$

Matrix Multiplication : Recursive

Suppose we have two matrices A and B of size 4x4 and we want to compute Ax B.

First divide each matrix into 2 matrices of size 2x2 as given below.

$$A=\left(\begin{array}{cc|cc} a11 & a12 & a13 & a14 \\ \multicolumn{2}{c|}{A11} & \multicolumn{2}{c}{A12} \\ a21 & a22 & a23 & a24 \\ \hline a31 & a32 & a33 & a34 \\ \multicolumn{2}{c|}{A21} & \multicolumn{2}{c}{A22} \\ a41 & a42 & a43 & a44 \end{array}\right) \qquad B=\left(\begin{array}{cc|cc} b11 & b12 & b13 & b14 \\ \multicolumn{2}{c|}{B11} & \multicolumn{2}{c}{B12} \\ b21 & b22 & b23 & b24 \\ \hline b31 & b32 & b33 & b34 \\ \multicolumn{2}{c|}{B21} & \multicolumn{2}{c}{B22} \\ b41 & b42 & b43 & b44 \end{array}\right)$$

Now, there will 4 matrices of size 2x2 for *matrix A* and for *matrix B*. These matrices are considered as *A11, A12, A21, A22* for A and *B11, B12, B21, B22* for B.

Strassen's Matrix Multiplication

Treat matrix *A11, A12, A21, A22* as an element of 2x2 matrix A and *B11, B12, B21, B22* as an element of 2x2 matrix B.

These 2 matrices of A and B can be easily multiplied using discussed formula.

$$A=\left(\begin{array}{cc|cc} a11 & a12 & a13 & a14 \\ \multicolumn{2}{c|}{A11} & \multicolumn{2}{c}{A12} \\ a21 & a22 & a23 & a24 \\ \hline a31 & a32 & a33 & a34 \\ \multicolumn{2}{c|}{A21} & \multicolumn{2}{c}{A22} \\ a41 & a42 & a43 & a44 \end{array}\right) \qquad B=\left(\begin{array}{cc|cc} b11 & b12 & b13 & b14 \\ \multicolumn{2}{c|}{B11} & \multicolumn{2}{c}{B12} \\ b21 & b22 & b23 & b24 \\ \hline b31 & b32 & b33 & b34 \\ \multicolumn{2}{c|}{B21} & \multicolumn{2}{c}{B22} \\ b41 & b42 & b43 & b44 \end{array}\right)$$

C11= A11*B11 + A12*B21

C12= A11*B12 + A12*B22

C21= A21*B11 + A22*B21

C22= A21*B12 + A22*B22

$$\begin{pmatrix} C11 & C12 \\ C21 & C22 \end{pmatrix}$$

Divide and Conquer Algorithm of multiplication

```
Function MM( A, B, n)
{
if(n<=2)
{
    C11= a11*b11 + a12*b21
    C12= a11*b12 + a12*b22
    C21= a21*b11 + a22*b21
    C22= a21*b12 + a22*b22
 }
else{
mid =n/2
MM( A11, B11, n/2)+ MM( A12, B21, n/2)
MM( A11, B12, n/2)+ MM( A12, B22, n/2)
MM( A21, B11, n/2)+ MM( A22, B21, n/2)
MM( A21, B12, n/2)+ MM( A22, B22, n/2)

}
```

```
C11= A11*B11 + A12*B21
C12= A11*B12 + A12*B22
C21= A21*B11 + A22*B21
C22= A21*B12 + A22*B22
```

A= $\begin{pmatrix} A11 & A12 \\ A21 & A22 \end{pmatrix}$ **B=** $\begin{pmatrix} B11 & B12 \\ B21 & B22 \end{pmatrix}$

Analysis of Divide and Conquer Algorithm of multiplication

```
Function MM( A, B, n)          ---------------- T(n)
{
if(n<=2)
{
    C11= a11*b11 + a12*b21
    C12= a11*b12 + a12*b22     ---------------- Constant
    C21= a21*b11 + a22*b21
    C22= a21*b12 + a22*b22
 }
else{
mid =n/2
MM( A11, B11, n/2)+ MM( A12, B21, n/2)
MM( A11, B12, n/2)+ MM( A12, B22, n/2) ---------  8T(n/2)+n^2
MM( A21, B11, n/2)+ MM( A22, B21, n/2)
MM( A21, B12, n/2)+ MM( A22, B22, n/2)
}}
```

Recurrence relation

$$T(n) = \begin{cases} 8T(n/2) + n^2 & \text{if } n>2 \\ 1 & n<=2 \end{cases}$$

Analysis of Strassen's Matrix Multiplication Algo

Recurrence relation

$$T(n) = \begin{cases} 8T(n/2) + n^2 & \text{if } n>2 \\ 1 & n<=2 \end{cases}$$

Case 1: if $\log_b a > k$, then $T(n)=O(n^{\log_b a})$

Use master theorem
a=8, b =2, f(n)= n^2 so, n=2
Find $\log_b a$ i.e., 3 which is greater than k=2

T(n)= O$(n^{\log_b a}) = O(n^3)$

So, can we improve the time complexity?
Yes, we can improve time complexity a bit by using Stassen's matrix multiplication formula.

Strassen's Matrix Multiplication

Strassen's Matrix Multiplication 7 formulas-

i. P= (A11+A12) *(B11+B22)

ii. Q= (A21+A22)*B11

iii. R= A11*(B12-B22)

iv. S= A22*(B21-B11)

v. T=(A11+A12)*B22

vi. U=(A21-A11)*(B11+B12)

vii. V=(A12-A22)*(B21+B22)

Matrix C after multiplication

$$\begin{pmatrix} C11= P+S-T+V & C12= R+T \\ C21= Q+S & C22=P+R-Q+U \end{pmatrix}$$

Strassen’s Matrix Multiplication

Observation about Strassen’s Matrix Multiplication

The standard matrix multiplication requires 8 matrix multiplication while Strassen’s Matrix Multiplication need 7 matrix multiplication but more matrix addition.

The matrix multiplication require more effort than matrix addition and subtraction. As, Strassen’s Matrix Multiplication need 7 matrix multiplication it will need less effort than standard matrix multiplication approach.

Analysis of Divide and Conquer Algorithm of multiplication

```
Function MM( A, B, n)  ------------------  T(n)
{if(n<=2){
   Use formula for 2x2 matrix  ----------  Constant
 }
else{
mid =n/2
P=MM( A11+A12, B11+B22,  n/2)
Q=MM( A21+A22, B11,  n/2)
R=MM( A11, B12-B22,  n/2)  ------------------  7*T(n/2)
S=MM( A22, B21-B11,  n/2)
T=MM( A11+A12, B22,  n/2)
U=MM( A21-A11, B11+B12,  n/2)
V=MM( A12-A22, B21-B22,  n/2)
}}
```

C11= P+S-T+V   C12= R+T

C21= Q+S   C22=P+R-Q+U

Recurrence relation

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

a=7, b=2, k=2 so, $\log_2 7$ = 2.81>2

Using master theorem

$T(n) = O(n^{\log_b a}) = O(n^{2.8})$