DIVIDE AND CONQUER II – MAX-MIN, STRASSEN'S MATRIX MULTIPLICATION

SUBJECT – DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY

WEEK 3

MIN MAX

- The aim is to find the ‘maximum’ and ‘minimum’ items in a set of ‘n’ elements with minimum number of operations.
- We use divide and conquer paradigm

DIVIDE AND CONQUER

- A general methodology for using recursion to design efficient algorithms is as follows:

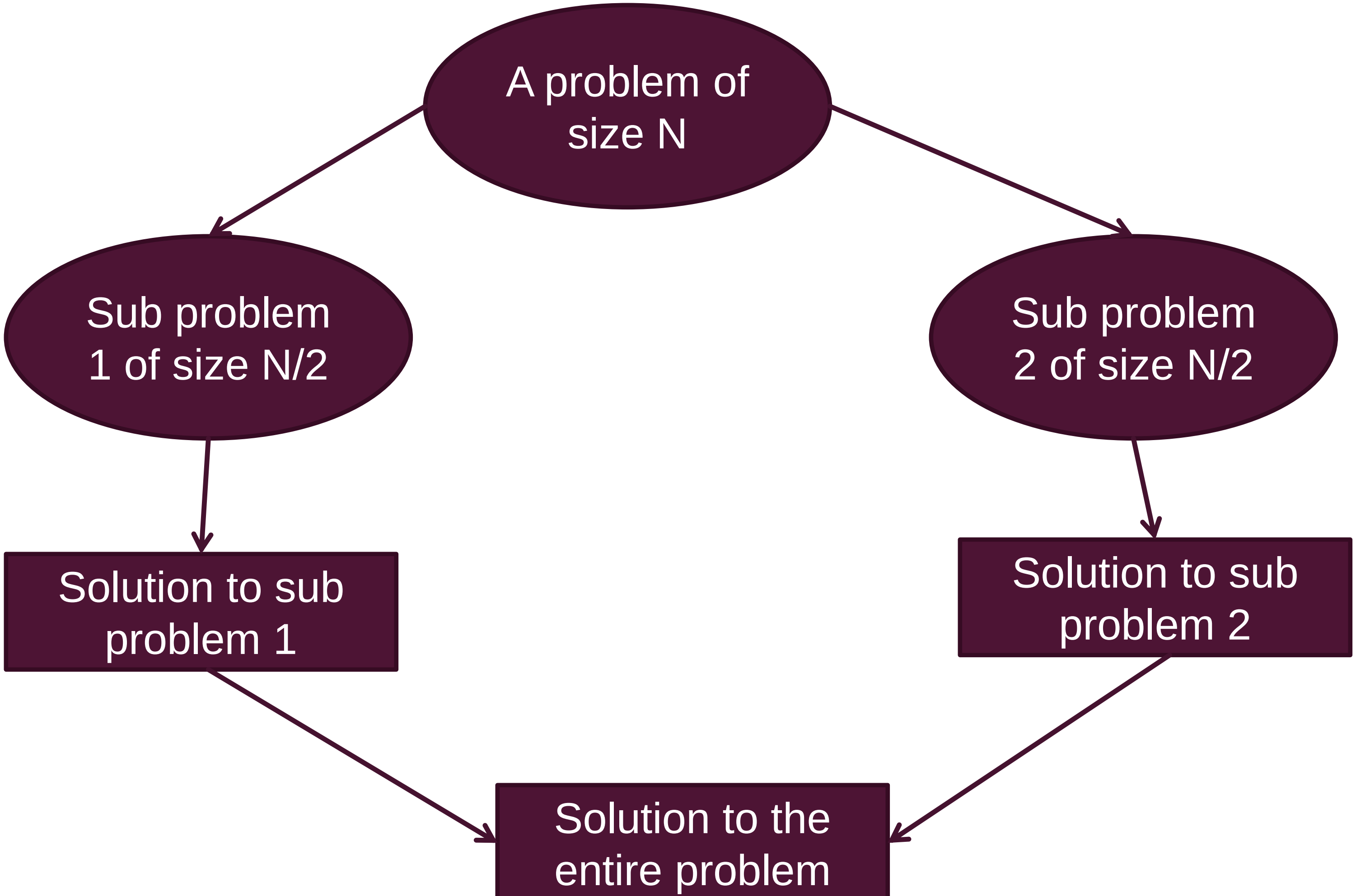
  It solves a problem by:

  – Diving the data into parts

  – Finding sub solutions for each of the parts

  – Constructing the final answer from the sub solutions

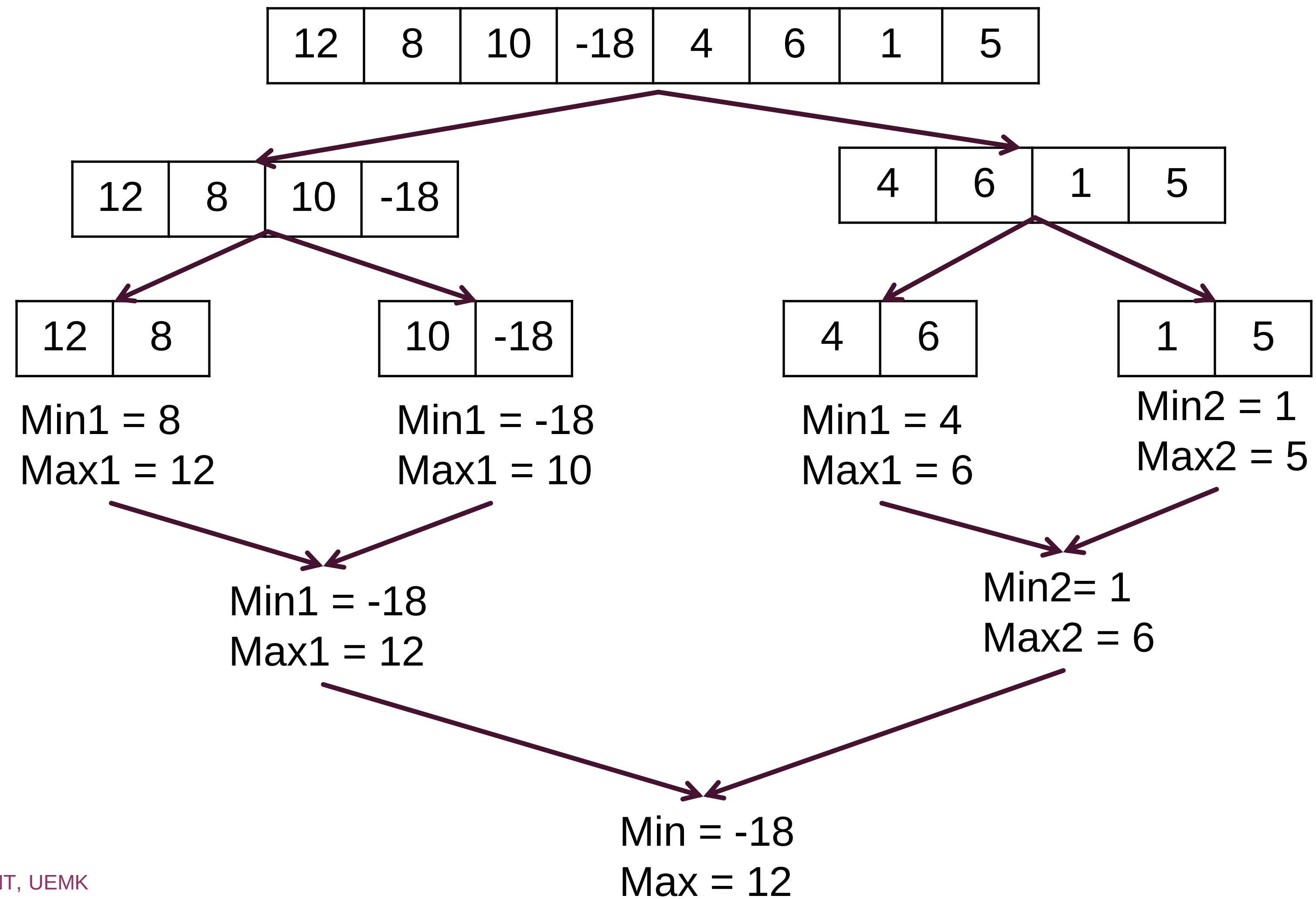
DIVIDE AND CONQUER



DIVIDE AND CONQUER

- For Divide-and-Conquer algorithms the running time is mainly affected by 3 criteria:
  - The **number of sub-problems into which** a problem is split.
  - The **ratio of initial problem size to sub-problem size.**
  - The **number of steps required to divide** the initial instance and to **combine the solutions.**

MIN MAX EXAMPLE



MIN MAX ALGORITHM

```
max_min(I,j,max,min)
begin
    if(i=j) then                    // Single element
      max = min=a[i]
      end if
    else
      if (i = j-1) than              // Double element
            if(a[i]<a[j]) then
                  max = a[j]
                  min = a[i]
            end if
            else
                  max = a[i]
                  min = a[j]
            end else
      end if
```

MIN MAX ALGORITHM

```
      else                              // More than two element
            mid = (i+j) / 2
            max_min(a, i, mid, max1, min1)
            max_min(a, mid+1,  j , max2, min2)
            if ( max1<max2) then
                  max1 = max2
            end if
            if (min1>min 2) then
                  min1 = min2
            end if
      end else
    end if
end
```

MIN MAX USING D&C– RECURRENCE RELATION

$T(1) = 0 \qquad ,n=1$

$T(2) = 1 \qquad ,n=2$

$T(n) = 2.T(n/2) + 2 \quad ,n>2$

MIN MAX USING D&C – COMPLEXITY ANALYSIS

T(n) = 2.T(n/2) + 2

= 2. (2. T(n/4) + 2) + 2 = 4.T(n/4) + 4 + 2

= 2. (2. (2. T(n/8) + 2) + 2) + 2 = 8T(n/8) + 8 + 4 + 2

= 2^k .T(n/2^k) + 2^k + ….. + 8 + 4 + 2

= 2^k .T(n/2^k) + 2.(2^k -1)/2-1 [sum of GP series]

= 2^k .T(n/2^k) + 2.2^k -2

Let us assume that $\boldsymbol{n / 2^k}$ =2

T(n) = (n/2).T(2) + 2(n/2)-2 = (n/2).1 + n-2 =n/2 + n -2

= 3n/2 – 2

= **O(n)**

MIN MAX COMPARATIVE ANALYSIS

- Simple Approach O(n)
- Divide & Conquer O(n)

```
Algorithm straight MaxMin (a, n, max, min)
// Set max to the maximum & min to the minimum of a [1: n]
{
Max = Min = a [1];
For i = 2 to n do
{
If (a [i] > Max) then Max = a [i];
If (a [i] < Min) then Min = a [i];
}}
```

RECURSIVE MATRIX MULTIPLICATION

- A N x N matrix can be viewed as a 2 x 2 matrix with entries that are (N/2) x (N/2) matrices.
- The recursive matrix multiplication algorithm recursively multiplies the (N/2) x (N/2) matrices and combines them using the equations for multiplying 2 x 2 matrices

Multiply 2 x 2 Matrices:

$$\begin{vmatrix} r & s \\ t & u \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} * \begin{vmatrix} e & g \\ f & h \end{vmatrix}$$

r = ae + bf
s = ag + bh
t = ce + df
u = cg + dh

MATRIX MULTIPLICATION
ANALYSIS

considering kth loop at step no 4

```
Algorithm : MULTIPLICATION(A,B,n)
{
 1  for( i :=0, i<n, i++)……………………………………………n+1
 2      for(j: 1 to n)……………………………………………n*(n+1)
 3         C[i][j] ← 0…………………………………………….n*n
 4             for(k: 1 to n)……………………………….......... n*n*(n+1)
 5              C[i][j]← C[i][j]  +  A[i][k] * B[k][j]……….n*n*n
    end while
 6   Return sum……………………………………………………....1
}
```

$T_{mul}(N)= 2n^3+3n^2+2n+2$

$O(n^3)$

RECURSIVE MATRIX MULTIPLICATION ALGORITHM

```
void multiply (int m1, int n1, int a[10][10], int m2, int n2, int b[10][10], int c[10][10])   //  Recursive
      function
{
static int i = 0, j = 0, k = 0;
if (i >= m1)
             {
                   return;
             }
else if (i < m1)
                {
                  if (j < n2)
                        {
                           if (k < n1)
                              {
                                 c[i][j] += a[i][k] * b[k][j];
                                 k++;
```

RECURSIVE MATRIX MULTIPLICATION ALGORITHM

```
multiply(m1, n1, a, m2, n2, b, c);
}
k = 0;
j++;
multiply(m1, n1, a, m2, n2, b, c);
}
j = 0;
i++;
multiply(m1, n1, a, m2, n2, b, c); }}
```

MATRIX MULTIPLICATION USING D&C

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix}$$

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

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

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

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

ALGORITHM

- Algorithm MMD&C(A,B,n)
- If (n<=2){ Direct equations are there}
- Else{
- mid =n/2

  MMD&C(A11,B11,n/2)+MMD&C(A12,B21,n/2)
- MMD&C(A11,B12,n/2)+MMD&C(A12,B22,n/2)
- MMD&C(A21,B11,n/2)+MMD&C(A22,B21,n/2)
- MMD&C(A21,B12,n/2)+MMD&C(A22,B22,n/2)
- }

STRASSEN'S MATRIX MULTIPLICATION

Multiply 2 x 2 Matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} r & s \\ t & u \end{bmatrix}$$

$r = p_6 + p_4 + p_5 – p_2$

$s = p_2 + p_1$

$t = p_3 + p_4$

$u = p_1 + p_5 – p_3 - p_7$

Where:

$p_1 = a * (f – h)$

$p_2 = h * (a + b)$

$p_3 = e * (c + d)$

$p_4 = d * (g – e)$

$p_5 = (a + d) * (e + h)$

$p_6 = (b - d) * (g + h)$

$p_7 = (a - c) * (e + f)$

EXAMPLE

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 8 \\ 5 & 10 \end{bmatrix}$$

$p_1 = 1 * (4 – 2) = 2$

$p_2 = 2 * (1 + 2) = 6$

$p_3 = 2 * (2 + 1) = 6$

$p_4 = 1 * (1 – 2) = -1$

$p_5 = (1 + 1) * (2 + 2) = 8$

$p_6 = (2 - 1) * (1 + 2) = 3$

$p_7 = (1 - 2) * (2 + 4) = - 6$

$r = p_6 + p_4 + p_5 – p_2$

$= 3 -1 + 8 – 6 = 4$

$s = p_2 + p_1 = 6 + 2 = 8$

$t = p_3 + p_4 = 6 -1 = 5$

$u = p_1 + p_5 – p_3 - p_7$

$= 2 + 8 – 6 + 6 = 10$

STRASSEN'S MATRIX MULTIPLICATION ANALYSIS

- General Matrix Multiplication:

  $T(n) = 8T(n/2) + \theta(n^2)$

  Order: $\theta(n^3)$ (master theorem)

- Strassen's Matrix Multiplication

  $T(n) = 7\,T(n/2) + \theta(n^2)$ [log 7= 2.81]

  Order: $\theta(n^{2.81})$ (master theorem)

THANK YOU