Analysis of recursive time:-

→ Represent time taken by a recursive function as recurrence relation.

Example:-

```
f(n) {
  if (n ≤ 1):
      print ("base condition")
  else:
      print("not base condition")
      f(n-1).
}
```

So Time taken by $f(n)$, $T(n) = \begin{cases} 1 + T(n-1) & ; n > 1 \\ 1 & ; n \leq 1 \end{cases}$

base condition.

1] $T(n) = \begin{cases} n + 2T(n/2) & ; n > 1 \\ 1 & ; n \leq 1 \end{cases}$

Recursion level

$T(n) = n + 2T(n/2) \longrightarrow 1$

$= n + 2\left[\frac{n}{2} + 2T\left(\frac{n}{2^2}\right)\right]$

$= 2n + 2^2 T\left(\frac{n}{2^2}\right) \longrightarrow 2.$

$= 2n + 2^2\left[\frac{n}{2^2} + 2T\left(\frac{n}{2^3}\right)\right]$

$= 3n + 2^3 T\left(\frac{n}{2^3}\right) \longrightarrow 3$

$\vdots$

$T(n) = kn + 2^k T\left(\frac{n}{2^k}\right) \longrightarrow k.$ (Recursion depth)

⇒ At recursion level K, base condition is reached.

$$\Rightarrow T\left(\frac{n}{2^K}\right) = T(1).$$

$$\Rightarrow \frac{n}{2^K} = 1 \Rightarrow 2^K = n \Rightarrow K = \log n.$$

So, $T(n) = Kn + 2^K \underbrace{T\left(\frac{n}{2^K}\right)}_{T(1)}$, here. $K = \log n$.

$$\Rightarrow T(n) = n \cdot \log n + 2^{\log_2 n} \times 1.$$

$$\Rightarrow T(n) = \underline{\underline{n \log n}} + n. \Rightarrow \underline{\underline{\Theta(n \log n)}}.$$

↳ tight bound.

2] $T(n) = \begin{cases} \log n + T(n-1) & ; n > 0 \\ 1 & ; n \leq 0. \end{cases}$ ← base condition.

	Recursion level.
$T(n) = \log n + T(n-1) \longrightarrow$	1
$= \log n + \log(n-1) + T(n-2) \longrightarrow$	2
$= \log n + \log(n-1) + \log(n-2) + T(n-3) \rightarrow$	3
$= \underbrace{\log n + \log(n-1) + \log(n-2) + \dots + \log[n-(K-1)]} + T(n-K) \longrightarrow$	K. ← Recursion depth.

$$T(n) = \log[n \cdot (n-1) \cdot (n-2) \cdots [n-(K-1)]] + T(n-K).$$

At recursion level K, base condition is reached. ⇒ $T(n-K) = T(0)$

$\Rightarrow n - K = 0 \Rightarrow K = n.$

↳ $T(n) = \log[n \cdot (n-1) \cdot (n-2) \cdots 1] + T(1).$

$\therefore T(n) = \log[1 \times 2 \times 3 \times \dots \times n] + 1.$

$T(n) = \underline{\underline{\log(n!)}} + 1 \leq \underbrace{\log(n^n)}_{\text{upper bound.}}$

no tight bound exists. $\Rightarrow O(\log(n^n)) \Rightarrow O(n \log n).$

upper bound.

NOTE:- There is no tight bound for $n!$, so upper bound for $n!$ is n^n.

⇒ upper bound for $\log(n!)$ is $\log(n^n)$.

3) $$T(n) = \begin{cases} n + T(n/4) + T(n/2) & ; n > 1 \\ 1 & ; n \leq 1. \end{cases}$$

$$T(n) = n + \underbrace{T(n/4)}_{\text{terminates faster.}} + \underbrace{T(n/2)}_{\text{terminates slower.}}$$

For such recurrence relation, we can't find tight bound (θ).

But we can find lower bound (Ω) and upper bound (O).

upper bound is more useful than lower bound.

lower bound $\rightarrow$ $T(n) \geq$ [Replace all terms with the term which terminates faster]

$\Rightarrow$ lower bound $\rightarrow$ $T(n) \geq n + T(n/4) + T(n/4)$.

$\Rightarrow$ lower bound (Ω) $\rightarrow$ $T(n) \geq n + 2T(n/4)$.

Same way,

upper bound $\rightarrow$ $T(n) \leq$ [Replace all terms with the term which terminates slower]

$\Rightarrow$ upper bound $\rightarrow$ $T(n) \leq n + T(n/2) + T(n/2)$

$\Rightarrow$ upper bound (O) $\rightarrow$ $T(n) \leq n + 2T(n/2)$.

<u>Finding lower bound</u> :-   <u>Recursion level</u>

$$T(n) \geq n + 2T(n/2^2) \longrightarrow 1$$

$$\geq n + 2\left[\frac{n}{2^2} + 2T\left(\frac{n}{2^4}\right)\right]$$

$$\geq n + \frac{n}{2} + 2^2 T\left(\frac{n}{2^4}\right) \longrightarrow 2.$$

$$\geq n + \frac{n}{2} + 2^2\left[\frac{n}{2^4} + 2T\left(\frac{n}{2^6}\right)\right]$$

$$\geq n + \frac{n}{2} + \frac{n}{2^2} + 2^3 T\left(\frac{n}{2^6}\right) \longrightarrow 3.$$

$\vdots$

$$T(n) \geq n \times \underbrace{\left[1 + \frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^{K-1}}\right]}_{\text{Geometric progression (G.P.)}} + 2^K T\left(\frac{n}{2^{2K}}\right) \longrightarrow K.$$

$a = 1,\ r = \frac{1}{2},\ n = K.$

$$\Rightarrow T(n) \geq n \times 1 \times \left[\frac{\left(\frac{1}{2}\right)^K - 1}{\frac{1}{2} - 1}\right] + 2^K \underbrace{T\left(\frac{n}{2^{2K}}\right)}_{T(1)}.$$

$$\cancel{\Rightarrow T(n) \geq -2n}$$

$$\Rightarrow T(n) \geq -2n\left[\frac{1}{2^K} - 1\right] + 2^K \cdot \underbrace{T\left(\frac{n}{2^{2K}}\right)}_{T(1)}.$$

$$\Rightarrow \frac{n}{2^{2K}} = 1 \Rightarrow 2^{2K} = n \Rightarrow 2K = \log n$$

$$\Rightarrow K = \left(\frac{1}{2}\right) \times \log n. \Rightarrow K = \log\left(n^{1/2}\right).$$

$$\therefore T(n) \geq -2n\left[\frac{1}{n^{1/2}} - 1\right] + n^{1/2}$$

$$T(n) \geq -2n^{1/2} + 2n + n^{1/2}.$$

lower bound.

$$T(n) \geq 2n - n^{1/2} \Rightarrow T(n) = \Omega(n).$$

Finding upper bound:- Recursion level

$$T(n) \leq n + 2T(n/2) \longrightarrow 1$$

$$\leq 2n + 2^2 T(n/2^2) \longrightarrow 2.$$

$$\leq 3n + 2^3 T(n/2^3) \longrightarrow 3$$

$$\boxed{\frac{n}{2^K} = 1 \Rightarrow K = \log n.}$$

$\vdots$

$$T(n) \leq Kn + 2^K \cdot \underbrace{T(n/2^K)}_{T(1)} \longrightarrow K.$$

upper bound.

$$T(n) \leq n \cdot \log n + n. \Rightarrow T(n) = O(n \log n).$$

4] For fibonacci function,

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

$T(n) = 1+T(n-1)+T(n-2) \Rightarrow$. No tight bound (θ).

Finding upper bound :- recursion level

$$T(n) \leq 1+2T(n-1). \longrightarrow 1$$

$$\leq 1+2+2^2T(n-2) \longrightarrow 2.$$

$$\leq 1+2+2^2+2^3T(n-3) \longrightarrow 3$$

$$\vdots$$

$$\leq \underbrace{1+2+2^2+\dots+2^{K-1}}_{GP,} + 2^K \underbrace{T(n-K)}_{T(1).} \longrightarrow K.$$

$$\leq 2^K - 1 + 2^K \qquad \begin{aligned} n-K &= 1 \\ K &= n-1. \end{aligned}$$

$$\leq 2^{K+1} - 1$$

$$T(n) \leq 2^n - 1. \Rightarrow T(n) = O(2^n).$$

Finding lower bound:- Recursion level

$$T(n) \geq 1+2T(n-2) \longrightarrow 1$$

$$\geq 1+2+2^2T(n-2\times2) \longrightarrow 2.$$

$$\geq 1+2+2^2+2^3T(n-2\times3) \longrightarrow 3$$

$$\vdots$$

$$\geq \underbrace{1+2+2^2+\dots+2^{K-1}}_{GP} + 2^K \underbrace{T(n-2K)}_{T(0)} \longrightarrow K.$$

$$T(n) \geq 2^K - 1 + 2^K. \qquad \begin{aligned} n-2K &= 0. \\ K &= \frac{n}{2}. \end{aligned}$$

$$\geq 2^{K+1} - 1$$

$$T(n) \geq 2^{\frac{n}{2}+1} - 1. \quad T(n) = \Omega(2^n)$$

5] $$T(n) = \begin{cases} 1+T(n/16)+T(n/8)+T(n/4)+T(n/2) & ; n>1 \\ 1 & ; n\leq 1. \end{cases}$$

$$T(n) = 1+\underbrace{T(n/16)}_{\text{terminates faster.}}+T(n/8)+T(n/4)+\underbrace{T(n/2)}_{\text{terminates slower.}} \Rightarrow \text{No tight bound } (\theta).$$

$$T(n) \leq 1+4T(n/2) \Rightarrow \text{upper bound } (O).$$

$$T(n) \geq 1+4T(n/16) \Rightarrow \text{lower bound } (\Omega).$$

6] $$T(n) = \begin{cases} 1+T(n-1)+T(n-2)+\dots+T(n-10) & ; n>1. \\ 1 & ; n\leq 1. \end{cases}$$

$$T(n) = 1+T(n-1)+T(n-2)+\dots+T(n-10) \Rightarrow \text{No tight bound } (\theta).$$

$$T(n) \leq 1+10T(n-1) \Rightarrow \text{upper bound } (O).$$

$$T(n) \geq 1+10T(n-10) \Rightarrow \text{lower bound } (\Omega).$$