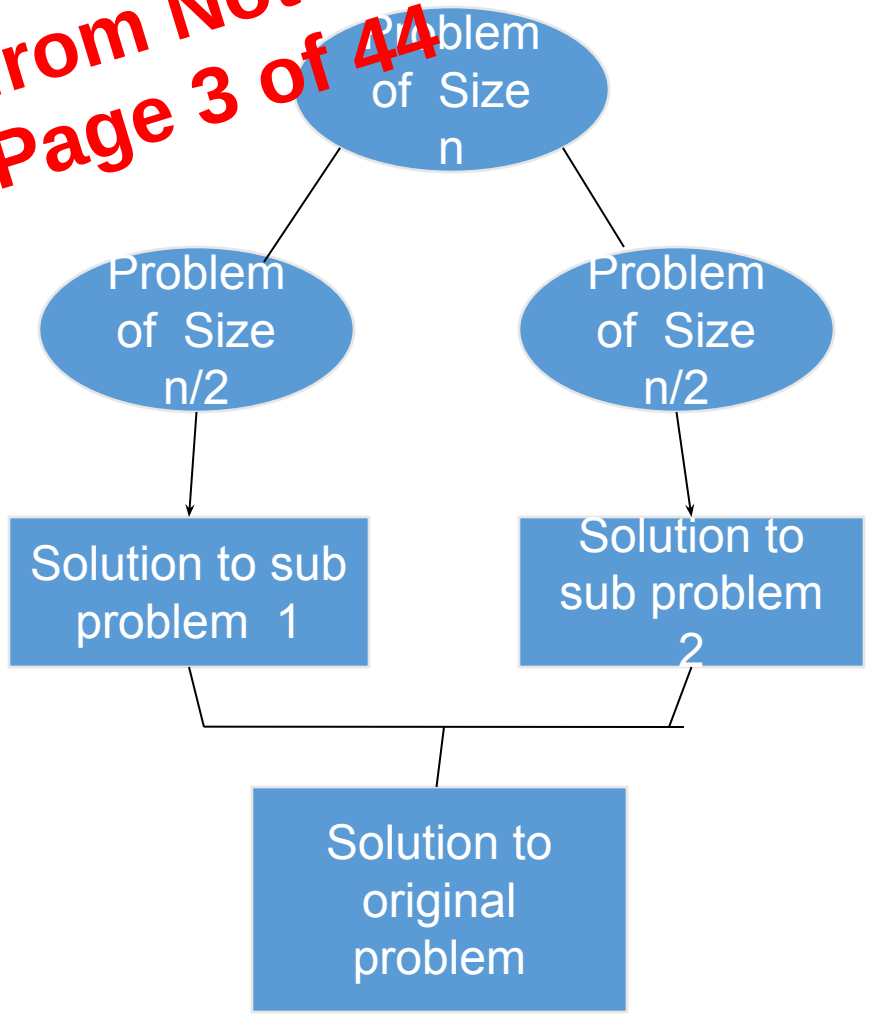
Introduction

- Diagram shows the general divide & conquer plan



Multiplying large Integers Problem

- There is one mathematical formula that we are going to apply here
  - (wz + xy) = (w + x)*(y + z) – wy – xz

- p = wy = 09 * 12 = 108
- q = xz = 81 * 34 = 2754
- r = (w + x) * (y + z) = 90 * 46 = 4140

- 0981 * 1234 = 10^4 p + 10^2 (r-p-q) + q
- = 1080000 + 127800 + 2754
- = 1210554

Merge Sort

- MergeSort(A,p,r)

```
if p>r
    return
    q=(p+r)/2
    MergeSort(A,p,q)
    MergeSort(A,q+1,r)
    MERGE(A,p,q,r)
```

Merge Sort

- MERGE (A, p, q, r)

$n_1 \leftarrow q - p + 1$

$n_2 \leftarrow r - q$

Create arrays L[1 . . $n_1 + 1$] and R[1 . . $n_2 + 1$]

for $i \leftarrow 1$ **to** n_1 then

do L[i] $\leftarrow$ A[$p + i - 1$]

for $j \leftarrow 1$ **to** n_2 then

do R[j] $\leftarrow$ A[$q + j$]

L[$n_1 + 1$] $\leftarrow \infty$

R[$n_2 + 1$] $\leftarrow \infty$

$i \leftarrow 1$

$j \leftarrow 1$

FOR $k \leftarrow p$ **TO** r

DO IF L[i] $\leq$ R[j]

THEN A[k] $\leftarrow$ L[i]

$i \leftarrow i + 1$

ELSE A[k] $\leftarrow$ R[j]

$j \leftarrow j + 1$

Merge Sort

- Time taken for divide is f(n)=2T(n/2) because of two half size partitions.
- For merge linear amount of work is done, so time taken is g(n)=O(n).
- Thus merge's are linear, and the overall time taken by merge sort is
  - T(n) = 2T(n/2) + O(n); or O(n log n)

Quick Sort

- Procedure partition(A,p,r)

```
{
    x<-A[r]
    i<-p-1

        for j<-p to r-1
        {
        if A[j]<=x
            then i<-i+1
            exchange A[i]<->A[j]
        }
        exchange A[i+1]<->A[r]
        return i+1
    }
```

Recursive Matrix Multiplication

```
Function MMult(A, B, n)
If n = 1
        Output A × B
Else
        Compute A11, B11, . . ., A22, B22 by computing m = n/2
        X1 ← MMult(A11, B11, n/2)
        X2 ← MMult(A12, B21, n/2)
        X3 ← MMult(A11, B12, n/2)
        X4 ← MMult(A12, B22, n/2)
        X5 ← MMult(A21, B11, n/2)
        X6 ← MMult(A22, B21, n/2)
        X7 ← MMult(A21, B12, n/2)
        X8 ← MMult(A22, B22, n/2)
        C 11 ← X1 + X2
        C 12 ← X3 + X4
        C 21 ← X5 + X6
        C 22 ← X7 + X8
        Output C
End If
```