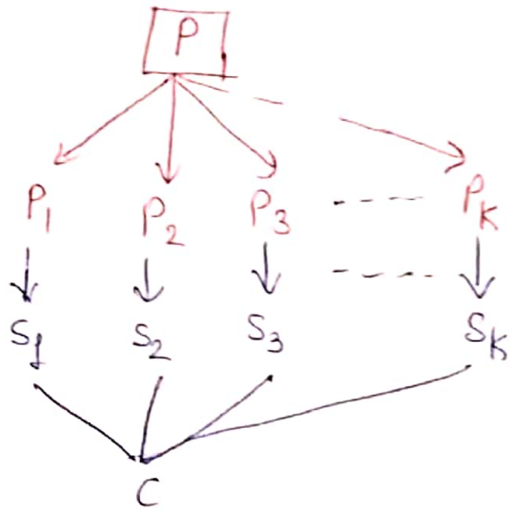
Divide & Conquer

Problem → i/p → size → P




```
DAC (P)
 {
   if (small (P))
    {
       S(P);
    }
   else
    {
      divide P into P1, P2, ... Pk
       Apply DAC(P1), DAC(P2)......
        Combine ( DAC(P1), DAC(P2), .... )
      }
 }
```

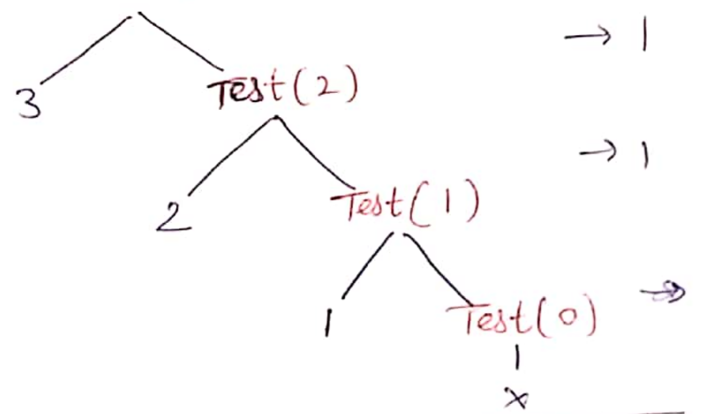
Problems depending upon Divide & Conquer:

- Binary Search
- Finding Maximum and minimum
- Merge sort
- Quick Sort
- Strassen's Matrix Multiplication

Recurrence Relation:

```
Void Test (int n)
{
  if (n>0)
  {
    printf ("%d", n);
    Test (n-1);
  }
}
```

Let Test (3) → 1 Unit of Time taking



3 + 1 → function calls

⇓

(n+1)

$f(n) = (n+1)$ calls

$O(n)$

Recurrence Relation :-

$$T(n) = \begin{cases} T(n-1) + 1 & ; \quad n>0 \\ 1 & ; \quad n=0 \end{cases}$$

Now, solve this Recurrence Relation. Using

→ Substitution Methode

$T(n) = T(n-1) + 1$

$\boxed{T(n-1) = T(n-2) + 1}$

$T(n) = [T(n-2)+1]+1$

$T(n) = [T(n-3)+1]+2 = [T(n-3)+3]$

$T(n) = [T(n-4)+1]+3 = [T(n-4)+4]$

upto k times

$T(n) = [T(n-k)+1]+k = [T(n-k)]+k$

Assume

$n-k=0$

$\boxed{n=k}$

$T(n) = [T(n-k)+k]$

$T(n) = [T(n-n)+n]$

$T(n) = [T(0)+n]$

$\boxed{T(n) = 1+n} \rightarrow$ calls.

$\Theta(n)$

T(n) ← void Test(int n)

```
1      ←     {
                if (n>0)
                {
n+1    ←       for(i=0; i<n; i++)
                  {
n      ←            printf("%d", n);
                  }
T(n-1) ←       Test(n-1);
                }
             }
```

$T(n) = T(n-1)+n+1+n+1$

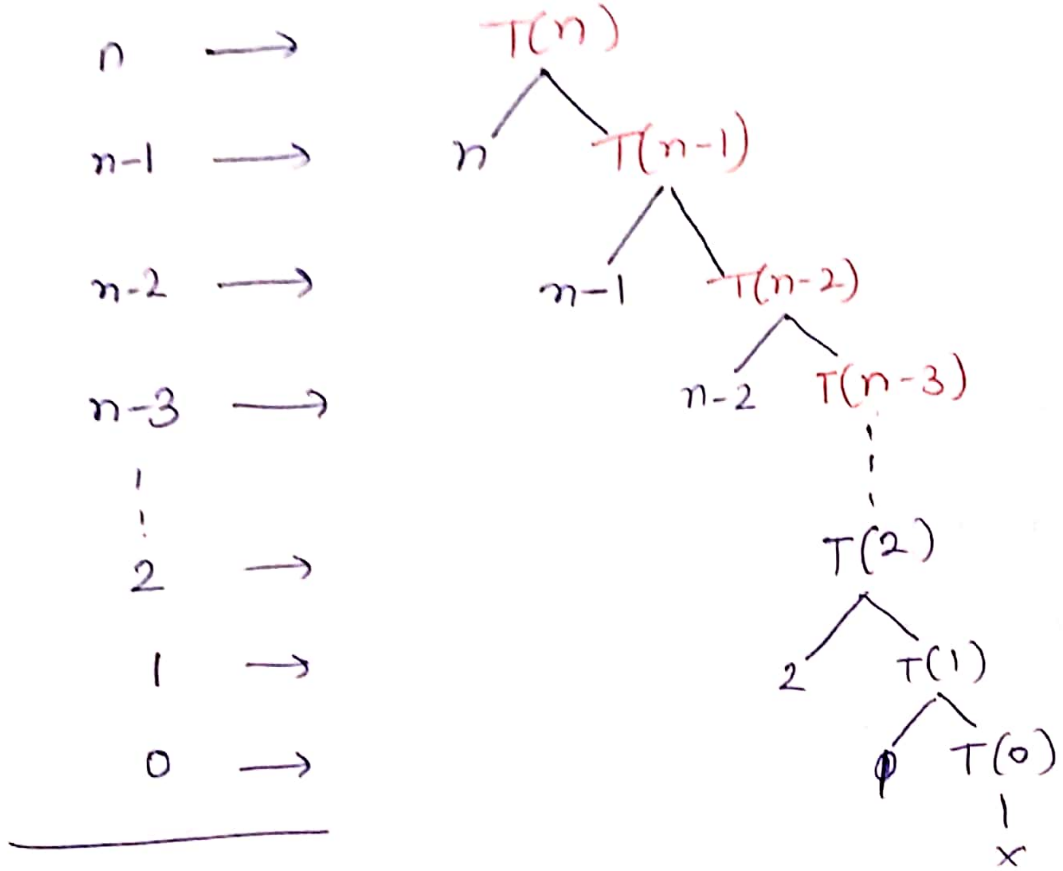
$= T(n-1)+2n+2$

$$T(n) = T(n-1) + n$$

$$\boxed{T(n) = \begin{cases} 1 & ; n = 0 \\ T(n-1) + n & ; n > 0 \end{cases}}$$

↳ Recurrence Relation.

Now, Solve it by using Recursion Tree :-



$$T(n) = 0 + 1 + 2 + \cdots + n$$

$$T(n) = \frac{n(n+1)}{2}$$

$$\boxed{\Theta(n^2)}$$

By substitution Method:

$$T(n) = T(n-1) + n \quad \text{— (I)}$$

$$T(n) = [T(n-2) + n-1] + n$$

$$T(n) = [T(n-2) + (n-1) + n] \quad \text{— (II)}$$

$$T(n) = [T(n-3) + (n+2)] + (n-1) + n$$

$$T(n) = [T(n-3) + (n+2) + (n-1) + n] \quad \text{— (III)}$$

$$\vdots \qquad \vdots$$

$$T(n) = [T(n-k) + (n-(k-1)) + (n-(k-2)) + \cdots + (n-1) + n]$$

Assume, $n-k=0$

$$\boxed{n=k}$$

$$T(n) = [T(\cancel{n}-\cancel{n}) + (\cancel{n}-(\cancel{n}-1)) + (\cancel{n}-(\cancel{n}-2)) + \cdots + (n-1) + n]$$

$$= T(0) + 1 + 2 + \cdots + (n-1) + n$$

$$T(n) = 1 + \frac{n(n+1)}{2}$$

$$\boxed{\Theta(n^2)}$$

#

```
void Test (int n)                    --> T(n)
{
  if (n>0)
  {
   for(i=1; i<n; i=i*2)
   {
     printf("%d", i);                --> log n
   }
   Test(n-1);                        --> T(n-1)
  }
}
```

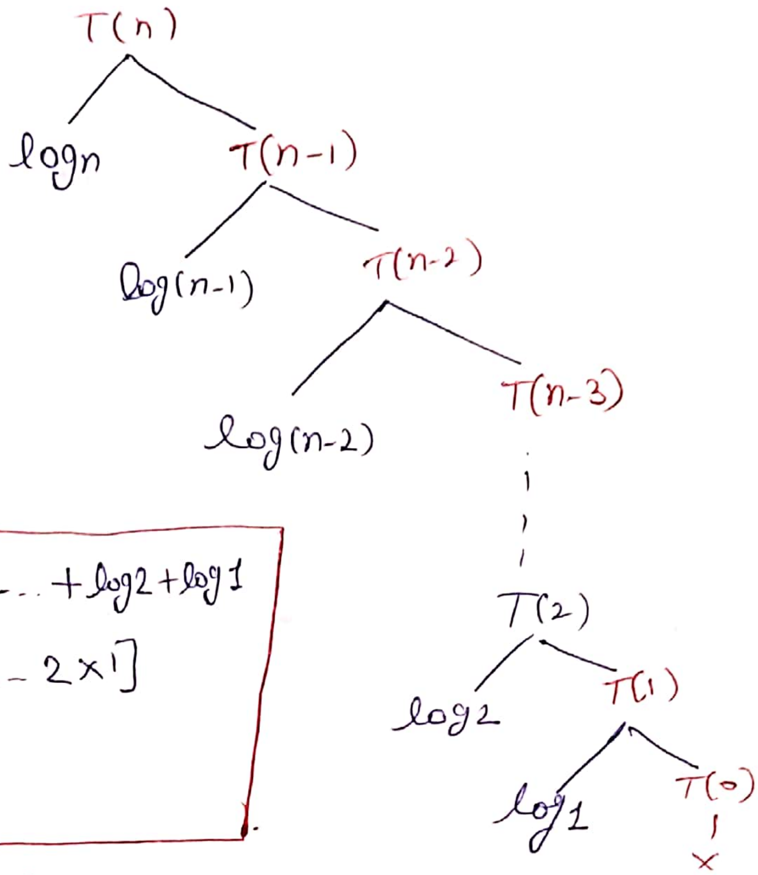
$$T(n) = T(n-1) + \log n$$

Scanned with CamScanner

Recurrence Relation:

$$T(n) = \begin{cases} 1 & ; n=0 \\ T(n-1)+\log n & ; n>0 \end{cases}$$

Recursion Tree:



$$\boxed{\begin{aligned} &\log n + \log(n-1) + \dots + \log 2 + \log 1 \\ &= \log[n \times (n-1) \times \dots 2 \times 1] \\ &= \log n! \end{aligned}}$$

$O(n \log n)$

By Substitution Method:

$$T(n) = \begin{cases} 1 & n=0 \\ T(n-1)+\log n ; & n>0 \end{cases}$$

$$T(n) = T(n-1) + \log n \quad \rightarrow (1)$$

$$T(n) = [T(n-2) + \log(n-1)] + \log n$$

$$T(n) = T(n-2) + \log(n-1) + \log n \quad \rightarrow (2)$$

$$T(n) = [T(n-3) + \log(n-2)] + \log(n-1) + \log n$$

$$T(n) = T(n-3) + \log(n-2) + \log(n-1) + \log n$$

$$T(n) = T(n-k) + \log 1 + \log 2 + \cdots + \log(n-1) + \log n$$

Let $n-k=0$

$$\boxed{n=k}$$

$$T(n) = T(\cancel{n}-\cancel{n}) + \log 1 + \log 2 + \cdots + \log(n-1) + \log n$$

$$T(n) = T(0) + \log n!$$

$$T(n) = 1 + \log n!$$

$$\boxed{O(n \log n)}$$

SUMMARY

$T(n) = T(n-1) + 1$	$O(n)$
$T(n) = T(n-1) + n$	$O(n^2)$
$T(n) = T(n-1) + \log n$	$O(n \log n)$
$T(n) = T(n-1) + n^2$	$O(n^3)$
$T(n) = T(n-2) + 1$	$\frac{n}{2} \to O(n)$
$T(n) = T(n-100) + n$	$O(n^2)$

Scanned with CamScanner

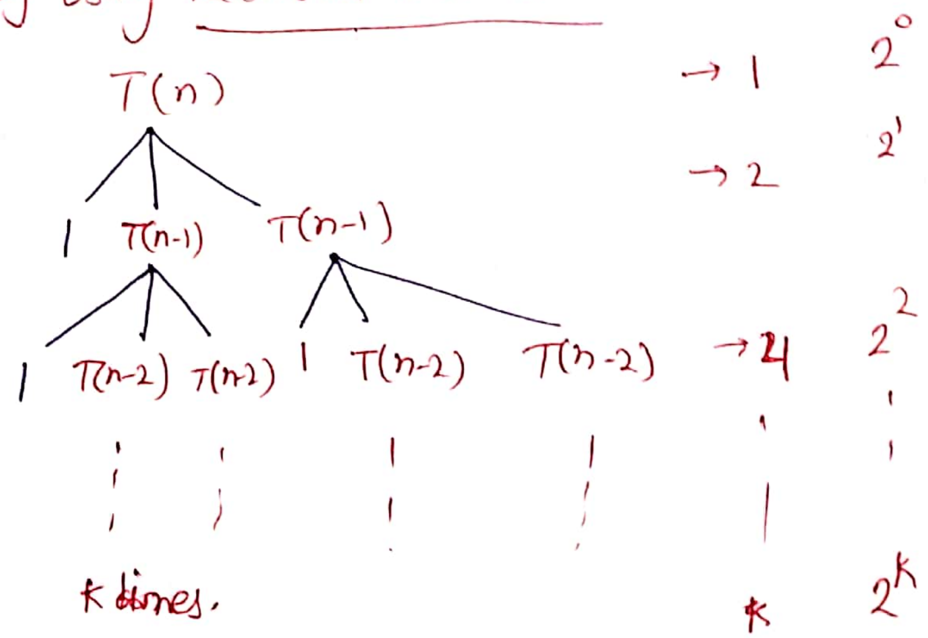
```
#   Algorithm Test (int n)                    → T(n)
    {
        if (n>0)
        {
            Printf("%d", n);                  → 1
            Test (n-1);                       → T(n-1)
            Test (n-1);                       → T(n-1)
        }
    }
```

$$\underline{T(n) = 2T(n-1)+1}$$

Recurrence Relation :-

$$T(n) = \begin{cases} 1 & n=0 \\ 2T(n-1)+1 \; ; & n>0 \end{cases}$$

Solution by using Recursion Tree Method:-



$$1+2+2^2+2^3+\cdots+2^k = 2^{k+1}-1$$

$$a+ar+ar^2+ar^3+\cdots+ar^k = \frac{a(r^{k+1}-1)}{r-1}$$

Let $n-k=0$

$\boxed{n=k}$

$2^{k+1}-1$

$2^{n+1}-1 \Rightarrow \boxed{O(2^n)}$

↳ By Substitution Method:-

$$T(n) = \begin{cases} 1 & ; n=0 \\ 2T(n-1)+1 & ; n>0 \end{cases}$$

$T(n) = 2T(n-1)+1 \rightarrow (1)$

$T(n) = 2[2T(n-2)+1]+1$

$= 2^2 T(n-2)+2+1 \rightarrow (2)$

$T(n) = 2^2[2T(n-3)+1]+2+1$

$= 2^3 T(n-3)+2^2+2+1 \rightarrow (3)$

|

|

$T(n) = 2^k T(n-k) + 2^{k-1} + 2^{k-2} + \dots + 2^2+2+1$

Let $n-k=0$

$\boxed{n=k}$

$T(n) = 2^n T(n-n) + 2^{n-1} + 2^{n-2} + \dots + 2^2 + 2 + 1$

$= 2^n T(0) + 1+2+2^2+\dots+2^{n-1}$

$= 2^n + 2^n - 1$

$2^{n+1} \cancel{+} 1 \Rightarrow \boxed{O(2^n)}$

Scanned with CamScanner

Master Theorem for Decreasing functions

$$\boxed{\begin{aligned} T(n) &= T(n-1)+1 &\rightarrow O(n) \\ T(n) &= T(n-1)+n &\rightarrow O(n^2) \\ T(n) &= T(n-1)+\log n &\rightarrow O(n\log n) \end{aligned}}$$

$$\boxed{\begin{aligned} T(n) &= 2T(n-1)+1 &\rightarrow O(2^n) \\ T(n) &= 3T(n-1)+1 &\rightarrow O(3^n) \\ T(n) &= 2T(n-1)+n &\rightarrow O(n2^n) \end{aligned}}$$

General form of Recurrence Relation:-

$$\boxed{T(n) = aT(n-b)+f(n)}$$

where $a>0, b>0$ & $f(n) = O(n^k)$

here, $k \geq 0$.

V. Imp.

if $a=1$	$O(n^{k+1})$ (or) $O(n * f(n))$
if $a>1$	~~$O(a^n)$~~ $O(n^k a^{n/b})$
if $a<1$	$O(n^k)$ or $O(f(n))$